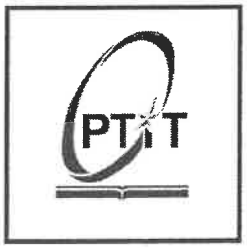
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



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

ĐỀ CƯƠNG ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC: PGS.TS HOÀNG XUÂN DẬU

HÀ NỘI – 2024

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi.

Các số liệu, kết quả nêu trong đề án tốt nghiệp là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tác giả đề án tốt nghiệp

LÊ NGỌC KHOA

MỤC LỤC

DANH MỤC CHỮ VÀ KÍ HIỆU VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
AI	Artificial intelligence	Trí tuệ nhân tạo
ANN	Artificial Neural Network	Mạng nơ-ron nhân tạo
API	Application Programming Interface	Giao diện lập trình ứng dụng
BoW	Bag of Words	Mô hình túi từ
CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
CSRF	Cross site request forgery	Một loại lỗ hổng web
CVE	Common Vulnerabilities and Exposures	Danh sách lỗ hổng bảo mật
DNN	Deep Neural Network	Mạng nơ ron sâu
FTP	File Transfer Protocol	Giao thức truyền tải tập tin
HTML	HyperText Markup Language	HyperText Markup Language
HTTP	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản
HTTPS	HyperText Transfer Protocol Secure	Giao thức bảo mật HTTP an toàn
IDS	Intrusion Detection System	Hệ thống phát hiện xâm nhập
IPS	Intrusion Prevention System	Hệ thống phát hiện xâm nhập và ngăn chặn
LDAP	Lightweight Directory Access Protocol	Giao thức ứng dụng truy cập các cấu trúc thư mục
LFI	Local File Inclusion	Một loại lỗ hổng web
LSTM	Long Short Term Memory networks	Mạng bộ nhớ ngắn dài
OS	Operating system	Hệ điều hành
OTP	One Time Password	Mật khẩu một lần
OWASP	Open Web Application Security Project	Dự án nguồn mở về bảo mật ứng dụng web
RCE	Remote Code Execution	Thực thi mã từ xa
RFI	Remote File Inclusion	Một loại lỗ hổng web
RNN	Recurrent neural networks	Mạng nơ ron hồi quy
SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc
SQLI	SQL Injection	Một loại lỗ hổng Web
SSRF	Server Side Request Forgery	Một loại lỗ hổng web

SVM	Support vector machine	Thuật toán học máy
TF-IDF	Term Frequency - Inverse Document Frequency	Mô hình trích xuất đặc trưng
URL	Uniform resource locator	Địa chỉ định vị tài nguyên thống nhất
WAF	Web Application Firewall	Tường lửa ứng dụng web
XSS	Cross-Site Scripting	Một loại lỗ hổng web

DANH MỤC BẢNG BIỂU

DANH MỤC HÌNH VẼ

LỜI MỞ ĐẦU

Sự phát triển của công nghệ thông tin trong thời đại cách mạng 4.0 đang ngày càng trở nên mạnh mẽ. Công nghệ đã và đang "phủ sóng" vào từng ngóc ngách của đời sống, tác động mạnh mẽ đến các lĩnh vực cốt lõi của xã hội như quốc phòng an ninh, y tế, giáo dục, quản lý hành chính. Các ứng dụng web cũng đang đóng vai trò quan trọng giúp người dùng dễ dàng tiếp cận với các nền tảng, ứng dụng phục vụ nhu cầu công việc và các tiện ích cuộc sống. Chính vì sự phổ biến của các ứng dụng web, yêu tố bảo mật và phòng chống tấn công được các tổ chức doanh nghiệp hết sức coi trọng.

Hiện nay, để đảm bảo an toàn cho các ứng dụng web các tổ chức doanh nghiệp đã kết hợp nhiều giải pháp an toàn bảo mật cùng với sự quản trị bởi các chuyên gia trong lĩnh vực an toàn thông tin. Tuy nhiên, tình trạng tội phạm mạng ngày càng gia tăng với việc các tin tặc sử dụng các kỹ thuật tinh vi nhằm vượt qua các lớp bảo mật. Giải pháp sử dụng học sâu trong an toàn thông tin đang được nghiên cứu và phát triển với độ chính xác cao.

Để giải quyết vấn đề phát hiện tấn công ứng dụng web, đề án này sử dụng học sâu (deep learning) để phát hiện các truy vấn HTTP tấn công. Học sâu là một chức năng của trí tuệ nhân tạo (AI), bắt chước hoạt động của bộ não con người trong việc xử lí dữ liệu và tạo ra các mẫu để sử dụng cho việc ra quyết định. Ứng dụng của học sâu đã mạng lại hiệu quả cao trong nhiều nhiệm vụ phân loại và nhận dạng trong các lĩnh vực xử lý ngôn ngữ tự nhiên (natural language processing) và thị giác máy tính (computer vision).

Đề án tập trung xây dựng mô hình phát hiện tấn công ứng dụng web dựa trên học sâu với nội dung được phân bổ trong 3 chương.

Chương 1: Các phương pháp tấn công ứng dụng web và phòng chống: chương này đưa ra khái niệm cơ bản về ứng dụng web, các phương pháp tấn công, một số lỗ hổng bảo mật và các phương pháp được sử dụng để phòng chống tấn công

Chương 2: Phát hiện tấn công web sử dụng học sâu: chương này trình bày tổng quan về học máy, học sâu, một số mô hình học sâu. Bên cạnh đó, mô hình Convolution Netural Network trong bài toán phát hiện tấn công ứng dụng web sẽ được trình bày cụ thể trong chương này.

Chương 3: Thử nghiệm và đánh giá: với tập dữ liệu lớn mô hình sẽ được cài đặt và thử nghiệm để đưa ra các kết quả và có những đánh giá về hiệu quả của mô hình.

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

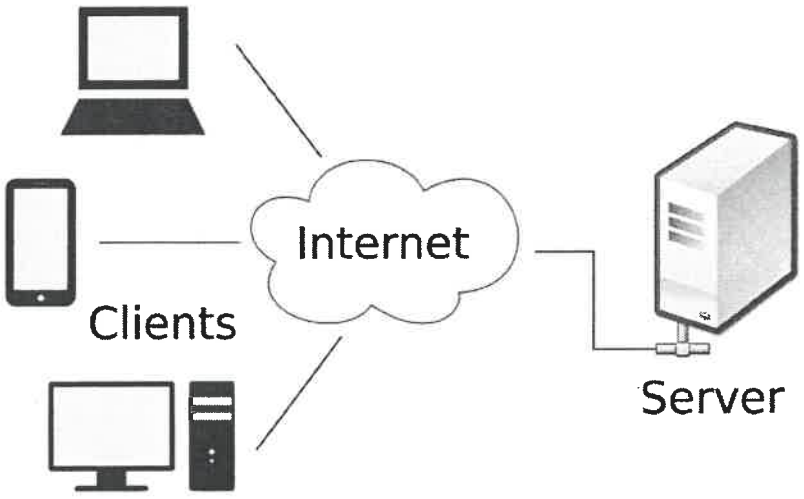
1.1 Tổng quan về ứng dụng web

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. Một số ứng dụng web phổ biến, quen thuộc nhất có thể kể đến như facebook.com, youtube.com, google.com,.. Không chỉ giới hạn trong giải trí, mà ứng dụng web còn lan rộng vào nhiều lĩnh vực quan trọng như tài chính, ngân hàng, bất động sản, y tế, giáo dục và mua sắm. Ứng dụng web là một yếu tố cốt lõi của cuộc cách mạng công nghệ 4.0, đóng vai trò quan trọng trong việc giúp con người nắm bắt thông tin và dữ liệu một cách nhanh chóng và hiệu quả.

1.1.2 Mô hình client – server của ứng dụng web

Mô hình cơ bản nhất của ứng dụng web:



Hình 1- 1 Mô hình client- server của một ứng dụng web [6]

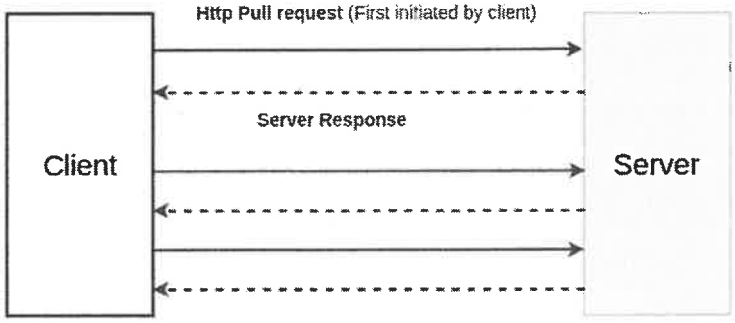
Mô hình client- server là một trong những mô hình phổ biến nhất tại thời điểm hiện tại. Không chỉ mang lại tính sẵn sàng cao, mô hình này cũng có khả năng mở rộng và phân phối nội dung tuyệt vời, chỉ cần người dùng có kết nối internet.

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).

Đối với client, máy khách trong mô hình web client-server,có thể nói là một ứng dụng hoặc trình duyệt web (như Chrome, Opera, Firefox, Safari…) được sử dụng để tương tác với các máy chủ Web theo yêu cầu của người dùng thông qua Internet. Về cơ bản, máy khách là một ứng dụng để gửi và nhận dữ liệu từ máy chủ.

Việc giao tiếp giữa Client và Server được thực hiện theo các gói tin HTTP. Đầu tiên, Client sẽ gửi một yêu cầu (HTTP request), mô tả công việc tới server. Khi yêu cầu HTTP được gửi đến, server dựa vào thông tin trong yêu cầu để xác định công việc cần phải thực thi. Sau đó, Server sẽ phản hồi cho client trong một gói tin HTTP trả lời (HTTP response).

Tổng quan quá trình gửi - nhận dữ liệu có thể mô tả bằng lược đồ như sau:



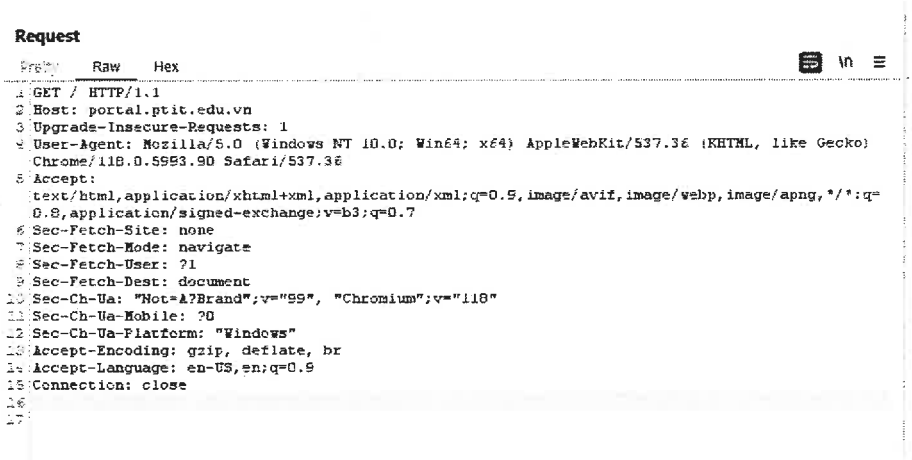
HTTP Pull

Hình 1- 2 Quy trình gửi nhận dữ liệu trong mô hình Client-Server.

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).

Thông thường, cổng được sử dụng trong ứng dụng web là cổng 80 (HTTP) hoặc HTTPS (443). Máy chủ HTTP tiếp nhận yêu cầu để xử lý.



Hình 1- 3 Ví dụ về yêu cầu HTTP

Các thành phần trong HTTP request có chứa đầy đủ các thành phần về kết nối và các thông tin cần truy vấn như:

- HTTP version: Phiên bản giao thức HTTP được sử dụng
- HTTP method: Phương thức được sử dụng trong yêu cầu, có thể là GET, POST, PUT, DELETE, OPTION,..
- URL: URL
- Host: domain của ứng dụng web
- Cookie: Thông tin về phiên làm việc của client và server

- User-agent: Thông tin về trình duyệt (client) được sử dụng trong request.

Sau khi xử lý yêu cầu, máy chủ phản hồi lại cho máy khách một HTTP response có định dạng như sau:

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Wed, 08 Nov 2023 02:21:02 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: close
6 Vary: Accept-Encoding
7 X-Pingback: https://portal.ptit.edu.vn/xmlrpc.php
8 Link: <https://portal.ptit.edu.vn/wp-json/>; rel="https://api.w.org/"
9 Link: <https://portal.ptit.edu.vn/>; rel=shortlink
10 Strict-Transport-Security: max-age=31536000;includeSubdomains;
11 Content-Length: 143149
12
13 <!DOCTYPE html>
14 <html class="no-js" lang="vi" prefix="og: http://ogp.me/ns#">
15
16   <!-- head -->
17   <head>
19     <meta name="google-site-verification" content="
       &-OiBGQbaAiS6bn579_tmrdlvyCHnWA6GdbCeAe3kiB" />
20     <!-- Begin EMC Tracking Code -->
20     <script type="text/javascript">
21       var _govaq = window._govaq || [];
22       _govaq.push(['trackPageView']);
23       _govaq.push(['enableLinkTracking']);
24       (function () {
25         _govaq.push(['setTrackerUrl', 'https://f-emc.ngsp.gov.vn/tracking']);
26         _govaq.push(['setSiteId', '15']);
27         var d = document,
28         g = d.createElement('script'),
29         s = d.getElementsByTagName('script')[0];
30         g.type = 'text/javascript';
31         g.async = true;
32         g.defer = true;
33         g.src = 'https://f-emc.ngsp.gov.vn/embed/gov-tracking.min.js';
34         s.parentNode.insertBefore(g, s);
35       }
```

Hình 1- 4 Ví dụ về phản hồi HTTP

Phản hồi thể hiện các thông tin như: Mã trạng thái phản hồi: 200 là mã thành công.

Có nhiều mã trạng thái

- 200 – 299: request thành công
- 300 – 399: Điều hướng gói tin
- 400 – 499: lỗi phía máy khách
- 500 – 599: lỗi phía máy chủ

Thông tin server: Một số máy chủ như apache, tomcat, nginx,..

Content – Length: Độ dài nội dung gói tin phản hồi HTTP

Giao thức HTTP này cũng nằm trong số các giao thức dễ bị tấn công trong ứng dụng web. Vì vậy, nên sử dụng giao thức HTTPS thay thế nhằm đảm bảo tính bảo mật của thông tin truyền đi.

1.2 Các lỗ hổng bảo mật trong ứng dụng web

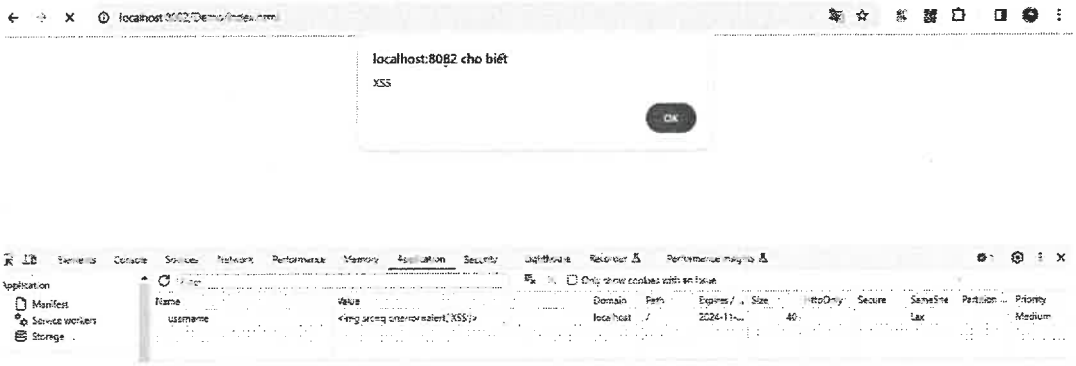
1.2.1 Lỗ hổng bảo mật ứng dụng web là gì?

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.

Khái niệm unstrusted data

Dữ liệu không tin cậy (Untrusted Data) là một trong những khái niệm căn bản nhất khi học về An Toàn Thông Tin nói chung. "Untrusted data" ám chỉ những dữ liệu mà người lập trình không kiểm soát được, chúng ta cần phải xác định hết tất cả Untrusted Data trên một ứng dụng Web. Vì nó chính là những cửa ngõ mà hacker bắt đầu tấn công vào.

Đây là một khái niệm cơ bản trong an toàn bảo mật nhưng là cần thiết để tránh trường hợp bỏ sót hoặc không lường hết được các mối nguy cơ có thể xảy ra đối với hệ thống. Các vị trí có thể trở thành dữ liệu không tin cậy trong truy vấn HTTP rất đa dạng. Nó có thể là các biến trong truy vấn và cũng có thể là các trường trong header hoặc thậm chí là cả các method.



Hình 1- 5 Ví dụ unstrusted data tồn tại ở biến username trong trường cookie thuộc header của truy vấn HTTP

1.2.2 Top 10 lỗ hổng, rủi ro theo OWASP [16]

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.

Trong lần cập nhật mới nhất năm 2021, danh sách 10 rủi ro/ lỗ hổng nghiêm trọng gồm có:

A01:2021-Broken Access Control

A02:2021-Cryptographic Failures

A03:2021-Injection

A04:2021-Insecure Design

A05:2021-Security Misconfiguration

A06:2021-Vulnerable and Outdated Components

A07:2021-Identification and Authentication Failures

A08:2021-Software and Data Integrity Failures

A09:2021-Security Logging and Monitoring Failures

A10:2021-Server-Side Request Forgery

A01:2021-Broken Access Control (Lỗ hổng kiểm soát truy cập)

Kiểm soát truy cập là một phần rất quan trọng trong các ứng dụng web. Trong đó hai khâu xác thực (authentication) và phân quyền (authorization), nếu hai khâu này không được xây dựng đủ mạnh thì hệ thống sẽ bị kẻ xấu truy cập trái phép, đánh cắp thông tin.

Một số ví dụ có thể kể đến như:

- Nhân viên có quyền truy cập đến các tài nguyên mà chỉ lãnh đạo mới có quyền truy cập và tác động.
- Truy cập vào các chức năng dành cho quản trị thông qua các URL, các API.
- Truy cập vào các thông tin, tài nguyên của người dùng ngang quyền thông qua mã định danh tài khoản (ví dụ: GET /user_infor.php?id=123).
- Cấu hình chia sẻ tài nguyên chưa chính xác (Cross-origin resource sharing).

A02:2021-Cryptographic Failures (Lỗ hổng mã hóa)

Những lỗi mã hóa dữ liệu có thể gây ra hậu quả nghiêm trọng, khi chúng có thể là nguyên nhân dẫn đến lộ các dữ liệu nhạy cảm, quan trọng hoặc xâm phạm hệ thống.

Một số ví dụ của lỗi mã hóa như:

- Lưu trữ mật khẩu với thuật toán mã hóa lỗi thời, có thể bị bẻ khóa (MD5,..)
- Sử dụng kết nối HTTP để nhận và truyền dữ liệu
- Các thuật toán sinh ngẫu nhiên chưa đủ mạnh

A03:2021-Injection (lỗ hổng chèn mã)

Chèn mã là lỗ hổng cho phép kẻ tấn công chèn vào dữ liệu gửi lên máy chủ và sau đó được thực thi trên hệ thống. Lỗ hổng chèn mã luôn nằm trong nhóm các lỗ hổng bảo mật nghiêm trọng nhất, bị khai thác nhiều nhất và hậu quả nếu bị khai thác cũng là rất nặng nề.

Một số dạng tấn công cụ thể của lỗ hổng chèn mã gồm: SQL Injection (chèn mã SQL), OS command injection (chèn mã lệnh OS command), Cross-site Scripting (XSS), LDAP injection,...

A04:2021-Insecure Design (thiết kế không an toàn)

Các ứng dụng không có thiết kế an toàn, yếu tố bảo mật không được chú trọng đến có nhiều khả năng gây rủi ro cho dữ liệu của người dùng, các tài nguyên hệ thống, tài chính và uy tín doanh nghiệp.

Ví dụ về thiết kế không an toàn:

Các rạp chiếu phim cho phép đặt tối đa 15 chỗ trước mà không yêu cầu đặt cọc tiền vé, nếu kẻ xấu lợi dụng để đặt nhiều vé ở nhiều rạp chiếu phim thì có thể sẽ gây thiệt hại lớn nếu các ghế đã được đặt mà không được thanh toán. Khi đó các rạp chiếu phim sẽ thiệt hại lớn về doanh thu.

A05:2021-Security Misconfiguration (Thiếu sót cấu hình bảo mật)

Trong quá trình triển khai vận hành các ứng dụng web, việc thiếu sót trong cấu hình bảo mật có thể gây ra những hậu quả nghiêm trọng. Một số lỗi thực tế do thiếu sót trong cấu hình như:

- Một số ứng dụng tạo mật khẩu mặc định cho người dùng, tuy nhiên hệ thống không yêu cầu người dùng phải đổi mật khẩu sau lần đăng nhập đầu tiên.
- Các cổng dịch vụ, các service chạy nội bộ nhưng lại được public ra internet
- Các cài đặt bảo mật không được thiết lập (giới hạn truy cập đến các hệ thống, chính sách đổi mật khẩu, ..)

A06:2021-Vulnerable and Outdated Components (Sử dụng các thành phần tồn tại lỗ hổng hoặc không còn hỗ trợ)

Đây là một lỗ hổng rất phổ biến hiện nay, do một ứng dụng web được tạo thành bởi nhiều thư viện, nền tảng khác nhau. Do đó khi một thư viện hay nền tảng tồn tại lỗ

hổng bảo mật hay đã không còn được các nhà cung cấp tiếp tục phát triển sẽ trở thành mục tiêu được rất nhiều kẻ tấn công hướng đến.

Ví dụ: Năm 2021, trên thư viện ghi log Log4j (CVE-2021-44228) được sử dụng trong rất nhiều ứng dụng web tồn tại lỗ hổng nghiêm trọng cho phép thực thi mã bất kỳ trên hệ thống.

A07:2021-Identification and Authentication Failures (Lỗ hổng xác minh danh tính và xác thực người dùng)

- Ứng dụng web cho phép sử dụng các mật khẩu mặc định, yếu hay phổ biến như admin/admin, root/root, admin/password.
- Lưu trữ mật khẩu ở dạng bản rõ, hoặc mã hóa với các thuật toán yếu.
- Không sử dụng đa xác thực (nhập username/password và nhập mã OTP).
- Chức năng tạo mới mật khẩu không an toàn.

A08:2021-Software and Data Integrity Failures(Lỗi toàn vẹn dữ liệu và phần mềm)

Đây là một lỗi mới, về việc cài đặt và triển khai các phần mềm hoặc phần cứng không đúng cách gây ra, ví dụ về việc triển khai CI/CD, hoặc CD không an toàn, gây ra những lỗi bảo mật nghiêm trọng đến hệ thống.

A09:2021- Security Logging and Monitoring Failures (Lỗ hổng lỗi ghi nhật ký bảo mật và giám sát lỗi)

Giám sát an toàn bảo mật và thu thập nhật ký là khâu vô cùng quan trọng trong đảm bảo an toàn ứng dụng web. Làm tốt khâu này có thể hạn chế, ngăn chặn các cuộc tấn công, điều tra nguyên nhân sự cố. Chính vì thế, nếu xảy ra lỗi giám sát và thu thập nhật ký sẽ dẫn đến những hậu quả nghiêm trọng.

A10:2021- Server Side Request Forgery (SSRF)

SSRF (Server Side Request Forgery) hay còn gọi là tấn công yêu cầu giả mạo từ phía máy chủ cho phép kẻ tấn công thay đổi tham số (thường là các url) được sử dụng trên ứng dụng web để truy cập trái phép, kiểm soát truy cập hệ thống hoặc làm bàn đạp để tấn công một hệ thống thứ 3.

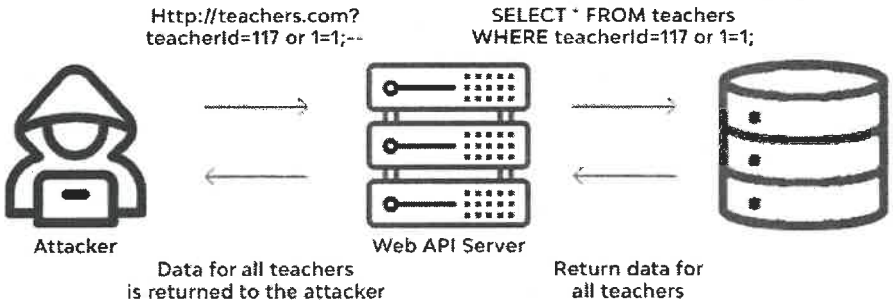
Ví dụ: Trang quản trị hệ thống chỉ cho phép truy cập từ mạng nội bộ thông qua domain localhost không thể truy cập thông qua internet. Lợi dụng điều này kẻ tấn công có thể tạo một url giả mạo chính là đường dẫn đến trang quản trị và truy cập trang quản trị thành công.

```
POST /product/stock HTTP/1.0
.....
stockApi=http://localhost/admin
```

1.2.3 Một số lỗ hổng bảo mật Web phổ biến

1.2.3.1 SQL injection

Tấn công chèn mã SQL (SQL Injection) là một kỹ thuật cho phép kẻ tấn công chèn mã SQL vào dữ liệu gửi đến máy chủ ứng dụng web và sau đó máy chủ cơ sở dữ liệu thực thi truy vấn. Khai thác thành công lỗ hổng chèn mã SQL có thể cho phép kẻ tấn công vượt qua bước đăng nhập, xem, sửa đổi, xóa dữ liệu, đánh cắp thông tin trong cơ sở dữ liệu và có thể chiếm quyền điều khiển máy chủ cơ sở dữ liệu. Đối với các ứng dụng web có kết nối cơ sở dữ liệu, chèn mã SQL là dạng tấn công phổ biến được tin tặc nhằm đến.



Hình 1- 6 Khai thác lỗ hổng SQL Injection

Một ví dụ về lợi dụng khai thác lỗ hổng chèn mã SQL để vượt qua xác thực người dùng:
Câu lệnh truy vấn SQL khi người dùng thực hiện đăng nhập:

$sql_get_user = "SELECT * FROM user WHERE username='$_POST['username']
' AND password='$_POST['password'] ';

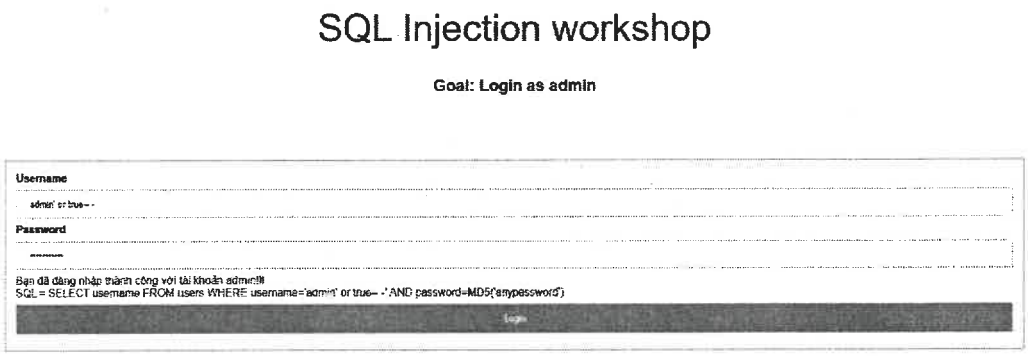
Kẻ tấn công có thể đăng nhập với thông tin username/password như sau:

username: admin'--

password: anyword

Khi đó câu truy vấn sẽ trở thành:

$sql_get_user = "SELECT * FROM user WHERE username='admin' or true--
AND password='anyword ';



Hình 1- 7 Lợi dụng lỗ hổng SQL Injection để vượt qua khâu xác thực

Kết quả thực thi truy vấn sẽ trả về tất cả các bản ghi trong bảng users do điều kiện OR true làm cho mệnh đề WHERE trở nên luôn đúng, điều kiện kiểm tra mật khẩu không được thực thi bởi ký hiệu (--). Trong ngôn ngữ truy vấn SQL, ký hiệu (--) đại diện cho phần ghi chú và sẽ không được thực thi. Nếu trong bảng users có chứa user admin, kẻ tấn công sẽ đăng nhập thành công vào hệ thống với user admin.

Do hậu quả của tấn công chèn mã SQL là rất nghiêm trọng, do đó nhiều giải pháp được đề xuất nhằm hạn chế ảnh hưởng và ngăn chặn triệt để lỗ hổng này. Nhiều phương án khắc phục lỗ hổng được kết hợp giúp đảm bảo an toàn cho hệ thống. Một số biện pháp, kỹ thuật có thể áp dụng gồm:

- Sử dụng các câu lệnh SQL được tham số hóa: Các truy vấn tham số hóa sẽ yêu cầu phải có câu lệnh SQL trước, sau đó dữ liệu được tham số hóa và truyền vào câu lệnh SQL. Cách này sẽ giúp cơ sở dữ liệu phân biệt được đâu là câu lệnh truy vấn và đâu là dữ liệu người dùng nhập. Câu lệnh này đảm bảo kẻ tấn công không thể thay đổi mục đích của câu truy vấn, ngay cả khi các đoạn mã truy vấn độc hại được nhập vào.
- Sử dụng Stored Procedure: Tuy không đảm bảo an toàn tuyệt đối trước các cuộc tấn công chèn mã SQL, tuy nhiên Stored Procedure cũng có hiệu quả nhất định.
- Cách hoạt động của Stored Procedure cũng khá tương tự như tham số hóa câu lệnh SQL, khác biệt ở đây là thay vì được lưu trữ trong mã của ứng dụng web,
- Stored Procedure được lưu trữ trực tiếp trong cơ sở dữ liệu, và được gọi ra từ ứng dụng.
- Xác thực đầu vào: Cách hữu hiệu nhất để ngăn chặn chèn mã tấn công SQL là xác thực đầu vào, trước khi thực hiện truy vấn. Có thể sử dụng các bộ lọc có sẵn, hoặc các thư viện, hay các framework cũng là một cách hữu hiệu.
- Không hiển thị các thông báo lỗi hoặc các ngoại lệ: Một trong những cách kẻ tấn công dùng đó là dựa vào các thông báo lỗi để thực hiện truy vết phiên bản cơ sở dữ liệu, kiểu lưu trữ,... vì vậy tốt hơn cả, không nên hiển thị chi tiết các thông báo lỗi hay các ngoại lệ.
- Tạo bản sao dữ liệu thường xuyên: Phòng trong trường hợp kẻ tấn công xóa hoặc thay đổi dữ liệu, ta luôn có một bản sao để khôi phục về trạng thái ban đầu.

1.2.3.2 Cross-Site Scripting (XSS)

Lỗ hổng Cross-Site Scripting (XSS) là một trong những lỗ hổng được tin tặc sử dụng phổ biến nhất để tấn công các ứng dụng web. Lợi dụng việc ứng dụng không kiểm soát được sự có mặt của các mã thực thi trái phép trong giá trị tham số đầu vào và/ hoặc trong phản hồi HTTP. Mã khai thác tồn tại trong các ứng dụng web chạy trên trình duyệt với quyền truy nhập của người dùng. Hậu quả của tấn công XSS có thể giúp tin

tặc đánh cắp thông tin người dùng cuối, ứng dụng web bị chèn các thông tin, hình ảnh xấu.

Có 3 loại tấn công XSS chính: Stored XSS, Reflected XSS, DOM-based XSS.

Ví dụ, một trang web có chức năng tìm kiếm như sau:

http://example.com/search?keyword=abc

Kẻ tấn công có thể chèn đoạn mã Javascript để khai thác lỗ hổng XSS như sau:

http://example.com/search?keyword=<script>alert(document.cookie)</script>

Khi đó, đoạn mã JavaScript sẽ được thực thi trên trình duyệt của người dùng.

Một vài biện pháp phòng chống của lỗ hổng này có thể kể đến như:

- Lọc dữ liệu đầu vào: lọc dữ liệu đầu vào luôn là một trong những cách phòng chống hữu hiệu nhất cho những cuộc tấn công chèn mã. Mọi dữ liệu người dùng đều phải qua quá trình lọc và xác thực, đặc biệt chú ý đến những ký tự đặc biệt.
- Sử dụng Entity Encode/ Escaped: Thay vì sử dụng trực tiếp các ký tự như '<', '>', … ta có thể sử dụng các nhóm ký tự mang ý nghĩa tương đương như &#60, &#61,... Các kí tự này sẽ ngăn chặn được việc thực thi đoạn mã script ở phía máy khách.
- Sử dụng các thư viện Javascript: Các thư viện Javascript hiện đại hầu hết đều được tích hợp sẵn các bộ lọc/ Escaped Javascript.

1.2.3.3 Cross-site request forgery

Cross site request forgery (CSRF) là một lỗ hổng bảo mật cho phép kẻ tấn công lừa người dùng thực hiện các hành động mà họ không có ý định thực hiện trên ứng dụng web mà họ đã xác thực. Các cuộc tấn công này thường kết hợp với phishing, social engineering để tiếp cận người dùng, lừa họ truy cập các URL một cách vô thức, thông qua tấn công CSRF có thể thực hiện chuyển tiền, thay đổi thông tin cá nhân, thay đổi thông tin đăng nhập…

Một ví dụ cho việc tấn công CSRF như sau:

Một URL có chức năng chuyển tiền có dạng như sau:

http://example_bank.com/transfer?username=alice&amount=100

Kẻ tấn công có thể lừa người dùng bằng cách gửi cho người dùng liên kết thông qua email hay tin nhắn, ví dụ như:

Free gift

Như vậy, chỉ cần người dùng click vào liên kết hay truy cập liên kết, khoản tiền 100$ sẽ được chuyển cho người dùng có username=alice

Cách phòng chống CSRF:

- Sử dụng CSRF token cho mỗi yêu cầu HTTP gửi lên máy chủ nhằm tránh kẻ tấn công giả mạo.
- Sử dụng captcha
- Sử dụng các thư viện phòng chống CSRF.

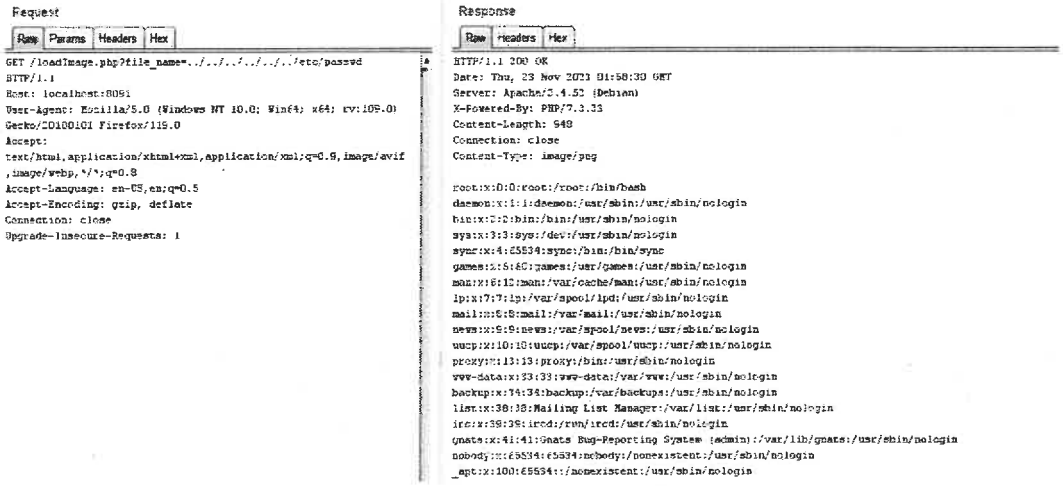
1.2.3.4 Path Traversal

Lỗ hổng web Path Traversal (còn được gọi là Directory Traversal) là một lỗ hổng bảo mật phổ biến trên các ứng dụng web. Lỗ hổng này cho phép kẻ tấn công truy cập vào các tệp tin và thư mục trên máy chủ web mà không được phép truy cập. Điều này có thể gây ra nhiều nguy hiểm cho hệ thống web và dữ liệu của người dùng.

Cơ chế hoạt động của lỗ hổng Path Traversal là kẻ tấn công sử dụng ký tự dot-dot-slash ("../") để truy cập vào các thư mục cha của thư mục hiện tại. Khi kẻ tấn công truy cập thành công vào thư mục cha, chúng có thể tiếp tục truy cập vào các thư mục, tập tin nhạy cảm trên máy chủ web (truy cập vào những tệp tin được cấp quyền cho user chạy ứng dụng web).

Ví dụ khai thác lỗ hổng Path Traversal:

https://example.com/?file_name=../../../../../etc/passwd



Hình 1- 8 Khai thác lỗ hổng Path Traversal đọc file nhạy cảm

Cách phòng chống tấn công Path Traversal:

- Xác thực dữ liệu đầu vào.
- Sử dụng danh sách các ký tự cho phép sử dụng.

1.2.3.5 File Inclusion

Lỗ hổng File Inclusion dựa trên chức năng include file được hỗ trợ bởi các ngôn ngữ lập trình.

Kẻ tấn công có thể lợi dụng để include những nội dung không muốn nhằm đánh cắp thông tin hệ thống, chèn các đoạn lệnh khiến server thực thi.

PHP: Include, require, include_once, require_once

JSP: <jsp: include…>, <c: import…>

Các dạng tấn công:

- LFI – Local File Inclusion: hacker có thể xem được rất nhiều thông tin của server như các file: passwd, php.ini, access_log,… Khi kết hợp với lỗ hổng upload file có thể dẫn đến RCE mà không cần thư mục có quyền thực thi).
- RFI – Remote File Inclusion: thực hiện RCE qua đường dẫn đến file mà hacker sử dụng để include.

Điều kiện xảy ra: Server cho phép include code từ bên ngoài.

Ví dụ: với cấu hình ngôn ngữ PHP php.ini:

- allow_url_include=On
- allow_url_open=On

Một ứng dụng web sử dụng chức năng include để import nội dung của file bất kỳ vào file hiện tại:

```
//File inclusion PHP
include('vendor/' . $_GET['file_name']);
```

Nếu kẻ tấn công truy cập và kết hợp với lỗ hổng Path Traversal để đọc file nhạy cảm trên hệ thống.

http://example.com?index.php?file_name=../../../../etc/passwd

Cách phòng chống:

- Kiểm tra chặt chẽ các file sử dụng để include.
- Hạn chế sử dụng include.
- Cấu hình không cho phép include từ bên ngoài nếu không cần thiết.

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

Lớp Bảo mật Mạng đảm nhận vai trò quan trọng trong việc đảm bảo sự an toàn của giao tiếp giữa máy chủ và máy khách. Để đạt được điều này, các thiết bị mạng như router, switch, tường lửa, và hệ thống ngăn chặn và phát hiện xâm nhập (IPS/IDS) cần được cấu hình và triển khai một cách an toàn và hiệu quả.

Lớp Bảo mật Máy chủ Web chịu trách nhiệm bảo vệ các thành phần và hệ thống con bên trong máy chủ, bao gồm hệ điều hành, cơ sở dữ liệu, phần mềm, và các tập tin khác. Điều này đòi hỏi các biện pháp bảo mật mạnh mẽ để ngăn chặn các mối đe dọa từ việc tấn công và xâm phạm.

Lớp Bảo mật Ứng dụng đảm bảo an toàn và bảo mật cho người dùng và dữ liệu của họ trong ứng dụng web. Các khía cạnh quan trọng của lớp này bao gồm quản lý quyền

truy cập, xác thực, cấu hình, và ghi nhật ký, nhằm ngăn chặn các mối đe dọa từ các vấn đề liên quan đến ứng dụng như đánh cắp thông tin cá nhân và truy cập trái phép.

Việc kết hợp chặt chẽ giữa các lớp này sẽ tạo nên một hệ thống bảo mật ứng dụng web toàn diện, đảm bảo rằng tất cả các khía cạnh của môi trường trực tuyến đều được bảo vệ hiệu quả và an toàn.

Triển khai các giải pháp bảo mật bảo vệ ứng dụng web

Thực hiện triển khai tường lửa ứng dụng web cùng các hệ thống phát hiện và ngăn chặn xâm nhập là một phương tiện quan trọng để tăng cường bảo mật cho ứng dụng web. Tường lửa ứng dụng web hay WAF, là một giải pháp thiết yếu nhằm đề phòng các lỗ hổng bảo mật tiềm ẩn. Được triển khai dưới dạng thiết bị phần cứng hoặc phần mềm trên máy chủ, WAF giữ vai trò theo dõi thông tin truyền tải qua giao thức http/https giữa trình duyệt của người dùng và máy chủ web.

WAF có khả năng thực hiện các chính sách bảo mật dựa trên việc phân tích các dấu hiệu tấn công, tuân thủ các giao thức tiêu chuẩn, và nhận diện lưu lượng truy cập ứng dụng web có sự bất thường. Điều này giúp ngăn chặn hiệu quả các mối đe dọa bảo mật như SQL injection và Cross-Site Scripting (XSS), đảm bảo an toàn cho dữ liệu và người dùng trong môi trường ứng dụng web.

WAF thường hoạt động thông qua một loạt các quy tắc được định nghĩa sẵn, hay còn gọi là chính sách. Chính sách này có thể dễ dàng thay đổi và cập nhật, do đó WAF có khả năng phản ứng nhanh đối với các lỗ hổng bảo mật mới.

Giống như WAF, các hệ thống phát hiện xâm nhập (IDS) và ngăn chặn xâm nhập (IPS) cũng có khả năng bảo vệ ứng dụng web khỏi các cuộc tấn công. Khác với WAF, IDS/IPS có khả năng phát hiện và ngăn chặn xâm nhập ở tầng mạng. Sự khác biệt giữa IDS/IPS đó là IDS chỉ có khả năng phát hiện tấn công, trong khi đó IPS có cả khả năng phát hiện và ngăn chặn. Cách hoạt động của IPS/IDS cũng khá tương đồng với WAF, cũng cần một bộ quy tắc để kiểm tra và ngăn chặn các giao thức mạng không hợp lệ. IPS/IDS thích hợp để thu thập thông tin, điều tra và ứng cứu sự cố xảy ra ở tầng mạng.

IPS/IDS và WAF thường được kết hợp với nhau để mang đến khả năng bảo mật tốt hơn cho các ứng dụng web.

Cấu hình, cập nhật phiên bản ứng dụng định kỳ

Một biện pháp cơ bản nhưng cực kỳ hữu hiệu trong việc bảo vệ ứng dụng web, đó là cấu hình đúng. Cấu hình ứng dụng web được hiểu như việc cấp quyền truy nhập, xác thực, trao quyền, và các cấu hình về thành phần mặc định, cách cập nhật hệ thống, cùng nhiều các thành phần khác. Một số cấu hình an toàn có thể kể đến như:

- Ngoại trừ các tài nguyên công cộng để tải trang web, chặn mọi truy cập đến các tài nguyên khác mà không có đúng quyền.
- Triển khai các cơ chế kiểm soát truy cập đến trang web.
- Tạo quyền tối thiểu cho những người dùng khác nhau đối với mỗi bản ghi, thay vì mọi người đều có thể chỉnh sửa và xóa bản ghi đó.
- Tắt chức năng duyệt thư mục theo mặc định, và ngăn chặn truy nhập đến các thư mục hệ thống và thư mục cấu hình/ cài đặt ứng dụng web.
- Cấu hình ghi nhật ký cho mỗi kết nối trong quá trình kết nối đến ứng dụng web.
- Các phiên làm việc nên được xóa bỏ khi người dùng đăng xuất.
- Các dữ liệu nhạy cảm nên được mã hóa bằng một thuật toán mã hóa mạnh trong cơ sở dữ liệu của ứng dụng web.
- Cấu hình giao thức HTTPS thay vì HTTP.
- Áp dụng các mẫu thiết kế an toàn.
- Các môi trường khác nhau (môi trường người dùng, môi trường kiểm thử) nên được cấu hình giống nhau, và đảm bảo an toàn giữa các môi trường đó.
- Loại bỏ các thành phần không được sử dụng ở ứng dụng web, các thư viện bên thứ ba, và thường xuyên cập nhật các thư viện cũng như thành phần nhằm đảm bảo các bản vá bảo mật luôn có sẵn.

- Đảm bảo quá trình ghi log được diễn ra đúng đủ và an toàn, phục vụ cho mục đích truy vết và ứng cứu khẩn cấp.
- Sử dụng chữ ký số đối với việc tải các tài nguyên bên thứ ba vào ứng dụng web, đảm bảo việc tải tài nguyên là đúng nguồn và không bị thay đổi.

Ngoài ra, rất nhiều ứng dụng được phát triển dựa trên các nền tảng được cung cấp bởi các hãng nổi tiếng và phát sinh lỗ hổng theo thời gian. Khi các lỗ hổng được tìm ra, các nhà cung cấp sẽ cung cấp các bản vá, chính vì vậy cần phải thực hiện cập nhật các bản vá để tránh bị ảnh hưởng bởi các lỗ hổng.

Không tin tưởng dữ liệu do người dùng cung cấp

Dữ liệu người dùng là một yếu tố không đáng tin cậy nhất, do đó cần được xác thực và tiến hành lọc bỏ các thành phần không hợp lệ, trước khi được máy chủ web xử lý. Việc lọc dữ liệu người dùng cần được triển khai trên cả máy khách và máy chủ, do việc chỉ đảm bảo dữ liệu đúng trên máy khách là không đủ để xác thực rằng dữ liệu sẽ an toàn khi xử lý trên máy chủ. Một ví dụ cơ bản cho trường hợp này, đó là kẻ tấn công có thể tạo form nhập liệu riêng, hoặc xóa các thành phần xác thực dữ liệu trên máy khách, tắt Javascript, hoặc phức tạp hơn, gửi trực tiếp yêu cầu đến máy chủ mà không thông qua bất cứ khâu xác thực nào.

Các khâu xác thực dữ liệu đầu vào có thể kể đến như: kiểm tra kích thước đầu vào, định dạng đầu vào. Ví dụ như, với dữ liệu yêu cầu là định dạng ngày, thì định dạng cần thiết sẽ là dd/mm/yyyy. Với dữ liệu có yêu cầu bắt buộc phải nhập, thì kích thước dữ liệu cần lớn hơn 0. Một số trường hợp đặc biệt có thể cần kiểm tra thêm nội dung và tính hợp lý của dữ liệu. Một số bộ lọc sẵn có có thể lọc được các dữ liệu không hợp lệ, dữ liệu được phân loại là dữ liệu chứa mã tấn công, để đảm bảo dữ liệu được xử lý là dữ liệu sạch và đúng. Một số bộ lọc khá nổi tiếng có thể kể đến như bộ lọc của OWASP, bộ lọc của CloudFlare hoặc một số bộ lọc công khai có trên các trang web quản lý mã nguồn (như Github, Gitlab…).

Phòng thủ theo chiều sâu

Đối với các hệ thống công nghệ thông tin nói chung và ứng dụng web nói riêng, phương pháp bảo mật hiệu quả các hệ thống là việc kết hợp nhiều lớp phòng vệ lại với nhau. Mỗi lớp bảo mật có tính năng tác dụng riêng và hỗ trợ cho nhau trong vấn đề đảm bảo an toàn tối đa cho ứng dụng web. Ngoài ra, các ứng dụng bảo mật web (như các phần mềm Acunetix, các hệ thống của CloudFlare, Microsoft..) cũng nên được cài đặt để đảm bảo tối đa cho ứng dụng web.

1.4 Kết chương

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

2.1.1.1 Khái niệm học máy

Định nghĩa về học máy: Arthur Samuel, một người Mỹ đi tiên phong trong lĩnh vực trò chơi máy tính và trí tuệ nhân tạo, đã đặt ra thuật ngữ "Machine Learning" vào năm 1959 khi còn ở IBM. Ông định nghĩa học máy là "lĩnh vực nghiên cứu cung cấp cho máy tính khả năng học hỏi mà không cần lập trình rõ ràng". Tuy nhiên, trên thực tế không có định nghĩa được chấp nhận rộng rãi cho học máy. Các tác giả khác nhau định nghĩa thuật ngữ khác nhau.

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.

Ví dụ, nếu chúng ta phát triển một mô hình học máy để dự đoán thành công của dự án phần mềm, chương trình có thể quan sát lịch sử các dự án trước đó. Số liệu về kích thước dự án, số lượng thành viên, và mức độ kinh nghiệm của đội ngũ phát triển được sử dụng để học từ những kết quả trước đó. Mô hình này có khả năng tự động điều chỉnh để ngày càng cải thiện dự đoán về xác suất thành công của dự án phần mềm dựa trên thông tin lịch sử.

Học máy là một nhánh nghiên cứu rất quan trọng của trí tuệ nhân tạo với khá nhiều ứng dụng thành công trong thực tế. Hiện nay, học máy là một trong những lĩnh vực phát triển mạnh nhất của trí tuệ nhân tạo. Mục tiêu của học máy nói chung là hiểu cấu trúc của dữ liệu và điều chỉnh dữ liệu đó thành các mô hình mà con người có thể hiểu và sử dụng được.

2.1.1.2 Phân loại học máy

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): Là dạng học máy trong đó cho trước tập dữ liệu huấn luyện dưới dạng các ví dụ cùng với giá trị đầu ra hay giá trị đích. Dựa trên dữ liệu huấn luyện, thuật toán học cần xây dựng mô hình hay hàm đích để dự đoán giá trị đầu ra (giá trị đích) cho các trường hợp mới.

- Nếu giá trị đầu ra là rời rạc thì học có giám sát được gọi là phân loại hay phân lớp (classification).
- Nếu đầu ra nhận giá trị liên tục, tức đầu ra là số thực, thì học có giám sát được gọi là hồi quy (regression). Trong phần tiếp theo, ta sẽ xem xét chi tiết hơn về học có giám sát.

Ví dụ:

Phân loại. Giả sử cho dữ liệu về một số loại ô tô con như trong bảng trên hình 1. Mỗi dòng trong bảng tương ứng với một loại xe cụ thể gồm hai thông số về xe là dung tích động cơ và loại xe. Ngoài ra, mỗi xe còn được xếp loại "cao cấp" hay "trung bình". Đây là dữ liệu huấn luyện tiêu biểu cho bài toán phân loại, trong đó mỗi ví dụ được gán một giá trị đích có thể nhận giá trị từ một trong hai giá trị "cao cấp" hoặc "trung bình". Trong trường hợp nói chung, số giá trị đích có thể nhiều hơn hai, nhưng giá trị đích vẫn là rời rạc và hữu hạn. Chẳng hạn, mỗi xe có thể xếp vào một trong ba loại "cao cấp", "trung bình", "bình dân" Nhiệm vụ của bài toán phân loại khi đó là tổng quát hóa trên dữ liệu huấn luyện đã cho. Từ đó xác định phân khúc xe cho các mẫu xe mới nếu biết dung tích động cơ và loại xe.

Bảng 2- 1 Dữ liệu huấn luyện cho bài toán phân loại

Dung tích động cơ Loại xe	Phân khúc
3200 Sedan	Cao cấp

2500 Sedan	Cao cấp
2500 SUV	Trung bình
2000 Sedan	Trung bình
3500 SUV	Cao cấp
1800 Sedan	Trung bình

Học không giám sát (Unsupervised learning): Mô hình được huấn luyện trên một tập dữ liệu không có nhãn. Mục tiêu là tìm kiếm cấu trúc ẩn trong dữ liệu, chẳng hạn như phân cụm hay giảm chiều dữ liệu. Học không giám sát sẽ không đưa ra kết quả cụ thể cho câu trả lời có hay không, hoặc một nhãn cụ thể nào đó. Các bài toán thường dùng cho học không giám sát là:

- Clustering (phân cụm): bài toán phân loại với tập dữ liệu X, tìm ra điểm tương đồng/ sự liên quan dữ liệu để phân nhóm. Ví dụ như, việc phân loại gen thành các nhóm về màu da, màu mắt, màu tóc… giống nhau
- Non-clustering (không phân cụm): bài toán tìm cấu trúc hoặc quy luật, dựa trên một dữ liệu có trước. Một ví dụ điển hình cho bài toán này đó là 'Cocktail Party Algorithm' - việc nhận dạng giọng nói/ âm thanh trong môi trường tạp âm.

Ví dụ: Hãy xem xét các dữ liệu sau đây liên quan đến bệnh nhân vào phòng khám. Dữ liệu bao gồm giới tính và độ tuổi của bệnh nhân và mỗi bệnh nhân được dán nhãn là "khỏe mạnh" hoặc "ốm yếu".

Bảng 2- 2 Bảng dữ liệu về giới tính, tuổi của bệnh nhân

Giới tính	Tuổi	Nhãn
M	48	sick
M	47	sick
F	18	healthy
M	23	sick
F	32	healthy
M	55	healthy
M	69	healthy

Học không giám sát tương tự như cách con người học, trong đó phương pháp này được sử dụng để nhận biết rằng các đối tượng hoặc sự kiện cụ thể thuộc cùng một nhóm, thường bằng cách quan sát sự tương đồng giữa chúng. Một số hệ thống mà bạn có thể gặp trên internet sử dụng loại học này để tự động hóa chiến lược tiếp thị.

Học nửa giám sát (Semi supervised learning): học nửa giám sát sẽ sử dụng những dữ liệu chỉ có một phần được gắn nhãn, và phần còn lại chưa được gắn nhãn. Phương thức này sẽ kết hợp học có giám sát và học không giám sát để đưa ra kết quả dự đoán cuối cùng.

Học tăng cường (reinforcement learning): Trong học tăng cường, hệ thống không nhận được kinh nghiệm dưới dạng đầu vào/đầu ra cho mỗi trạng thái hoặc hành động một cách trực tiếp. Thay vào đó, nó nhận được một giá trị khuyến khích (reward) là kết quả của một chuỗi hành động cụ thể. Thuật toán trong học tăng cường nhằm mục đích học cách hành động để cực đại hóa giá trị khuyến khích.

Ví dụ, trong một phòng lab với các chướng ngại vật, robot nhận dạng trạng thái của mình dựa trên tọa độ và thông tin từ cảm biến. Hành động của robot bao gồm di chuyển, quẹo trái, quẹo phải và dừng lại. Robot nhận được giá trị khuyến khích dựa trên khoảng cách đến điểm đích. Trong quá trình học, robot thực hiện các hành động, nhận được phản hồi, và điều chỉnh chiến lược để tối ưu hóa việc đạt được mục tiêu. Thông qua nhiều thử nghiệm, robot ngày càng cải thiện khả năng tự điều hướng và di chuyển một cách hiệu quả trong môi trường không gian, mặc dù không biết trước được bản đồ chi tiết của nó.

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.

Mô hình học sâu thường được xây dựng dựa trên các kiến trúc mạng nơ-ron sâu, mô phỏng cách mà não người xử lý thông tin. Các mô hình này có khả năng học từ dữ liệu thông qua nhiều lớp nơ-ron, tăng cường khả năng biểu diễn và hiểu các đặc trưng phức tạp của dữ liệu.

Ứng dụng của học sâu rộng rãi, từ nhận diện giọng nói, nhận diện hình ảnh, dịch ngôn ngữ tự nhiên đến tự động lái xe. Học sâu đã đưa ra những tiến bộ lớn trong nhiều lĩnh vực, mở ra những cánh cửa mới cho sự đổi mới và cải tiến công nghệ. Nó không chỉ là một công cụ mạnh mẽ cho các nhà nghiên cứu, mà còn là nguồn động viên lớn đối với những người mong muốn khám phá và ứng dụng sức mạnh của máy học trong thế giới thực.

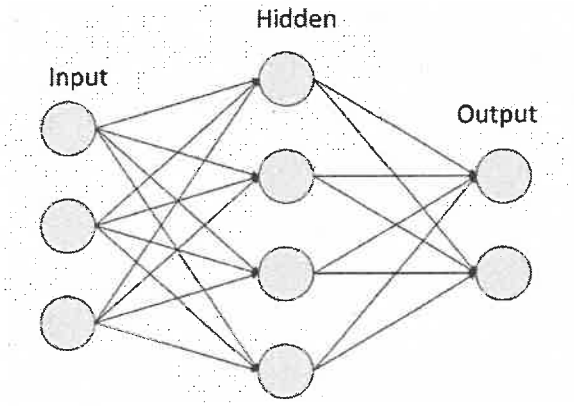
Giới thiệu mạng nơ ron [4]

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.

Quá trình học của ANN thường dựa trên giả định "học giám sát", nơi mô hình được đào tạo bằng cách so sánh đầu ra dự đoán với đầu ra mong muốn. Các thuật toán như lan truyền ngược (backpropagation) thường được sử dụng để điều chỉnh trọng số và cải thiện hiệu suất của mạng.

Kiến trúc chung của một mạng neuron nhân tạo gồm 3 thành phần đó là: Input Layer, Hidden Layer và Output Layer



Hình 2- 1 Kiến trúc mạng neuron

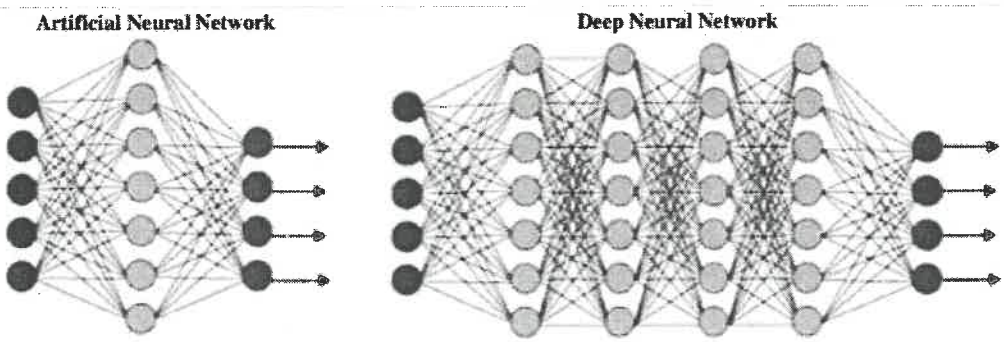
Trong đó, lớp ẩn (Hidden Layer) gồm các nơ ron nhận dữ liệu input từ các nơ ron ở lớp (Layer) trước đó và chuyển đổi các input này cho các lớp xử lý tiếp theo. Trong một ANN có thể có nhiều lớp ẩn.

Trong đó các Processing Elements (PE) của ANN gọi là nơ ron, mỗi nơ ron nhận các dữ liệu vào (Inputs) xử lý chúng và cho ra một kết quả (Output) duy nhất. Kết quả xử lý của một nơ ron có thể làm Input cho các nơ ron khác.

2.1.3 Một số phương pháp học sâu

2.1.3.1 Mạng nơ ron sâu (Deep Neural Network-DNN)

Mạng nơ ron sâu (Deep Neural Network - DNN) là một dạng cụ thể của lĩnh vực học sâu. Mạng nơ ron sâu là một mạng nơ ron nhân tạo nhưng có kiến trúc phức tạp và "sâu" hơn nhiều so với kiến trúc của mạng nơ ron truyền thống. Mạng DNN tận dụng thành phần của mạng nơ ron nhân tạo ANN. Nghĩa là nó có số nút trong mỗi lớp và số lớp ẩn lớn hơn rất nhiều và cách thức hoạt động của nó phức tạp hơn so với kiến trúc mạng nơ ron truyền thống.



Hình 2- 2 Tương quan mạng ANN và DNN

DNN cho phép mô hình thực hiện với độ chính xác cao hơn. Chúng cho phép mô hình thao tác với tập dữ liệu và trả về một kết quả với độ chính xác được cải thiện so với mạng nơ ron truyền thống.

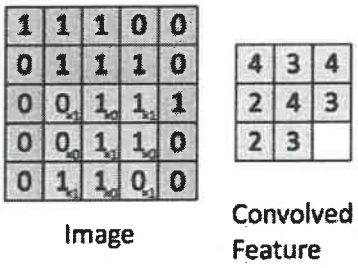
2.1.3.2 Mạng nơ ron tích chập (Convolutional Neural Network) [7]

CNN là từ viết tắt của cụm Convolutional Neural Network hay là mạng nơ ron tích chập. Đây là mô hình vô cùng tiên tiến được áp dụng nhiều trong lĩnh vực học sâu Deep learning. Mạng CNN cho phép người dùng xây dựng những hệ thống phân loại và dự đoán với độ chính xác cực cao. Hiện nay, mạng CNN được ứng dụng nhiều hơn trong xử lý ảnh, cụ thể là nhận diện đối tượng trong ảnh.

Convolution (tích chập)

Tích chập được sử dụng đầu tiên trong xử lý tín hiệu số (Signal processing). Nhờ vào nguyên lý biến đổi thông tin, các nhà khoa học đã áp dụng kĩ thuật này vào xử lý ảnh và video số.

Để dễ hình dung, ta có thể xem tích chập như một cửa sổ trượt (sliding window) áp đặt lên một ma trận. Cơ chế của tích chập qua hình minh họa bên dưới.

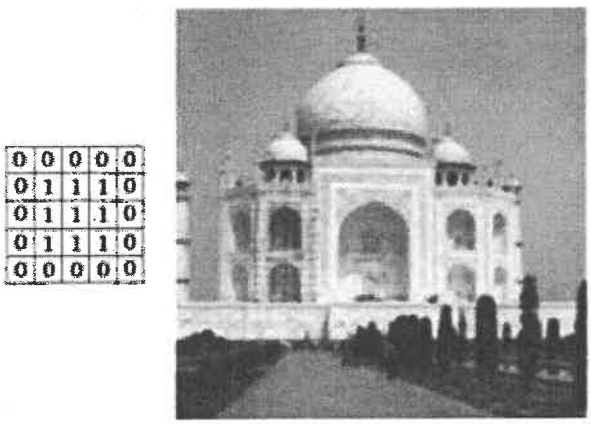


Hình 2- 3 Minh họa cơ chế tích chập

Ma trận bên trái là một bức ảnh đen trắng. Mỗi giá trị của ma trận tương đương với một điểm ảnh (pixel), 0 là màu đen, 1 là màu trắng (nếu là ảnh grayscale thì giá trị biến thiên từ 0 đến 255).

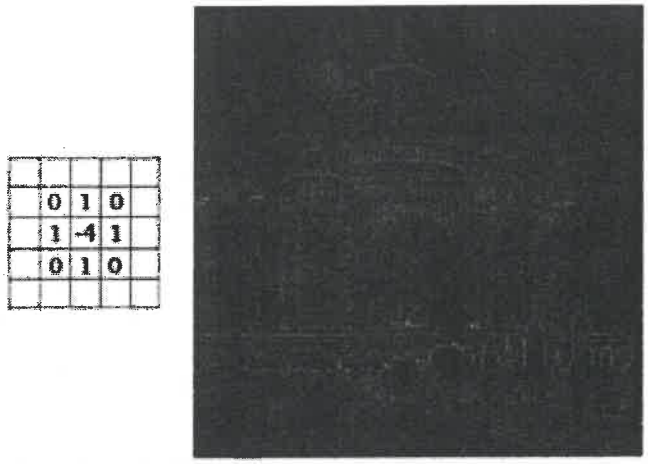
Sliding window còn có tên gọi là kernel, filter hay feature detector. Ở đây, ta dùng một ma trận filter 3×3 nhân từng thành phần tương ứng (element-wise) với ma trận ảnh bên trái. Giá trị đầu ra do tích của các thành phần này cộng lại. Kết quả của tích chập là một ma trận (convoled feature) sinh ra từ việc trượt ma trận filter và thực hiện tích chập cùng lúc lên toàn bộ ma trận ảnh bên trái. Dưới đây là một vài ví dụ của phép toán tích chập.

- Ta có thể làm mờ bức ảnh ban đầu bằng cách lấy giá trị trung bình của các điểm ảnh xung quanh cho vị trí điểm ảnh trung tâm.



Hình 2- 4 Làm mờ bức ảnh

- Ngoài ra, ta có thể phát hiện biên cạnh bằng cách tính vi phân (độ dị biệt) giữa các điểm ảnh lân cận.



Hình 2- 5 Phát hiện các cạnh trong hình ảnh

Mô hình mạng neural tích chập

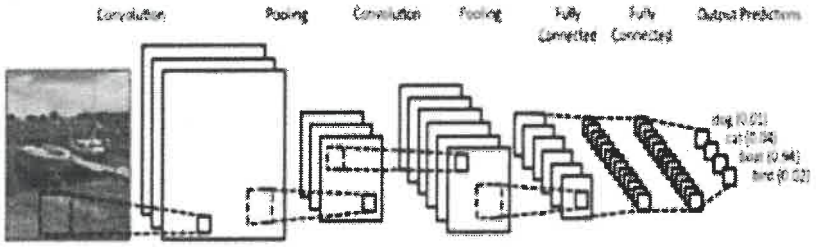
Bây giờ, Chúng ta đã biết thế nào là tích chập. Vậy CNNs là gì? CNNs chỉ đơn giản gồm một vài lớp của tích chập kết hợp với các hàm kích hoạt phi tuyến (nonlinear activation function) nhờ ReLU hay tanh để tạo ra thông tin trừu tượng hơn (abstract/higher-level) cho các layer tiếp theo.

Trong mô hình CNNs các layer liên kết được với nhau thông qua cơ chế convolution. Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Nghĩa là mỗi nơ-ron ở layer tiếp theo sinh ra từ filter áp đặt lên một vùng ảnh cục bộ của nơ-ron layer trước đó.

Mỗi layer nhờ vậy được áp đặt các filter khác nhau, thông thường có vài trăm đến vài nghìn filter như vậy. Một số layer khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu). Tuy nhiên, ta sẽ không đi sâu vào khái niệm của các layer này.

Trong suốt quá trình huấn luyện, CNNs sẽ tự động học được các thông số cho các filter. Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối qua cho

các filter tương ứng theo thứ tự raw pixel > edges > shapes > facial > high-level features. Layer cuối cùng được dùng để phân lớp ảnh.



CNNs có tính bất biến và tính kết hợp cục bộ (Location Invariance and Compositionality). Với cùng một đối tượng, nếu đối tượng này đợợc chiếu theo các gốc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể. Pooling layer sẽ cho bạn tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling).

Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter. Đó là lý do tại sao CNNs cho ra mô hình với độ chính xác rất cao. Cũng giống như cách con người nhận biết các vật thể trong tự nhiên. Ta phân biệt được một con chó với một con mèo nhờ vào các đặc trong từ mức độ thấp (có 4 chân, có đuôi) đến mức độ cao (dáng đi, hình thể, màu lông) [2].

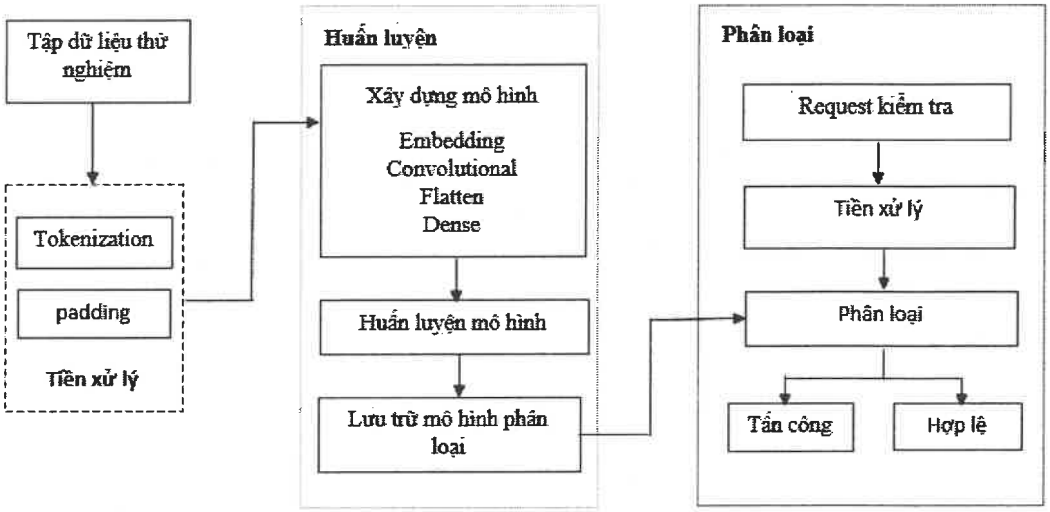
2. 2 Phát hiện tấn công ứng dụng web dựa trên mô hình học sâu CNN

CNN sử dụng các lớp tích chập để áp dụng các bộ lọc nhỏ lên dữ liệu, giúp chúng có thể nhận biết các mẫu cục bộ trong dữ liệu mà không phụ thuộc vào vị trí cụ thể. Điều này rất hữu ích trong việc phát hiện các biểu hiện tấn công ứng dụng web mà có thể xuất hiện ở bất kỳ vị trí nào trong dữ liệu lưu lượng mạng.

Đề án lựa chọn sử dụng mạng học sâu CNN để xây dựng và thử nghiệm mô hình phát hiện tấn công web sử dụng học sâu là do CNN đã được sử dụng rất rộng rãi và hiệu quả cho các bài toán, như phân loại ảnh, phân loại văn bản. Ngoài ra, mô hình

CNN có tốc độ xử lý nhanh, yêu cầu tài nguyên tính toán thấp hơn các mô hình học sâu khác, như LSTM, hoặc BiLSTM, phù hợp với mô hình phát hiện tấn công web cần xử lý lượng dữ liệu rất lớn.

2.2.1 Giới thiệu mô hình



Hình 2- 6 Mô hình phát hiện tấn công ứng dụng web sử dụng học sâu CNN

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.

Dữ liệu từ các HTTP request sẽ trải qua các bước tiền xử lý bao gồm tokenization và padding. Sau đó, các request sẽ được đưa qua chuỗi các lớp xử lý nhúng, tích chập, phẳng hóa, và cuối cùng là một lớp kết nối đầy đủ ở giai đoạn huấn luyện. Kết quả từ quá trình huấn luyện sẽ được lưu trữ để sử dụng cho việc phân loại tiếp theo.

Trong quá trình phần loại, request đầu vào cũng sẽ được tiền xử lý để phù hợp với mô hình học sâu. Sau khi được phân loại request hợp lệ sẽ được tiếp tục xử lý, request tấn công sẽ bị chặn và trả lại thông báo cho người dùng.

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.

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

Hình 2- 7 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.

Từ các giá trị TP, TN, FP, FN, ta có thể tính toán một số độ đo hiệu quả của mô hình phân loại như sau:

Chỉ số Precision được xác định bằng số request tấn công được phân loại chính xác là tấn công chia cho tổng của số request được phân loại chính xác và số request hợp lệ được phân loại là các request tấn công.

$$\text{Precision} = \frac{TP}{TP+FP} * 100\%$$

Recall đại diện cho số request tấn công được phân loại chính xác chia cho tổng của số request tấn công được phân loại chính xác và số request hợp lệ được phân loại là hợp lệ.

$$\text{Recall} = \frac{TP}{TP+FN} * 100\%$$

F1-score cung cấp đánh giá tổng thể về hiệu suất của mô hình. Công thức tính F1-score như sau:

$$\text{F1-score} = 2 * \frac{Precision*Recall}{Precision+Recall} * 100\%$$

Chỉ số loss là một phép đo đánh giá sự khác biệt giữa giá trị dự đoán của mô hình và giá trị thực tế trong dữ liệu huấn luyện.

Ngoài ra tập dữ liệu dùng để validation sử dụng các giá trị val_recall, val_loss, val_precision để đánh giá hiệu quả.

2. 3 Kết chương

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

Trong chương 3, mô hình phát hiện tấn công web sử dụng học sâu CNN sẽ được cài đặt và thử nghiệm để đưa ra các kết quả của mô hình học sâu. Các phép tính toán sẽ chỉ ra ưu nhược điểm của mô hình, từ đó có hướng đi mở rộng bài toán trong tương lai.

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, 12.130 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 gán nhãn "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--	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 công được gán nhãn "sqli", "xss", "path-traversal"

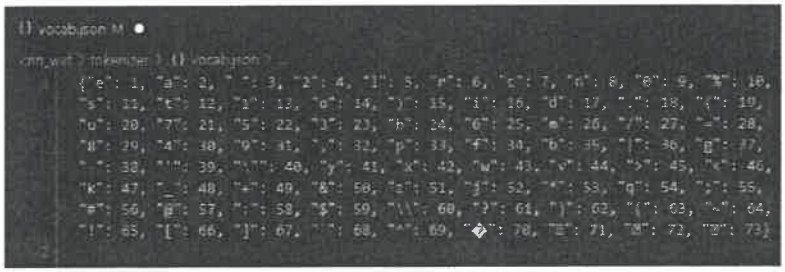
Tập dữ liệu sẽ được chia thành hai tập dữ liệu con là tập Train và tập Validation, trong đó có 75% dữ liệu sử dụng cho tập Train và 25% dữ liệu được sử dụng cho tập Validation.

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.

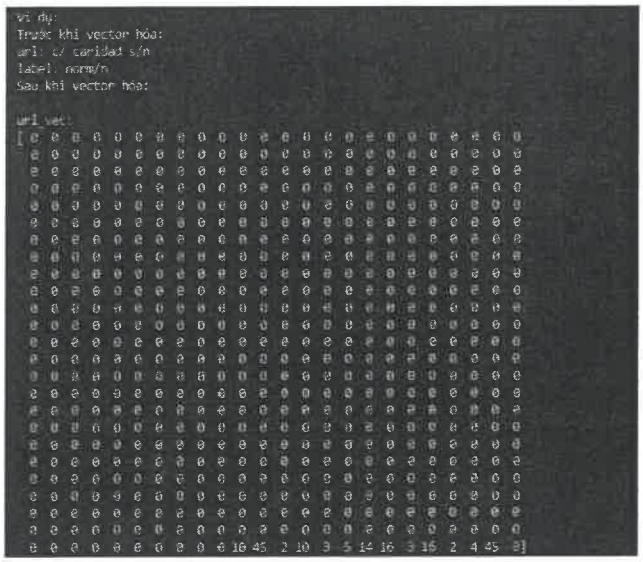
Công cụ Tokenizer từ thư viện keras.preprocessing.text được sử dụng để thực hiện tokenization.

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.



Hình 3- 4 Ví dụ payload được vector hóa

3.3 Huấn luyện và kiểm tra

3.3.1 Môi trường thử nghiệm

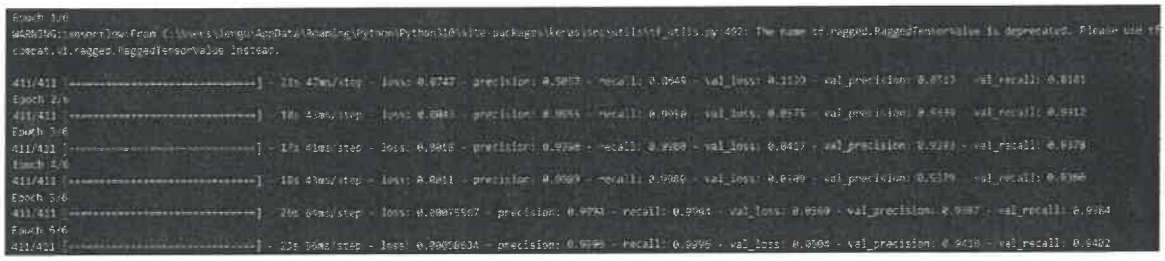
Môi trường thử nghiệm được mô tả trong bảng dưới đây:

Bảng 3- 1 Môi trường thử nghiệm mô hình học sâu CNN phát hiện tấn công web

Thông số	Mô tả
CPU	11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz
RAM	16.0 GB
Hệ điều hành	Windows 11 64GB
Platform	Visual Studio Code
Ngôn ngữ lập trình	Python 3.10.0

3.3.2 Kết quả và nhận xét

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



Hình 3- 5 Quá trình huấn luyện dữ liệu

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- 2 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.

Tiếp theo, chúng ta sẽ so sánh kết quả này với hiệu suất của phương pháp học máy truyền thống để có góc nhìn tổng quan về hai phương pháp.

Bảng 3- 3 So sánh hiệu suất phát hiện tấn công web dựa trên CNN trong đề án và dựa trên học máy truyền thống trong [3]

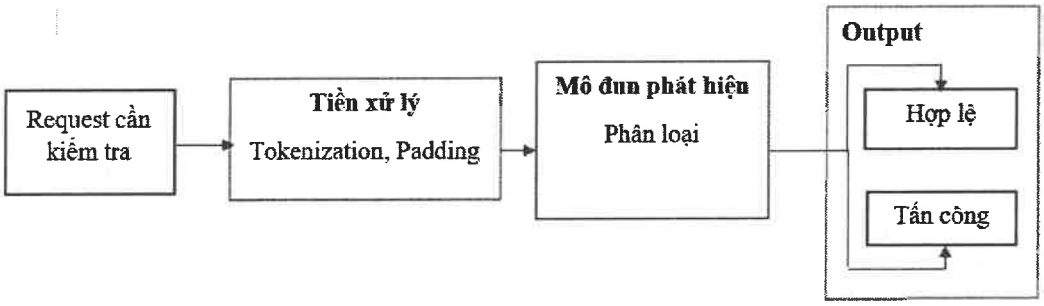
Thuật toán	Precision	Recall	F1-score
SVM với mô hình TF-IDF	0.98	0.978	0.978
SVM với mô hình BoW	0.99	0.978	0.984
Decision Tree với mô hình BoW	0.974	0.982	0.978
Decision Tree với mô hình TF-IDF	0.968	0.974	0.97
Học sâu CNN trong đề án	**0.9996**	**0.9996**	**0.9996**

Thông qua bảng 3-3, kết quả cho thấy các thuật toán học máy cũng đạt được hiệu suất cao trong việc phát hiện tấn công ứng dụng web. Tuy nhiên, điểm đáng chú ý là trong học sâu, đặc biệt là mô hình CNN, đã thể hiện sự ấn tượng với khả năng tự động học và trích xuất đặc trưng từ dữ liệu mà không cần bước trích xuất đặc trưng độc lập. Trong khi đó, học máy thường đòi hỏi một quá trình trích xuất đặc trưng riêng biệt từ dữ liệu ban đầu. Sự khác biệt này thúc đẩy sự quan tâm ngày càng tăng trong việc nghiên cứu và ứng dụng của học sâu trong việc phát hiện và ngăn chặn các cuộc tấn công vào 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- 6 Sơ đồ mô hình phát hiện tấn công ứng dụng web

Hình 3-6 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 trên 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.

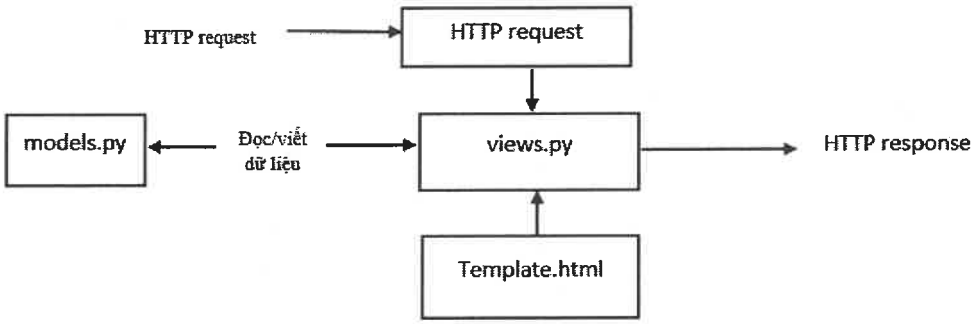
Django là một framework lập trình web bậc cao rất phổ biến và đầy đủ tính năng. Nó cho phép phát triển nhanh chóng các website và tối ưu hoá việc bảo trì chúng. Được xây dựng bởi các kỹ sư giàu kinh nghiệm, Django giúp giải quyết nhiều vấn đề trong việc phát triển web, từ đó có thể tập trung vào công việc viết ứng dụng.

Django Python sử dụng mô hình MVT (Model-View-Template) thay vì mô hình MVC (Model-View-Controller). Đây là một mô hình bao gồm mã HTML với Django Template Language.

- Model (M) là lớp có chức năng truy cập và lưu trữ dữ liệu: Từ cách thức truy cập, phương thức dữ liệu, cho đến mối quan hệ các dữ liệu.

- View (V) là lớp chứa các logic, giúp truy cập dữ liệu qua Model và truyền đến Template tương ứng.
- Template (T) là lớp hiển thị. Nó lưu trữ những gì liên quan đến việc hiển thị dữ liệu trên web hoặc các nền tảng khác.

Các ứng dụng web Django thường nhóm mã xử lý cho từng bước này vào các tệp riêng biệt:



Hình 3- 7 Cơ chế hoạt động của Django

- URLs: Mặc dù có thể xử lý các yêu cầu từ mọi URL thông qua một hàm duy nhất, nhưng việc viết một hàm xem (view) riêng để xử lý mỗi tài nguyên là cách quản lý tối ưu hơn. Một bộ URL được sử dụng để định hướng các yêu cầu HTTP đến chức năng view thích hợp dựa trên URL yêu cầu. Bộ URL cũng có thể phù hợp với một chuỗi ký tự hoặc số cụ thể có trong URL và chuyển chúng đến một view dưới dạng dữ liệu.
- View: Một hàm xem (view) là một hàm xử lý yêu cầu. View nhận yêu cầu HTTP và trả về phản hồi HTTP. Hàm view truy cập dữ liệu cần thiết để đáp ứng các yêu cầu thông qua các mô hình (model) và cung cấp dữ liệu trả về cho các mẫu (template).
- Model: Mô hình (model) là các class trong Python dùng để tạo các cấu trúc dữ liệu của ứng dụng và cung cấp các phương pháp để quản lý (thêm, sửa, xóa) và truy cập vào cơ sở dữ liệu.

- Template: Mẫu (template) là một file giúp định dạng cấu trúc hoặc layout của file (ví dụ một trang HTML) với các vị trí ô đã được chỉ định để trình bày một nội dung cụ thể. Một hàm view có thể tự động tạo một trang HTML bằng cách sử dụng HTML template với dữ liệu được thêm vào từ model. Một template có thể sử dụng để tạo ra một cấu trúc cho bất kỳ loại file nào (không chỉ HTML).

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.

Ứng dụng web có cấu trúc như hình dưới đây:

Hình 3- 8 Cấu trúc ứng dụng 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.

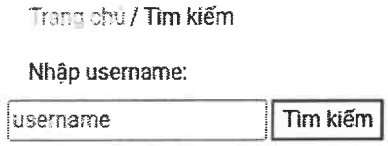
- Tất cả chức năng sẽ được tích hợp mô hình phát hiện tấn công ứng dụng web để có thể phát hiện tấn công nếu request chứa payload khai thác lỗ hổng SQL Injection, Cross-site Scriptting và Path-Traversal. Đây đều là các lỗ hổng phổ biến và đã được đề cập ở chương đầu tiên của đề án.

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]

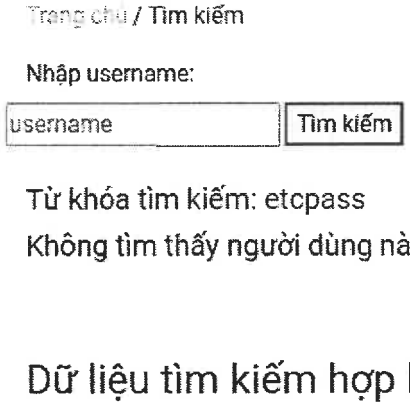
Kết quả thử nghiệm các mẫu thử cho kết quả dự đoán đúng 10/10 mẫu, kết quả cụ thể như sau:

Hình 3- 9 Mẫu thử "khoaln" cho kết quả hợp lệ

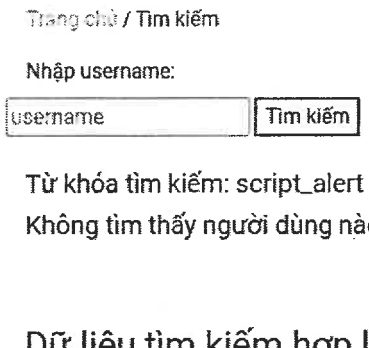


Hình 3- 10 Mẫu thử "admin_union_select" cho kết quả hợp lệ

Hình 3- 11 Mẫu thử "user123" cho kết quả hợp lệ



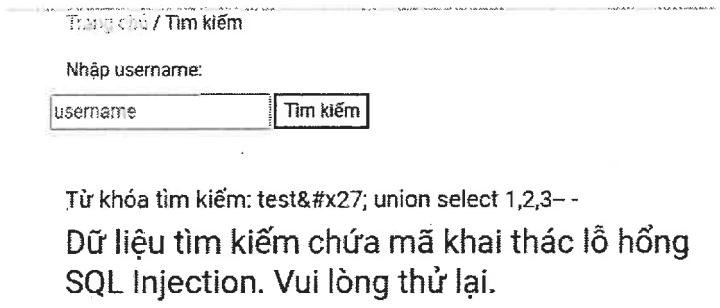
Hình 3- 12 Mẫu thử "etcpass" cho kết quả hợp lệ



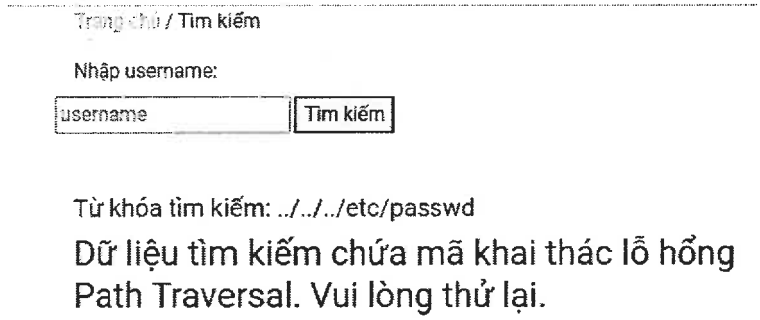
Hình 3- 13 Mẫu thử "script_alert" cho kết quả hợp lệ

Hình 3- 14 Mẫu thử "admin' or 1=1-- -" cho kết quả SQL Injection

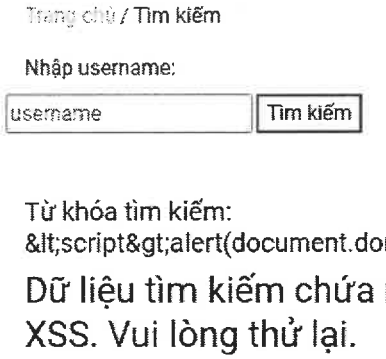
Hình 3- 15 Mẫu thử "admin'><script>alert(1)</script>" cho kết quả XSS



Hình 3- 16 Mẫu thử "test' union select 1,2,3-- -" cho kết quả SQL Injection



Hình 3- 17 Mẫu thử "../../../etc/passwd" cho kết quả Path Traversal



Hình 3- 18 *Mẫu thử* "<script>alert(document.domain)</script>" cho kết quả XSS

3.5 Kết chương

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.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] Hoàng Xuân Dậu, *Bài Giảng An Toàn Ứng Dụng Web và Cơ Sở Dữ Liệu*, Hà Nội: Học viện Công nghệ Bưu chính Viễn Thông, 2017.

[2] Từ Minh Phương, *Giáo trình nhập môn trí tuệ nhân tạo*, Hà Nội: Học viện Công nghệ Bưu chính Viễn thông, 2014.

[3] Nguyễn Ngọc Đoàn, NGHIÊN CỨU CÁC PHƯƠNG PHÁP PHÁT HIỆN TẤN CÔNG WEB DỰA TRÊN HỌC MÁY, Học Viện Công Nghệ Bưu Chính Viễn Thông, Hà Nội, 2022.

[4] "[Cẩm nang AI] Artificial Neural Network là gì? Cấu trúc, cách hoạt động và ứng dụng của mô hình này," 2022. [Online]. Available: https://viettelidc.com.vn/tin-tuc/cam-nang-ai-artificial-neural-network-la-gi-cau-truc-cach-hoat-dong-va-ung-dung-cua-mo-hinh-nay. [Accessed 02 12 2023].

[5] "Django – Framework phát triển web," Django – Framework phát triển web, 2023. [Online]. Available: https://sec.vnpt.vn/2023/03/django_framework_phat_trien_web/. [Accessed 20 12 2023].

[6] "Mô Hình Client Server là gì? Tìm hiểu về mô hình mạng máy khách – máy chủ," 2021. [Online]. Available: https://vietnix.vn/mo-hinh-client-server/. [Accessed 10 12 2023].

[7] "Thuật toán CNN là gì? Cấu trúc mạng Convolutional Neural Network," [Online]. Available: https://topdev.vn/blog/thuat-toan-cnn-convolutional-neural-network/. [Accessed 10 12 2023].

[8] J. H. An, Z. Wang and I. Joe, "A CNN-based automatic vulnerability detection" EURASIP Journal on Wireless Communications and Networking, 2023.

[9] H. Yan, L. Feng, Y. Yu, W. Liao, L. Feng, J. Zhang, D. Liu, Y. Zou, C. Liu, L. Qu and X. Zhang, "Cross-site scripting attack detection based on a modified convolution neural network" Computational Neuroscience, vol. 13, pp. 5-8, 2022.

[10] R. Lu, S. Wang and Y. Li, "Research on SQL Injection Detection Model Based on CNN" 2021 International Conference on Intelligent Computing, Automation and Applications (ICAA), Nanjing, China, pp. 111-114, 2021.

[11] S. Benke, K. Abhijit, P. Sanket, M. Vaibhav and T. Siddhant, "NEURAL NETWORK-BASED WEB SECURITY FIREWALL" International Research Journal of Modernization in Engineering Technology and Science, 2023.

[12] "Detecting Malicious Requests with Keras & Tensorflow," 2017. [Online]. Available: https://medium.com/slalom-build/detecting-malicious-requests-with-keras-tensorflow-5d5db06b4f28. [Accessed 03 01 2024].

[13] "HTTP DATASET CSIC 2010," 2010. [Online]. Available: https://www.tic.itefi.csic.es/dataset/. [Accessed 12 2023].

[14] "Http Params Dataset," [Online]. Available: https://github.com/Morzeux/HttpParamsDataset/blob/master/payload_train.csv. [Accessed 20 12 2023].

[15] "Introduction to deep neural networks" 7 2023. [Online]. Available: https://www.datacamp.com/tutorial/introduction-to-deep-neural-networks. [Accessed 12 1 2024].

[16] "OWASP Top Ten," 2021. [Online]. Available: https://owasp.org/www-project-top-ten/. [Accessed 20 12 2023].

BẢN CAM ĐOAN

Tôi cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung đề án qua phần mềm kiemtratailieu.vn một cách trung thực và đạt kết quả mức độ tương đồng 14% toàn bộ nội dung đề án. Bản đề án kiểm tra qua phần mềm là bản cứng đề án đã nộp để bảo vệ trước hội đồng. Nếu sai tôi xin chịu các hình thức kỷ luật theo quy định hiện hành của Học viện.

Hà Nội, ngày 01 tháng 03 năm 2024

HỌC VIÊN CAO HỌC

LÊ NGỌC KHOA

KiểmTraTàiLiệu

BÁO CÁO KIỂM TRA TRÙNG LẶP

Thông tin tài liệu

Tên tài liệu:	Đề án_Le_Ngoc_Khoa_final
Tác giả:	Lê Ngọc Khoa
Điểm trùng lặp:	14
Thời gian tải lên:	14:10 29/02/2024
Thời gian sinh báo cáo:	14:11 29/02/2024
Các trang kiểm tra:	63/63 trang

Kết quả kiểm tra trùng lặp

Có 14% nội dung trùng lặp

Có 86% nội dung không trùng lặp

Có 0% nội dung người dùng loại trừ

Có 0% nội dung hệ thống bỏ qua

Nguồn trùng lặp tiêu biểu

123docz.net tailieu.vn aptech.fpt.edu.vn

Lê Khoa

Lê Ngọc Khoa

Người Hướng Dẫn

PGS.TS Hoàng Xuân Dậu

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BIÊN BẢN
HỌP HỘI ĐỒNG CHẤM ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Căn cứ quyết định số Quyết định số 289/QĐ-HV ngày 06 tháng 03 năm 2024 của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ. Hội đồng đã họp vào hồi ..10.. giờ ..00.. phút, ngày 20 tháng 03 năm 2024 tại Học viện Công nghệ Bưu chính Viễn thông để chấm đề án tốt nghiệp thạc sĩ cho:

Học viên: **Lê Ngọc Khoa**

Tên đề án tốt nghiệp: **Nghiên cứu phương pháp phát hiện tấn công Web dựa trên mô hình học sâu CNN**

Chuyên ngành: **Khoa học máy tính** Mã số: **8.48.01.01**

Các thành viên của Hội đồng chấm đề án tốt nghiệp có mặt: ..05../ 05

TT	HỌ VÀ TÊN	TRÁCH NHIỆM TRONG HĐ	GHI CHÚ
1	**TS. Nguyễn Duy Phương**	Chủ tịch	
2	**TS. Đào Thị Thúy Quỳnh**	Thư ký	
3	**TS. Lê Quốc Hưng**	Phản biện 1	
4	**PGS.TS. Đỗ Trung Tuấn**	Phản biện 2	
5	**PGS.TS. Nguyễn Mạnh Hùng**	Uỷ viên	

Các nội dung thực hiện:

1. Chủ tịch Hội đồng điều khiển buổi họp. Công bố quyết định của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ.
2. Người hướng dẫn khoa học hoặc thư ký đọc lý lịch khoa học và các điều kiện bảo vệ đề án tốt nghiệp của học viên (Có bản lý lịch khoa học và kết quả các môn học cao học của học viên kèm theo).
3. Học viên trình bày tóm tắt đề án tốt nghiệp.
4. Phản biện 1 đọc nhận xét (có văn bản kèm theo)
5. Phản biện 2 đọc nhận xét (có văn bản kèm theo)
6. Các câu hỏi của thành viên Hội đồng:

1) Đóng góp mới của học viên trong mô hình đề xuất?

2) Thay có xử lý các hình ảnh, có thể sử dụng p.p học thông thường đối với bài toán đặt ra không?

7. Trả lời của học viên:

- Các phương pháp sẽ được áp dụng vào các câu hỏi nghiên cứu CSDL công trong mô hình học sâu CNN cho những kết quả tốt hơn

- Đóng góp mới của học viên chưa có, trong tương lai học viên sẽ tiếp tục nghiên cứu và bổ sung

8. Thư ký đọc nhận xét về quá trình thực hiện đề án tốt nghiệp của học viên *(có văn bản kèm theo).*

9. Hội đồng họp riêng:

- Bầu Ban kiểm phiếu:
  1. Trưởng Ban kiểm phiếu: PGS.TS. Đỗ Trung Tuấn
  2. Uỷ viên Ban kiểm phiếu: TS. Đào Thị Thúy Quỳnh
  3. Uỷ viên Ban kiểm phiếu: TS. Lê Quốc Hùng
- Hội đồng chấm đề án tốt nghiệp bằng bỏ phiếu kín.
- Ban kiểm phiếu làm việc:
- Trưởng Ban kiểm phiếu báo cáo kết quả kiểm phiếu *(có Biên bản họp Ban kiểm phiếu kèm theo)*
- **Điểm trung bình của đề án tốt nghiệp:** 8,0

Kết luận:

1. Các nội dung cần chỉnh sửa, hoàn thiện sau bảo vệ đề án tốt nghiệp:

- Chỉnh sửa một số lỗi chính tả

- Xem xét bổ sung thêm công so sánh thực nghiệm với một số phương pháp phát hiện tấn công sử dụng học máy

2. Đề nghị Học viện công nhận (hoặc không) và cấp bằng (hoặc không) thạc sĩ cho học viên: Đề nghị Học viện công nhận và cấp bằng Th.S cho học viên

3. Đề án tốt nghiệp có thể phát triển thành đề tài nghiên cứu cho NCS: không

Buổi làm việc kết thúc vào 10h45 cùng ngày.

Chủ tịch	**Thư ký**
TS. Nguyễn Duy Phương	**TS. Đào Thị Thúy Quỳnh**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

BẢN NHẬN XÉT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Tên đề tài đề án tốt nghiệp: NGHIÊN CỨU PHÁT HIỆN TẤN CÔNG WEB DỰA TRÊN MÔ HÌNH HỌC SÂU CNN

Chuyên ngành: Khoa học máy tính

Mã chuyên ngành: 8.48.01.01

Họ và tên học viên: LÊ NGỌC KHOA

Họ và tên người nhận xét: Lê Quốc Hưng

Học hàm, học vị: Tiến sĩ

Chuyên ngành: Toán học

Cơ quan công tác: Tạp chí Thông tin và Truyền thông - Bộ Thông tin và Truyền thông

Số điện thoại: 0936366056 E-mail: hung_le31@yahoo.com

NỘI DUNG NHẬN XÉT

I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài

Đảm bảo an toàn thông tin (ATTT) cho hệ thống thông tin nói chung là vấn đề quan trọng trong thời đại hiện nay khi mà công cuộc chuyển đổi số đang diễn ra mạnh mẽ, gần như mọi mặt của cuộc sống đều được đưa lên môi trường số. Trong đó việc bảo vệ an toàn thông tin cho ứng dụng Web lại càng rất quan trọng khi mọi thứ đều có thể truy cập, khai thác, chia sẻ qua mạng Internet.

Khi ứng dụng Web bị tấn công có thể gây ra nhiều hậu quả nghiêm trọng đối với người dùng, tổ chức như mất thông tin cá nhân, doanh nghiệp, dịch vụ bị từ chối,... Những điều này sẽ nên gây nên nhiều hiểm họa, hậu quả nặng nề đối với cá nhân hay tổ chức (như mất danh tính, thiệt hại tài chính, dịch vụ bị gián đoạn, doanh nghiệp mất uy tín, rủi ro pháp lý,...).

Việc đảm bảo ATTT cho hệ thống thống thông tin nói chung, ứng dụng Web nói riêng cần phải được thực hiện với nhiều giải pháp, trong đó phát hiện tấn công để chủ động đối phó là một giải pháp quan trọng. Hiện tại có nhiều phương pháp, giải pháp được ứng dụng để phát hiện các cuộc tấn công Web, trong đó giám sát hệ thống và dựa trên các kỹ thuật học máy, học sâu là một giải pháp hiệu quả khi các biện pháp, kỹ thuật tấn công ứng dụng Web ngày càng tinh vi, biến đổi không ngừng.

Đề án có ý nghĩa thực tiễn trong việc ứng dụng để phát hiện tấn công ứng dụng Web nhằm chủ động đối phó để đảm bảo ATTT.

II/ Nội dung của đề án tốt nghiệp, các kết quả đã đạt được

Trên cơ sở nghiên cứu bài toán đảm bảo ATTT cho ứng dụng Web đề án đã tập trung xây dựng mô hình phát hiện tấn công ứng dụng web dựa trên học sâu và cài đặt thử nghiệm. Kết quả của đề án cơ bản phù hợp với đề cương được duyệt và được trình bày trong 3 chương chính:

1- Các phương pháp tấn công ứng dụng web và phòng chống tấn công. Đề án đã tổng hợp, hệ thống hóa các phương pháp tấn công, một số lỗ hổng bảo mật và các phương pháp được sử dụng để phòng chống tấn công.

2- Phát hiện tấn công web sử dụng học sâu. Trình bày tổng quan về học máy, học sâu, một số mô hình học sâu.

3- Thử nghiệm và đánh giá.

III/ Những vấn đề cần giải thích thêm

- Trong đề cương có đề cập đến nghiên cứu, sử dụng các phương pháp học sâu, gồm CNN (Convolutional Neural Network - Mạng nơ ron tích chập), LSTM (Long Short Term Memory) để xây dựng mô hình phát hiện tấn công ứng dụng web nhưng trong đề án lại chỉ nghiên cứu phương pháp phát hiện tấn công ứng dụng web sử dụng học sâu CNN, cần xem lại để nhất quán nội dung này.

- Mô hình đề xuất sử dụng trong đề án chưa thấy trình bày về thông số cụ thể của các lớp (ví dụ: đầu vào của lớp tích chập (convolution) là một tensor 4D có kích thước (số lượng mẫu, chiều cao, chiều rộng, kênh) như thế nào,...).

- Đề án chưa đưa ra được so sánh (độ chính xác, thời gian xử lý,..) giữa mô hình đề xuất với các mô hình tham khảo.

- Đề án còn lỗi đánh máy, lỗi định dạng (format) văn bản cần rà soát, chỉnh sửa.

* Câu hỏi: Đóng góp mới của đề tài là gì?

IV/ Kết luận

Đồng ý cho phép học viên bảo vệ đề án tốt nghiệp.

Hà Nội, ngày 20 tháng 3 năm 2024

NGƯỜI NHẬN XÉT

Lê Quốc Hưng

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ
BƯU CHÍNH VIỄN THÔNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập Tự do Hạnh phúc

BẢN NHẬN XÉT LUẬN VĂN TỐT NGHIỆP THẠC SĨ

Tên đề tài luận văn : **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

Tên học viên : **Lê Ngọc Khoa**

Họ và tên người nhận xét: Đỗ Trung Tuấn,

Đơn vị, Cơ quan công tác: Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Hà Nội

Ý KIẾN NHẬN XÉT

1. Tính thời sự, cấp thiết, ý nghĩa khoa học và thực tiễn của đề tài

Học viên thực hiện luận văn về an toàn ứng dụng web. Tấn công ứng dụng web là một phần của tấn công mạng máy tính. Một cuộc tấn công không gian mạng là bất kỳ hình thức tấn công nào của các quốc gia, cá nhân, nhóm hoặc tổ chức nhắm vào các hệ thống thông tin máy tính, cơ sở hạ tầng, mạng máy tính hoặc các thiết bị máy tính cá nhân bằng nhiều cách khác nhau của các hành vi độc hại thường có nguồn gốc từ một nguồn giấu tên, mà đánh cắp, thay đổi, hoặc hủy hoại một mục tiêu cụ thể bằng cách xâm nhập tái phép vào một hệ thống dễ bị tổn thương

Do vậy đề tài luận văn của học viên có ý nghĩa thực tế.

Công nghệ học viên sử dụng là học sâu. Học sâu là một phần trong một nhánh rộng hơn các phương pháp học máy dựa trên mạng thần kinh nhân tạo kết hợp với việc học biểu diễn đặc trưng. Việc học này có thể có giám sát, nửa giám sát hoặc không giám sát. Trong học sâu, một mạng thần kinh tích chập là một lớp của mạng thần kinh sâu, áp dụng phổ biến nhất để phân tích hình ảnh trực quan.

Luận văn của học viên được thầy PGS. TS. Hoàng Xuân Dậu hướng dẫn, đáp ứng mã đào tạo về công nghệ thông tin sau đại học về khoa học máy tính, mã 8 48 01 01.

Hiện người nhận xét thấy kết quả của luận văn không trùng lặp với các luận văn hay đề tài nghiên cứu khoa học đã bảo về tại cơ sở đào tạo.

2. Về nội dung, chất lượng của luận văn, các kết quả đã đạt được

Học viên đã có các tài liệu liên quan đến việc bảo vệ kết quả nghiên cứu và ứng dụng, gồm bản viết luận văn và bản tóm tắt luận văn, đề cương luận văn đã được bộ môn chuyên ngành thẩm định và góp ý.

Nội dung bản viết luận văn thể hiện các ý chính của bản viết luận văn. Luận văn của học viên chia thành các chương, liên quan đến các vấn đề sau :

- Các dạng tấn công ứng dụng web, liên quan đến các lỗ hổng bảo mật của các ứng dụng;
- Vai trò của học sâu trong việc phát hiện tấn công web. Học viên đề xuất mô hình học sâu. Tuy nhiên liệu có thể sử dụng phương pháp học thông thường ?
- Học viên trình bày phần thử nghiệm, đánh giá kết quả thử nghiệm trong chương cuối của luận văn, từ trang 35 đến trang 46.

3. Những vấn đề cần giải thích thêm

Liên quan đến kết quả của học viên, qua bản viết luận văn, người nhận xét thấy :

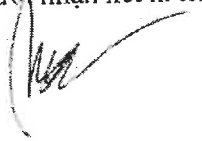
- Tuy có xử lí các hình ảnh, có thể sử dụng phương pháp học thông thường đối với bài toán mà học viên đặt ra không ?

4. Kết luận

Theo các kết quả đạt được của luận văn về nghiên cứu và thực nghiệm, căn cứ vào yêu cầu của luận án thạc sĩ chuyên ngành, tôi đồng ý để học viên Lê Ngọc Khoa được bảo vệ luận văn trước Hội đồng chấm luận văn thạc sĩ.

Hà nội, 15 tháng 3 năm 2024

Người nhận xét kí tên



Đỗ Trung Tuấn

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ
BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BÁO CÁO GIẢI TRÌNH
SỬA CHỮA, HOÀN THIỆN ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Họ và tên học viên: Lê Ngọc Khoa.....

Chuyên ngành: Khoa học máy tính.....

Khóa: 2022 đợt I. .....

Tên đề tài: Nghiên cứu phương pháp phát hiện tấn công ứng dụng web sử dụng học sâu CNN

Người hướng dẫn khoa học: PGS. TS. Hoàng Xuân Dậu.....

Ngày bảo vệ: 20/03/2024 .....

Các nội dung học viên đã sửa chữa, bổ sung trong đề án tốt nghiệp theo ý kiến đóng góp của Hội đồng chấm đề án tốt nghiệp:

TT	Ý kiến hội đồng	Sửa chữa của học viên
1	Rà soát chỉnh sửa lỗi chính tả	Đã thực hiện sửa lỗi chính tả trong đề án. Ví dụ: Dòng 26 trang 14, dòng 6 trang 21, dòng 4 trang 42.
2	Bổ sung so sánh thực nghiệm giữa mô hình học sâu CNN với một số phương pháp phát hiện tấn công ứng dụng web sử dụng học máy.	Đã bổ sung so sánh với một số phương pháp học máy ở mục 3.3.2 thuộc trang 41.

Hà Nội, ngày 27 tháng 03 năm 2024

Ký xác nhận của

CHỦ TỊCH HỘI ĐỒNG CHẤM LUẬN VĂN	THƯ KÝ HỘI ĐỒNG	NGƯỜI HƯỚNG DẪN KHOA HỌC	HỌC VIÊN
TS. Nguyễn Duy Phương	TS. Đào Thị Thúy Quỳnh	PGS.TS. Hoàng Xuân Dậu	Lê Ngọc Khoa

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ
BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BÁO CÁO GIẢI TRÌNH
SỬA CHỮA, HOÀN THIỆN ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Họ và tên học viên: Lê Ngọc Khoa.....

Chuyên ngành: Khoa học máy tính.....

Khóa: 2022 đợt I. .....

Tên đề tài: Nghiên cứu phương pháp phát hiện tấn công ứng dụng web sử dụng học sâu CNN

Người hướng dẫn khoa học: PGS. TS. Hoàng Xuân Dậu.....

Ngày bảo vệ: 20/03/2024 .....

Các nội dung học viên đã sửa chữa, bổ sung trong đề án tốt nghiệp theo ý kiến đóng góp của Hội đồng chấm đề án tốt nghiệp:

TT	Ý kiến hội đồng	Sửa chữa của học viên
1	Rà soát chỉnh sửa lỗi chính tả	Đã thực hiện sửa lỗi chính tả trong đề án. Ví dụ: Dòng 26 trang 14, dòng 6 trang 21, dòng 4 trang 42.
2	Bổ sung so sánh thực nghiệm giữa mô hình học sâu CNN với một số phương pháp phát hiện tấn công ứng dụng web sử dụng học máy.	Đã bổ sung so sánh với một số phương pháp học máy ở mục 3.3.2 thuộc trang 41.

Hà Nội, ngày 27.. tháng 03 năm 2024

Ký xác nhận của

CHỦ TỊCH HỘI ĐỒNG CHẤM LUẬN VĂN	THƯ KÝ HỘI ĐỒNG	NGƯỜI HƯỚNG DẪN KHOA HỌC	HỌC VIÊN
TS. Nguyễn Duy Phương	TS. Đào Thị Thúy Quỳnh	PGS.TS. Hoàng Xuân Dậu	Lê Ngọc Khoa