BỘ GIÁO DỤC
VÀ ĐÀO TẠO

VIỆN HÀN LÂM KHOA HỌC
VÀ CÔNG NGHỆ VIỆT NAM

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ

Nguyễn Trọng Hưng

NGHIÊN CỨU CÁC GIẢI PHÁP PHÁT HIỆN TẤN CÔNG WEB SỬ DỤNG WEB LOG VÀ NỘI DUNG KẾT HỢP ẢNH MÀN HÌNH TRANG WEB

LUẬN ÁN TIẾN SĨ HỆ THỐNG THÔNG TIN

Hà Nội - 2024

BỘ GIÁO DỤC
VÀ ĐÀO TẠO

VIỆN HÀN LÂM KHOA HỌC
VÀ CÔNG NGHỆ VIỆT NAM

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ

Nguyễn Trọng Hưng

NGHIÊN CỨU CÁC GIẢI PHÁP PHÁT HIỆN TẤN CÔNG WEB SỬ DỤNG WEB LOG VÀ NỘI DUNG KẾT HỢP ẢNH MÀN HÌNH TRANG WEB

LUẬN ÁN TIẾN SĨ HỆ THỐNG THÔNG TIN

Mã số: 9 48 01 04

Xác nhận của Học viện Khoa học và Công nghệ	Người hướng dẫn 1	Người hướng dẫn 2
KT. GIÁM ĐỐC PHÓ GIÁM ĐỐC	*(Ký, ghi rõ họ tên)*	*(Ký, ghi rõ họ tên)*
VIỆN HÀN LÂM KHOA HỌC VÀ CÔNG NGHỆ VIỆT NAM HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ		
Nguyễn Thị Trung		

Hà Nội - 2024

LỜI CAM ĐOAN

Tôi xin cam đoan luận án: *"Nghiên cứu các giải pháp phát hiện tấn công web sử dụng web log và nội dung kết hợp ảnh màn hình trang web"* là công trình nghiên cứu của chính mình dưới sự hướng dẫn khoa học của tập thể thầy hướng dẫn. Luận án sử dụng thông tin trích dẫn từ nhiều nguồn tham khảo khác nhau và các thông tin trích dẫn được ghi rõ nguồn gốc. Các kết quả nghiên cứu của tôi được công bố chung với các tác giả khác đã được sự nhất trí của đồng tác giả khi đưa vào luận án. Các số liệu, kết quả được trình bày trong luận án là hoàn toàn trung thực và chưa từng được công bố trong bất kỳ một công trình nào khác ngoài các công trình công bố của tác giả. Kết quả thực nghiệm của luận án được lưu trữ trên tài khoản Github của NCS https://github.com/tronghung-nguyen/PhD.

Luận án được hoàn thành trong thời gian tôi làm nghiên cứu sinh tại Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam.

Hà Nội, Ngày tháng năm 20

Tác giả luận án

Nguyễn Trọng Hưng

LỜI CẢM ƠN

Thực hiện luận án tiến sĩ là một thách thức rất lớn, một quá trình nghiên cứu đòi hỏi sự tập trung và kiên trì. Hoàn thành chương trình nghiên cứu sinh và được công bố những kết quả trong quá trình nghiên cứu tôi thực sự thấy hạnh phúc. Đây không chỉ là nỗ lực cá nhân, mà còn là sự hỗ trợ và giúp đỡ nhiệt tình của các Thầy hướng dẫn, Học viện, bộ môn, các đơn vị hỗ trợ đào tạo, đồng nghiệp và gia đình.

Trước hết, tôi xin gửi lời cảm ơn chân thành và sâu sắc tới PGS.TS. Hoàng Xuân Dậu và PGS.TS. Nguyễn Đức Dũng đã quan tâm hướng dẫn và giúp đỡ tôi trong suốt quá trình thực hiện và hoàn thành luận án.

Tôi xin chân thành cảm ơn Lãnh đạo Viện Công nghệ thông tin – Viện Hàn lâm Khoa học và Công nghệ Việt Nam, Học viện Khoa học và Công nghệ – Viện Hàn lâm Khoa học và Công nghệ Việt Nam, đã tạo điều kiện thuận lợi cho tôi trong thời gian nghiên cứu và hoàn thành luận án. Tôi cũng xin cảm ơn Lãnh đạo Khoa An ninh mạng và PCTPSDCNC – Học viện An ninh nhân dân và đồng nghiệp đã hỗ trợ, động viên tôi trong quá trình nghiên cứu và thực hiện luận án.

Cuối cùng, tôi xin gửi lời cảm ơn vô hạn tới gia đình đã luôn ở bên cạnh, chia sẻ, động viên tội những lúc khó khăn, hỗ trợ cả về vật chất lẫn tinh thần trong suốt quá trình nghiên cứu.

Hà Nội, Ngày tháng năm 20

Tác giả luận án

Nguyễn Trọng Hưng

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

STT	Từ viết tắt	Từ gốc	Tiếng Việt
1	A05	A05	Cục An ninh mạng và phòng chống sử dụng công nghệ cao
2	API	Application Programning Interface	Giao diện lập trình ứng dụng
3	CGI	Common Gateway Interface	Giao diện cổng giao tiếp chung
4	CSRF	Cross-Site Request Forgery	Tấn công CSRF
5	CSS	Cascade Style Sheet	Định dạng CSS
6	DOM	Document Object Model	Mô hình DOM
7	HTML	Hyper Text Markup Language	Ngôn ngữ đánh dấu siêu văn bản
8	HTTP	Hyper Text Transfer Protocol	Giao thức truyền siêu văn bản
9	HTTPS	Secure HTTP	Giao thức HTTP an toàn
10	IDS	Intrusion Detection System	Hệ thống phát hiện xâm nhập
11	IIS	Internet Information Services	Dịch vụ thông tin Internet
12	IP	Internet Protocol	Giao thức Internet
13	OSI	Open Systems Interconnect	Hệ thống kết nối mở
14	OWASP	Open Web Application Security Project	Dự án cho đảm bảo an toàn cho ứng dụng web mở
15	SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc
16	SQLi	SQL injection	Tấn công chèn mã SQL
17	SVM	Support Vector Machine	Máy véc tơ hỗ trợ
18	TCP	Transfer Control Protocol	Giao thức điều khiển truyền
19	UDP	User Datagram Protocol	Giao thức truyền gói tin người dùng
20	URI	Uniform Resource Identifier	Tên nhận dạng tài nguyên đồng nhất
21	URL	Uniform Resource Locator	Bộ định vị tài nguyên đồng nhất
22	VNCS	Vietnam Cyber Security	Công ty VNCS
23	WAF	Web Application Firewall	Tường lửa ứng dụng web
24	XML	eXtensible Markup Language	Ngôn ngữ đánh dấu mở rộng
25	XSS	Cross Site Scripting	Tấn công XSS

26	CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
27	LSTM	Long Short-Term Memory	Bố nhớ dài-ngắn hạn
28	BiLSTM	Bidirectional Long Short-Term Memory	Bố nhớ dài-ngắn hạn hai chiều
29	PPV	Positive Predictive Value	Giá trị dự đoán dương tính
30	TPR	True Positive Rate	Tỷ lệ dương tính thật
31	FPR	False Positive Rate	Tỷ lệ dương tính giả
32	FNR	False Negative Rate	Tỷ lệ âm tính giả
33	ACC	Accuracy	Độ chính xác tổng thể
34	LRFN	Learning Rate Schedule	Hàm thiết lập lịch trình học

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

MỞ ĐẦU

1. TÍNH CẤP THIẾT CỦA LUẬN ÁN

Ngày nay, các ứng dụng trên nền web (gọi tắt là ứng dụng web) gồm các website và web portal đã và đang đóng góp rất lớn vào việc phổ cập thông tin, hoạt động quảng bá tin tức, các cơ sở dữ liệu, và nhiều ứng dụng trực tuyến trên mạng như: các gian hàng trực tuyến, trò chơi điện tử trực tuyến và mạng xã hội [69]. Các ứng dụng web đã làm thay đổi cả thế giới từ khi xuất hiện vào đầu những năm 90 của thế kỷ trước. Theo thống kê từ Statista[1], tính đến cuối năm 2022, thế giới có khoảng trên 5,3 tỷ người dùng các ứng dụng trên Internet, với số lượng website trên toàn thế giới là gần 2 tỷ trang web. Đó là những số liệu nói lên sự bùng nổ, phát triển mạnh mẽ của các ứng dụng web và người dùng trên đó. Đi kèm với sự phát triển này là những nguy cơ, thách thức mà các tổ chức và người sử dụng cá nhân phải đối mặt, như các hình thức tấn công mạng nói chung và các hình thức tấn công ứng dụng web nói riêng [81] [2]. Số liệu thống kê đến quý 3 năm 2018 của CyStack[2] ghi nhận 129.722 website trên toàn cầu đã bị tin tặc tấn công và chiếm quyền điều khiển. Các hình thức tấn công chủ yếu khai thác các lỗ hổng bảo mật ứng dụng web như: SQLi (SQL injection), XSS (Cross Site Scripting), CSRF (Cross-Site Request Forgery), CMDi (Command Injection), duyệt đường dẫn, webshell, thay đổi giao diện, HTTP DDoS [79] .

Theo báo cáo an ninh mạng từ Cystask[3], trong năm 2019 trên thế giới có hơn 560.000 vụ tấn công vào các trang web, trong đó Việt Nam có 9.300 trang web bị xâm nhập, xếp thứ 11 trên thế giới và thứ 3 tại Đông Nam Á. Theo số liệu báo cáo "Mối đe dọa từ API và ứng dụng web năm 2022"từ công ty công nghệ Akamai[4] chuyên về an ninh mạng, cung cấp các dịch vụ bảo mật web và internet cho thấy, chỉ tính riêng nửa đầu năm 2022 số cuộc tấn công khai thác ứng dụng web và API trên toàn cầu là khoảng 9 tỷ lượt, số lượng này đã tăng gấp 3 lần so với nửa đầu năm 2021. Trong số các cuộc tấn công này thì hình thức tấn công khai thác chủ yếu là SQLi, duyệt đường dẫn, khai thác các tệp cục bộ, khai thác XSS. Tại Việt Nam theo số liệu từ Cục An toàn thông tin, trong 11 tháng đầu năm 2022, đã có tới 11.213 cuộc tấn công mạng hướng vào Việt Nam, tăng 44,2% so với cùng kỳ năm 2021. Trong đó, có

---

[1] A. Petrosyan, "Global number of internet users 2005-2022," Statista, 23 2 2023. [Online]. Available: https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/. [Accessed 7 2023].
[2] "CyStack Security Report Q3 2018," CyStack, [Online]. Available: https://s.cystack.net/resource/home/wp-content/uploads/sites/4/CyStack_Security_Report_Q3_2018-1.pdf. [Accessed 9 2021].
[3] N. Dang, "Báo cáo an ninh website thực hiện bởi CyStack," CyStack, 2023. [Online]. Available: https://cystack.net/vi/blog/viet-nam-co-hon-9300-trang-web-bi-tan-cong-trong-nam-2019. [Accessed 5 2023].
[4] Akamai, "Akamai Web Application and API Threat Report," 2022.

3.930 cuộc tấn công giả mạo (phishing), đặc biệt có 1.524 cuộc tấn công thay đổi giao diện trang web (defacement), 5.759 cuộc tấn công sử dụng phần mềm độc hại (malware).

Do tính chất nguy hiểm của tấn công web đối với các cơ quan, tổ chức và cá nhân, nhiều giải pháp đã được nghiên cứu, phát triển và triển khai để phát hiện, phòng chống tấn công web, như sử dụng tường lửa web (WAF), hệ thống phát hiện xâm nhập web (Web IDS - Intrusion Detection System), kiểm thử xâm nhập [40] [72] [86] Nhìn chung, hiện nay có hai hướng tiếp cận chính trong phát hiện tấn công web: (1) phát hiện dựa trên dấu hiệu, chữ ký và (2) phát hiện dựa trên bất thường [40] [49] [80]. Các giải pháp theo hướng tiếp cận (1) sử dụng các quy tắc, tập luật, chữ ký để phát hiện các cuộc tấn công web. Phương pháp này cho độ chính xác cao, tỷ lệ dương tính giả thấp, tuy nhiên nó lại không phát hiện được những cuộc tấn công mới do những tấn công này chưa được mô tả bởi các quy tắc, tập luật, chữ ký đã có. Các giải pháp theo hướng tiếp cận (2) phát hiện dựa trên bất thường là "vấn đề tìm ra các mẫu trong dữ liệu không phù hợp với hành vi mong muốn - the problem of finding patterns in data that do not conform to expected behavior" [20] [73]. Các thuật toán dựa trên thống kê từ lâu đã được sử dụng để phát hiện các bất thường [19] [20]. Ngoài ra, phát hiện bất thường còn dựa trên một số kỹ thuật và thuật toán như: dựa trên hoạt động hoặc ngưỡng, mô hình Markov, mô hình Moment hoặc độ lệch chuẩn trung bình, mô hình học máy và các thuật toán di truyền [20] [48]. Ưu điểm của phát hiện dựa trên bất thường là nó cho phép phát hiện các cuộc tấn công mới do không yêu cầu có trước các thông tin về các cuộc tấn công. Nhược điểm chính của phát hiện tấn công dựa trên bất thường là tỷ lệ cảnh báo sai (gồm tỷ lệ dương tính giả và tỷ lệ âm tính giả) còn tương đối cao so với kỹ thuật phát hiện dựa trên dấu hiệu, chữ ký.

Học máy là một trong nhiều kỹ thuật được sử dụng trong phát hiện bất thường [20] [48]. Đặc biệt, với sự phát triển mạnh mẽ của công nghệ trong thời đại 4.0, hiện nay các mô hình học máy, học sâu ngày càng được sử dụng như một trong những phương pháp tiếp cận phổ biến trong phát hiện bất thường [73]. Các kỹ thuật học máy được sử dụng để xây dựng mô hình phân biệt giữa các lớp bình thường và các lớp bất thường. Phụ thuộc vào sự sẵn có của dữ liệu được dán nhãn, có thể sử dụng các mô hình học máy có giám sát, bán giám sát hoặc không giám sát. Trong khi các mô hình học máy có giám sát yêu cầu toàn bộ dữ liệu được dán nhãn, các mô hình học máy bán giám sát chỉ yêu cầu một phần dữ liệu được dán nhãn, còn các mô hình học máy không giám sát có thể xử lý dữ liệu không được dán nhãn. Nhờ sử dụng dữ liệu được dán nhãn, các mô hình học máy có giám sát thường cho độ chính xác cao, tỷ lệ cảnh báo sai thấp và tốc độ xử lý tương đối nhanh [73]. Trên thực tế, hướng phát hiện các

dạng tấn công web sử dụng học máy, học sâu dựa trên việc phân tích log, phân tích nội dung, kết hợp hình ảnh chụp màn hình trang web được quan tâm nghiên cứu trong những năm gần đây và cho nhiều kết quả khả quan [14] [38] [40] [44] [43] [58] [76] [85] [91].

Từ các phân tích trên, luận án tập trung nghiên cứu các kỹ thuật phát hiện tấn công web dựa trên học máy và học sâu - một biến thể thuộc hướng tiếp cận (2) – phát hiện bất thường. Ngoài khả năng phát hiện được các dạng tấn công chưa xuất hiện trong dữ liệu huấn luyện, có thể tự động hóa quá trình xây dựng mô hình phát hiện tấn công web từ tập dữ liệu huấn luyện. Nhờ vậy, có thể giảm nhân lực chuyên gia cho việc xây dựng thủ công các tập luật, tập dấu hiệu, chữ ký phát hiện.

Cụ thể hơn, luận án tập trung nghiên cứu theo hai hướng chính: hướng (i) phát hiện các dạng tấn công web cơ bản, bao gồm *SQLi, XSS, duyệt đường dẫn, CMDi* và hướng (ii) là phát hiện tấn công thay đổi giao diện trang web. Theo hướng (i), có thể liệt kê các đề xuất cho phát hiện tấn công web tiêu biểu, như AMNESIA [39], Swaddler [22], CANDID [16] và Torrano-Gimenez và cộng sự [100]. Các nghiên cứu này sử dụng các phương pháp như rà quét mã nguồn ứng dụng web [39], hay như phân tích trạng thái bên trong của ứng dụng web và tìm mối quan hệ giữa điểm thực thi quan trọng của ứng dụng web và trạng thái bên ngoài. Một cách tiếp cận khác trong phát hiện tấn công web trong hướng (i) là sử dụng học máy, học sâu, tiêu biểu và có tiềm năng gồm Betarte và cộng sự [14], Liang và cộng sự [58], Pan và cộng sự [76], Saiyu Hao và cộng sự [40]. Các nghiên cứu này sử dụng các phương pháp học máy truyền thống và một số thuật toán học sâu để xây dựng mô hình phát hiện tấn công web. Tuy vậy, chưa có nhiều công trình sử dụng bộ dữ liệu từ web log và các nghiên cứu này thường chỉ thực hiện phát hiện được một hình thức tấn công trên một tập dữ liệu thử nghiệm cụ thể. *Do đó, luận án này tiếp tục nghiên cứu phát hiện đồng thời các dạng tấn công web thường gặp, bao gồm SQLi, XSS, duyệt đường dẫn, CMDi dựa trên dữ liệu web log sử dụng các mô hình học máy có giám sát.*

Theo hướng (ii), các kỹ thuật thường được sử dụng để phát hiện tấn công thay đổi giao diện trang web bao gồm các kỹ thuật đơn giản, như so sánh Checksum, so sánh diff, phân tích cây DOM (Document Object Model) và các kỹ thuật phức tạp, như sử dụng các thuật toán học máy, học sâu, hoặc phương pháp thống kê [44]. Phát hiện tấn công thay đổi giao diện trang web dựa trên các kỹ thuật đơn giản chỉ có thể áp dụng hiệu quả với các trang web tĩnh – là những trang ít có sự thay đổi về hình thức và nội dung. Ngược lại, phát hiện tấn công thay đổi giao diện trang web dựa trên các kỹ thuật phức tạp có thể áp dụng hiệu quả với cả các trang web tĩnh và trang web

động - là những trang có sự thay đổi, cập nhật thường xuyên về hình thức và nội dung. Một số đề xuất tiêu biểu có thể liệt kê là các nghiên cứu [13] [23] [38] [44] [43] [54]. Tuy vậy, một số đề xuất có độ phức tạp cao, yêu cầu tài nguyên tính toán lớn. Ngoài ra, hầu hết các nghiên cứu đã có chỉ tập trung sử dụng một loại đặc trưng liên quan đến nội dung trang web mà chưa có sự kết hợp các loại đặc trưng điển hình, gồm nội dung và hình ảnh của của trang web bị tấn công thay đổi giao diện. *Do vậy, luận án tập trung nghiên cứu phương pháp phát hiện tấn công thay đổi giao diện trang web sử dụng các thuật toán học sâu và kết hợp các đặc trưng văn bản/nội dung và hình thức thể hiện - là ảnh chụp màn hình trang web để cải thiện hiệu suất phát hiện của mô hình, có xem xét đến thời gian phát hiện để mô hình đề xuất có khả năng triển khai thực tế.*

2. MỤC TIÊU CỦA LUẬN ÁN

Mục tiêu chung của luận án là nghiên cứu, đề xuất mô hình phát hiện tấn công web dựa trên kỹ thuật học máy và học sâu. Cụ thể, luận án tập trung vào các mục tiêu sau:

- Nghiên cứu, đánh giá, các phương pháp, kỹ thuật, giải pháp, công cụ phát hiện tấn công web.

- Nghiên cứu đề xuất mô hình phát hiện các dạng tấn công web thường gặp dựa trên kỹ thuật học máy có giám sát sử dụng dữ liệu web log, nhằm nâng cao độ chính xác, giảm cảnh báo sai, đồng thời cho phép phát hiện nhiều loại tấn công web.

- Nghiên cứu đề xuất mô hình phát hiện tấn công thay đổi giao diện trang web dựa trên kỹ thuật học sâu và kết hợp hai loại đặc trưng văn bản và hình ảnh của trang web, nhằm nâng cao độ chính xác, giảm cảnh báo sai.

- Cài đặt, thử nghiệm và đánh giá các mô hình phát hiện tấn công web đã đề xuất sử dụng các tập dữ liệu đã được công bố và tập dữ liệu thu thập thực tế.

3. ĐỐI TƯỢNG NGHIÊN CỨU VÀ PHẠM VI NGHIÊN CỨU

- Đối tượng nghiên cứu là các dạng tấn công web, bao gồm: SQLi, XSS, CMDi, duyệt đường dẫn và tấn công thay đổi giao diện trang web.

- Phạm vi nghiên cứu giới hạn trong các kỹ thuật, giải pháp phát hiện tấn công web, cụ thể:

  - Phát hiện tấn công web cơ bản như: SQLi, XSS, CMDi, duyệt đường dẫn sử dụng web log;
  - Phát hiện tấn công thay đổi giao diện trang web dựa trên việc sử dụng đặc trưng văn bản và ảnh màn hình trang web.

- Các thuật toán, mô hình học máy truyền thống, học sâu sử dụng trong các mô hình phát hiện tấn công web.

4. PHƯƠNG PHÁP NGHIÊN CỨU

Luận án sử dụng phương pháp nghiên cứu lý thuyết kết hợp với phương pháp thực nghiệm. Trong đó, phương pháp nghiên cứu lý thuyết được sử dụng để thực hiện các công việc sau:

- Nghiên cứu nền tảng lý thuyết về tấn công web, bao gồm khái quát về web và dịch vụ web, tổng quan về tấn công web, các dạng tấn công web thường gặp, khảo sát đánh giá các phương pháp phát hiện tấn công web hiện có;

- Nghiên cứu nền tảng lý thuyết về học máy, học sâu cho luận án, bao gồm khái quát về học máy, một số thuật toán học máy có giám sát, một số thuật toán học sâu, phương pháp đánh giá và các độ đo đánh giá mô hình phát hiện dựa trên học máy và học sâu;

- Khảo sát, đánh giá các đề xuất, giải pháp đã có cho phát hiện tấn công web, trên cơ sở đó tổng hợp các ưu điểm, nhược điểm làm cơ sở cho đề xuất của luận án;

- Lựa chọn, đề xuất các đặc trưng, xây dựng các mô hình phát hiện các dạng tấn công web.

Phương pháp thực nghiệm được sử dụng trong luận án để thực hiện các phần việc sau:

- Khảo sát và xây dựng các tập dữ liệu về tấn công web dựa trên web log và lựa chọn tập dữ liệu phù hợp cho thực nghiệm;

- Cài đặt và thực nghiệm các mô hình phát hiện tấn công web đề xuất trong luận án, đánh giá, so sánh các mô hình đề xuất với các mô hình, đề xuất đã có.

5. CÁC ĐÓNG GÓP CỦA LUẬN ÁN

Đóng góp thứ nhất của luận án là đề xuất mô hình phát hiện các dạng tấn công web dựa trên học máy sử dụng các đặc trưng ký tự trong dữ liệu truy vấn URI trích xuất từ web log (cụ thể là các ?query_string trong URI, lý do lựa chọn truy vấn này được phân tích tại mục *2.3.1. Giới thiệu mô hình*). Các thuật toán học máy có giám sát được sử dụng gồm Rừng ngẫu nhiên, Cây quyết định, Naïve Bayes và SVM. Mô hình đề xuất cho độ chính xác cao, tỷ lệ cảnh báo sai thấp, thời gian xử lý nhanh, phù hợp bài toán giám sát một lượng web log rất lớn trong thực tế. Kết quả của đóng góp này được phân tích tại mục *2.3.6. Nhận xét*.

Đóng góp thứ hai của luận án là đề xuất mô hình phát hiện tấn công thay đổi giao diện trang web dựa trên học sâu sử dụng các đặc trưng văn bản trích xuất từ trang web kết hợp với các đặc trưng hình ảnh màn hình trang web. Thuật toán học sâu sử dụng là BiLSTM (Bidirectional LSTM) cho xử lý đặc trưng văn bản thuần và EfficientNet cho xử lý ảnh màn hình. Trong đề xuất này, các đặc trưng văn bản và ảnh màn hình trang web được lựa chọn làm dữ liệu tương ứng cho các mô hình học sâu thành phần là BiLSTM và EfficientNet; Kết quả của mô hình là sự kết hợp của 2 mô hình phát hiện thành phần. Kết quả của đóng góp này được phân tích tại mục *3.5.5. Nhận xét*.

6. BỐ CỤC CỦA LUẬN ÁN

Luận án được bố cục thành ba chương với nội dung như sau:

Chương 1. Tổng quan về phát hiện tấn công web giới thiệu khái quát về web và dịch vụ web, các lỗ hổng bảo mật web, các dạng tấn công web thường gặp. Tiếp theo là phần khảo sát một số giải pháp, công cụ và kỹ thuật phát hiện tấn công web hiện có. Trên cơ sở kết quả khảo sát, chương chỉ rõ các ưu điểm và hạn chế của các giải pháp đã có làm cơ sở cho 2 bài toán sẽ được giải quyết trong luận án. Phần cuối của chương giới thiệu khái quát về học máy, học sâu và mô tả một số giải thuật học máy có giám sát và học sâu sử dụng trong các mô hình phát hiện tấn công web được đề xuất trong Chương 2 và Chương 3.

Chương 2. Phát hiện tấn công web dựa trên học máy sử dụng web log giới thiệu khái quát về web log, một số đề xuất phát hiện tấn công web sử dụng học máy, đánh giá ưu nhược điểm của các đề xuất. Phần cuối của chương mô tả việc xây dựng, cài đặt, thử nghiệm và đánh giá mô hình phát hiện tấn công web thường gặp dựa trên học máy sử dụng web log.

Chương 3. Phát hiện tấn công thay đổi giao diện trang web giới thiệu khái quát về tấn công thay đổi giao diện, các phương pháp phát hiện tấn công thay đổi giao diện, so sánh các phương pháp phát hiện thay đổi giao diện sử dụng đặc trưng ảnh chụp màn hình trang web. Phần cuối của chương mô tả việc xây dựng, cài đặt, thử nghiệm và đánh giá mô hình phát hiện tấn công thay đổi giao diện trang web dựa trên học sâu sử dụng kết hợp đặc trưng ảnh chụp màn hình và đặc trưng nội dung văn bản của trang web.

Cuối cùng là phần Kết luận của luận án.

CHƯƠNG 1. TỔNG QUAN VỀ PHÁT HIỆN TẤN CÔNG WEB

Chương 1 giới thiệu khái quát về web và dịch vụ web, các lỗ hổng bảo mật, các dạng tấn công web thường gặp. Tiếp theo là phần khảo sát một số giải pháp, công cụ và kỹ thuật phát hiện tấn công web hiện có. Trên cơ sở kết quả khảo sát, chương chỉ rõ các ưu điểm và hạn chế của các giải pháp đã có làm cơ sở cho 2 bài toán sẽ được giải quyết trong luận án. Phần cuối của chương giới thiệu khái quát về học máy, học sâu và mô tả một số giải thuật học máy có giám sát và học sâu sử dụng trong các mô hình phát hiện tấn công web được đề xuất trong Chương 2 và Chương 3.

1.1. Khái quát về web và dịch vụ web

1.1.1. Các định nghĩa

Dịch vụ web (Web service): Tổ chức World Wide Web Consortium (W3C) định nghĩa Dịch vụ web là hệ thống phần mềm cho phép các máy khác nhau tương tác với nhau thông qua mạng. Các dịch vụ web đạt được nhiệm vụ này với sự trợ giúp của các tiêu chuẩn mở, bao gồm XML, SOAP, WSDL và UDDI [33]. Tuy nhiên, theo một nghĩa hẹp hơn Dịch vụ web là hệ thống dịch vụ mạng dựa trên giao thức HTTP, cung cấp nội dung trên nền web.

Ứng dụng web (Web application) là một hệ thống phần mềm ứng dụng chạy trên nền web [102]. Ứng dụng web cũng được vận hành dựa trên giao thức HTTP theo mô hình khách chủ (Client/Sever).

Website là tập hợp của các trang web được cài đặt và chạy (host) trên máy chủ web. Như vậy, website là một phần của ứng dụng web. *Trang web* (Web page) là một phần của một website cung cấp một đầu mục nội dung hay một tính năng cụ thể của website. Ngôn ngữ thường dùng để tạo các trang web là HTML.

Trong nội dung luận án này, nghiên cứu sinh tập trung nghiên cứu các dạng tấn công cơ bản lên các ứng dụng web và các website.

1.1.2. Giao thức HTTP

Giao thức truyền siêu văn bản (HTTP – Hyper-Text Transfer Protocol) là giao thức thuộc tầng ứng dụng thuộc bộ giao thức TCP/IP được sử dụng cho truyền *siêu văn bản* (Hyper-Text). HTTP là giao thức nền tảng trong vận hành dịch vụ web và các ứng dụng web. Ngoài HTTP, HTTPS (Secure HTTP) còn được sử dụng cho các ứng dụng web có yêu cầu đảm bảo an toàn thông tin truyền giữa máy khách (Client) và máy chủ (Server). Cổng dịch vụ chuẩn của HTTP và HTTPS tương ứng là 80 và 443. Giao thức HTTP có 3 đặc điểm cơ bản, bao gồm không hướng kết nối, độc lập

với thông tin truyền và không trạng thái. Giao thức HTTP hỗ trợ một số phương thức (method) để máy khách có thể gửi yêu cầu lên máy chủ. Các phương thức bao gồm: GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS và TRACE.

Phương thức GET được sử dụng để truy vấn thông tin từ máy chủ sử dụng một địa chỉ web. Các yêu cầu sử dụng phương thức GET chỉ nên truy vấn dữ liệu và không nên có ảnh hưởng (thay đổi) đến dữ liệu.

Phương thức HEAD tương tự như phương thức GET, nhưng chỉ có dòng trạng thái và phần tiêu đề được chuyển từ máy chủ đến máy khách.

Phương thức POST được sử dụng để gửi dữ liệu đến máy chủ, chẳng hạn thông tin khách hàng, file tải lên,… được gửi lên máy chủ sử dụng HTML form.

Phương thức PUT được sử dụng để thay thế tất cả các biểu diễn hiện tại của tài nguyên đích bằng nội dung tải lên.

Phương thức DELETE được sử dụng để xóa tất cả các biểu diễn hiện tại của tài nguyên đích cho bởi một địa chỉ web.

Phương thức CONNECT được sử dụng để thiết lập đường hầm tới máy chủ được xác định bởi một địa chỉ web nhất định.

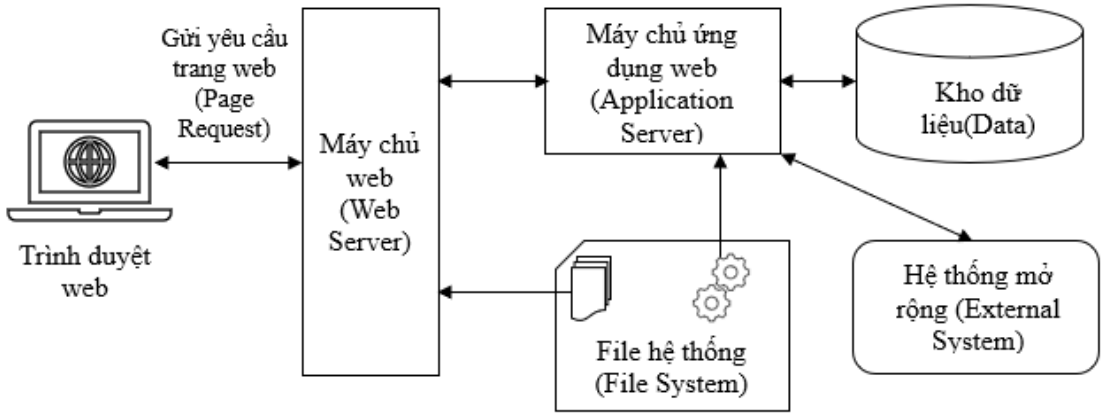
Phương thức OPTIONS được sử dụng để mô tả các tùy chọn truyền thông cho tài nguyên đích.

Phương thức TRACE được sử dụng để thực hiện một phép kiểm tra vòng lặp lại (loop-back) theo đường dẫn đến tài nguyên đích.

1.1.3. Kiến trúc ứng dụng web và các thành phần

Hình 1. 1 biểu diễn kiến trúc chuẩn của hệ thống ứng dụng web (hay ngắn gọn là ứng dụng web), trong đó mô tả các thành phần của một ứng dụng web và giao tiếp giữa chúng. Theo đó, các thành phần của một ứng dụng web gồm Web Browser (Trình duyệt web), Web Server (Máy chủ web), Application Server (Máy chủ ứng dụng), Data (Kho chứa dữ liệu – thường là cơ sở dữ liệu), File System (Hệ thống file trên máy chủ) và External System (Các hệ thống bên ngoài). Web Browser tạo và gửi yêu cầu về trang web (Page Request) đến Web Server. Nếu đó là yêu cầu trang web tĩnh, Web Server sẽ đọc nội dung trang từ File System và gửi trang web cho Web Browser. Nếu đó là yêu cầu trang web động, Web Server sẽ chuyển yêu cầu cho Application Server xử lý. Application Server sẽ dịch và thực hiện mã script trong trang web để tạo kết quả. Application Server có thể cần truy nhập Data, File System, hoặc External

System để xử lý yêu cầu. Kết quả xử lý yêu cầu được chuyển lại cho Web Server để tạo trang web và gửi cho Web Browser. [37]



Hình 1. 1. Kiến trúc chuẩn của ứng dụng web [37]

Một ứng dụng web có thể gồm các thành phần: Máy khách web/trình duyệt web (Web client/web browser), Máy chủ web (HTTP/web server), URL/URI, Web session và cookie, Bộ diễn dịch và thực hiện các server script, Các server script (CGI – Common Gateway Interface) và Máy chủ cơ sở dữ liệu [37]. Cụ thể:

- *Trình duyệt web*: Trình duyệt web là bộ phần mềm chạy trên máy khách có chức năng tạo yêu cầu, gửi yêu cầu và hiển thị phản hồi/kết quả trả về từ máy chủ web. Trình duyệt web có khả năng hiển thị nhiều loại dữ liệu của trang web: văn bản, hình ảnh, âm thanh, video,... Trình duyệt cũng hỗ trợ khả năng lập trình bằng các ngôn ngữ script (như Javascript), xử lý các ngôn ngữ HTML, XML, CSS,... Một số trình duyệt thông dụng bao gồm: Microsoft Internet Explorer, Google Chrome, Mozilla Firefox, Opera, Apple Safari,... [37]

- *Máy chủ web*: Máy chủ web tiếp nhận yêu cầu từ trình duyệt web, xử lý yêu cầu và trả về *đáp ứng*. Các đáp ứng thường là các trang web. Nếu là yêu cầu truy nhập các file tĩnh, máy chủ web truy nhập hệ thống file cục bộ, đọc nội dung file và gửi kết quả cho trình duyệt. Nếu là yêu cầu truy nhập các file script, máy chủ web chuyển các script cho bộ xử lý script. Script có thể bao gồm các lệnh truy nhập cơ sở dữ liệu để xử lý dữ liệu. Kết quả thực hiện script được chuyển lại cho máy chủ web để tạo thành đáp ứng và gửi cho trình duyệt [37].

Có nhiều loại máy chủ web được triển khai sử dụng trên thực tế, trong đó các máy chủ web thông dụng nhất bao gồm: Mozilla Apache web server, Microsoft Internet Information Services (IIS), nginx (NGINX, Inc), Google web services, IBM Websphere và Oracle web services.

- *URL và URI:* URL (Uniform Resource Locator) còn gọi là địa chỉ web, là một chuỗi ký tự cho phép tham chiếu đến một tài nguyên. Dạng thông dụng của URL:

scheme://domain:port/path?query_string#fragment_id, trong đó:

+ scheme: chỉ giao thức truy nhập (http, https, ftp,...)

+ domain: tên miền, ví dụ www.google.com

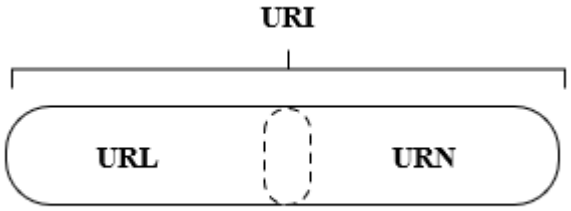
+ port: số hiệu cổng dịch vụ; với cổng chuẩn (http 80 hoặc https 443) thì không cần chỉ ra số hiệu cổng

+ path: đường dẫn đến tên file/trang

+ ?query_string: chuỗi truy vấn, gồm một hoặc một số cặp tên biến=giá trị. Ký tự và (&) được dùng để ngăn cách các cặp

+ fragment_id: một tên liên kết định vị đoạn trong trang.

URI (Uniform Resource Identifier) là một chuỗi ký tự dùng để nhận dạng một địa chỉ web hoặc một tên. URI có thể là URL hoặc URN (Uniform Resource Name), trong đó URN được dùng để nhận dạng tên của tài nguyên, còn URL được dùng để tìm địa chỉ, hoặc vị trí của tài nguyên [37].



Hình 1. 2. Các thành phần của URI

- *Web session và cookie:* Web session (phiên làm việc web) là một kỹ thuật cho phép tạo ra ứng dụng web có trạng thái (stateful) vận hành trên giao thức HTTP không trạng thái (stateless). Máy chủ web tạo ra và lưu một chuỗi định danh (ID) cho mỗi phiên (Session) theo yêu cầu của máy khách. Phiên cho phép máy chủ web nhận dạng người dùng và xâu chuỗi các yêu cầu HTTP của mỗi người dùng. Thời gian hoạt động của mỗi phiên tùy thuộc vào cấu hình máy chủ web. Ví dụ, sau đăng nhập thành công, máy chủ web tạo một phiên làm việc cho người dùng và không yêu cầu thông tin đăng nhập với các yêu cầu truy nhập tiếp theo cho đến khi kết thúc phiên làm việc.

Cookie còn gọi là HTTP cookie, hay Browser cookie là một mẩu thông tin do website gửi xuống và được lưu trên trình duyệt khi người dùng thăm website. Khi người dùng thăm website trong tương lai, website có thể đọc lại thông tin trong cookie

để biết các hoạt động trước đó của người dùng. Cookie thường được sử dụng để lưu thông tin phiên làm việc và duy trì trạng thái phiên làm việc [37].

- *Bộ diễn dịch và thực hiện các server script:* Các bộ diễn dịch và thực hiện các server script (script engine), hay mô tơ script có nhiệm vụ nạp, dịch và thực hiện từng dòng lệnh script trên máy chủ web. Do hầu hết các mô tơ script làm việc theo chế độ thông dịch nên tốc độ thường chậm so với các ứng dụng đã được biên dịch ra mã thực hiện. Nói chung, nhiều bộ diễn dịch và thực hiện các server script có thể được cài đặt và làm việc với một máy chủ web. Có thể kể đến một số mô tơ script thông dụng: Microsoft ASP, ASP.NET, PHP engine, Python engine, JVM/JSP [37].

- *Các server script:* Các server script là các đoạn mã được nhúng vào các trang web HTML để thực hiện các công việc xử lý dữ liệu và trả về kết quả để tạo nội dung cho trang web. Các server script được máy chủ web chuyển cho các mô tơ script để dịch và thực hiện. Kết quả thực hiện script được chuyển lại cho máy chủ web. Một số ngôn ngữ lập trình cho server script: ASP (VBScript), ASP.NET (C#), PHP, Python, JSP (Java) [37],…

- *Máy chủ cơ sở dữ liệu:* Máy chủ cơ sở dữ liệu thường được sử dụng để quản trị các cơ sở dữ liệu chứa dữ liệu tạo các trang web động. Một trang web động là trang web mà nội dung của nó chỉ được tạo ra khi có yêu cầu từ người dùng thông qua máy khách. Nội dung của các trang web động thường được lưu trữ trong cơ sở dữ liệu. Khi có yêu cầu truy vấn của người dùng, máy chủ web thực hiện các server script để truy nhập và xử lý dữ liệu từ cơ sở dữ liệu. Kết quả thực hiện script được chuyển lại cho web server để tạo nội dung trang web [37].

1.2. Tổng quan về tấn công web

1.2.1. Giới thiệu về tấn công web

Tấn công web, hay tấn công ứng dụng web là việc lợi dụng những điểm yếu, lỗ hổng tồn tại trên hệ thống website, ứng dụng web để thực hiện các hành vi khai thác, đánh cắp dữ liệu nhạy cảm tồn tại trên hệ thống [68]. Cũng theo [68], gần đây có tới 75% cuộc tấn công mạng được thực hiện ở cấp độ ứng dụng web. Luận án này tập trung nghiên cứu các giải pháp phát hiện các dạng tấn công web thường gặp, bao gồm tấn công SQLi, tấn công XSS, tấn công CMDi, tấn công duyệt đường dẫn, tấn công thay đổi giao diện.

Phần tiếp theo Luận án sẽ trình bày cụ thể hơn các nguy cơ và lỗ hổng bảo mật web được liệt kê theo tổ chức OWASP, gồm top 10 nguy cơ và lỗ hổng bảo mật web các phiên bản 2013, 2017 và 2021.

1.2.2. Top 10 nguy cơ và lỗ hổng bảo mật web theo OWASP

Các lỗ hổng bảo mật trong các ứng dụng web là các điểm yếu cho phép tin tặc tấn công đánh cắp dữ liệu người dùng, dữ liệu hệ thống, kiểm soát ứng dụng web, hoặc thậm chí kiểm soát cả hệ thống máy chủ chạy ứng dụng web. OWASP (Open Web Application Security Project - http://www.owasp.org) [41] là một dự án cộng đồng mở hoạt động với mục đích tăng cường an toàn cho các ứng dụng web. Năm 2021, OWASP khởi động dự án "OWASP Top 10 - 2021" nhằm đưa ra danh sách top 10 lỗ hổng bảo mật nghiêm trọng nhất trong các ứng dụng web năm 2021 nhằm thay thế cho danh sách top 10 lỗ hổng bảo mật nghiêm trọng nhất đưa ra năm 2017. Trước bản cập nhật năm 2017, tổ chức OWASP đã có các lần cập nhật danh sách các lỗ hổng nghiêm trọng cho ứng dụng web vào các năm 2003, 2004, 2007, 2010, 2013. Trong nội dung Luận án, NCS chỉ làm rõ sự thay đổi trong các lần cập nhật Top 10 lỗ hổng bảo mật của OWASP qua các năm gần nhất, gồm 2017 và 2021.

Bảng 1. 1. So sách thay đổi trong Top 10 lỗ hổng theo OWASP 2017, 2021

OWASP Top 10 – 2017	**OWASP Top 10 – 2021**
A1:2017-Injection	A01:2021-Broken Access Control
A2:2017-Broken Authentication	A02:2021-Cryptographic Failures
A3:2017-Sensitive Data Exposure	A03:2021-Injection
A4:2017-XML External Entities (XXE)	A04:2021-Insecure Design
A5:2017-Broken Access Control	A05:2021-Security Misconfiguration
A6:2017-Security Misconfiguration	A06:2021-Vulnerable and Outdated Components
A7:2017-Cross-Site Scripting (XSS)	A07:2021-Identification and Authentication Failures
A8:2017-Insecure Deserialization	A08:2021-Software and Data Integrity Failures
A9:2017-Using Components with Known Vulnerabilities	A09:2021-Security Logging and Monitoring Failures
A10:2017-Insufficient Logging&Monitoring	A10:2021-Server-Side Request Forgery

Dưới đây là mô tả ngắn gọn top 10 lỗ hổng bảo mật năm 2021 theo OWASP:

- A1. Broken Access Control

Vị trí của lỗ hổng này năm 2017 là A5, tuy nhiên đến năm 2021 đã tăng lên A1. Quản lý điều khiển truy cập không được kiểm soát chặt chẽ, tạo điều kiện cho người dùng này có thể truy cập được các thông tin nhạy cảm của người dùng khác hoặc sử dụng các quyền trái phép, chẳng hạn thay đổi thông tin người dùng khác, xóa

sửa các thông tin,… Ví dụ: khi một người dùng thông thường của một ứng dụng web có thể truy cập trái phép vào các chức năng quản trị - các chức năng mà chỉ có nhân viên, hoặc quản trị ứng dụng web mới có quyền truy cập.

- A2. Cryptographic Failures

Vị trí của lỗ hổng này năm 2017 là A3, tuy nhiên đến năm 2021 đã tăng lên A2. Nhiều ứng dụng web không có các cơ chế xác thực hoặc sử dụng các hàm mật mã đủ mạnh để bảo vệ các dữ liệu nhạy cảm, như thông tin thẻ tín dụng, số an sinh xã hội và thông tin xác thực người dùng. Kẻ tấn công có thể đánh cắp, hoặc chỉnh sửa các thông tin nhạy cảm để lạm dụng, hoặc trục lợi. Do vậy, cần có các cơ chế bổ sung, hoặc an toàn hơn để bảo vệ các thông tin nhạy cảm, như mã hóa và hạn chế quyền truy nhập vào các files chứa thông tin nhạy cảm (file lưu mật khẩu,…).

- A3. Injection

Vị trí của lỗ hổng này năm 2017 là A01, tuy nhiên đến năm 2021 đã xuống A03. Chèn mã là dạng lỗ hổng bảo mật cho phép tin tặc chèn mã vào dữ liệu gửi đến và được thực hiện trên hệ thống nạn nhân. Trong nhiều năm, chèn mã luôn được đánh giá là dạng lỗ hổng bảo mật nghiêm trọng nhất, bị khai thác phổ biến nhất và hậu quả của khai thác lỗi chèn mã cũng thường nặng nề nhất. Các dạng lỗ hổng chèn mã thường gặp bao gồm: Buffer overflow (Tràn bộ đệm), SQL injection (chèn mã SQL), XPath/XQuery injection (chèn mã XPath/XQuery), LDAP lookups / injection (chèn mã LDAP) và Shell command injection (chèn các lệnh Shell).

- A4. Insecure Design

Đây là một lỗ hổng mới xuất hiện, không thuộc Top 10 năm 2017. Lỗ hổng dạng này xuất hiện từ khâu thiết kế phần mềm do tính bảo mật không được chú ý trong khâu thiết kế. Để giảm thiểu lỗi này, các giải pháp bảo mật cần được tích hợp vào ngay từ khâu thiết kế phần mềm.

- A5. Security Misconfiguration

Vị trí của lỗ hổng này năm 2017 là A6, tuy nhiên đến năm 2021 đã tăng lên A5. Lỗi cấu hình bảo mật sai là vấn đề thường gặp và đây thường là kết quả của cấu hình mặc định không an toàn, cấu hình không đầy đủ, hoặc việc lưu trên hệ thống cloud, tiêu đề HTTP được cấu hình sai và thông báo lỗi chứa các thông tin nhạy cảm. Tất cả các hệ điều hành, thư viện và ứng dụng không chỉ được cấu hình an toàn mà còn phải được cập nhất các bản vá kịp thời.

- A6. Vulnerable and Outdated Components

Vị trí của lỗ hổng này năm 2017 là A9, tuy nhiên đến năm 2021 tăng lên A6. Các thành phần, bao gồm các thư viện, các framework và các mô đun phần mềm hầu như được chạy với quyền truy nhập đầy đủ như người dùng kích hoạt ứng dụng. Nếu một thành phần có chứa lỗ hổng bị khai thác có thể gây ra việc mất mát nhiều dữ liệu, hoặc máy chủ có thể bị chiếm quyền điều khiển. Các ứng dụng sử dụng các thành phần chứa lỗ hổng đã biết có thể làm suy giảm khả năng phòng vệ của ứng dụng và cho phép thực hiện nhiều loại tấn công lên hệ thống.

- A7. Identification and Authentication Failures

Vị trí của lỗ hổng này năm 2017 là A2, tuy nhiên đến năm 2021 đã giảm xuống A7. Khâu xác thực (authentication) và trao quyền (authorisation) được sử dụng khá phổ biến trong các ứng dụng web. Nếu các khâu xác thực và trao quyền không đủ mạnh thì đó là lỗ hổng để kẻ tấn công truy nhập đánh cắp thông tin.

- A8. Software and Data Integrity Failures

Đây là một lỗ hổng mới, tuy nhiên nó cũng bao gồm một phần từ lỗ hổng A8 thuộc Top 10 năm 2017. Các lỗi về tính toàn vẹn của phần mềm và dữ liệu liên quan đến mã và cơ sở hạ tầng không được bảo vệ trước các cuộc tấn công vào tính toàn vẹn. Một ví dụ về điều này là khi một ứng dụng dựa vào các plugin, thư viện hoặc mô-đun từ các nguồn, kho lưu trữ và mạng phân phối nội dung (CDN) không đáng tin cậy. Đường dẫn CI/CD không an toàn có thể dẫn đến khả năng truy cập trái phép, mã độc hại hoặc xâm nhập hệ thống. Cuối cùng, nhiều ứng dụng hiện bao gồm chức năng tự động cập nhật, nơi các bản cập nhật được tải xuống mà không cần xác minh tính toàn vẹn đầy đủ và được áp dụng cho ứng dụng đáng tin cậy trước đó. Những kẻ tấn công có khả năng tải lên các bản cập nhật của riêng họ để được phân phối và chạy trên tất cả các bản cài đặt.

- A9. Security Logging and Monitoring Failures

Vị trí của lỗ hổng này năm 2017 là A10, tuy nhiên đến năm 2021 tăng lên A09. Lỗi ghi nhật ký và giám sát không đầy đủ cùng với việc tích hợp thiếu hoặc chưa hiệu quả đồng bộ với quá trình phản ứng sự cố, cho phép kẻ tấn công vào hệ thống và có thể chuyển hướng sang nhiều mục tiêu khác trong hệ thống.

- A10. Server Side Request Forgery (SSRF)

SSRF là một lỗ hổng mới, xảy ra bất cứ khi nào ứng dụng web đang tìm nạp tài nguyên từ xa mà không xác thực URL do người dùng cung cấp. Nó cho phép kẻ tấn công ép ứng dụng gửi một yêu cầu thủ công đến một điểm đến không mong muốn,

ngay cả khi được bảo vệ bởi tường lửa, VPN hoặc một loại danh sách kiểm soát truy cập mạng (ACL) khác.

Nhận xét: Qua các lần cập nhật 2013, 2017 và 2021, một trong các lỗ hổng ứng dụng phổ biến là chèn mã (A1-Injection). Tuy chèn mã có thay đổi về vị trí trong Top 10 2021, tuy nhiên chúng vẫn nằm trong Top 10. Một số lỗ hổng mới xuất hiện cũng được OWASP liệt kê trong danh sách Top 10. Trên thực tế đã xuất hiện các loại hình tấn công web khai thác các lỗ hổng này.

Phần tiếp theo của Luận án sẽ mô tả chi tiết các các dạng tấn công web thường gặp.

1.2.3. Các dạng tấn công web thường gặp

Có thể kể đến các dạng tấn công, xâm nhập phổ biến vào các website, ứng dụng web (gọi tắt là tấn công web), bao gồm tấn công chèn mã SQL (SQLi – SQL injection), tấn công XSS (Cross-Site Scripting), tấn công CSRF (Cross-site Request Forgery) [89], tấn công chèn dòng lệnh (CMDi – Command injection), tấn công duyệt đường dẫn, tấn công DoS/DDoS và tấn công thay đổi giao diện [21] [37] [41]. Các tấn công web, gồm SQLi, XSS, CMDi, Duyệt đường dẫn được phát hiện sử dụng thông tin trích xuất từ dữ liệu weblog; Tấn công thay đổi giao diện trang web được phát hiện dựa trên đặc trưng văn bản và ảnh màn hình trang web.

1.2.3.1. Tấn công SQLi

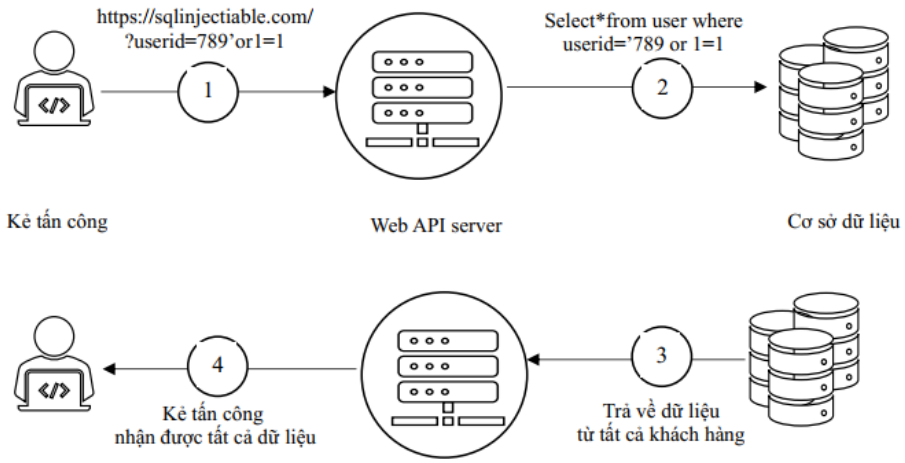
Tấn công chèn mã SQL là một kỹ thuật cho phép kẻ tấn công chèn mã SQL vào dữ liệu gửi đến máy chủ và cuối cùng được thực hiện trên máy chủ cơ sở dữ liệu [3] [55] [62]. Tùy vào mức độ tinh vi, tấn công chèn mã SQL có thể cho phép kẻ tấn công (1) vượt qua các khâu xác thực người dùng, (2) chèn, sửa đổi, hoặc xóa dữ liệu, (3) đánh cắp các thông tin trong cơ sở dữ liệu và (4) chiếm quyền điều khiển hệ thống máy chủ cơ sở dữ liệu. Tấn công chèn mã SQL là dạng tấn công thường gặp ở các ứng dụng web, các trang web có kết nối đến cơ sở dữ liệu. Hình 1. 3 biểu diễn một dạng tấn công SQLi nhằm trích xuất dữ liệu trái phép từ cơ sở dữ liệu.

Có 2 nguyên nhân của lỗ hổng trong ứng dụng cho phép thực hiện tấn công chèn mã SQL: (1) Dữ liệu đầu vào từ người dùng hoặc từ các nguồn khác không được kiểm tra hoặc kiểm tra không kỹ lưỡng và (2) Sử dụng các câu lệnh SQL động trong ứng dụng, trong đó có thao tác nối dữ liệu người dùng với mã lệnh SQL gốc.

1.2.3.2. Tấn công XSS

Tấn công Cross-Site Scriting (XSS – Mã script liên site, liên miền) là một trong các dạng tấn công phổ biến nhất vào các ứng dụng web. XSS xuất hiện từ khi

trình duyệt bắt đầu hỗ trợ ngôn ngữ JavaScript. Mã tấn công XSS được nhúng trong trang web chạy trong lòng trình duyệt với quyền truy nhập của người dùng, có thể truy nhập các thông tin nhạy cảm của người dùng lưu trong trình duyệt [35] [50] [55]. Do mã XSS chạy trong lòng trình duyệt nên nó miễn nhiễm với các trình quét các phần mềm độc hại và các công cụ bảo vệ hệ thống.



Hình 1. 3. Một dạng tấn công SQLi (SQL Injection)

Tấn công XSS thường xuất hiện khi trang web cho phép người dùng nhập dữ liệu và sau đó hiển thị dữ liệu lên trang. Kẻ tấn công có thể khéo léo chèn mã script vào trang và mã script của kẻ tấn công được thực hiện khi người dùng khác thăm lại trang web đó, nếu ứng dụng không được lọc các mã đặc biệt này thì có thể sẽ là nguồn cho đối tượng tấn công khai thác chiếm đoạt tài khoản, đầu độc cookie, hay tấn công từ chối dịch vụ [55]. Tùy theo mục đích và mức độ tinh vi, XSS có thể cho phép kẻ tấn công thực hiện các thao tác sau trên hệ thống nạn nhân:

- Đánh cắp thông tin nhạy cảm của người dùng lưu trong Cookie của trình duyệt

- Giả mạo hộp đối thoại đăng nhập để đánh cắp mật khẩu

- Bắt phím gõ từ người dùng để đánh cắp thông tin về tài khoản ngân hàng, email, và thông tin đăng nhập các dịch vụ trả tiền,...

- Sử dụng trình duyệt để quét các cổng dịch vụ trong mạng LAN

- Lén lút cấu hình lại bộ định tuyến nội bộ để bỏ qua tường lửa của mạng nội bộ

- Tự động thêm người dùng ngẫu nhiên vào tài khoản mạng xã hội

- Tạo môi trường cho tấn công CSRF.

Có thể chia tấn công XSS thành 3 loại chính: Stored XSS (XSS lưu trữ), Reflected XSS (XSS phản chiếu) và DOM-based/Local XSS (XSS dựa trên DOM hoặc cục bộ) [37] [55].

1.2.3.3. Tấn công CMDi

Tấn công CMDi là việc khai thác vào lỗ hổng ứng dụng web cho phép kẻ tấn công đưa các lệnh của hệ điều hành trên máy chủ ứng dụng vào dữ liệu người dùng dưới dạng cookie, biểu mẫu, tiêu đề HTTP để xâm phạm ứng dụng và dữ liệu trên máy chủ [4] [96]. Tấn công CMDi có thể được chia thành 2 loại [4] [96]: (1) Chèn lệnh dựa trên kết quả: kẻ tấn công sẽ chèn một mã độc hại vào ứng dụng và ứng dụng này phản hồi với lỗ hổng. Dựa trên đầu ra là lỗ hổng được phản hồi kẻ tấn công có thể sửa đổi thông tin đầu vào để thu thập thông tin cần thiết; và (2) Blind Command Injections: kẻ tấn công chèn mã độc hại vào ứng dụng dễ bị tấn công và ứng dụng này không gửi lại phản hồi nào. Do đó, kết quả không được hiển thị trên màn hình nên lúc này kẻ tấn công có thể sử dụng hai phương pháp là kỹ thuật dựa trên thời gian (Time based technique) hoặc kỹ thuật dựa trên tệp (File based technique) để thu thập thông tin cần thiết.

1.2.3.4. Tấn công duyệt đường dẫn

Tấn công duyệt đường dẫn là việc kẻ tấn công khai thác lỗ hổng ứng dụng web để thực hiện đọc các tệp tùy ý trên máy chủ ứng dụng web, dữ liệu có thể là mã ứng dụng và cơ sở dữ liệu, thông tin đăng nhập của hệ thống và tệp nhạy cảm. Trong một số trường hợp, kẻ tấn công có thể ghi vào các tệp tùy ý trên máy chủ, cho phép sửa đổi dữ liệu và cuối cùng là kiểm soát hoàn toàn máy chủ [52] [53] [108]. Bảng 1. 2 liệt kê một số mã tấn công duyệt đường dẫn vào máy chủ web [53] [108].

Bảng 1. 2. Một số mẫu URL tấn công duyệt đường dẫn vào máy chủ web [108]

http://example.com/../../../../../some/file
http://example.com/..%255c..%255c..%255csome/file
http://example.com/..%u2216..%u2216some/file

1.2.3.5. Tấn công thay đổi giao diện

Tấn công thay đổi giao diện website (Website Defacement Attack) là dạng tấn công làm thay đổi nội dung trang web và thông qua đó thay đổi hình thức hiển thị của trang web [29] [82]. Tấn công thay đổi giao diện trang web thường được thực hiện thông qua việc khai thác một số lỗ hổng tồn tại trên hệ thống web, như SQLi hoặc

XSS. Có nhiều động cơ dẫn đến các cuộc tấn công thay đổi giao diện các website. Các động cơ tấn công chính có thể được chia thành 3 nhóm chính:

- Cảnh báo các lỗ hổng trên các website: các cuộc tấn công thuộc nhóm này thường được thực hiện bởi các tin tặc "mũ trắng", hoặc các nhóm bảo mật mạng, nhằm cảnh báo về các lỗ hổng bảo mật tồn tại trên các website đến người quản trị.

- Thể hiện bản thân: các cuộc tấn công thuộc nhóm này thường được thực hiện bởi các tin tặc trẻ tuổi, hoặc các cá nhân mới tìm hiểu về bảo mật, thích thể hiện bản thân, thích được nổi tiếng.

- Trả thù, hoặc các mục đích tôn giáo, chính trị: các cuộc tấn công thuộc nhóm này thường được thực hiện bởi các tin tặc chuyên nghiệp, nhằm trả thù cá nhân, hoặc có liên quan đến các xung đột tôn giáo, chính trị, hoặc ý thức hệ.

1.2.3.6. Tấn công Phishing

Tấn công phishing là một hình thức tấn công, trong đó kẻ tấn công thực hiện giả mạo các trang web hợp pháp để thu thập thông tin nhạy cảm của người dùng nhằm thực hiện các hành vi phạm pháp, như chiếm đoạt các thông tin nhạy cảm, hoặc tài sản tài chính. Hình thức tấn công này thường bắt đầu khi kẻ tấn công gửi một email, hoặc tin nhắn có vẻ đáng tin cậy, yêu cầu người dùng nhấp vào đường link (URL) để cập nhật hoặc xác minh thông tin cá nhân. Khi người dùng nhấp vào đường link, họ sẽ bị chuyển hướng đến một trang web giả mạo và được yêu cầu cung cấp thông tin cá nhân hoặc tài khoản ngân hàng. Trong nghiên cứu [48] Liu và cộng sự đã tạo ra một trang web độc hại giống với trang thật, trong trang web độc hại này đã sử dụng các mã XSS tạo liên kết độc hại với cửa sổ đăng nhập, sau đó gửi cho người dùng qua email và dụ người dùng nhấp vào liên kết đó.

Hình thức tấn công phishing cũng là một hình thức diễn ra khá phổ biến hiện nay và có sử dụng chèn mã XSS như trong nghiên cứu [48].Tuy nhiên trong nội dung của luận án không thực hiện phát hiện hình thức tấn công này. Luận án chỉ tập trung phát hiện các hình thức tấn công web thường gặp như: SQLi, XSS, CMDi, Duyệt đường dẫn và hình thức tấn công thay đổi giao diện.

1.3. Phát hiện tấn công web

1.3.1. Khái quát về phát hiện tấn công web

Trước mức độ nguy hiểm của các cuộc tấn công vào dịch vụ web, nhiều giải pháp phòng chống đã được nghiên cứu và ứng dụng vào thực tế nhằm phát hiện và ngăn chặn các cuộc tấn công này nhằm bảo vệ website, ứng dụng web và người dùng

web. Nói chung, có 3 hướng tiếp cận phòng thủ đối với các cuộc tấn công này, bao gồm (1) kiểm tra, xác thực tất cả dữ liệu đầu vào, (2) giảm các bề mặt tấn công và (3) sử dụng chiến lược "phòng thủ theo chiều sâu" [8] [41] [111]. Cụ thể, hướng tiếp cận (1) yêu cầu tất cả dữ liệu đầu vào cho các ứng dụng web phải được kiểm tra kỹ lưỡng sử dụng các bộ lọc dữ liệu đầu vào và chỉ những đầu vào hợp pháp mới được chuyển sang các bước tiếp theo để xử lý. Mặt khác, hướng tiếp cận (2) yêu cầu chia ứng dụng web thành nhiều phần và sau đó áp dụng các biện pháp điều khiển truy cập phù hợp để hạn chế quyền truy cập của người dùng. Đối với hướng tiếp cận (3), một số biện pháp phòng thủ được triển khai trong các lớp kế tiếp nhau để bảo vệ các trang web, ứng dụng web và người dùng web.

Trong các mục tiếp theo, luận án khảo sát một số giải pháp, công cụ và kỹ thuật cho giám sát, phát hiện các dạng tấn công web – là một phần quan trọng trong hướng tiếp cận (3) sử dụng chiến lược "phòng thủ theo chiều sâu".

1.3.2. Các giải pháp và công cụ phát hiện tấn công web

Có nhiều giải pháp, công cụ phát hiện tấn công web được phát triển và triển khai ứng dụng trên thực tế, như VNCS Web Monitoring [61], Nagios Web Application Monitoring Software [103], Site24x7 Website Defacement Monitoring [78], ModSecurity [25], Snort IDS [104], Acunetix Vulnerability Scanner, App Scanner [6] và Abbey Scan[5], WebOrion Defacement Monitor[6], Visualping[7], Imperva Application Security [109], Fluxguard [77].

VNCS Web monitoring [61] là giải pháp cho phép giám sát nhiều website đồng thời dựa trên thu thập, xử lý và phân tích log truy cập sử dụng nền tảng Splunk do công ty cổ phần Công nghệ An ninh không gian mạng Việt Nam phát triển. Hạn chế của VNCS Web monitoring là vấn đề vận chuyển khối lượng log từ các máy chủ về trung tâm xử lý đòi hỏi đường truyền ổn định với thông lượng lớn. Ngoài ra, đây là giải pháp thương mại nên chi phí lắp đặt và vận hành tương đối cao.

Nagios Web Application Monitoring Software [103] là bộ công cụ cho phép giám sát các website, các ứng dụng web, các giao dịch web và dịch vụ web, bao gồm các tính năng như giám sát tính sẵn dùng, giám sát địa chỉ URL, giám sát trạng thái HTTP, giám sát nội dung, phát hiện chiếm quyền điều khiển website và giám sát

[5] "Abbey Scan," Misterscanner, [Online]. Available: https://misterscanner.com. [Accessed 5 2021].
[6] "WebOrion Defacement Monitor," Banff Cyber Technologies, [Online]. Available: https://www.weborion.io/website-defacement-monitor/. [Accessed 5 2021].
[7] "What is Visualping?," Visualping, [Online]. Available: https://visualping.io/. [Accessed 7 2023].

chứng chỉ số SSL. Hạn chế chủ yếu của Nagios là giải pháp thương mại nên chi phí lắp đặt và vận hành cao.

Site24x7 Website Defacement Monitoring [78] là dịch vụ giám sát, phát hiện tấn công thay đổi giao diện website. Ưu điểm của dịch vụ này là cài đặt đơn giản và chi phí đầu tư ban đầu thấp. Tuy nhiên, dịch vụ chỉ phù hợp với các trang web có nội dung tĩnh, hoặc ít thay đổi và không phù hợp với các trang có nội dung động, như các trang thương mại điện tử hay các diễn đàn.

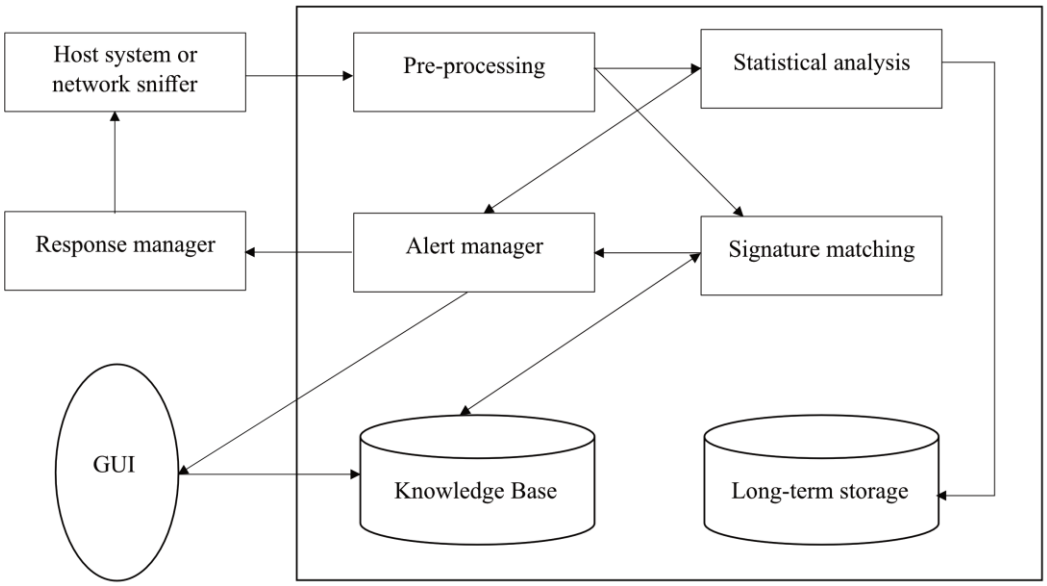
ModSecurity [25] là một dạng tường lửa ứng dụng web (WAF - Web Application Firewall) được sử dụng để lọc các truy vấn người sử dụng gửi đến các máy chủ web. ModSecurity có thể được cài đặt trên máy chủ để bảo vệ nhiều website, ngăn chặn hầu hết các dạng tấn công vào website như SQLi và XSS [15]. Ưu điểm của ModSecurity so với các giải pháp WAF thương mại là mã mở, miễn phí, gọn nhẹ và được tích hợp sâu vào các máy chủ web, như máy chủ Apache HTTP. Hạn chế của ModSecurity là khả năng tương thích với các máy chủ web.

1.3.3. Các kỹ thuật phát hiện tấn công web

Có nhiều kỹ thuật phát hiện tấn công web được đề xuất và ứng dụng trong những năm qua. Mục này trình bày 2 nhóm kỹ thuật phát hiện tấn công web sử dụng phổ biến, bao gồm (1) phát hiện dựa trên chữ ký, mẫu hoặc tập luật [1] và (2) phát hiện dựa trên bất thường [32] nói chung và phát hiện sử dụng các mô hình học máy và học sâu nói riêng.

1.3.3.1. Phát hiện dựa trên chữ ký và tập luật

Phát hiện tấn công, xâm nhập nói chung và phát hiện các dạng tấn công vào ứng dụng web nói riêng dựa trên dấu hiệu, chữ ký (signature), mẫu (pattern), hoặc luật (rule) là phương pháp phát hiện tấn công, xâm nhập dựa trên việc tìm hay so khớp tập chữ ký của các tấn công, xâm nhập đã biết với các dữ liệu giám sát thu thập được. Một tấn công, xâm nhập được phát hiện khi có ít nhất một so khớp chữ ký thành công.



Hình 1. 4. Kiến trúc giám sát phát hiện tấn công, xâm nhập dựa trên chữ ký

Hình 1. 4 mô tả kiến trúc giám sát, phát hiện tấn công, xâm nhập dựa trên chữ ký. Theo đó, chữ ký của các tấn công, xâm nhập đã biết được xây dựng và lưu trữ vào cơ sở dữ liệu (Knowledge Base) trong giai đoạn xây dựng. Trong giai đoạn giám sát, các dữ liệu được thu thập từ hệ thống, hoặc giao tiếp mạng được đưa qua các khâu tiền xử lý để làm sạch và chuẩn hóa. Tiếp theo, các dữ liệu giám sát được so khớp (Signature matching) với cơ sở dữ liệu chữ ký để phát hiện tấn công, xâm nhập. Kết quả phát hiện được chuyển cho các thành phần cảnh báo (Alert manager) và phản hồi (Response manager) để xử lý. Để hệ thống có khả năng phát hiện các dạng tấn công mới, cơ sở dữ liệu chữ ký cần được cập nhật thường xuyên.

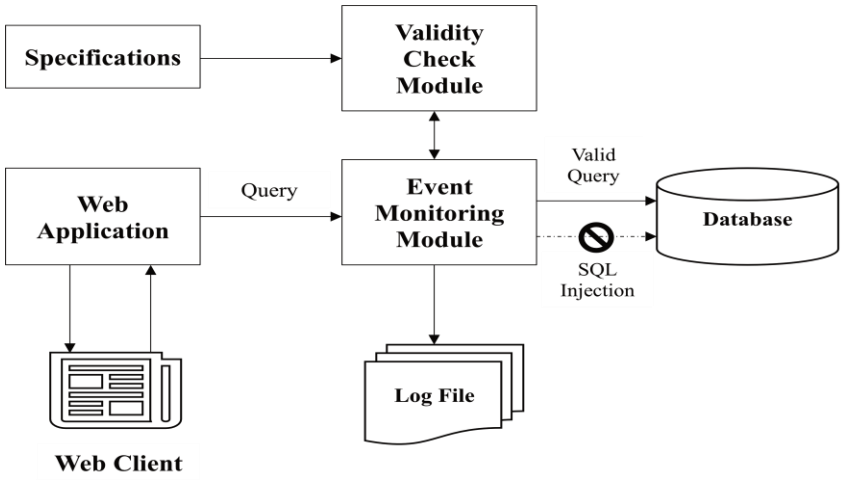
Kỹ thuật phát hiện tấn công, xâm nhập dựa trên chữ ký có ưu điểm là có khả năng phát hiện nhanh và chính xác các dạng tấn công đã biết. Tuy nhiên, kỹ thuật này có nhược điểm là không có khả năng phát hiện các dạng tấn công chưa có trong tập chữ ký phát hiện, ví dụ tấn công khai thác lỗ hổng zero-day do chữ ký của chúng chưa được cập nhật vào cơ sơ dữ liệu. Ngoài ra, việc xây dựng và cập nhật cơ sở dữ liệu chữ ký thường được thực hiện thủ công, nên tốn nhiều công sức.

Có thể liệt kê một số nghiên cứu phát hiện tấn công web sử dụng các tập chữ ký hoặc luật như: SQLCheck [96], OWASP ModSecurity Core Rule Set [25], Abhishek Kumar Baranwal [9] và cộng sự, Jesús Díaz-Verdejo [26] và cộng sự, SQL-IDS [51], XSS-GUARD [75] và SQLGuard [17].

OWASP ModSecurity Core Rule Set (CRS) [25] là một bộ luật được phát triển bởi dự án OWASP để phát hiện nhiều loại tấn công web được liệt kê trong Top 10 của OWASP với tỷ lệ cảnh báo sai thấp. CRS có thể được sử dụng với ModSecurity –mô-

đun đi kèm với máy chủ web Mozilla Apache. Ưu điểm của CRS là được hỗ trợ và cập nhật thường xuyên bởi OWASP và cộng đồng bảo mật web toàn cầu. Tuy nhiên, vì CRS bao gồm một số lượng khá lớn các luật nên nó tương đối cồng kềnh và có thể gặp sự cố tương thích khi được tích hợp vào các tường lửa ứng dụng web khác hoặc được sử dụng với các máy chủ web khác, chẳng hạn như máy chủ web Microsoft IIS.

SQL-IDS [51] là một hệ thống phát hiện tấn công SQLi dựa trên đặc tả. Ý tưởng chính của kỹ thuật này là xây dựng một tập luật đặc tả cấu trúc của các câu lệnh SQL hợp lệ mà ứng dụng sinh và chuyển đến máy chủ cơ sở dữ liệu để thực hiện. Trên cơ sở tập luật mô tả cấu trúc câu truy vấn SQL hợp lệ, hệ thống giám sát các truy vấn SQL, tiến hành phân tích từ vựng, cú pháp và cấu trúc các câu lệnh SQL gửi đến để phân loại chúng. Hình *Hình 1. 5* minh họa kiến trúc SQL-IDS.

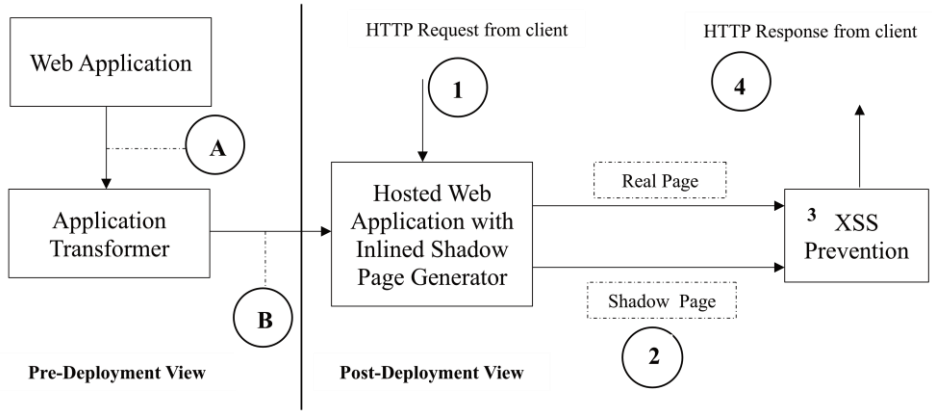


Hình 1. 5. Kiến trúc hệ thống SQL-IDS

Các kết quả thử nghiệm ban đầu cho thấy hệ thống đề xuất có độ trễ thấp và phát hiện chính xác tất cả các tấn công SQLi với tỷ lệ cảnh báo sai là 0%. Ngoài ra, SQL-IDS được cài đặt như một proxy giữa máy chủ web và máy chủ cơ sở dữ liệu nên có thể sử dụng để bảo vệ nhiều website mà không yêu cầu chỉnh sửa mã nguồn của trang web và cơ sở dữ liệu. Tuy nhiên, hệ thống đề xuất chưa được thử nghiệm với các tập dữ liệu lớn và cần được đánh giá toàn diện hơn trong môi trường thử nghiệm thực tế. Ngoài ra, việc xây dựng tập luật được thực hiện thủ công nên tiêu tốn nhiều thời gian, đặc biệt với các hệ thống ứng dụng web có quy mô lớn.

XSS-GUARD [75] là một framework cho phép giám sát và ngăn chặn các dạng tấn công XSS thông qua việc sinh và so sánh một trang bóng (shadow page) với trang thực (real page) từ phản hồi của máy chủ web trước khi gửi trang thực cho máy khách. Trang bóng là một trang được sinh song hành với trang thực từ phản hồi của máy chủ web với cùng mã gốc của trang, nhưng với dữ liệu đầu vào sạch (không có mã script)

được tạo tự động có cùng độ dài với dữ liệu đầu vào thực. Hình 1. 6 biểu diễn kiến trúc của XSS-GUARD.



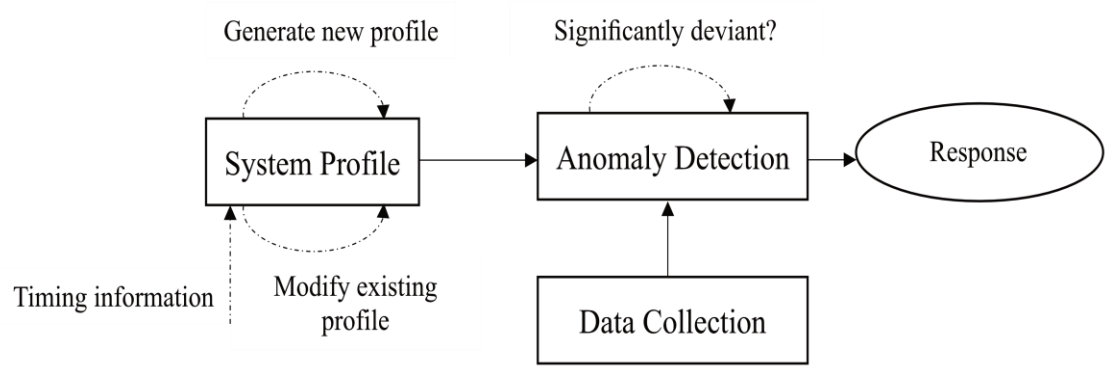
Hình 1. 6. Kiến trúc của XSS-GUARD

Ưu điểm của XSS-GUARD là không cần phải sử dụng các bộ lọc XSS cồng kềnh và phải cập nhật thường xuyên. Các thử nghiệm cho thấy, XSS-GUARD có khả năng ngăn chặn khá hiệu quả các dạng tấn công XSS và các thủ thuật vượt qua các bộ lọc đề xuất bởi OWASP [41]. Tuy nhiên, kỹ thuật đề xuất làm tăng tải đáng kể cho máy chủ khi phải liên tục sinh các trang bóng song hành với trang thực cho mỗi yêu cầu từ người dùng. Ngoài ra, hệ thống XSS-GUARD dựa vào trình duyệt để thực hiện một số tính năng trên yêu cầu HTTP, như quét, tách từ, giảm nội dung,… nên sẽ gặp phải vấn đề tương thích giữa các nền tảng trình duyệt khác nhau.

SQLGuard [17] và *SQLCheck* [96] rất giống nhau vì cả hai đều sử dụng xác thực cây cú pháp của các lệnh SQL để phát hiện các cuộc tấn công SQLi. Do đó, luận án chỉ thực hiện đánh giá về SQLGuard. SQLGuard là một hệ thống phát hiện và ngăn chặn SQLi dựa trên việc xác thực cây cú pháp của lệnh SQL. SQLGuard xây dựng và so sánh cây cú pháp của lệnh SQL trước khi chèn dữ liệu đầu vào của người dùng và cây cú pháp của nó sau khi chèn dữ liệu đầu vào của người dùng. SQLGuard có thể phát hiện các cuộc tấn công SQLi vì đầu vào SQLi thay đổi cây cú pháp của lệnh SQL trong khi đầu vào hợp lệ không thay đổi cây cú pháp của lệnh SQL. Các thử nghiệm xác nhận rằng SQLGuard có thể phát hiện các cuộc tấn công SQLi một cách hiệu quả. Tuy nhiên, phương pháp được đề xuất yêu cầu xây dựng thủ công các cây cú pháp của tất cả các lệnh SQL hợp lệ của ứng dụng web. Hơn nữa, nó yêu cầu sửa đổi mã nguồn Java của ứng dụng web, điều này không phải khi nào cũng có thể thực hiện được.

1.3.3.2. Phát hiện dựa trên bất thường

Phát hiện tấn công web dựa trên bất thường, hay dị thường dựa trên giả thiết các hành vi tấn công, xâm nhập vào ứng dụng web có quan hệ mật thiết với các hành vi bất thường. Quá trình xây dựng và triển khai một hệ thống phát hiện xâm nhập dựa trên bất thường gồm 2 giai đoạn: (1) huấn luyện và (2) phát hiện. Trong giai đoạn huấn luyện, hồ sơ (profile) của đối tượng trong chế độ làm việc bình thường được xây dựng. Để thực hiện giai đoạn huấn luyện, cần giám sát đối tượng trong một khoảng thời gian đủ dài để thu thập được đầy đủ dữ liệu mô tả các hành vi của đối tượng trong điều kiện bình thường làm dữ liệu huấn luyện. Tiếp theo, thực hiện huấn luyện dữ liệu để xây dựng mô hình phát hiện, hay hồ sơ của đối tượng. Trong giai đoạn phát hiện, thực hiện giám sát hành vi hiện tại của hệ thống và cảnh báo nếu có khác biệt rõ nét giữa hành vi hiện tại và các hành vi lưu trong hồ sơ của đối tượng. *Hình 1. 7* mô tả mô hình phương pháp phát hiện xâm nhập dựa trên bất thường.



Hình 1. 7. Mô hình phương pháp phát hiện xâm nhập dựa trên bất thường

Ưu điểm của phát hiện xâm nhập dựa trên bất thường là có tiềm năng phát hiện các loại tấn công, xâm nhập mới mà không yêu cầu biết trước thông tin về chúng. Tuy nhiên, phương pháp này có tỷ lệ cảnh báo sai tương đối cao so với phương pháp phát hiện dựa trên chữ ký. Điều này làm giảm khả năng ứng dụng thực tế của phát hiện xâm nhập dựa trên bất thường. Ngoài ra, nó cũng tiêu tốn nhiều tài nguyên hệ thống cho việc xây dựng hồ sơ đối tượng và phân tích hành vi hiện tại do cần xử lý một lượng lớn dữ liệu. Mục tiếp theo khảo sát một số đề xuất tiêu biểu trong nhóm phát hiện tấn công web dựa trên bất thường theo 2 dạng: (a) các đề xuất phát hiện tấn công web thường gặp, như SQLi, XSS và (b) các đề xuất phát hiện tấn công thay đổi giao diện.

a. Các đề xuất phát hiện tấn công web thường gặp, như SQLi, XSS

Các đề xuất phát hiện tấn công web thường gặp tiêu biểu có thể kể đến, gồm AMNESIA [39], Swaddler [22], CANDID [16] và Torrano-Gimenez và cộng sự

[100], S. Sharma và cộng sự [91], S. Saleem và cộng sự [85], Saiyu Hao cùng cộng sự [40], Betarte và cộng sự [14], Liang và cộng sự [58], và Pan và cộng sự [76].

AMNESIA [39] là một hệ thống phát hiện tấn công web dựa trên sự bất thường, đầu tiên hệ thống quét mã ứng dụng web để tìm và phân tích tất cả các truy vấn SQL được sử dụng. Sau đó, mỗi truy vấn SQL được mô hình hóa bằng phương pháp tự động hóa hữu hạn không xác định (NDFA). Các thử nghiệm xác nhận rằng AMNESIA có thể phát hiện tất cả các cuộc tấn công SQLi trong các tình huống thử nghiệm. Tuy nhiên, nó yêu cầu quyền truy cập vào mã nguồn của ứng dụng web, điều này không phải khi nào cũng khả thi trong thực tế.

Cùng nhóm với AMNESIA [39], Swaddler [22] là một hệ thống phát hiện tấn công web dựa trên sự bất thường sử dụng một cách tiếp cận khá khác. Swaddler phân tích trạng thái bên trong của ứng dụng web và tìm hiểu mối quan hệ giữa các điểm thực thi quan trọng của ứng dụng và trạng thái bên trong của nó để phát hiện các trạng thái không nhất quán hoặc bất thường. Ưu điểm của Swaddler là nó có thể được sử dụng để bảo vệ nhiều trang web chạy trên cùng một mô tơ PHP và việc xây dựng mô hình phát hiện có thể được thực hiện tự động. Tuy nhiên, nó yêu cầu sửa đổi mô tơ PHP để theo dõi luồng thực thi của ứng dụng web và đây có thể là một khó khăn trong triển khai thực tế.

Theo một hướng khác, CANDID [16] đầu tiên sử dụng phân tích động để trích xuất các truy vấn SQL hợp pháp của ứng dụng web trong thời gian chạy và sau đó xây dựng cấu hình phát hiện bằng phương pháp cây cú pháp. Mỗi truy vấn SQL cần giám sát được chuyển đổi thành một cây cú pháp và sau đó cây cú pháp này được so sánh với cây tiêu chuẩn của cấu hình phát hiện để tìm kiếm sự khác biệt. Nếu tìm thấy sự không phù hợp, truy vấn SQL sẽ bị chặn và ghi lại. Ưu điểm của CANDID so với AMNESIA [39] là không cần truy cập mã nguồn của ứng dụng web. Tuy nhiên, phương pháp đề xuất chỉ hoạt động với các ứng dụng web được phát triển và vận hành trên nền tảng Java.

Torrano-Gimenez và cộng sự [100] đề xuất xây dựng một hệ thống phát hiện các dạng tấn công vào ứng dụng web dựa trên bất thường. Hệ thống được cài đặt dưới dạng một Proxy hay tường lửa ứng dụng web (WAF) đứng giữa máy khách và máy chủ web. Trong giai đoạn huấn luyện, một file XML mô tả tập các yêu cầu hợp lệ của một website được sinh tự động từ dữ liệu huấn luyện bằng phương pháp thống kê. Sau đó, trong giai đoạn phát hiện, file XML được sử dụng để phân loại các yêu cầu gửi đến máy chủ web. Nếu yêu cầu được xác định là bình thường thì được chuyển tiếp tới máy chủ web xử lý. Ngược lại, các yêu cầu được xác định là bất thường sẽ bị

chặn. Các thử nghiệm cho thấy, hệ thống đề xuất cho độ chính xác cao và tỷ lệ sai thấp khi lượng dữ liệu huấn luyện đủ lớn (từ 10.000 yêu cầu trở lên).

Trong nhóm các đề xuất phát hiện tấn công dựa trên bất thường, theo hướng (3) các giải pháp ứng dụng học máy là phương pháp đang được sử dụng rộng rãi và có tiềm năng nhất [14] [73]. Một số công trình nghiên cứu trong nhóm này gồm có: Betarte và cộng sự [14], Liang và cộng sự [58], và Pan và cộng sự [76], Sharma và cộng sự [91], Saleem và cộng sự [85], Hao cùng cộng sự [40]. Đây là các đề xuất sử dụng các thuật toán học máy, học sâu để xây dựng các mô hình phát hiện và sau đó sử dụng các mô hình này để phát hiện các cuộc tấn công web có thể xảy ra. Các thuật toán học máy được sử dụng có thể là phương pháp học truyền thống, chẳng hạn như naïve bayes, cây quyết định, SVM, rừng ngẫu nhiên [10] [71], hoặc các thuật toán học sâu, như CNN và RNN hay BiLSTM [40] [94].

Betarte và cộng sự [14] đề xuất sử dụng các mô hình phân loại 1-lớp và mô hình phân loại dựa trên n-gram để nâng cao khả năng phát hiện và độ chính xác phát hiện tấn công, xâm nhập website cho tường lửa ứng dụng web ModSecurity [25]. Các thử nghiệm được thực hiện bao gồm (1) phát hiện chỉ sử dụng OWASP Core Rule Set, (2) phát hiện chỉ sử dụng mô hình phân loại 1-lớp, hoặc mô hình phân loại dựa trên n-gram và (3) phát hiện kết hợp việc sử dụng OWASP Core Rule Set và các mô hình học máy. Các kết quả thử nghiệm trên các tập dữ liệu CSIC2010, ECML/ PKDD2007 và tập dữ liệu tự tạo cho thấy các mô hình học máy và kết hợp có độ chính xác cao hơn nhiều so với OWASP Core Rule Set.

Liang và cộng sự [58] đề xuất mô hình phát hiện tấn công web dựa trên phương pháp học sâu RNN sử dụng URL. Phương pháp đề xuất sử dụng kỹ thuật tách từ (Tokenizer) để chia URL thành hai phần: phần đầu là các *ups* chứa thông tin cấu trúc của đường dẫn URL, phần thứ hai là *qps* chứa thông tin cấu trúc các truy vấn. Trong quá trình thực hiện kỹ thuật tách từ với URL, Liang và cộng sự thực hiện giải mã các URL và chuyển các ký tự hoa thành ký tự thường; tiếp theo, trong phần *ups* thay thế các giá trị trong truy vấn thành ký hiệu <VAL>, trong phần *qps* thay thế các giá trị số trong truy vấn thành ký hiệu <NV>, giá trị chuỗi thành ký hiệu <SV>. Các thử nghiệm trên tập dữ liệu CSIC 2010 [5] cho thấy phương pháp đề xuất đạt độ chính xác cao nhất khoảng 98%. Ưu điểm của mô hình đề xuất là có tỉ lệ phát hiện tấn công cao và tỉ lệ cảnh báo sai thấp. Tuy vậy, nhược điểm của nó là thời gian huấn luyện và phát hiện của mô hình tương đối dài do sử dụng kết hợp hai mô hình RNN và một mô hình MLP.

Theo một hướng khác, Pan và cộng sự [76] đề xuất sử dụng công cụ RSMT (Robust Software Modeling Tool) để giám sát và trích xuất thông tin thực thi của ứng

dụng web, sau đó sử dụng thông tin đã thu thập để huấn luyện bộ mã hóa tự động khử nhiễu xếp chồng (stacked denoising autoencoder) nhằm xây dựng mô hình phát hiện. Các thử nghiệm cho thấy phương pháp được đề xuất có thể phát hiện nhiều loại tấn công web khác nhau và độ đo F1 trung bình trên 91%. Tuy nhiên, mô hình có độ phức tạp cao, thời gian huấn luyện và phát hiện tương đối dài. Ngoài ra, hệ thống giám sát thực thi RSMT được cài đặt trên máy chủ web có thể ảnh hưởng nhiều đến hiệu năng hoạt động của máy chủ, đặc biệt là đối với các hệ thống web có lượng truy cập lớn.

Sharma và cộng sự [91] đề xuất phương pháp phát hiện tấn công web dựa trên các thuật toán học máy truyền thống, bao gồm cây quyết định J48, Naïve Bayes, One rule (OneR) trên tập dữ liệu CSIC HTTP 2010 [5]. Đề xuất của nhóm tác giả gồm ba phần chính: (1) Tiền xử lý dữ liệu bao gồm loại bỏ các dữ liệu lỗi, thừa, khoảng trắng … trong tập dữ liệu; (2) Trích chọn 20 đặc trưng, bao gồm: độ đài truy vấn các trường truy vấn GET, POST, hoặc số lượng từ khóa trong các trường SELECT, DROP, UNION, DELETE, MODIFY,… và (3) Huấn luyện, trong đó tập dữ liệu với 20 đặc trưng sẽ được huấn luyện và phân loại sử dụng các thuật toán học máy, gồm J48, Naïve Bayes, OneR với phương pháp kiểm thử chéo với k=10 trên nền tảng Weka 3.8. Kết quả thử nghiệm cho thấy, thuật toán cây quyết định J48 cho kết quả tỷ lệ phát hiện chung cao nhất là 94,5%.

Tương tự, Saleem và cộng sự [85] cũng sử dụng các thuật toán học máy để phát hiện các dạng tấn công, như SQLi, XSS, và DoS vào máy chủ web. Trong nghiên cứu của mình, Saleem và cộng sự sử dụng tập dữ liệu được xây dựng bằng cách giả lập các cuộc tấn công SQLi, XSS, DoS vào hệ thống máy chủ web được cài đặt với ứng dụng XAMP trên hệ điều hành window 10, lịch sử các cuộc tấn công này sẽ được lưu lại vào file log của hệ thống. Tập dữ liệu gồm có 20.000 bản ghi dữ liệu tấn công và bình thường, trong đó dữ liệu bình thường là chiếm đa số, gồm trên 12.000 bản ghi, dữ liệu tấn công XSS và DoS chiếm khoảng 2000 bản ghi, còn lại là dữ liệu tấn công SQLi. Saleem và cộng sự [85] sử dụng 2955 đặc trưng n-gram và phương pháp TF-IDF để tính giá trị cho các đặc trưng này. Các thử nghiệm với tập dữ liệu do nhóm tác giả xây dựng cho độ chính xác chung và độ đo F1 cao nhất với thuật toán cây quyết định đều là 98%. Hạn chế của mô hình đề xuất là chỉ được thử nghiệm với bộ dữ liệu tự thu thập, chưa được kiểm chứng trong thực tế.

Hao cùng cộng sự [40] đề xuất mô hình BL-IDS cho phát hiện tấn công web dựa trên mạng BRNN (Bidirectional recurrent neural networks) với các đơn vị Bi-LSTM (Bi-directional Long-Short Term Memory). Mô hình đề xuất nhận đầu vào là các URL và bổ sung thêm thông tin HTTP FORM khi request là HTTP POST. Sau

khi các URL được tiền xử lý bằng cách tách thành các thành phần riêng biệt (tokenizer), chúng sẽ được vector hóa sử dụng phương pháp word2vec. Sau đó, lớp BiLSTM sẽ học từ các mẫu request bình thường. Và cuối cùng mạng nơ-ron đã được huấn luyện sẽ dựa trên đầu ra của BiLSTM được sử dụng để xác định các request đầu vào có bị tấn công hay không.

Bảng 1. 3 đánh giá các ưu điểm và những tồn tại của các công trình liên quan gần với đề xuất nghiên cứu của luận án bao gồm: Sharma và cộng sự [91], Saleem và cộng sự [85], Hao cùng cộng sự [40], Betarte và cộng sự [14], Liang và cộng sự [58], và Pan và cộng sự [76], Ming Zhang và cộng sự [111].

Bảng 1. 3. Đánh giá các nghiên cứu liên quan

Đề xuất	Cơ chế	Hạn chế
Sharma và cộng sự [91]	- Trích xuất 20 đặc trưng từ dữ liệu thành file csv gồm các đặc trưng như độ dài các trường, độ dài user-agent, độ dài nội dung, GET và POST request, độ dài cookie. - Sử dụng các thuật toán học máy trên WEKA (J48, Naive Bayes, OneR)	- Số lượng đặc trưng hạn chế - Chỉ phát hiện được 2 loại tấn công, gồm SQLi, XSS.
Saleem và cộng sự [85]	- Sử dụng TF-IDF để trích xuất 2955 đặc trưng đối với từ, Decision Tree, SVM, Naive Bayes với MultinomialNB và AdaBoost để phân loại.	- Sử dụng bộ dữ liệu tự thu thập - Chưa chứng minh tính hiệu quả trên thực tế - Chỉ phát hiện được 2 loại tấn công, gồm SQLi, XSS
Hao cùng cộng sự [40]	- Giải mã các truy vấn HTTP - Sử dụng word2vec để chuyển các từ đã được tiền xử lý dữ liệu về dạng vector. - Sử dụng thuật toán BiLSTM đối với các dữ liệu sau quá trình Word Embedding để huấn luyện và phát hiện	- Yêu cầu nhiều tài nguyên cho huấn luyện và phát hiện - Chưa phát hiện nhiều hình thức tấn công (SQLi, XSS, CMDi, duyệt đường dẫn)
Betarte và cộng sự [14]	- Tokenizer bag of words: Phân chia chỉ bằng sử dụng khoảng trắng, giữ nguyên các ký tự đặc biệt sau khi tokenizer. - Sau khi tokenizer, mô hình sẽ chuyển các từ này thành vector bằng cách sử dụng TF-IDF. - Mô hình sử dụng thuật toán Information gain để giảm chiều dữ liệu sau khi tính TF-IDF.	- Sử dụng Information gain để giảm chiều dữ liệu đã làm giảm hiệu suất phát hiện tấn công của mô hình. - Chưa phát hiện nhiều hình thức tấn công (SQLi, XSS, CMDi, duyệt đường dẫn)

Đề xuất	Cơ chế	Hạn chế
	- Cuối cùng sử dụng các thuật toán học máy để phân loại tấn công bao gồm SVM, KNN, Rừng ngẫu nhiên.	
Liang và cộng sự [58]	- Chỉ sử dụng URL trong các request để phân loại tấn công. - Tokenizer URL - Word Embedding - Mô hình: RNN	- Tiền xử lý và huấn luyện sử dụng kết hợp mô hình phức tạp - Chưa phát hiện nhiều hình thức tấn công (SQLi, XSS, CMDi, duyệt đường dẫn)
Pan và cộng sự [76]	- Sử dụng tác nhân RSMT để thu thập lượng lớn dữ liệu không được đánh nhãn. - Từ dữ liệu đó mô hình sử dụng mô hình bán giám sát với autoencoder và một lượng nhỏ dữ liệu được đánh nhãn bình thường và bất thường để phân loại.	- Thời gian huấn luyện và phát hiện của mô hình dài. - Ảnh hưởng nhiều đến hiệu năng hoạt động của máy chủ web - Chưa phát hiện nhiều hình thức tấn công (SQLi, XSS, CMDi, duyệt đường dẫn)

b. Các đề xuất phát hiện tấn công thay đổi giao diện

Các đề xuất phát hiện tấn công thay đổi giao diện tiêu biểu có thể liệt kê, gồm Kim và cộng sự [54], Bartoli và cộng sự [13], Davanzo và cộng sự [24], Hoang [38] và Hoang và cộng sự [43] [44], phát hiện dựa trên so sánh checksum, so sánh DIFF và phân tích DOM tree trên các trang web [44].

Kim và cộng sự [54] đề xuất phương pháp thống kê để theo dõi và phát hiện các cuộc tấn công thay đổi giao diện trang web. Phương pháp được đề xuất sử dụng kỹ thuật 2-gram để học "hồ sơ" từ tập dữ liệu đào tạo của các trang web hoạt động bình thường. Phương pháp đề xuất được thực hiện qua 2 giai đoạn: Giai đoạn huấn luyện và giai đoạn phát hiện. Để tạo tập "hồ sơ" trong giai đoạn huấn luyện, nội dung HTML của mỗi trang web trong tập dữ liệu huấn luyện được vector hóa bằng cách sử dụng các chuỗi con 2-gram và tần suất xuất hiện tương ứng của chúng. Dựa trên các thực nghiệm, thì 300 2-gram có tần suất xuất hiện cao nhất được chọn để đại diện cho một trang web để phát hiện thay đổi giao diện. Tiếp theo trong giai đoạn phát hiện, đầu tiên trang web cần giám sát sẽ được tải về sau đó nội dung HTML của trang web sẽ được vector hóa bằng kỹ thuật xử lý được thực hiện đối với các trang web trong quá trình huấn luyện. Tiếp đó, vector đặc trưng của trang web được giám sát sẽ đem so sánh với vector của trang web tương ứng trong "hồ sơ" để tính điểm tương tự bằng cách sử dụng khoảng cách cosin. Nếu điểm tương tự được tính toán nhỏ hơn ngưỡng phát hiện được xác định trước thì sẽ có cảnh báo về tấn công. Một thuật toán sinh và cập nhật ngưỡng động được đề xuất để giảm tỷ lệ phát hiện sai. Ưu điểm chính của

đề xuất là có thể tạo và điều chỉnh ngưỡng phát hiện động do đó nó có thể giảm tỷ lệnh cảnh báo sai. Tuy nhiên nhược điểm của phương pháp này là: (i) đối với các trang web có nội dung thay đổi thường xuyên, các ngưỡng được điều chỉnh định kỳ có thể không phù hợp do đó phương pháp này sẽ tạo ra nhiều cảnh báo sai hơn, (ii) phương pháp này cũng đòi hỏi tài nguyên máy tính lớn để điều chỉnh ngưỡng động cho mỗi trang web được giám sát.

Bartoli và cộng sự [13] *và Davanzo và cộng sự* [24] đề xuất sử dụng kỹ thuật lập trình Gen để xây dựng "hồ sơ" nhằm phát hiện các cuộc tấn công làm thay đổi giao diện trang web. Để thu thập dữ liệu của các trang web, trong bước đầu tiên họ sử dụng 43 cảm biến phân thành 5 nhóm để theo dõi và trích xuất thông tin của các trang web được giám sát. Trong bước tiếp theo, thông tin thu thập từ mỗi trang web sẽ được chuyển đổi thành vector gồm 1466 phần tử. Cách tiếp cận của đề xuất được thực hiện trong hai giai đoạn: (i) giai đoạn huấn luyện và (ii) giai đoạn phát hiện. Trong giai đoạn (i) thông tin của trang web hoạt động bình thường được thu thập và vector hóa để xây dựng "hồ sơ" phát hiện bằng kỹ thuật lập trình Gen. Trong giai đoạn (ii) thu thập thông tin của trang web được giám sát, vector hóa và so sánh với "hồ sơ" phát hiện để tìm ra sự khác biệt. Cảnh báo tấn công sẽ được kích hoạt nếu tìm thấy bất kỳ sự khác biệt đáng kể nào. Tuy nhiên phương pháp này đòi hỏi tài nguyên tính toán rất lớn để xây dựng "hồ sơ" phát hiện do các vector đặc trưng của các trang web có kích thước lớn và việc sử dụng kỹ thuật lập trình Gen cũng đòi hỏi chi phí tính toán lớn.

Hoang [38] đề xuất sử dụng các kỹ thuật học máy truyền thống có giám sát để xây dựng mô hình phát hiện tấn công thay đổi giao diện trang web. Trong cách tiếp cận này, các mã HTML của mỗi trang web được vector hóa bằng cách sử dụng kỹ thuật n-gram và tần suất xuất hiện của từ tương ứng. Phương pháp này sử dụng tập dữ liệu thử nghiệm bao gồm 100 trang web "bình thường" và 300 trang web "deface" để sử dụng cho quá trình huấn luyện và thử nghiệm. Kết quả thử nghiệm trên các kịch bản khác nhau sử dụng thuật toán học máy Cây quyết định J48 và Naïve Bayes cho thấy phương pháp đề xuất cho tỷ lệ phát hiện cao và tỷ lệ cảnh báo sai thấp. Tuy nhiên, nhược điểm chính của Hoang [38] là: (i) tập dữ liệu thử nghiệm tương đối nhỏ, làm giảm độ tin cậy của kết quả và (ii) phương pháp chỉ xử lý với mã HTML của trang web trong khi các thành phần quan trọng khác của trang web như là: mã JavaScript, mã CSS, hình ảnh không được xử lý.

Để giải quyết các vấn đề tồn tại trong đề xuất của Hoang [38], Hoang và cộng sự [44] đã đề xuất mô hình mở rộng cho phát hiện tấn công thay đổi giao diện trang

web bằng cách sử dụng kết hợp phát hiện dựa trên chữ ký và phát hiện dựa trên học máy. Trong bước đầu tiên, thành phần phát hiện dựa trên chữ ký sẽ tìm kiếm các chữ ký tấn công được xác định trước có trong mã HTML của trang web được giám sát, và điều này giúp cải thiện hiệu xuất xử lý. Trong bước tiếp theo, thành phần phát hiện dựa trên học máy được sử dụng để phân loại trang web dựa trên bộ phân loại được tích hợp trong quá trình huấn luyện. Cuối cùng, sử dụng hàm băm để kiểm tra tính toàn vẹn trong các tệp được nhúng trong trang web. Các thử nghiệm được sử dụng tập dữ liệu bao gồm 1200 trang web "bình thường" và 1200 trang web "deface" cho thấy mô hình đề xuất đạt hiệu suất phát hiện cao. Mặc dù mô hình kết hợp kiểm tra tính toàn vẹn của các tệp nhúng trong trang web, kỹ thuật dựa trên hàm băm chỉ có thể hoạt động hiệu quả với các tệp nhúng tĩnh.

Trong phần mở rộng hơn nữa từ các công trình trước đó [38] [44], Hoang và cộng sự [43] đề xuất mô hình đa lớp để phát hiện tấn công thay đổi giao diện trang web. Trong mô hình đa lớp [43], mô hình tích hợp dựa trên học máy được sử dụng để phát hiện các cuộc tấn công thay đổi giao diện đối với thành phần văn bản có trong trang web, bao gồm mã HTML, mã JavaScript và mã CSS. Đối với hình ảnh nhúng trong trang web, hàm băm được sử dụng để kiểm tra tính toàn vẹn. Các thử nghiệm cho thấy mô hình đa lớp có thể phát hiện các cuộc tấn công thay đổi giao diện một cách hiệu quả trên các thành phần văn bản của trang web. Tuy nhiên, khả năng phát hiện tấn công thay đổi giao diện của mô hình đề xuất trên các hình ảnh của trang web bị hạn chế vì chỉ kiểm tra được tính toàn vẹn dựa trên hàm băm và trên thực tế, nhiều trang web hình ảnh nhúng là yếu tố quan trọng trong nội dung trang.

Các phương pháp phát hiện dựa trên so sánh checksum, diff và cây DOM [23] [24] [44] là các phương pháp phát hiện được sử dụng từ lâu, tuy nhiên thường chỉ phù hợp cho các trang web tĩnh hoặc ít thay đổi về cấu trúc. Theo đó, phát hiện dựa trên so sánh checksum là phương pháp đơn giản nhất để phát hiện các thay đổi trên trang web. Trước hết, nội dung (hoặc các thành phần của nội dung) các trang web được tính checksum (sử dụng các giải thuật băm như MD5 hoặc SHA1) và lưu trữ vào hồ sơ. Sau đó trang web được giám sát, tính checksum và so sánh với giá trị đã lưu trữ trong hồ sơ. Chỉ một thay đổi nhỏ trong nội dung trang web cũng dẫn đến sự thay đổi giá trị checksum và dẫn đến một cảnh báo tấn công. Phương pháp phát hiện dựa trên so sánh checksum hoạt động tốt với các trang web tĩnh, hoặc ít thay đổi nội dung. Với các trang web động, như các diễn đàn, hoặc các trang thương mại điện tử có nội dung được cập nhật thường xuyên, phương pháp này không còn phù hợp.

Trong phương pháp phát hiện dựa trên so sánh diff, diff là một công cụ so sánh tìm sự khác biệt giữa nội dung 2 trang web, được hỗ trợ phổ biến trên các nền tảng hệ điều hành Linux và Unix. Việc cần thực hiện là xác định một ngưỡng phát hiện bất thường làm đầu vào cho mỗi trang web được giám sát. Phương pháp này khá hiệu quả và làm việc tốt trên hầu hết các trang web động nếu ngưỡng phát hiện bất thường được xác định phù hợp. Trong khi các c*ác phương pháp phát hiện dựa trên so sánh checksum, diff* tập trung vào sự thay đổi nội dung, *phương pháp phát hiện dựa trên so sánh cây DOM* (Document Object Model) tập trung vào phát hiện sự thay đổi cấu trúc trang web. DOM là một giao diện lập trình ứng dụng cho phép định nghĩa cấu trúc logic của các tài liệu HTML – hay các trang web. DOM có thể được sử dụng để duyệt và phân tích các thành phần của một trang web. Tương tự như các kỹ thuật dựa trên bất thường khác, cây DOM của trang khi hoạt động bình thường được tạo và lưu. Sau đó, cây DOM của trang được giám sát sẽ được so sánh với cây DOM đã lưu để tìm sự khác biệt. Nhìn chung, phương pháp này có khả năng hoạt động tốt với các trang web có cấu trúc ổn định. Việc xác định ngưỡng thay đổi cấu trúc cũng đơn giản và ổn định hơn so với ngưỡng thay đổi nội dung.

Bảng 1. 4 phân tích và đánh giá ưu nhược điểm, tồn tại của các giải pháp, công trình nghiên cứu gần với mô hình đề xuất bao gồm: nghiên cứu từ Kim và cộng sự [54], Bartoli và cộng sự [13] và Davanzo và cộng sự [24], Hoang [38], Hoang và cộng sự [44], Hoang và cộng sự [43], Siyan Wu và cộng sự [106], các giải pháp đơn giản hơn như phát hiện dựa trên so sánh diff và cây DOM [23] [24] [44].

Bảng 1. 4. Đánh giá ưu nhược điểm các nghiên cứu liên quan

Đề xuất	**Cơ chế**	**Hạn chế**
Nhóm các giải pháp DIFF, DOM, Checksum	- Checksum: + Sử dụng các giải thuật MD5, SHA1 tính checksum nội dung trang web, lưu vào hồ sơ + Trang web cần giám sát, tính checksum và so sánh với hồ sơ tìm sự khác biệt -DIFF: so sánh sự khác biệt giữa nội dung 2 trang web, xác định ngưỡng bất thường - DOM: Phát hiện dựa trên thay đổi cấu trúc, thay đổi nội dung trang web	- Checksum: Không phù hợp với trang web động, được cập nhật thường xuyên. - DIFF: Phải xác định được ngưỡng phù hợp. - DOM: Chỉ phù hợp với trang web có cấu trúc ổn định.
Kim và cộng sự [54]	- Sử dụng kỹ thuật 2-gram - Lựa chọn 300 2-gram xuất hiện nhiều nhất	- Không phù hợp với trang web động, nội dung thay đổi thường xuyên

Đề xuất	Cơ chế	Hạn chế
	- So sánh tính khoảng cách tương tự bằng khoảng cách Cosi.	- Tài nguyên tính toán lớn cho điều chỉnh ngưỡng động với mỗi trang web được giám sát.
Bartoli và cộng sự [13] và Davanzo và cộng sự [24]	- Sử dụng 43 cảm biến, phân thành 5 nhóm theo dõi và trích xuất thông tin của trang web giám sát - Thông tin thu thập chuyển thành vector 1466 đặc trưng. - Sử dụng lập trình gen để xây dựng mô hình	- Số lượng đặc trưng sử dụng rất lớn - Phương pháp xây dựng mô hình đòi hỏi chi phí tính toán lớn
Hoang [38]	- Sử dụng học máy truyền thống - Xử lý mã HTML, với n-gram và tính tần suất xuất hiện của ký tự - Tập dữ liệu: 100 web bình thường và 300 web tấn công.	- Tập dữ liệu nhỏ, giảm độ tin cậy của kết quả - Chỉ xử lý với mã HTML chưa xử lý mã JavaScript, mã CSS, hình ảnh không được xử lý.
Hoang và cộng sự [44]	- Sử dụng học máy truyền thống kết hợp sử dụng chữ ký - Tìm mẫu chữ ký trong mã HTML - Phân loại trang web với thuật toán học máy - Sử dụng hàm băm MD5 xác nhận tính toàn vẹn của tệp nhúng - Tập dữ liệu bao gồm 1200 trang web bình thường và 1200 trang web tấn công	- Tập dữ liệu tương đối nhỏ - Kỹ thuật sử dụng hàm băm chỉ phù hợp với các tệp nhúng tĩnh - Gây ra nhiều cảnh báo sai hơn bình thường do hàm băm quá nhạy với các thay đổi - Tập chữ ký nhỏ với 50 dấu hiệu tấn công
Hoang và cộng sự [43]	- Sử dụng mô hình đa lớp - Sử dụng mã HTML, Javascript, CSS, sử dụng hàm băm kiểm tra tính toán vẹn của hình ảnh trên trang web - Tập dữ liệu với 2700 trang web thông thường và 2100 trang web bị tấn công	- Phát hiện với ảnh nhúng chỉ sử dụng hàm băm còn hạn chế - Tập dữ liệu chưa thực sự đủ lớn
Siyan Wu và cộng sự [106]	- Sử dụng học máy có giám sát để huấn luyện và phát hiện - Trích chọn 28 đặc trưng trong HTML với 2 nhóm chính là đặc trưng từ của mã trang web và mã trojan được nhúng trong trang web	- Hiệu suất giảm với tập dữ liệu lớn do thời gian huấn luyện tăng lên - Chưa xem xét yếu tố cấu trúc trang, chưa xử lý ảnh nhúng trong trang web.

1.4. Hướng nghiên cứu của luận án

1.4.1. Ưu điểm và nhược điểm của các giải pháp phát hiện tấn công web

Công cụ phát hiện tấn công web điển hình, gồm VNCS Web Monitoring [61], Nagios Web Application Monitoring Software [103], Site24x7 Website Defacement Monitoring [78], ModSecurity [25] có ưu điểm nổi bật, như: đây là giải pháp thương mại nên được hỗ trợ thường xuyên từ các nhà sản xuất nên dễ dàng triển khai. Tuy nhiên, các công cụ này cũng tồn tại những nhược điểm, như: chi phí cho các hệ thống này tương đối cao, một số hệ thống gặp vấn đề tương thích, như ModSecurity [25], hoặc gặp vấn đề về xử lý và vận chuyển khối lượng log lớn như VNCS Web monitoring [61], hoặc chỉ phù hợp với những trang web tĩnh ít có sự thay đổi, như Site24x7 Website Defacement Monitoring [78].

Các đề xuất phát hiện dựa trên chữ ký và tập luật đã được khảo sát tại mục *1.3.3.1. Phát hiện dựa trên chữ ký và tập luật*, có những ưu điểm và nhược điểm sau:

- Ưu điểm: các đề xuất có khả năng phát hiện nhanh các dạng tấn công web đã biết; *OWASP ModSecurity Core Rule Set* [25] được cập nhật tập luật từ OWASP và cộng đồng bảo mật web, *SQL-IDS* [51] có thể bảo vệ được nhiều trang web mà không cần chỉnh sửa mã nguồn trang web, *XSS-GUARD* [75] phát hiện hiệu quả tấn công XSS, *SQLGuard* [17] và *SQLCheck* [96] phát hiện tốt tấn công SQLi.

- Nhược điểm: các đề xuất sử dụng tập luật cho quá trình phát hiện nên đòi hỏi phải cập nhật thường xuyên, *OWASP ModSecurity Core Rule Set* [25] có tập luật cồng kềnh gặp vấn đề khi tích hợp với các máy chủ ứng dụng web khác; *SQL-IDS* [51] chỉ phát hiện được tấn công SQLi, tập dữ liệu cần được đánh giá toàn diện với dữ liệu thực tế; *XSS-GUARD* [75] chỉ phát hiện tấn công XSS, bộ lọc cồng kềnh tăng tải cho hệ thống thực tế; một số nghiên cứu đòi hỏi truy cập mã nguồn Java của ứng dụng web như *SQLGuard* [17] và *SQLCheck* [96].

Các đề xuất phát hiện dựa trên bất thường đã được khảo sát tại mục *1.3.3.2. Phát hiện dựa trên bất thường*, có những ưu điểm và nhược điểm sau:

- Ưu điểm: Có khả năng phát hiện được những dạng tấn công mới chưa có trong cơ sở dữ liệu. AMNESIA [39], Swaddler [22], CANDID [16], Torrano-Gimenez và cộng sự [100] phát hiện tốt các tấn công SQLi; một số đề xuất sử dụng các kỹ thuật học máy và học sâu tiên tiến, gồm Betarte và cộng sự [14], Liang và cộng sự [58], và Pan và cộng sự [76], S. Sharma và cộng sự [91], S. Saleem và cộng sự [85], Saiyu Hao cùng cộng sự [40], Kim và cộng sự [54], Bartoli và cộng sự [13]

và Davanzo và cộng sự [24], Hoang [38], Hoang và cộng sự [43] là hướng nghiên cứu cho kết quả tốt và hướng nghiên cứu được quan tâm trong những năm gần đây.

- Nhược điểm: Tỷ lệ cảnh báo giả còn tương đối cao, cũng như những giải pháp sử dụng các kỹ thuật học sâu [38], [40], [43], [44], [58] đòi hỏi nhiều tài nguyên hệ thống, có quá trình xử lý dữ liệu tương đối phức tạp; các đề xuất chỉ phát hiện được một loại tấn công cụ thể, như [16], [22], [39], [100] chỉ phát hiện tốt tấn công SQLi; [85], [91]phát hiện tốt tấn công SQLi và XSS; các đề xuất [13], [24], [38], [54] phát hiện tốt các tấn công thay đổi giao diện. Một số đề xuất có quá trình xử lý phức tạp như Saiyu Hao cùng cộng sự [40], Hoang và cộng sự [43], và các đề xuất sử dụng tập dữ liệu nhỏ như [38], [43].

Tựu trung, có thể tóm tắt các tồn tại của các đề xuất phát hiện dựa trên bất thường: (1) các đề xuất phát hiện tấn công web (SQLi, CMDi, XSS, duyệt đường dẫn) thường chỉ phát hiện được một hoặc hai loại tấn công phổ biến, như SQLi và/hoặc XSS; chưa có nghiên cứu phát hiện đồng thời nhiều dạng tấn công web; một số đề xuất có quá trình xử lý dữ liệu tương đối phức tạp, hoặc hiệu suất phát hiện chưa cao *(cụ thể là độ chính xác tổng thể chưa cao (khoảng 90-95% hoặc thấp hơn) và tỷ lệ phát hiện sai còn tương đối cao (khoảng 7-10% hoặc cao hơn));* (2) các đề xuất phát hiện tấn công thay đổi giao diện sử dụng tập dữ liệu nhỏ; chưa có đề xuất thực hiện kết hợp các đặc trưng văn bản và hình ảnh chụp màn hình trang web; hiệu suất phát hiện còn tương đối thấp. Do đó, trong phần tiếp theo NCS sẽ đề xuất các vấn đề nghiên cứu trong Luận án nhằm giải quyết các tồn tại nói trên.

1.4.2. Các vấn đề giải quyết trong luận án

Qua phân tích các đề xuất cho phát hiện các dạng tấn công web thường gặp và tấn công thay đổi giao diện có thể kết luận việc tiếp tục nghiên cứu các giải pháp hiệu quả cho phát hiện tấn công web thường gặp và tấn công thay đổi giao diện là rất cần thiết. Luận án cũng đã nghiên cứu, khảo sát các kỹ thuật phát hiện tấn công web thường gặp và tấn công thay đổi giao diện dựa trên dấu hiệu, tập luật, chữ ký; dựa trên bất thường và một số công cụ giải pháp cho giám sát phát hiện tấn công web thường gặp và tấn công thay đổi giao diện web. Mỗi phương pháp, giải pháp và các công cụ lại có những ưu nhược điểm riêng như đã trình bày trong mục *1.4.1. Ưu điểm và nhược điểm của các giải pháp phát hiện tấn công web.*

Hướng nghiên cứu của luận án là phát hiện tấn công web thường gặp và tấn công thay đổi giao diện web dựa trên bất thường nói chung và cụ thể là sử dụng các mô hình học máy, học sâu do phương pháp này có khả năng phát hiện các dạng tấn công web mới, đồng thời có khả năng tự động hóa việc xây dựng mô hình phát hiện.

Trên cơ sở khảo sát, phân tích các ưu điểm và hạn chế của các đề xuất đã có, luận án tập trung nghiên cứu, giải quyết các vấn đề sau: (1) Xây dựng mô hình phát hiện tấn công web thường gặp dựa trên học máy sử dụng web log và (2) Xây dựng mô hình phát hiện tấn công thay đổi giao diện trang web dựa trên học sâu sử dụng kết hợp dữ liệu văn bản nội dung trang web và ảnh màn hình trang web. Lý do thực hiện (1) là do một số kỹ thuật phát hiện dựa trên bất thường phát hiện được một hoặc hai loại tấn công phổ biến, như SQLi và/hoặc XSS trên một tập dữ liệu cụ thể, mà không phát hiện được đồng thời nhiều loại tấn công web, như: XSS, SQLi, duyệt đường dẫn, CMDi. Ngoài ra, một số đề xuất phát hiện dựa trên bất thường có tỷ lệ phát hiện đúng còn thấp và tỷ lệ cảnh báo sai còn cao. Tương tự, việc thực hiện (2) là bởi 2 lý do: (i) hình thức tấn công thay đổi giao diện là một loại tấn công web, tuy vậy đây là loại tấn công web đặc biệt có cơ chế thực hiện cũng như hậu quả của dạng tấn công này rất khác biệt so với các dạng tấn công web thường gặp khác, như SQLi và XSS. Do vậy, các kỹ thuật giám sát, phát hiện dạng tấn công này có đặc thù riêng, nên NCS tách ra thành một mục và được trình bày riêng; (ii) nhằm nâng cao tỷ lệ phát hiện đúng và giảm tỷ lệ cảnh báo sai cho mô hình phát hiện tấn công thay đổi giao diện sử dụng dữ liệu đầu vào kết hợp giữa văn bản trích xuất từ nội dung trang web và ảnh màn hình trang web. *Việc lựa chọn kết hợp đặc trưng văn bản và ảnh màn hình trang web được NCS luận giải cụ thể tại mục 3.1.3. Phát hiện tấn công thay đổi giao diện.*

Từ các hướng nghiên cứu trên trong luận án NCS đề xuất hai bài toán cần giải quyết:

Bài toán 1: Phát hiện tấn công web thường gặp dựa trên học máy sử dụng web log; Dữ liệu đầu vào là các dòng web logs và kết quả phát hiện là dòng log bình thường hay có chứa dữ liệu tấn công SQLi, XSS, duyệt đường dẫn, hoặc CMDi.

Bài toán 2: Phát hiện tấn công thay đổi giao diện trang web; Dữ liệu đầu vào là các trang web cần giám sát, trong đó sử dụng kết hợp hai đặc trưng là văn bản thuần trích xuất từ trang và ảnh màn hình trang web. Kết quả phát hiện là trang web bình thường hay bị tấn công thay đổi giao diện.

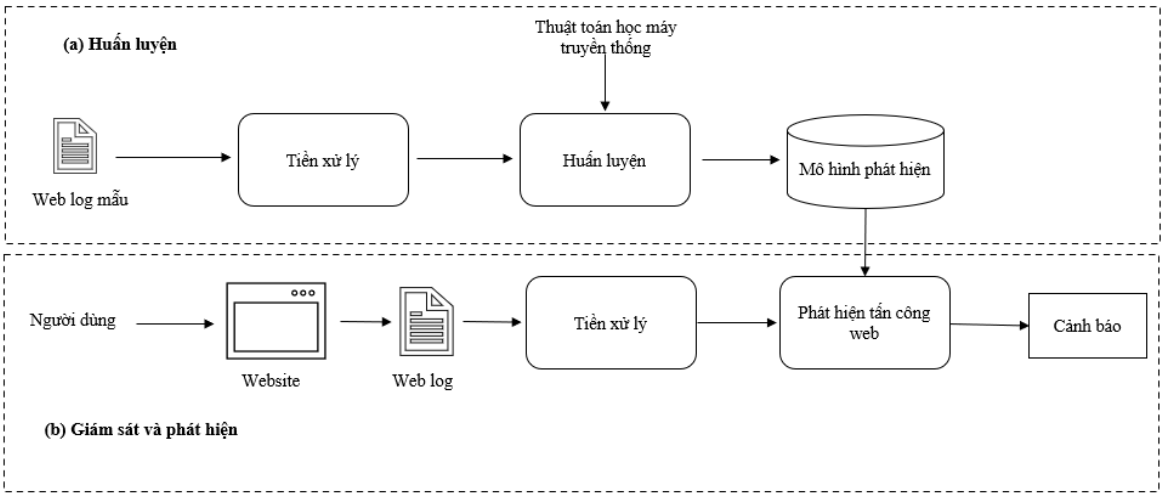
1.4.3. Kiến trúc mô hình tổng thể cho các hướng nghiên cứu của luận án

Trong mục này luận án sẽ đưa ra mô hình tổng thể cho các hướng nghiên cứu được đề xuất tại mục *1.4.2. Các vấn đề giải quyết trong luận án.*

1.4.3.1. Kiến trúc phát hiện tấn công web thường gặp dựa trên học máy sử dụng web log

Hình 1. 8 kiến trúc tổng thể phát hiện tấn công web thường gặp. Đầu vào của mô đun này là các dòng web logs. Nhiệm vụ chính của mô đun phát hiện tấn công

web thường gặp là phân tích các dòng web logs nhằm phát hiện các dấu hiệu của tấn công SQLi, XSS, duyệt đường dẫn và CMDi. Mô đun này sẽ được phát triển thành 2 giai đoạn: giai đoạn huấn luyện và giai đoạn phát hiện được trình bày chi tiết hơn trong chương 2 của luận án.



Hình 1. 8. Kiến trúc tổng thể cho phát hiện tấn công web dựa trên học máy sử dụng dữ liệu weblog

Chức năng chính của mô đun này:

- Xây dựng mô hình phát hiện từ dữ liệu huấn luyện sẵn có.

Chức năng này sẽ thực hiện các khâu tiền xử lý tập dữ liệu huấn luyện có sẵn, huấn luyện mô hình phát hiện và kiểm thử độ chính xác phát hiện sử dụng tập dữ liệu kiểm thử.

Đầu vào: Tập dữ liệu huấn luyện và kiểm thử.

Xử lý: Tiền xử lý, huấn luyện, lưu mô hình phát hiện và kiểm thử.

Đầu ra: File lưu mô hình phát hiện và độ chính xác kiểm thử phát hiện.

- Tự động đọc dữ liệu web log từ CSDL.

Chức năng này sẽ định kỳ đọc dữ liệu web log từ CSDL theo khoảng thời gian cho trước. Dữ liệu đọc được chuyển cho khâu phân tích, phát hiện.

Đầu vào: Thời gian bắt đầu và kết thúc (tem thời gian của web log) của đợt xử lý.

Xử lý: Thực hiện truy vấn CSDL để đọc ra tập dòng log cho xử lý.

Đầu ra: Tập dòng log (hoặc tập rỗng).

- Kiểm tra, phân tích từng dòng web log để phát hiện tấn công.

Chức năng này nhận tập dòng log, xử lý từng dòng nhằm phát hiện các dấu hiệu tấn công web.

Đầu vào: Tập dòng web log, mô hình phát hiện.

Xử lý: Tách URI truy nhập, tiền xử lý từng dòng web log và phát hiện trạng thái URI truy nhập.

Đầu ra: Trạng thái của dòng web log (bình thường hay bị tấn công, gồm loại tấn công).

- Lưu trữ toàn bộ kết quả phân tích, xử lý web log theo từng đợt vào CSDL.

Chức năng này lưu toàn bộ kết quả phân tích, xử lý web log theo từng đợt vào CSDL.

Đầu vào: Thông tin đợt xử lý web log và kết quả xử lý tổng hợp.

Xử lý: Kết nối đến CSDL, lưu thông tin đợt xử lý web log và kết quả xử lý tổng hợp vào CSDL.

Đầu ra: Dữ liệu lưu thành công trong CSDL.

- Lưu thông tin và sinh cảnh báo khi phát hiện tấn công.

Chức năng này lưu thông tin chi tiết về tấn công cho mỗi dòng log và sinh cảnh báo khi phát hiện tấn công cho cả đợt.

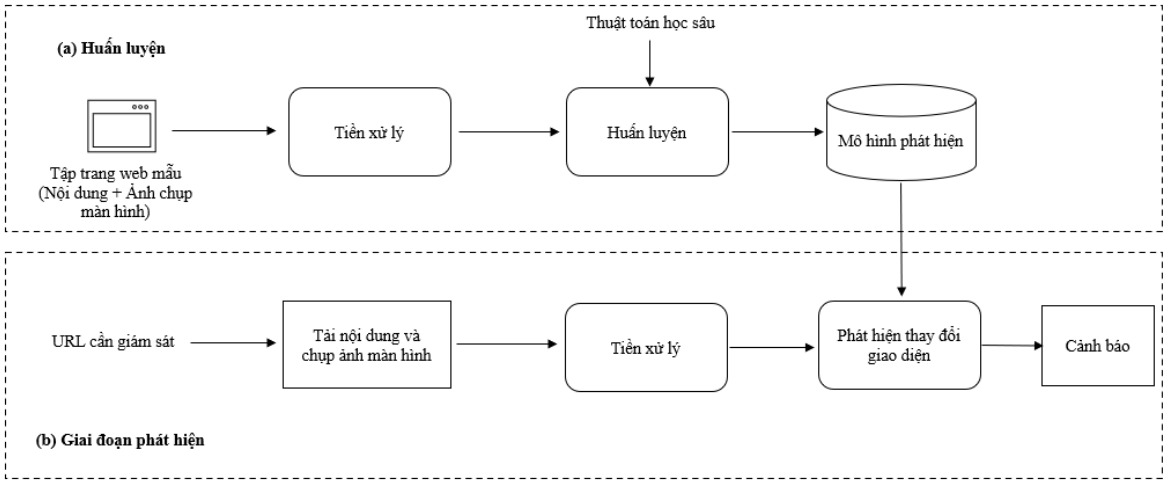
Đầu vào: Thông tin chi tiết về tấn công cho từng dòng log

Xử lý: Kết nối đến CSDL, lưu thông tin chi tiết về tấn công cho từng dòng log, sinh các cảnh báo (pop-up, SMS, email) và lưu vào bảng thông tin cảnh báo theo định dạng cho trước.

Đầu ra: Dữ liệu lưu thành công trong CSDL.

1.4.3.2. Kiến trúc phát hiện tấn công thay đổi giao diện trang web

Hình 1. 9 kiến trúc tổng thể phát hiện tấn công thay đổi giao diện trang web. Đầu vào của mô đun này là các trang web cần giám sát. Nhiệm vụ chính của mô đun phát hiện tấn công thay đổi giao diện là định kỳ giám sát các trang web nhằm phát hiện các dấu hiệu của tấn công thay đổi giao diện. Mô đun này được phát hiển thành 2 giai đoạn: giai đoạn huấn luyện và giai đoạn phát hiện được trình bày chi tiết hơn trong chương 2 của luận án.



Hình 1. 9. Kiến trúc tổng thể cho phát hiện tấn công thay đổi giao diện trang web

Chức năng chính của mô đun này là:

- Xây dựng mô hình phát hiện từ dữ liệu huấn luyện sẵn có.

Chức năng này thực hiện các khâu tiền xử lý tập dữ liệu huấn luyện có sẵn, huấn luyện mô hình phát hiện bằng các thuật toán học sâu.

Đầu vào: Tập dữ liệu huấn luyện như mô tả trong mục.

Xử lý: Tiền xử lý, huấn luyện, lưu mô hình phát hiện và kiểm thử.

Đầu ra: File lưu mô hình phát hiện và độ chính xác kiểm thử phát hiện.

- Quản lý các trang web được giám sát.

Chức năng này cho phép quản lý các trang web được giám sát, bao gồm các tính năng xem danh sách trang, thêm mới, sửa, xoá mỗi trang.

Đầu vào: Thông tin các trang web được giám sát.

Xử lý: Hỗ trợ các tính năng xem danh sách trang, thêm mới, sửa, xoá mỗi trang.

Đầu ra: Thông tin các trang web được giám sát lưu trong CSDL.

- Giám sát tự động đồng thời nhiều trang web

Chức năng này thực hiện nạp danh sách các trang web cần giám sát từ CSDL, tải mã HTML và phân tích từng trang web nhằm phát hiện tấn công thay đổi giao diện

Đầu vào: Danh sách các trang web cần giám sát lưu trong CSDL.

Xử lý: Tải mã HTML từng trang, lưu mã HTML và màn hình, tiền xử lý, phát hiện.

Đầu ra: Trạng thái từng trang web được giám sát (bình thường, lỗi tải trang, hay bị thay đổi giao diện).

- Lưu trữ toàn bộ kết quả giám sát vào CSDL

Chức năng này cho phép lưu trữ toàn bộ kết quả giám sát từng trang vào CSDL. Người dùng có thể xem kết quả giám sát trên giao diện của phân hệ quản trị.

Đầu vào: Thông tin kết quả giám sát các trang

Xử lý: Kết nối đến CSDL, lưu toàn bộ kết quả giám sát từng trang vào CSDL.

Đầu ra: Các thông tin được lưu thành công vào CSDL.

- Lưu thông tin và sinh cảnh báo khi phát hiện tấn công

Chức năng này lưu thông tin chi tiết và sinh cảnh báo khi phát hiện tấn công cho mỗi trang web được giám sát.

Đầu vào: Thông tin chi tiết về tấn công cho mỗi trang web được giám sát

Xử lý: Kết nối đến CSDL, lưu thông tin chi tiết về tấn công cho từng mỗi trang web được giám sát, sinh các cảnh báo (pop-up, SMS, email) và lưu vào bảng thông tin cảnh báo theo định dạng cho trước.

Đầu ra: Dữ liệu lưu thành công trong CSDL.

1.5. Một số thuật toán học máy và học sâu sử dụng trong luận án

Mục này trình bày vắn tắt một số thuật toán học máy có giám sát truyền thống và một số thuật toán học sâu được sử dụng trong các mô hình phát hiện tấn công web đề xuất tại chương 2 và chương 3 của luận án bao gồm: Naïve Bayes, Cây quyết định, Rừng ngẫu nhiên, SVM và CNN, LSTM, BiLSTM và EfficientNet.

1.5.1. Naïve Bayes

Naïve Bayes là một thuật toán dựa trên định lý Bayes về lý thuyết xác suất để đưa ra các phán đoán cũng như phân loại dữ liệu dựa trên các dữ liệu được quan sát và thống kê. Naïve Bayes là một trong những thuật toán được ứng dụng nhiều trong các lĩnh vực học máy dùng để đưa các dự đoán chính xác nhất dựa trên một tập dữ liệu đã được thu thập, vì nó tương đối đơn giản và cho độ chính xác cao. Naïve Bayes thuộc vào nhóm các thuật toán học có giám sát, tức là học từ các dữ liệu đã được gán nhãn [70] [101].

1.5.2. Cây quyết định

Cây quyết định là một thuật toán học máy có giám sát có thể được áp dụng vào cả hai bài toán phân loại và hồi quy. Việc xây dựng một cây quyết định trên dữ liệu huấn luyện cho trước là việc xác định các câu hỏi và thứ tự của chúng. Một điểm

đáng lưu ý của cây quyết định là nó có thể làm việc với các đặc trưng (các đặc trưng thường được gọi là thuộc tính – attribute), thường là rời rạc và không có thứ tự. Ví dụ, mưa, nắng hay xanh, đỏ, v..v. Cây quyết định cũng làm việc với dữ liệu có vector đặc trưng bao gồm cả thuộc tính dạng rời rạc và liên tục. Một điểm đáng lưu ý nữa là cây quyết định ít yêu cầu việc chuẩn hoá dữ liệu.

1.5.3. Rừng ngẫu nhiên

Rừng ngẫu nhiên (Random Forest) là một thuật toán học máy phổ biến thuộc nhóm học máy có giám sát. Tương tự cây quyết định, rừng ngẫu nhiên có thể được sử dụng giải quyết hai vấn đề phân loại và hồi quy trong học máy [64]. Rừng ngẫu nhiên dựa trên khái niệm học tập theo nhóm, là một quá trình kết hợp nhiều bộ phân loại để giải quyết một vấn đề phức tạp và để cải thiện hiệu suất của mô hình. Thay vì dựa vào một cây quyết định, rừng ngẫu nhiên lấy dự đoán từ mỗi cây và dựa trên đa số phiếu, và dự đoán kết quả cuối cùng. Số lượng cây lớn hơn trong rừng dẫn đến độ chính xác cao hơn và ngăn ngừa vấn đề quá vừa.

1.5.4. SVM

Máy véc tơ hỗ trợ (SVM) là một trong những thuật toán học máy có giám sát phổ biến nhất, được sử dụng cho các bài toán phân loại cũng như hồi quy. Tuy nhiên, SVM chủ yếu được sử dụng để giải quyết các bài toán phân loại. Mục tiêu của thuật toán SVM là tạo đường hoặc mặt ranh giới quyết định tốt nhất có thể tách không gian n chiều thành các lớp để có thể dễ dàng đặt điểm dữ liệu mới vào đúng danh mục trong tương lai. Ranh giới quyết định tốt nhất này được gọi là siêu phẳng.

1.5.5. CNN

CNN (Convolutional Neural Network) là một thuật toán phổ biến và được sử dụng rộng rãi trong học sâu. Nó đã được áp dụng rộng rãi trong các ứng dụng khác nhau như NLP, xử lý giọng nói, và thị giác máy tính,… Mạng CNN có kiến trúc được cấu tạo bởi một số lớp bao gồm: Convolutional layer, Pooling layer và Fully connected layer. Convolutional layer hay Lớp tích chập (CONV) sử dụng các bộ lọc để thực hiện phép tích chập khi đưa chúng đi qua đầu vào I theo các chiều của nó. Các siêu tham số của các bộ lọc này bao gồm kích thước bộ lọc F và độ trượt (stride) S. Kết quả đầu ra O được gọi là bản đồ đặc trưng (feature map) hay bản đồ kích hoạt (activation map). Lớp pooling (POOL) là một phép lấy mẫu xuống (downsampling), thường được sử dụng sau tầng tích chập, giúp tăng tính bất biến không gian. Có thể liệt kê một số loại cấu trúc CNN như: LetNet, AlexNet, ZFNet, VGGNet và GoogleNet.

1.5.6. LSTM

LSTM (Long Short-Term Memory) là một biến thể nâng cao hơn của mạng RNN (Recurrent Neural Networks) với việc thiết kế để giải quyết các bài toán về phụ thuộc dài hạn. LSTM được giới thiệu lần đầu vào năm 1997 và được cải tiến vào năm 2013, đạt được sự phổ biến đáng kể trong cộng đồng học sâu. So với các RNN tiêu chuẩn, các mô hình LSTM đã được chứng minh là hiệu quả hơn trong việc lưu giữ và sử dụng thông tin qua các chuỗi dài hơn [93]. LSTM khắc phục được rất nhiều những hạn chế của RNN trước đây về triệt tiêu đạo hàm. Tuy nhiên cấu trúc của chúng có phần phức tạp hơn mặc dù vẫn dữ được tư tưởng chính của RNN là sự sao chép các kiến trúc theo dạng chuỗi [30]. LSTM bao gồm ba cổng: cổng đầu vào, cổng quên và cổng đầu ra. Mỗi cổng thực hiện một chức năng cụ thể trong điều khiển luồng thông tin [110].

1.5.7. BiLSTM

BiLSTM (Bidirectional LSTM) là một phần mở rộng của kiến trúc LSTM giải quyết giới hạn của các mô hình LSTM tiêu chuẩn bằng cách xem xét cả quá khứ và bối cảnh tương lai trong các nhiệm vụ lập mô hình trình tự. Trong khi các mô hình LSTM truyền thống chỉ xử lý dữ liệu đầu vào trong hướng về phía trước, BiLSTM khắc phục hạn chế này bằng cách đào tạo mô hình theo hai hướng: tiến và lùi [92]. BiLSTM bao gồm hai lớp LSTM song song: (1) xử lý chuỗi đầu vào theo hướng thuận, (2) xử lý nó theo hướng ngược lại. Phía trước lớp LSTM đọc dữ liệu đầu vào từ trái sang phải. Đồng thời, lớp LSTM ngược lại đọc dữ liệu đầu vào từ phải sang bên trái [59]. Quá trình xử lý hai chiều này cho phép mô hình nắm bắt thông tin từ cả bối cảnh quá khứ và tương lai, cho phép hiểu rõ hơn sự hiểu biết toàn diện về sự phụ thuộc thời gian trong trình tự.

1.5.8. EfficientNet

Mạng tích chập (ConvNet – hay CNN) đã trở thành một trong những mô hình được sử dụng nhiều nhất trong lĩnh thực thị giác máy tính. Mạng ConvNet thường được phát triển với ngân sách tài nguyên cố định và sau đó được mở rộng quy mô để có độ chính xác cao hơn nếu có sẵn nhiều tài nguyên hơn [98]. Do đó nhằm tăng độ chính xác mô hình thì chúng ta thường có 3 hướng sau : (1) Tăng độ sâu của mô hình, (2) Tăng độ rộng của từng layer trong mô hình, (3) Cải thiện chất lượng của đầu vào (tăng chất lượng, kích thước ảnh). Mặc dù 3 hướng trên có thể giúp mở rộng mô hình, nhưng có thể sẽ khó khăn khi cần tối ưu đối với một mạng ConvNet lớn.

Mạng EfficientNet là mô hình mạng được mở rộng từ mô hình ConvNet nhằm đạt kết quả tốt hơn (accuracy) với số lượng tham số (params) ít hơn, số FLOPS tăng lên (số lượng floating points cần tính toán per seconds) [98]. Mingxing Tan, Quoc V. Le [98] cũng đưa ra thêm các mô hình EfficientNet B0 đến B7 (B0 là mô hình cơ sở còn B1->B7 là mô hình được điều chỉnh của B0) đã hoàn toàn vượt trội so với các mô hình ConvNet về hiệu suất.

1.6. Các độ đo đánh giá

Để đánh giá khả năng phát hiện của các mô hình đề xuất trong các Chương 2 và Chương 3, luận án sử dụng sáu độ đo bao gồm: PPV (Positive Pedictive Value), TPR (True Positive Rate), FPR (False Positive Rate), FNR (False Negative Rate), F1 và ACC (Accuracy). Các độ đo này được tính toán sử dụng các tham số TP, TN, FP và FN trong ma trận nhầm lẫn cho trên Bảng 1. 5. Theo đó, TP (True Positives) là số lượng các mẫu tấn công được dự đoán đúng, TN (True Negatives) là số lượng các mẫu bình thường được dự đoán đúng, FP (False Positives) là số lượng mẫu bình thường được dự đoán sai thành tấn công và FN (False Negatives) là số lượng mẫu tấn công được dự đoán sai thành bình thường.

Bảng 1. 5. Bảng ma trận nhầm lẫn

		Lớp thực tế	
		Tấn công	*Bình thường*
Lớp dự đoán	*Tấn công*	TP	FP
	Bình thường	FN	TN

Các độ đo này được định nghĩa như sau:

- Giá trị dự đoán dương tính (PPV-Positive Predictive Value, hay Precision), còn gọi là độ chính xác, được tính theo công thức:

$$PPV = \frac{TP}{TP + FP} \quad (1.\ 1)$$

- Tỷ lệ dương tính thật (TPR-True Positive Rate, hay Recall) còn gọi là độ nhạy, hay độ bao phủ, được tính theo công thức:

$$TPR = \frac{TP}{TP + FN} \quad (1.\ 2)$$

- Tỷ lệ dương tính giả (FPR-False Positive Rate), hay còn gọi là tỷ lệ nhầm lẫn, được tính theo công thức:

$$FPR = \frac{FP}{FP + TN} \quad (1.3)$$

- Tỷ lệ âm tính giả (FPR- False Negative Rate), hay còn gọi tỷ lệ bỏ sót, được tính theo công thức:

$$FNR = \frac{FN}{FN + TP} \quad (1.4)$$

- Độ đo F1 là trung bình điều hòa giữa Precision và Recall, được tính theo công thức:

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (1.5)$$

- Độ chính xác toàn cục (ACC) hay độ chính xác chung, được tính theo công thức:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (1.6)$$

1.7. Kết luận chương

Trong phần đầu, chương 1 đã trình bày khái quát về web và dịch vụ web, tổng quan về tấn công web; đồng thời khái quát về phát hiện tấn công web, các giải pháp, công cụ và kỹ thuật chính; các vấn đề về học máy và học sâu sử dụng trong luận án.

Nội dung chính của chương 1 của luận án phân tích các nghiên cứu trong các kỹ thuật phát hiện tấn công web, đưa ra nhận xét và từ đó đề xuất hai bài toán cần giải quyết trong luận án được trình bày trong các chương 2 và chương 3.

Nội dung của chương này đã được công bố trong bài báo:

Hoang Xuan Dau, Ninh Thi Thu Trang, **Nguyen Trong Hung**. "A Survey of Tools and Techniques for Web Attack Detection". Journal of Science and Technology on Information security, Special Issue CS (15) 2022, pp. 109-118.

CHƯƠNG 2. PHÁT HIỆN TẤN CÔNG WEB DỰA TRÊN HỌC MÁY SỬ DỤNG WEB LOG

Chương 2 giới thiệu khái quát về web log, một số đề xuất phát hiện tấn công web sử dụng học máy, đánh giá ưu nhược điểm của các đề xuất. Phần cuối của chương mô tả việc xây dựng, cài đặt, thử nghiệm và đánh giá mô hình phát hiện tấn công web thường gặp dựa trên học máy sử dụng web log.

2.1. Khái quát về web log

2.1.1. Giới thiệu về web log

Web log là các bản ghi lưu lại các hoạt động của máy chủ web. Các bản ghi web log thường được lưu trữ trong các file log dưới dạng văn bản thuần. Các máy chủ web thông dụng hiện nay, như Mozilla Apache HTTP Server, Microsoft IIS, Nginx đều cung cấp cơ chế để ghi lại log hoạt động vào file.

Web log có nhiều định dạng khác nhau, chứa những thông tin liên quan tới thời gian, nội dung trao đổi giữa máy khách và máy chủ web. Dữ liệu log có thể được lưu vào một file, hoặc chia thành các file log riêng biệt, như log truy cập, log lỗi, log tham chiếu,… tuỳ thuộc vào loại máy chủ web. Web log có thể được xử lý bởi các công cụ phân tích log giúp trích xuất nhiều thông tin hữu ích về hành vi người dùng web. Ngoài ra, khi hệ thống gặp sự cố, web log cũng là một nguồn cung cấp các dữ liệu quan trọng cho quản trị viên tìm hiểu nguyên nhân và khắc phục sự cố [11].

File log của các loại máy chủ web khác nhau chứa các loại dữ liệu không hoàn toàn giống nhau. Tuy nhiên, dữ liệu log chứa trong file log của hầu hết các loại máy chủ web bao gồm các mục thông tin [112]:

- Tên người dùng (Username): Xác định tên người dùng (hay định danh người dùng) đã truy cập website. Trong trường hợp không xác định được tên người dùng, địa chỉ IP máy khách được sử dụng để xác định định danh của người dùng.

- Đường dẫn truy cập (Path): Xác định tài nguyên mà người dùng truy cập trên website.

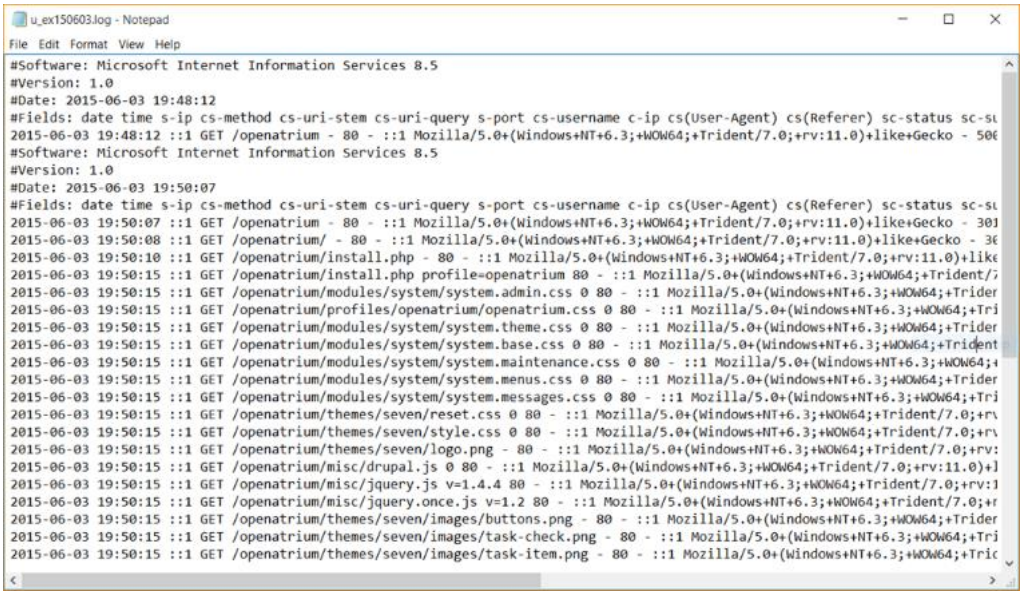
- Thời gian truy cập (Date and Time): Cho biết thời điểm và khoảng thời gian người dùng thăm một trang web. Thông số này thường dùng trong việc xác định các phiên làm việc của người dùng.

- Trang xem cuối cùng (Last page visited): Cho biết trang mà người dùng xem trước khi rời website.

- Máy khách người dùng (User Agent): Một chuỗi mô tả loại, phiên bản của máy khách web (thường là trình duyệt web) người dùng sử dụng để truy cập website.

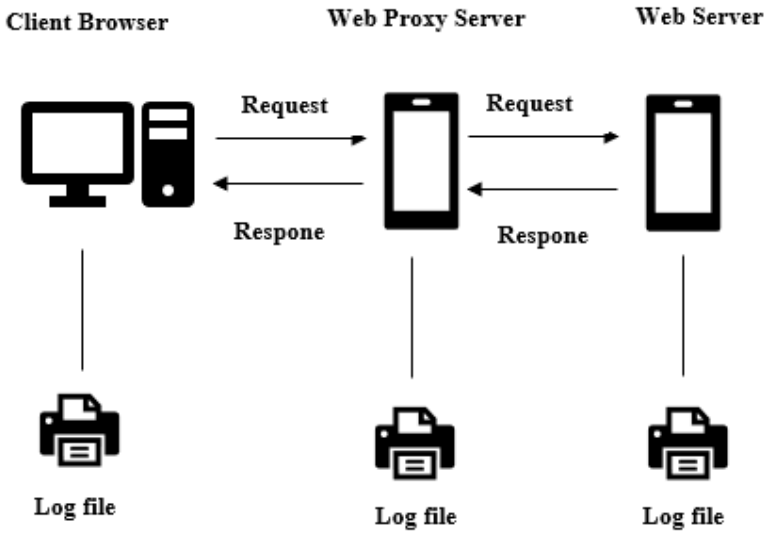
- URL (Uniform Resource Locator): Địa chỉ tài nguyên được truy cập bởi người dùng, có thể là một trang HTML, một chương trình CGI, hoặc một đoạn mã,….

- Loại yêu cầu (HTTP method): Phương thức yêu cầu gửi đến từ máy khách, có thể là GET, POST, HEAD,…



Hình 2. 1. Các bản ghi web log trên máy chủ web Microsoft IIS

Các nguồn sinh web log chính gồm: Client Browser (Trình duyệt máy khách), Web Proxy Server (Máy chủ web proxy) và Web Server (Máy chủ web) [27].



Hình 2. 2. Các nguồn sinh web log

2.1.2. Một số dạng web log

Các định dạng web log được sử dụng phổ biến nhất hiện nay bao gồm định dạng web log chuẩn của NCSA (NCSA Common Log Format), định dạng web log kết hợp (NCSA Combined Log Format), định dạng web log mở rộng của W3C (W3C Extended Log Format) và định dạng web log của máy chủ web Microsoft IIS (Microsoft IIS Log Format) [27] [112]. Tuy nhiên, trong thực tế mỗi máy chủ web đều hỗ trợ một số định dạng web log trong số các dạng kể trên do đó người quản trị có thể lựa chọn định dạng web log sử dụng để máy chủ sinh các file web log phù hợp.

2.1.2.1. NCSA Common Log Format

NCSA Common Log Format hay Common Log Format, là định dạng web log với các trường cố định mà không thể tùy chỉnh. Môt số thông tin cơ bản có trong dạng web log này gồm: yêu cầu người dùng, hostname (tên) của máy khách, tên người dùng, ngày, giờ, loại yêu cầu, mã trạng thái HTTP trả về, số lượng byte gửi bởi server. Các trường trong mỗi bản ghi log được phân cách bởi dấu trắng. Những trường không chứa dữ liệu sẽ được biểu diễn bằng dấu (-), các ký tự không in được sẽ được biểu diễn bởi dấu (+).

Với máy chủ Apache HTTP Server, định dạng Common Log Format có thể được cấu hình nhờ chuỗi định dạng như sau [57]:

LogFormat "%h %l %u %t \"%r\" %>s %b" common

Ví dụ, với Common Log Format thì một đầu mục (entry) sẽ có dạng như sau:

127.0.1.1 - frank [10/Oct/2021:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326

Trong đó, các trường thông tin của đầu mục này gồm:

- 127.0.1.1 (tương ứng kí hiệu %h): Địa chỉ IP của máy khách gửi yêu cầu đến máy chủ.

- Trống (-) (tương ứng kí hiệu %l): Định danh của máy khách.

- frank (tương ứng kí hiệu %u): Định danh/tên của người dùng gửi yêu cầu được xác định nhờ thủ tục xác thực HTTP.

- [10/Oct/2021:13:55:36 -0700] (tương ứng kí hiệu %t): Thời gian máy chủ kết thúc xử lý yêu cầu, theo định dạng sau: [day/month/year:hour:minute:second zone], hay ngày/tháng/năm:giờ:phút: giây và múi giờ. Trong đó, day = 2*digit, month

= 3*letter; year = 4*digit; hour = 2*digit; minute = 2*digit; second = 2*digit và zone = (‘+’ | ‘-‘) 4*digit.

- "GET /apache_pb.gif HTTP/1.0" (tương ứng kí hiệu \"%r\"): Yêu cầu của máy khách gửi lên máy chủ.

- 200 (tương ứng kí hiệu %>s): Mã trạng thái mà máy chủ gửi trả về cho máy khách.

- 2326 (tương ứng kí hiệu %b): Kích thước của gói tin trả về cho máy khách, không bao gồm header.

2.1.2.2. NCSA Combined Log Format

NCSA Combined Log Format gọi tắt là Combined Log Format về cơ bản tương tự như Common Log Format, ngoại trừ việc nó bổ sung thêm hai trường thông tin ở cuối là Referrer (Liên kết tham chiếu) và User agent (Máy khách người dùng). Với Apache HTTP Server, định dạng này có thể được cấu hình nhờ chuỗi định dạng như sau:

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" combined [90]

Ví dụ, với Combined Log Format, một đầu mục sẽ như sau:

127.0.1.1 - frank [10/Oct/2021:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326 "http://www.example.com/start.html" "Mozilla/4.08 [en] (Win98; I ;Nav)"

Các trường được bổ sung bao gồm:

- "http://www.example.com/start.html" (tương ứng kí hiệu \"%{Referer}i\"): Cho biết trang web người dùng đã thăm trước khi đến trang hiện tại.

- Mozilla/4.08 [en] (Win98; I ;Nav)" (tương ứng kí hiệu \"%{User-agent}i\"): Cho biết thông tin về trình duyệt và hệ điều hành máy khách đang sử dụng.

2.1.2.3. W3C Extended Log Format

Hiện nay, W3C Extended Log Format đề xuất bởi The World Wide Web Consortium (W3C) là định dạng được sử dụng rộng rãi nhất và được hầu hết các máy chủ web hỗ trợ [97]. Định dạng web log này có các khả năng:

- Hỗ trợ kiểm soát những thông tin sẽ được ghi trong web log.

- Hỗ trợ một định dạng web log chung cho cả máy chủ proxy, máy khách và máy chủ web.

- Cung cấp một cơ chế mạnh mẽ xử lý các vấn đề về các ký tự thoát (character escaping).

- Cho phép trao đổi dữ liệu nhân khẩu học (demographic).

- Hỗ trợ tổng hợp dữ liệu.

Một file log theo định dạng W3C Extended Log chứa một tập hợp các dòng văn bản thuần gồm các ký tự theo chuẩn ASCII (hoặc UniCode) được phân tách nhau bởi ký tự xuống dòng (LF hoặc CRLF). Các file log khác nhau sẽ có ký tự kết thúc dòng khác nhau tùy thuộc vào quy ước kết thúc dòng của nền tảng hoạt động. Trên mỗi dòng thường có một chỉ thị (directive) hoặc một đầu mục (entry).

2.1.2.4. Microsoft IIS Log Format

Microsoft IIS là máy chủ web chạy trên hệ điều hành Microsoft Windows Server. Máy chủ Microsoft IIS hỗ trợ nhiều định dạng web log khác nhau như: NCSA Common Log Format, W3C Extended Log Format và Microsoft IIS Log Format.

Microsoft IIS Log Format chứa các thông tin cơ bản như: Địa chỉ IP của máy khách, tên người dùng, ngày giờ thực hiện yêu cầu, mã trạng thái dịch vụ, số lượng byte đã nhận. Ngoài ra, nó còn chứa các thông tin chi tiết như hành động thực hiện, file đích, thời gian thực hiện. Các trường trong mỗi bản ghi log được phân cách bởi dấu phẩy, những trường không chứa thông tin thay bằng dấu '-', các ký tự không in được thay bằng dấu '+'. Ví dụ, với Microsoft IIS Log Format thì một đầu mục của web log sẽ như sau:

192.168.114.201, -, 03/20/01, 7:55:20, W3SVC2, SALES1, 172.21.13.45, 4502, 163, 3223, 200, 0, GET, /DeptLogo.gif, -,

Trong đó:

- 192.168.114.201 là địa chỉ IP máy khách

- 03/20/01, 7:55:20 là ngày và giờ thực hiện yêu cầu

- W3SVC2 chỉ tiến trình chạy dịch vụ web

- SALES1 là tên máy chủ web

- 172.21.13.45 là địa chỉ IP máy chủ web

- 4502 là thời gian xử lý tính bằng mili giây

- 163 là số byte của yêu cầu

- 3223 là số byte của phản hồi (kết quả) máy chủ gửi máy khách

- 200 là mã trạng thái thực hiện yêu cầu (thành công)

- GET là phương thức yêu cầu

- /DeptLogo.gif là file được yêu cầu.

2.1.2.5. Các chuỗi định dạng log của Apache HTTP Server

Apache HTTP Server cung cấp khả năng ghi log rất toàn diện và linh hoạt, bao gồm các loại log như: Security Warning (cảnh báo an ninh), Error Log (log các lỗi), Access Log (log truy cập), Log Rotation (quay vòng log), Piped Logs (các log đường ống). Log truy cập của Apache HTTP Server ghi lại tất cả các yêu cầu được xử lý bởi máy chủ. Định dạng của log truy cập có thể được cấu hình bởi các chuỗi định dạng, như liệt kê trong Bảng 2. 1.

Bảng 2. 1. Các chuỗi định dạng của Apache HTTP Server

Chuỗi định dạng	**Mô tả**
%%	Dấu phần trăm
%a	Địa chỉ IP máy khách
%A	Địa chỉ IP của máy cục bộ
%B	Kích thước của gói tin trả lời tính bằng byte, không tính HTTP header
%b	Kích thước của gói tin trả lời tính bằng byte, không tính HTTP header. Trong định dạng Common Log, nếu không có byte nào được gửi thì được biểu diễn bởi dấu '-' chứ không phải số 0.
%{VARNAME}C	Nội dung của cookie VARNAME trong yêu cầu gửi đến server.
%D	Thời gian xử lý yêu cầu gửi đến, tính bằng micro giây.
%{VARNAME}e	Giá trị của biến môi trường VARNAME
%f	Tên file
%h	Hostname của máy khách, nếu thuộc tính HostnameLookups được đặt là off thì sẽ thay bằng địa chỉ IP
%H	Giao thức của gói tin yêu cầu gửi đến
%{VARNAME}i	Giá trị trường VARNAME trong header của yêu cầu gửi đến máy chủ.
%k	Số lượng yêu cầu giữ kết nối (keepalive) được xử lý trong kết nối.
%l	Logname, nếu không bật thuộc tính IdentityCheck thì sẽ thay bằng dấu '-'
%L	Nếu yêu cầu đến hoặc gói trả lời phát sinh lỗi thì sẽ là Log ID từ Error log tương ứng, nếu không sẽ là dấu '-'
%m	Phương thức của gói tin yêu cầu
%{VARNAME}n	Giá trị của chú thích VARNAME ở các module khác

Chuỗi định dạng	Mô tả
%{VARNAME}o	Nội dung của trường VARNAME trong header của gói tin trả lời
%p	Canonical port (cổng chính tắc) của máy chủ nhận yêu cầu đến
%{format}p	Các định dạng khả dụng: canonical (chính tắc), local (cục bộ) và remote (từ xa).
%P	ID của tiến trình xử lý yêu cầu
%{format}P	ID của tiến trình hoặc luồng xử lý yêu cầu. Các định dạng khả dụng: pid, tid và hextid.
%q	Chuỗi truy vấn
%r	Dòng đầu tiên của truy vấn
%R	Bộ phận xử lý tạo gói tin trả lời (nếu có)
%s	Trạng thái
%t	Thời gian nhận yêu cầu đến
%{format}t	Thời gian ở định dạng format, gồm các dạng sau: sec, msec, usec, msec_frac, usec_frac.
%T	Thời gian cần để xử lý yêu cầu, tính bằng giây.
%{UNIT}T	Thời gian cần để xử lý yêu cầu tính bằng UNIT, các UNIT có thể là: milli giây (ms), micro giây (us) và giây (s).
%u	Người dùng ở máy khách, nếu như yêu cầu đã được xác thực
%U	URL được yêu cầu, không bao gồm chuỗi truy vấn.
%v	Tên của máy chủ phục vụ yêu cầu.
%V	Tên của máy chủ
%X	Trạng thái của kết nối sau khi hoàn thành gói trả lời: x: Kết nối bị huỷ bỏ trước khi gói trả lời hoàn thành. +: Kết nối có thể được giữ sau khi gói trả lời gửi đi. -: Kết nối được đóng khi gói trả lời gửi đi.
%I	Số byte nhận, bao gồm cả header
%O	Số byte gửi, bao gồm cả header
%S	Số lượng byte được chuyển (cả nhận và gửi).
%{VARNAME}^ti	Nội dung của dòng cuối trong yêu cầu gửi đến máy chủ
%{VARNAME}^to	Nội dung của dòng cuối trong gói trả lời gửi từ máy chủ

2.2. Phát hiện tấn công web dựa trên học máy

Kết quả nghiên cứu và khảo sát tại mục *1.3.3.2. Phát hiện dựa trên bất thường* nhận thấy, các giải pháp đề xuất phát hiện tấn công web dựa trên dữ liệu web log là một hướng hiệu quả. Đặc biệt, hướng nghiên cứu sử dụng học máy là nhánh có triển vọng do mô hình phát hiện đơn giản, có thể được xây dựng tự động từ tập dữ liệu huấn luyện. Đây cũng chính là nhánh nghiên cứu của luận án lựa chọn thực hiện.

Mặc dù các nghiên cứu đã có cho phát hiện tấn công web đạt kết quả khả quan, tuy nhiên vẫn còn một số vấn đề cần tiếp tục nghiên cứu như: (1) một số đề xuất tuy sử dụng cơ chế đơn giản, nhưng chỉ cho độ chính xác phát hiện cao với tập dữ liệu cụ

thể hoặc với một loại tấn công web cụ thể, và số lượng đặc trưng sử dụng quá nhiều, điển hình như Sharma và cộng sự [91], Saleem và cộng sự [85]; (2) một số đề xuất sử dụng mô hình học sâu hoặc sử dụng bộ công cụ giám sát máy chủ nên đòi hỏi chi phí tính toán lớn cho quá trình xây dựng mô hình, cũng như quá trình giám sát phát hiện và điều này làm giảm khả năng triển khai ứng dụng trên các hệ thống thực [58] [76]; và (3) một số đề xuất sử dụng cơ chế phức tạp, sử dụng mô hình học sâu, đòi hỏi nhiều tài nguyên tính toán, nhưng không phát hiện được nhiều hình thức tấn công web (SQLi, XSS, CMDi, duyệt đường dẫn), như [40] [58].

Do đó, luận án này đề xuất mô hình có khả năng phát hiện được nhiều loại tấn công web (SQLi, XSS, CMDi, duyệt đường dẫn), cài đặt tương đối đơn giản, có độ chính xác cao, yêu cầu tài nguyên tính toán tương đối thấp và có khả năng ứng dụng thực tế. Chi tiết về mô hình phát hiện tấn công đề xuất được mô tả trong mục tiếp theo của luận án.

2.3. Xây dựng và thử nghiệm mô hình phát hiện tấn công web dựa trên học máy sử dụng web log

2.3.1. Giới thiệu mô hình

Giao thức HTTP là nền tảng truyền thông dữ liệu cho web. Theo đó, máy khách gửi yêu cầu HTTP đến máy chủ và máy chủ xử lý yêu cầu và trả về phản hồi HTTP cho máy khách. Nếu máy chủ bị tấn công, điều đó có nghĩa là nó nhận được một hoặc nhiều yêu cầu độc hại. Những yêu cầu này thường được lưu lại trên tệp tin log của máy chủ web. Do yêu cầu HTTP có nhiều thành phần, nên NCS chỉ tập trung vào nội dung truy vấn URI trong yêu cầu HTTP. Hình 2. 3 hiển thị một số mẫu truy vấn URI tấn công được gửi đến máy chủ web và được lưu lại trong weblog.

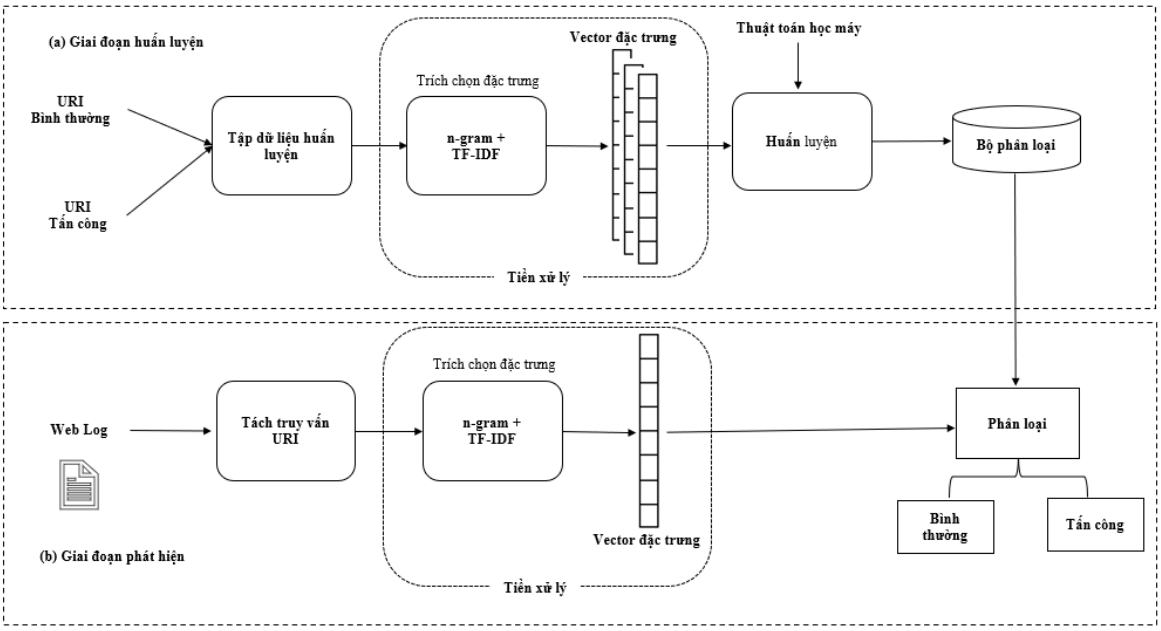
```
84.55.41.57 - - [17/Apr/2016:07:16:06 +0100] "GET
/wordpress/wp-admin/update.php?action=install-plugin&plugin=file-manager&_wpnonce=3c6c8a7fca HTTP/1.1" 200 5698
"http://www.example.com/wordpress/wp-admin/plugin-install.php?tab=search&s=file+permission"
84.55.41.57 - - [17/Apr/2016:07:18:19 +0100] "GET
/wordpress/wp-admin/plugins.php?action=activate&plugin=file-manager%2Ffile-manager.php&_wpnonce=bf932ee530
HTTP/1.1" 302 451
"http://www.example.com/wordpress/wp-admin/update.php?action=install-plugin&plugin=file-manager&_wpnonce=3c6c8a7f
84.55.41.57 - - [17/Apr/2016:07:21:46 +0100] "GET
/wordpress/wp-admin/admin-ajax.php?action=connector&cmd=upload&target=l1_d3AtY29udGVudA&name%5B%5D=r57.php&FILES=
1460873968131 HTTP/1.1" 200 731 "http://www.example.com/wordpress/wp-admin/admin.php?page=file-manager_settings"


84.55.41.57 - - [17/Apr/2016:07:22:53 +0100] "GET /wordpress/wp-content/r57.php HTTP/1.1" 200 9036 "-"
84.55.41.57 - - [17/Apr/2016:07:32:24 +0100] "POST /wordpress/wp-content/r57.php?14 HTTP/1.1" 200 8030
"http://www.example.com/wordpress/wp-content/r57.php?14"
84.55.41.57 - - [17/Apr/2016:07:29:21 +0100] "GET /wordpress/wp-content/r57.php?29 HTTP/1.1" 200 8391
"http://www.example.com/wordpress/wp-content/r57.php?28"
84.55.41.57 - - [17/Apr/2016:07:57:31 +0100] "POST /wordpress/wp-admin/admin-ajax.php HTTP/1.1" 200 949
"http://www.myw ebsite.com/wordpre ss/wp-admin/admin.php?page=file-manager_settings"
```

Hình 2. 3. Truy vấn URI trong web log

Phân tích kỹ hơn cấu trúc một truy vấn URI từ mẫu dữ liệu web log tại Hình 2. 3, có thể thấy URI chứa các thông tin yêu cầu cài đặt và kích hoạt một plugin từ kẻ tấn công được gửi tới máy chủ web:

"/wordpress/wp-admin/update.php?action=install-plugin&plugin=file-manager&_wpnonce=3c6c8a7fca", thì chuỗi truy vấn (**?query_string**) trong URI là: *"?action=install-plugin&plugin=file-manager&_wpnonce=3c6c8a7fca"*

Dựa trên điều này, phương pháp phát hiện được thiết kế bằng cách kiểm tra các truy vấn URI trong dữ liệu web log (cụ thể là các **?query_string** trong URI) để phát hiện tấn công web. Hình 2. 4 biểu diễn mô hình phát hiện tấn công web dựa trên học máy sử dụng dữ liệu web log. Mô hình đề xuất sử dụng tập đặc trưng ký tự được trích xuất từ truy vấn URI trong dữ liệu web log. Mô hình phát hiện tấn công web đề xuất được triển khai trong 2 giai đoạn: (a) giai đoạn huấn luyện và (b) giai đoạn phát hiện. Trong giai đoạn huấn luyện dữ liệu URI tấn công và bình thường được thu thập, tiếp theo sẽ tiến hành tiền xử lý dữ liệu thu thập được nhằm trích xuất các đặc trưng cho quá trình huấn luyện. Trong bước huấn luyện, các thuật toán học máy có giám sát, như Naïve bayes, SVM, Cây quyết định, Rừng ngẫu nhiên được áp dụng để học ra bộ phân loại hay mô hình phát hiện, thuật toán cho kết quả tốt nhất sẽ được sử dụng cho mô hình phát hiện. Trong giai đoạn Phát hiện, các truy vấn URI sẽ được trích lọc từ dữ liệu weblog, qua quá trình tiền xử lý như giai đoạn Huấn luyện và đến bước phân loại sử dụng Bộ phân loại từ giai đoạn Huấn luyện để xác định truy vấn Bình thường hay Tấn công.



Hình 2. 4. Mô hình phát hiện tấn công web dựa trên dữ liệu web log

2.3.2. Tiền xử lý dữ liệu

Các nghiên cứu [14] [40] [58] [85] sử dụng các kỹ thuật Word2vec, bag of words, tập trung vào xử lý đặc trưng với cấp độ từ, trong toàn truy vấn URI *(*cấu trúc URI được trình bày tại mục *1.1.3. Kiến trúc ứng dụng web và các thành phần)*. Luận án tập trung khai thác đặc trưng thống kê ký tự của cụm n-gram xuất hiện trong một thành phần của truy vấn URI là **?query_string**. Trong Luận án, NCS sẽ thực nghiệm với 2-gram, 3-gram, 4-gram và 5-gram để chọn ra n-gram cho kết quả tốt nhất.

2-gram, hay bi-gram là cụm 2 ký tự liên tiếp nhau trong chuỗi ký tự. Ví dụ, với truy vấn URI với phương thức GET "1%union" gồm các bi-gram {1%, %u, un, ni, io, on}. Một truy vấn trong URI có thể chứa các ký tự trong tập 26 ký tự chữ cái (a-z), 10 ký tự số (0-9), và 30 ký tự đặc biệt / + ? & ; = , ‘ " ( ) < > * ! $ # | ^ { } \ -- % ~ @ . ` [ ] : do đó số bi-gram có thể là TS(bi-gram) = 66^2 = 4.356.

3-gram, hay tri-gram là cụm 3 ký tự liên tiếp nhau trong chuỗi ký tự. Ví dụ, với truy vấn URI với phương thức GET "1%union" gồm các tri-gram {1%u, %un, uni, nio, ion}. Tương tự cách tính tổng số bi-gram, tổng số tri-gram có thể có là TS(tri-gram)=66^3=287.496.

4-gram, 5-gram là cụm 4 và 5 ký tự liên tiếp nhau trong chuỗi ký tự. Ví dụ với truy vấn vấn URI với phương thức GET "1%union" gồm các 4-gram {1%un, %uni, unio, nion} và 5-gram {1%uni, %unio, union}. Tương tự cách tính tổng số bi-gram, tổng số 4-gram có thể có là TS(4-gram)=66^4=18.974.736 và tổng số 5-gram có thể có là TS(5-gram)=66^5=1.252.332.576.

Tiếp đó, mỗi đặc trưng n-gram đơn nhất trích xuất từ URI được tính giá trị bằng phương pháp TF-IDF (Term Frequency-Inverse Document Frequency) [105] và kết quả sẽ thu được là một vector đặc trưng đại diện cho URI. Tuy nhiên, có thể nhận thấy với n-gram lớn (chẳng hạn 4-gram, 5-gram) thì tập đặc trưng n-gram cũng rất lớn dẫn tới vector đặc trưng của URI cũng rất lớn dẫn đến thời gian cho quá trình huấn luyện sẽ kéo dài. Để khắc phục vấn đề này, NCS sử dụng phương pháp giảm chiều dữ liệu để giảm kích thước vector đặc trưng, qua đó làm giảm thời gian huấn luyện đồng thời không gây ảnh hưởng đáng kể tới hiệu suất của mô hình. Luận án sẽ thực nghiệm ba phương pháp giảm chiều dữ liệu, bao gồm: phương pháp hệ số tương quan, phương pháp Information Gain và phương pháp PCA (Principal Component Analysis). Trên cơ sở thực nghiệm, luận án đánh giá và lựa chọn phương pháp giảm chiều cho hiệu suất tốt nhất.

Phương pháp giảm chiều bằng hệ số tương quan: Hệ số tương quan là một chỉ số thống kê đo lường sức mạnh của mối quan hệ giữa hai biến. Có nhiều loại hệ số tương quan khác nhau. Trong luận án sử dụng hệ số tương quan Pearson. Hệ số tương quan Pearson giữa hai biến X và Y được tính bằng công thức sau [28]:

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} \quad (2.\ 1)$$

$Cov(X,Y)$: Là hiệp phương sai của X và Y

σ_X: Độ lệch chuẩn của X

σ_y: Độ lệch chuẩn của Y

Phương pháp giảm chiều dữ liệu bằng Information Gain: Information Gain (IG) là một phương pháp đánh giá đặc trưng dựa trên hàm nhiễu và được sử dụng rộng rãi trong học máy. IG được định nghĩa như một đại lượng đo lường lượng thông tin thu được về một lớp từ một đặc trưng. Giá trị thông tin được tính toán dựa trên đại lượng entropy. Hàm entropy được định nghĩa như sau [84]: Cho một phân phối xác suất của một biến rời rạc x có thể nhận n giá trị khác nhau $\{x_1, x_2, .., x_n\}$. Giả sử xác suất để nhận được giá trị x là $p_x = p(x = x_i)$ với $0 \le p_i \le 1$ và $\sum p_i = 1$. Hàm phân phối sẽ là $P = (p_1, p_2, \dots, p_n)$. Giá trị entropy là:

$$H(P) = -\sum_{i=1}^{n} p_i \log p_i \quad (2.\ 2)$$

Từ công thức về entropy, nguyên tắc tính toán của IG như sau [28]:

Bước 1: Xét một bài toán với C lớp khác nhau. Giả sử dữ liệu với một nút không có lá với dữ liệu các điểm tạo thành tập hợp S có số phần tử là $S\ v\ N$. Giả sử trong N dữ liệu này thì có N_c điểm dữ liệu (với $c = 1,2, \dots, C$) thuộc lớp c. Xác suất cho mỗi dữ liệu thuộc lớp c xấp xỉ $\frac{N_c}{N}$. Như vậy, entropy ở nút này được tính như sau:

$$H(S) = -\sum_{C=1}^{C} \frac{N_c}{N} \log_2 \frac{N_c}{N} \quad (2.\ 3)$$

Bước 2: Giả sử tập dữ liệu được chia thành các tập con theo một đặc điểm x. Dựa trên x, điểm dữ liệu trong S được chia thành các nút con: $S_1, S_2, \dots, S_k$ với $m_1, m_2, \dots, m_k$ điểm trong mỗi nút con. Luận án xác định công thức sau là tổng entropy có trọng số của mỗi nút con.

$$H(x,S) = -\sum_{k=1}^{K} \frac{m_k}{N} H(S_k) \quad (2.4)$$

Bước 3: Tính toán giá trị thu được thông tin dựa trên đặc điểm x.

$$G(x,S) = H(S) - H(x,S) \quad (2.5)$$

Phương pháp giảm chiều dữ liệu bằng PCA: Để đơn giản hóa tính toán, PCA sẽ tìm một cơ sở trực chuẩn để tạo ra một cơ sở mới, trong đó các thành phần quan trọng nhất nằm ở một số tọa độ của thành phần đầu tiên. Các bước thực hiện PCA như sau [28]:

Bước 1: Tính vectơ trung bình của tất cả dữ liệu.

$$\bar{X} = \frac{1}{N}\sum_{n=1}^{N} X_n \quad (2.6)$$

Bước 2: Trừ vectơ trung bình khỏi mỗi điểm dữ liệu.

$$\bar{X} = X_n - \overline{X_n} \quad (2.7)$$

Bước 3: Tính ma trận hiệp phương sai:

$$S = \frac{1}{N}\hat{X}\hat{X}^T \quad (2.8)$$

Bước 4: Tính các giá trị riêng và vector riêng với norm bằng 1 của ma trận này, sắp xếp chúng theo thứ tự giảm dần của giá trị riêng.

Bước 5: Chọn K vector riêng có K giá trị riêng cao nhất để xây dựng ma trận UK với các cột tạo thành một không gian trực giao. K vector này còn được gọi là các thành phần quan trọng tạo thành một không gian gần với phân phối của dữ liệu gốc đã chuẩn hóa.

Bước 6: Chiếu dữ liệu gốc đã chuẩn hóa xuống không gian đã tìm thấy.

Bước 7: Tính toán tọa độ của dữ liệu mới. Dữ liệu mới là tọa độ của các điểm dữ liệu trên không gian mới theo công thức.

$$Z = U_K^T \hat{X} \quad (2.9)$$

Dữ liệu gốc có thể được xấp xỉ theo dữ liệu mới như trong công thức.

$$X \approx U_K Z + \hat{X} \quad (2.\ 10)$$

Như vậy, quá trình tiền xử lý dữ liệu web log dựa trên kỹ thuật n-gram, TF-IDF và giảm chiều được thực hiện theo các bước như sau:

Bước 1: Tách các truy vấn **?query_string** trong các truy vấn URI

Bước 2: Từ các truy vấn này thực hiện tách các đặc trưng n-gram

Bước 3: Tính giá trị cho các đặc trưng n-gram sử dụng phương pháp TF-IDF [105]. Với mỗi n-gram, giá trị tf-idf được tính toán như sau:

$$tf(t,d) = \frac{f(t,d)}{\max\{f(\omega,d) : \omega \in d\}} \quad (2.\ 11)$$

$$idf(t,D) = log\frac{N}{|\{d \in D : t \in d\}|} \quad (2.\ 12)$$

$$tf - idf(t,d,D) = tf(t,d) \times idf(t,D) \quad (2.\ 13)$$

Trong đó, $tf(t,d)$ là tần suất của n-gram *t* trong URI *d*; $f(t,d)$ là số lần n-gram *t* xuất hiện trong URI *d*; $\max\{f(\omega,d) : \omega \in d\}$ là số lần xuất hiện lớn nhất của một n-gram bất kỳ trong URI *d*; *D* là tập tất cả các URI và *N* là tổng số lượng các URI.

Bước 4: Giảm chiều dữ liệu sử dụng phương pháp hệ số tương quan, phương pháp Information Gain, hoặc phương pháp PCA.

2.3.3. Huấn luyện và phát hiện

Mở đầu giai đoạn huấn luyện, các truy vấn URI bình thường và URI tấn công xây dựng bộ dữ liệu huấn luyện có số lượng URI là *M*. Trong khâu tiền xử lý tiếp theo, tách các truy vấn trong bộ dữ liệu huấn luyện thành các n-gram và sử dụng phương pháp TF-IDF để tính giá trị cho các đặc trưng n-gram này. Tuy nhiên với n-gram càng cao thì số lượng đặc trưng càng nhiều nên vector đặc trưng càng lớn. Do đó, bước tiếp theo sẽ tiến hành giảm chiều vector đặc trưng này bằng phương pháp giảm chiều dữ liệu, sau quá trình giảm chiều mô hình thu được ma trận đặc trưng là Mx(N+1) trong đó M là số lượng URI, N là số đặc trưng sau khi giảm chiều và thêm 1 cột dán nhãn của các truy vấn.

Tiếp đến, ma trận đặc trưng của các URI được huấn luyện sử dụng một số thuật toán học máy có giám sát truyền thống được hỗ trợ bởi thư viện Sklearn trong Python để xây dựng và kiểm thử các mô hình phát hiện. Các thuật toán học máy được sử

dụng bao gồm: Naive Bayes, SVM, cây quyết định và rừng ngẫu nhiên. Đối với mỗi thuật toán, lấy ngẫu nhiên 80% tập dữ liệu dùng cho quá trình huấn luyện để xây dựng mô hình phát hiện và 20% dữ liệu còn lại để kiểm thử để tính toán các độ đo đánh giá. Luận án sử dụng phương pháp cross-validation với k=10 để tính độ đo đánh giá cuối cùng là trung bình của các độ đo sau 10 lần chạy. Cuối cùng, mô hình phát hiện sử dụng thuật toán học máy nào cho kết quả tốt nhất dựa trên đánh giá các độ đo sẽ được lựa chọn để xây dựng mô hình phát hiện tấn công web dựa trên web log.

Trong giai đoạn phát hiện, các truy vấn URI được tách từ web log cần giám sát, thực hiện tiền xử lý như quá trình huấn luyện, tiếp theo sử dụng bộ phân loại từ quá trình huấn luyện để phát hiện URI là tấn công hay bình thường.

Để đánh giá kết quả thử nghiệm, ngoài các độ đo được trình bày tại mục *1.6. Các độ đo đánh giá,* luận án còn sử dụng độ đo Detection Rate (DR) để đo lường hiệu quả của mô hình phát hiện đề xuất cho từng loại tấn công web. DR cho mỗi loại tấn công web được tính như sau:

$$DR = \frac{Số\ tấn\ công\ phát\ hiện\ đúng\ của\ mỗi\ loại}{Tổng\ số\ tấn\ công\ của\ mỗi\ loại} \qquad (2.\ 14)$$

2.3.4. Tập dữ liệu thử nghiệm

Các thử nghiệm trong luận án được thực hiện trên tập dữ liệu được dán nhãn HTTP Param Dataset [88] có các truy vấn bình thường được lọc từ bộ dữ liệu HTTP CISC 2010 [5] và các truy vấn tấn công SQLi, XSS, CMDi, duyệt đường dẫn được thực hiện từ các môi trường tấn công SQLmap, XSSya, Vega Scaner, FuzzDB repository. Bộ dữ liệu này gồm 31.067 chuỗi truy vấn **?query_string** trong URI của các yêu cầu web, bao gồm độ dài và nhãn của truy vấn. Có 2 loại nhãn truy vấn là Norm (Bình thường) và Anom (Tấn công). Nhãn Anom lại gồm 4 loại tấn công cụ thể: SQLi, XSS, CMDi và duyệt đường dẫn. Số lượng từng loại URI cho như trong Bảng 2. 2. Độ dài của các URI rất đa dạng, có thể từ 1 đến 1058 ký tự, như minh họa trên *Bảng 2. 3. Độ dài các truy vấn và nhãn trong HTTP Param Dataset*.

Bảng 2. 2. Số lượng từng loại trọng tải trong HTTP Param Dataset [90]

Truy vấn	Số lượng	Nhãn
Bình thường	19.304	Norm
SQLi	10.852	Anom
XSS	532	Anom
CMDi	89	Anom
Duyệt đường dẫn	290	Anom

Bảng 2. 3. Độ dài các truy vấn và nhãn trong HTTP Param Dataset

URI / Payload	Length	Attack_type	Label
c/ caridad s/n	14	norm	norm
campello, el	12	norm	norm
campello, el	12	norm	norm
40184	5	norm	norm
-3136%') or 3400=6002	21	sqli	anom
1')) as gfzb where 7904=7904;begin dbms_lock.sleep(5); end--	60	sqli	anom
1))) union all select null,null,null#	37	sqli	anom
-5622" where 7970=7970 union all select 7970,7970,7970,7970,7970--	66	sqli	anom
location=?javascript:alert(1)>click	35	xss	anom
<svg><script>alert(/1/)</script>	32	xss	anom
</script><script>alert(1)</script>	34	xss	anom
5rt(0);'>rhainfosec	19	xss	anom
+|+dir+c:",10,cmdi,anom $+|+dir+c:"	35	cmdi	anom
&&+|+dir c:",12,cmdi,anom $&&dir c:"	36	cmdi	anom
&&+|+dir c:/	12	cmdi	anom
/../../../../../../../../../../../etc/passwd	47	path-traversal	anom
/etc/passwd	11	path-traversal	anom

2.3.5. Thử nghiệm và kết quả

2.3.5.1. Kịch bản thử nghiệm

Mục này mô tả 4 kịch bản thử nghiệm nhằm đánh giá toàn diện ảnh hưởng của các tham số sử dụng đến hiệu suất của mô hình phát hiện tấn công web đề xuất. Chi tiết về các kịch bản này như sau:

- Kịch bản 1: Đánh giá ảnh hưởng của các tham số 2-gram, 3-gram, 4-gram, 5-gram sử dụng trong khâu tiền xử lý lên hiệu suất của mô hình đề xuất với thuật toán học máy Rừng ngẫu nhiên, từ đó lựa chọn tham số n-gram cho kết quả tốt nhất. Trong kịch bản này Luận án sẽ sử dụng tập dữ liệu với 80% cho huấn luyện và 20% cho kiểm thử và phương pháp kiểm tra chéo 10 lần để tính kết quả trung bình hiệu suất phát hiện của mô hình đề xuất. Trong kịch bản này, tập đặc trưng được giữ nguyên và các phương pháp giảm chiều dữ liệu không được sử dụng.

- Kịch bản 2: Đánh giá ảnh hưởng của ba phương pháp giảm chiều dữ liệu là PCA, Information Gain, Hệ số tương quan lên tập đặc trưng thu được từ Kịch bản 1 *(thuật toán Random Forest sử dụng với n-gram cho kết quả tốt nhất)*. Từ đó lựa chọn được phương pháp giảm chiều dữ liệu cho kết quả tốt nhất. Trong kịch bản này Luận

án sử dụng tập dữ liệu với 80% cho huấn luyện và 20% cho kiểm thử và phương pháp kiểm tra chéo 10 lần để tính hiệu suất phát hiện trung bình của mô hình đề xuất.

- Kịch bản 3: Đánh giá kết quả của mô hình huấn luyện sử dụng các thuật toán học máy có giám sát Navie Bayes và SVM, Cây quyết định, Rừng ngẫu nhiên (10, 30, 50, 60 cây) với n-gram cho kết quả tốt nhất từ Kịch bản 1 và phương pháp giảm chiều dữ liệu cho kết quả tốt nhất từ Kịch bản 2, từ đó lựa chọn thuật toán cho kết quả tốt nhất, sẽ được sử dụng cho quá trình phát hiện. Trong kịch bản này Luận án sẽ sử dụng tập dữ liệu với 80% cho huấn luyện và 20% cho kiểm thử và phương pháp kiểm tra chéo 10 lần để tính hiệu suất phát hiện trung bình của mô hình đề xuất.

- Kịch bản 4: Đánh giá mô hình phát hiện đề xuất với thuật toán học máy có giám sát cho kết quả tốt nhất từ Kịch bản 3 với các nghiên cứu liên quan.

2.3.5.2. Kết quả thử nghiệm

Kết quả thử nghiệm theo Kịch bản 1 được thể hiện tại Bảng 2. 4 đánh giá các đặc trưng 2-gram, 3-gram, 4-gram, 5-gram với thuật toán Rừng ngẫu nhiên, cùng thời gian huấn luyện của mô hình. Vì trong kịch bản 1 đang đánh giá và lựa chọn số lượng n-gram phù hợp nên NCS chưa tính thời gian phát hiện. Tuy nhiên, trong Kịch bản 4 NCS có tính thời gian huấn luyện và phát hiện của mô hình đề xuất cuối cùng *(với các tham số tốt nhất đã lựa chọn từ các kịch bản trước đó)* khi so sánh với các nghiên cứu liên quan khác.

Bảng 2. 4. Kết quả đánh giá Kịch bản 1

Thuật toán	**Đặc trưng n-gram**	**PPV (%)**	**TPR (%)**	**FPR (%)**	**FNR (%)**	**ACC (%)**	**F1 (%)**	**Time(s)**
Rừng ngẫu nhiên	2-gram	98,94	99,32	0,64	0,68	99,34	99,13	17,90
	3-gram	**100**	**99,14**	**0**	**0,86**	**99,68**	**99,57**	**92,99**
	4-gram	99,91	99,1	0,05	0,9	99,63	99,51	132,56
	5-gram	100	98,80	0	1,20	99,55	99,40	135,23

Kết quả từ Bảng 2. 4 cho thấy với thuật toán Rừng ngẫu nhiên khi sử dụng đặc trưng 3-gram cho độ chính xác chung ACC và độ đo F1 cao nhất so với khi mô hình lần lượt sử dụng các đặc trưng 2-gram, 4-gram và 5-gram. Cụ thể, độ đo ACC và F1 lần lượt là 99,68% và 99,57 so với 99,34% và 99,13%, 99,63% và 99,51%, 99,55% và 99,40% khi sử dụng tương ứng với các mô hình lần lượt sử dụng 3-gram, 2-gram, 4-gram và 5-gram. Đồng thời, thời gian chạy của mô hình dựa trên thuật toán Rừng ngẫu nhiên khi sử dụng các đặc trưng 2-gram và 3-gram là ngắn hơn rất nhiều so với

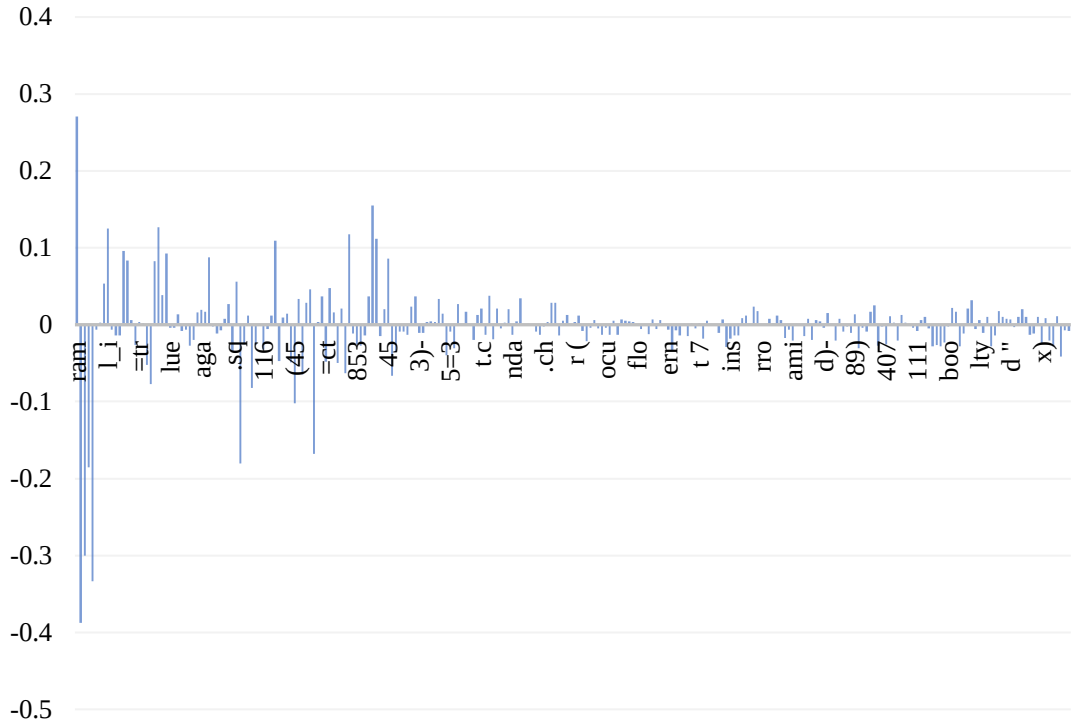
khi sử dụng 4-gram và 5-gram. Luận án không sử dụng kết hợp đồng thời các đặc trưng 2-gram, 3-gram, 4-gram và 5-gram vì 2 lý do chính sau: (1) việc kết hợp nhiều đặc trưng n-gram không làm tăng đáng kể hiệu suất phát hiện của mô hình (sử dụng 3-gram đã đạt độ chính xác chung ACC trên 99%) và (2) việc kết hợp làm số lượng đặc trưng huấn luyện sẽ rất lớn và điều này dẫn đến yêu cầu cao về tài nguyên tính toán và thời gian huấn luyện và phát hiện kéo dài, làm giảm khả năng phát hiện tấn công web theo thời gian thực. Luận án kế thừa từ các nghiên cứu trước đây đã sử dụng 3-gram và cho kết quả tốt, như trong [44]- [43]; NCS cũng tiến hành thực nghiệm với các n-gram khác nhau (gồm 2, 3, 4, 5 gram), thì 3-gram đều cho kết quả tổng thể tốt nhất, gồm cả hiệu suất phát hiện và thời gian xử lý nhanh – phù hợp với các ứng dụng giám sát theo thời gian thực. Do đó, Luận án lựa chọn đặc trưng 3-gram – là đặc trưng cho hiệu suất phát hiện cao nhất cho các quá trình xử lý dữ liệu tiếp theo tại Kịch bản 2.

Kết quả thử nghiệm theo Kịch bản 2 được thể hiện tại Bảng 2. 5 - đánh giá ảnh hưởng của ba phương pháp giảm chiều dữ liệu là PCA, Information Gain, Hệ số tương quan khi sử dụng thuật toán Rừng ngẫu nhiên và tập đặc trưng 3-gram từ Kịch bản 1.

Bảng 2. 5. Kết quả Kịch bản 2

Thuật toán	PP giảm chiều dữ liệu	PPV (%)	TPR (%)	FPR (%)	FNR (%)	ACC (%)	F1 (%)
Rừng ngẫu nhiên	**PCA**	**98,97**	**98,72**	**0,62**	**1,28**	**99,13**	**98,84**
	Information Gain	99,28	94,53	0,41	5,47	97,68	96,85
	Hệ số tương quan	99,59	92,77	0,23	7,23	97,14	96,06

Kết quả từ Bảng 2. 5 cho thấy sau khi thực hiện giảm chiều dữ liệu, phương pháp giảm chiều với PCA cho kết quả ACC, F1, TPR cao nhất so với các phương pháp giảm chiều Information Gain và Hệ số tương quan. Các độ đo ACC, F1, TPR lần lượt là 99,13%, 98,84%, 98,72% sử dụng PCA so với 97,68%, 96,85%, 94,53% và 97,14%, 96,06%, 92,77% sử dụng Information Gain và Hệ số tương quan. Như vậy, kết quả của Kịch bản 2 khẳng định phương pháp giảm chiều dữ liệu PCA cho kết quả tốt nhất và được sử dụng cho quá trình tiền xử lý dữ liệu trong mô hình đề xuất.



Hình 2. 5. Biểu đồ giá trị đặc trưng sử dụng phương pháp PCA

Kết quả thử nghiệm theo Kịch bản 3 được thể hiện tại Bảng 2. 6 - đánh giá kết quả của mô hình huấn luyện sử dụng các thuật toán học máy có giám sát Cây quyết định, Navie Bayes, SVM và Rừng ngẫu nhiên (10, 30, 50, 60 cây) với 3-gram và phương pháp giảm chiều dữ liệu PCA từ Kịch bản 1 và Kịch bản 2.

Bảng 2. 6. Kết quả Kịch bản 3

Thuật toán	**PPV (%)**	**TPR (%)**	**FPR (%)**	**FNR (%)**	**ACC (%)**	**F1 (%)**
Navie Bayes	89,48	96,41	6,84	3,59	94,38	92,82
SVM	99,87	98,50	0,08	1,50	99,09	99,18
Cây quyết định	96,48	98,42	2,17	1,58	98,05	97,44
Rừng ngẫu nhiên - 10	98,13	98,85	1,14	1,15	98,86	98,49
Rừng ngẫu nhiên - 30	98,68	98,80	0,80	1,20	99,05	98,80
Rừng ngẫu nhiên - 50	**98,97**	**98,72**	**0,62**	**1,28**	**99,13**	**98,84**
Rừng ngẫu nhiên - 60	98,80	98,76	0,72	1,24	99,08	98,78

Kết quả tại Bảng 2. 6 cho thấy mô hình sử dụng thuật toán Rừng ngẫu nhiên (50 cây) với đặc trưng 3-gram kết hợp phương pháp giảm chiều dữ liệu PCA cho hiệu suất phát hiện tốt nhất, ngược lại mô hình sử dụng thuật toán Navie Bayes cho kết quả thấp nhất. Cụ thể, độ đo ACC và F1 của mô hình dựa trên Rừng ngẫu nhiên (50

cây) là 99,13% và 98,84% so với 94,38% và 92,82% cho bởi mô hình dựa trên Navie Bayes. Các mô hình dựa trên Rừng ngẫu nhiên (30, 60 cây) và SVM đều cho kết quả độ đo ACC tương đồng, từ 99,05% đến 99,09%. Các mô hình dựa trên Rừng ngẫu nhiên (10 cây) và cây quyết định cho kết quả độ đo ACC thấp hơn đáng kể, lần lượt là 98,86% và 98,05%.

Lý do lựa chọn số lượng cây như trên thực nghiệm: hiện chưa có cơ sở lý thuyết cho việc lựa chọn số lượng cây cho thuật toán RF vào các bài toán thực tế do số lượng cây cho hiệu suất mô hình cao nhất phụ thuộc vào tập dữ liệu huấn luyện. Do vậy, NCS thực hiện chạy nhiều lần với dải các số lượng cây khác nhau để tìm ra số lượng cây cho hiệu suất cao nhất. Mặc dù vậy, số lượng cây trong RF không làm thay đổi quá lớn sự chênh lệch về hiệu suất phát hiện và đây chỉ là một phương pháp tinh chỉnh (fine-tuning) mô hình.

Kịch bản 4 đánh giá kết quả của mô hình đề xuất với các nghiên cứu liên quan về các độ đo và thời gian huấn luyện, phát hiện. Tuy nhiên, một số công trình không công bố cấu hình thực nghiệm và các tham số trong quá trình thực nghiệm nên luận án thực hiện lại các nghiên cứu này trên môi trường tương đồng với việc thực hiện mô hình đề xuất. Cụ thể các thực nghiệm này được thực hiện trên tài khoản Kaggle với cấu hình RAM 13G, GPU P100.

Bảng 2. 7. Kết quả Kịch bản 4

Mô hình	**PPV (%)**	**TPR (%)**	**FPR (%)**	**FNR (%)**	**ACC (%)**	**F1 (%)**	**Thời gian huấn luyện**	**Thời gian phát hiện**
Mô hình đề xuất - Rừng ngẫu nhiên (50 cây)	**98,97**	**98,72**	**0,62**	**1,28**	**99,13**	**98,84**	**27,52**	**1,49**
Liang và cộng sự [58]	99,04	96,88	1,13	3,12	97,78	97,95	1177,20	5,67
Ming Zhang và cộng sự [111]	98,59	93,35	1,37	6,65	96,49	95,92	151,00	4,18
Saiyu Hao cùng cộng sự [40]	98,77	93,71	0,62	6,29	97,41	96.17	13063,56	15,05
Pan và cộng sự [76]	90,60	92,80				91,80		
S. Sharma và cộng sự [91]	99,60	91,52	0,20	8,48	96,91	95,39		

Bảng 2. 7 cho thấy mô hình đề xuất dựa trên Rừng ngẫu nhiên (50 cây) cho hiệu suất phát hiện tốt hơn đáng kể so với các đề xuất [40] [58] [76] [91] [111] . Cụ

thể, các độ đo ACC, F1, TPR của mô hình đề xuất lần lượt là 99,13%, 98,84%, 98,72% so với 97,78%, 97,95%, 96,88% của Liang và cộng sự [58]; và 96,61%, 95,39%, 91,52% của Sharma và cộng sự [91]; và 096,49%, 95,92%, 93,35% của Ming Zhang và cộng sự [111]; và 97,41%, 96,17%, 93,71% của Saiyu Hao cùng cộng sự [40]. Đồng thời, mô hình đề xuất cho độ âm tính giả (FNR) và dương tính giả thấp (FPR). Cụ thể, FNR của mô hình đề xuất là thấp nhất là 1,28% so với 3,12%, 8,48%, 6,65% và 6,29% lần lượt của [58], [91], [111] và [40]. Độ đo FPR của mô hình đề xuất là 0,62% bằng với Saiyu Hao cùng cộng sự [40], thấp hơn 1,13% và 1,37% của Liang và cộng sự [58] và Ming Zhang và cộng sự [111]. Thời gian huấn luyện của mô hình đề xuất là 27,52 giây nhanh hơn nhiều so với các đề xuất [58], [111] và [40]. Thời gian phát hiện của mô hình đề xuất 1,49 giây nhanh hơn nhiều so với các đề xuất [58], [111] và [40].

Trong quá trình thực nghiệm các kịch bản, NCS đã tiến hành thực nghiệm nhiều lần để lựa chọn các siêu tham số phù hợp đối với mỗi thuật toán học máy. Quá trình huấn luyện và kiểm tra khả năng phát hiện của các mô hình, NCS đồng thời tiến hành tinh chỉnh các tham số chính của các thuật toán học máy để tìm ra mô hình phù hợp nhất. Dưới đây là giải thích chi tiết về việc sử dụng và lựa chọn các tham số, siêu tham số trong thực nghiệm gồm các thuật toán (TF-IDF, PCA, RF):

TF-IDF Vectorizer chuyển đổi văn bản thành ma trận TF-IDF, nơi mỗi giá trị đại diện cho tần suất xuất hiện của một từ/ngữ kết hợp với mức độ quan trọng của từ/ngữ đó trong văn bản. Các siêu tham số sử dụng:

- min_df=0.0: Tham số này xác định số lần tối thiểu một từ/ngữ phải xuất hiện trong tập dữ liệu để được giữ lại. Giá trị 0.0 có nghĩa là không bỏ qua từ/ngữ nào.

- analyzer="char": Xác định mức độ phân tích, ở đây là ký tự.

- sublinear_tf=True: Sử dụng TF dưới dạng logarit, giúp giảm bớt ảnh hưởng của các từ xuất hiện quá nhiều lần.

- ngram_range=(3, 3): Sử dụng n-gram ở cấp độ 3 ký tự (trigram).

- max_features=8000: Giới hạn số lượng tính năng (features) tối đa được sử dụng trong ma trận TF-IDF là 8000.

PCA giảm chiều dữ liệu bằng cách chọn các thành phần chính (principal components) để giữ lại nhiều thông tin nhất có thể.

- n_components=256: Xác định số lượng thành phần chính được giữ lại là 256. Việc chọn 256 thành phần này có thể dựa trên một số tiêu chí như giữ lại một tỷ lệ

nhất định của tổng phương sai hoặc thử nghiệm để xác định số lượng thành phần phù hợp với dữ liệu.

Random Forest là một thuật toán học máy sử dụng nhiều cây quyết định để thực hiện phân loại.

- n_estimators=50: Số lượng cây quyết định trong rừng.

- criterion='gini': Xác định tiêu chí dùng để đánh giá chất lượng phân chia tại mỗi nút trong cây quyết định.

- max_depth=None: Xác định độ sâu tối đa của mỗi cây trong rừng. None có nghĩa là không giới hạn độ sâu, cho phép cây phát triển cho đến khi tất cả các lá chỉ chứa một lớp dữ liệu hoặc các điều kiện dừng khác được thỏa mãn.

- min_samples_split=2: Xác định số lượng mẫu tối thiểu cần có tại một nút trước khi nút đó có thể được chia thành các nút con. Giá trị nhỏ hơn có thể dẫn đến cây quyết định quá phức tạp và dễ bị overfitting.

- min_samples_leaf=1: Xác định số lượng mẫu tối thiểu cần có tại một lá (leaf node). Giá trị nhỏ hơn cho phép cây phân chia đến các lá nhỏ hơn, trong khi giá trị lớn hơn giúp làm cho mô hình tổng quát hơn.

- min_weight_fraction_leaf=0.0: Xác định tỷ lệ trọng số tối thiểu của các mẫu trong một lá.

- max_features='sqrt': Xác định số lượng đặc trưng được xem xét tại mỗi phân chia nút. sqrt có nghĩa là số lượng đặc trưng sẽ là căn bậc hai của tổng số đặc trưng, giúp giảm sự tương quan giữa các cây trong rừng và cải thiện tính tổng quát.

- max_leaf_nodes=None: Xác định số lượng lá tối đa trong cây quyết định. None có nghĩa là không giới hạn số lượng lá.

- min_impurity_decrease=0.0: Xác định mức giảm tối thiểu của độ không tinh khiết (impurity) cần thiết để thực hiện một phân chia. Giúp điều chỉnh mức độ tinh khiết mà cây cần đạt được để thực hiện phân chia.

- bootstrap=True: Quyết định có sử dụng phương pháp bootstrap (tạo mẫu với thay thế) để tạo ra các tập con dữ liệu cho mỗi cây hay không. True nghĩa là sử dụng bootstrap.

- oob_score=False: Xác định có tính toán điểm số Out-Of-Bag (OOB) hay không. OOB score là một phương pháp đánh giá mô hình mà không cần một tập dữ liệu kiểm tra riêng biệt.

- n_jobs=None: Xác định số lượng lõi CPU sử dụng để tính toán. None có nghĩa là sử dụng một lõi, trong khi giá trị khác có thể chỉ định số lượng lõi để tăng tốc độ tính toán.

- verbose=0: Quyết định mức độ thông báo chi tiết trong quá trình huấn luyện. 0 có nghĩa là không có thông báo, trong khi giá trị lớn hơn có thể cung cấp thông tin chi tiết hơn.

- warm_start=False: Quyết định có sử dụng trạng thái đã huấn luyện trước đó để tiếp tục huấn luyện thêm hay không. False nghĩa là không sử dụng warm start.

- class_weight=None: Xác định trọng số cho các lớp khác nhau trong dữ liệu huấn luyện. None có nghĩa là tất cả các lớp có trọng số bằng nhau. Trọng số có thể giúp cải thiện mô hình khi dữ liệu không cân bằng.

- ccp_alpha=0.0: Xác định tham số pruning (cắt tỉa) để giảm độ phức tạp của cây quyết định. 0.0 nghĩa là không áp dụng pruning. Giá trị lớn hơn sẽ giúp loại bỏ các nhánh cây ít quan trọng hơn.

- max_samples=None: Xác định số lượng mẫu tối đa được sử dụng để huấn luyện mỗi cây quyết định. None nghĩa là sử dụng tất cả các mẫu trong tập huấn luyện.

- monotonic_cst=None: Xác định các ràng buộc đơn điệu cho các đặc trưng. None có nghĩa là không áp dụng ràng buộc nào. Khi có, nó giúp đảm bảo rằng mối quan hệ giữa các đặc trưng và nhãn là đơn điệu.

Bảng 2. 8 thể hiện kết quả của mô hình đề xuất dựa trên rừng ngẫu nhiên trong phát hiện từng loại tấn công cụ thể: SQLi, XSS, CMDi, duyệt đường dẫn (Path) và tỷ lệ phát hiện trung bình. Có thể thấy, mô hình cho tỷ lệ phát hiện tấn công SQLi cao nhất và tỷ lệ phát hiện tấn công CMDi là thấp nhất.

Bảng 2. 8. Tỷ lệ phát hiện (DR) cho các cuộc tấn công web trên thuật toán học máy

SQLi(%)	**XSS(%)**	**CMDi(%)**	**Path(%)**	**Trung bình (%)**
99,90	98,68	82,02	98,62	99,67

2.3.6. Nhận xét

Trong mục này, luận án đánh giá về hiệu suất phát hiện của mô hình đề xuất dựa trên các khía cạnh sau: (1) ảnh hưởng của việc phân bố số lượng tấn công web đến tỷ lệ phát hiện, (2) hiệu suất phát hiện của mô hình dựa trên các thuật toán học máy khác nhau và (3) so sánh giữa mô hình đề xuất với các đề xuất trước đó.

Như đã trình bày trong mục *2.3.4. Tập dữ liệu thử nghiệm*, bộ dữ liệu có phân bố không đều về số lượng các loại tấn công web, cụ thể: Số lượng tấn công SQLi chiếm đa số với khoảng 97% trong tổng số các loại tấn công, vì đây là loại hình tấn công phổ biến nhất, trong khi các kiểu tấn công khác, gồm XSS, CMDi và duyệt đường dẫn chỉ chiếm tỷ lệ khoảng 3%. Số lượng các loại tấn công web cụ thể trong tập dữ liệu phân bố không cân bằng do đó ảnh hưởng tới hiệu suất phát hiện với từng loại tấn công web. Điều này dẫn đến tỷ lệ phát hiện cho tấn công SQLi là cao nhất và tỷ lệ phát hiện cho tấn công CMDi là thấp nhất. Cụ thể, tỷ lệ phát hiện SQLi, duyệt đường dẫn, XSS và CMDi sử dụng thuật toán rừng ngẫu nhiên lần lượt là 99,90%, 98,62%, 98,68% và 82,02%, như cho trên Bảng 2. 8. Hiệu suất phát hiện tấn công CMDi không cao do lượng dữ liệu huấn luyện trong tập dữ liệu cho loại tấn công web này không đủ để xây dựng một mô hình phát hiện tốt.

Như đã trình bày trong mục *2.3.1. Giới thiệu mô hình*, mô hình phát hiện đề xuất trong luận án được xây dựng dựa trên các thuật toán học máy có giám sát sử dụng dữ liệu huấn luyện do đó không yêu cầu phải xây dựng, cập nhật thường xuyên các tập luật phát hiện như trong [17] [25] [51] [75]. Ngoài ra, mô hình đề xuất không yêu cầu truy cập vào mã nguồn của các trang web như trong [39] cũng như không cần các cơ chế đặc biệt như [76] để thu thập thông tin đầu vào vì mô hình sử dụng dữ liệu web log là dữ liệu đầu vào để phát hiện các dạng tấn công web. Về hiệu suất phát hiện, mô hình đề xuất cho độ chính xác ACC là 99,13%, F1 là 98,84%, Recall là 98,72% cao nhất so [40] [58] [76] [91] [111], đồng thời thời gian huấn luyện của mô hình đề xuất cũng nhanh hơn nhiều so với các nghiên cứu [40] [58] [91] khi thực nghiệm trên cùng một môi trường như cho trên Bảng 2. 7.

Kết quả thực nghiệm trong luận án cho kết quả tốt hơn các nghiên cứu liên quan về các độ đo và thời gian do các lý do: (1) Trong quá trình xử lý dữ liệu, luận án đã tập trung xử lý chuỗi truy vấn (**?query_string**) trong URI truy cập trong khi các nghiên cứu khác lại sử dụng toàn bộ URI. Chuỗi truy vấn là các thành phần tin tặc thường sử dụng để nhúng các đoạn mã tấn công vào địa chỉ URL của trang web. Việc sử dụng toàn bộ URI có thể làm giảm tỷ trọng của dữ liệu tấn công chứa trong chuỗi truy vấn dẫn đến làm giảm hiệu suất phát hiện. (2) Luận án sử dụng phương pháp trích chọn đặc trưng n-gram là phương pháp tương đối đơn giản và sử dụng các thuật toán học máy truyền thống cho quá trình huấn luyện và phát hiện do đó thời gian xử lý sẽ ngắn hơn, thích hợp với các hệ thống giám sát web log để phát hiện tấn công theo thời gian thực.

2.4. Kết luận chương

Trong phần mở đầu, Chương 2 đã trình bày khái quát về web log và một số dạng web log phổ biến; khảo sát một số nghiên cứu liên quan trong phát hiện tấn công web sử dụng các kỹ thuật học máy và học sâu từ đó có những đánh giá về ưu nhược điểm và đề xuất mô hình phát hiện của luận án.

Nội dung chính của chương 2 của luận án đề xuất mô hình phát hiện tấn công web dựa trên các kỹ thuật học máy có giám sát sử dụng dữ liệu web log. Mô hình có khả năng phát hiện 4 kiểu tấn công web nguy hiểm, bao gồm SQLi, XSS, CMDi và duyệt đường dẫn. Các thử nghiệm trên tập dữ liệu được gán nhãn và nhật ký web thực cho thấy mô hình phát hiện dựa trên thuật toán rừng ngẫu nhiên đạt được độ chính xác tổng thể (ACC) và độ đo F1 cao, lần lượt là 99,13% và 98,84%. Kết quả hiệu suất chung cho thấy mô hình đề xuất có thể phát hiện các cuộc tấn công web phổ biến một cách hiệu quả và nó hoạt động tốt hơn các mô hình phát hiện được khảo sát [40] [58] [76] [91] [111], đồng thời thời gian huấn luyện của mô hình cũng nhanh hơn đáng kể so với các đề xuất [40] [58] [91].

Ngoài hiệu suất phát hiện cao, mô hình được đề xuất còn có một số ưu điểm khác so với các đề xuất đã có: (1) mô hình được xây dựng sử dụng các thuật toán học máy có giám sát truyền thống nên có chi phí tính toán thấp nhưng đạt được hiệu suất phát hiện tốt. Điều này rất quan trọng đối với việc triển khai thực tế vì hệ thống phát hiện tấn công web thường cần phải xử lý một lượng rất lớn nhật ký web; (2) việc xây dựng mô hình có thể được thực hiện tự động và không yêu cầu cập nhật thường xuyên; đồng thời, thời gian huấn luyện của mô hình đề xuất ngắn hơn nhiều lần so với các đề xuất trước đó.

Một số hạn chế trong mô hình đề xuất bao gồm: (1) hiện tại mô hình đang được huấn luyện bằng tập dữ liệu chưa đủ lớn nên chưa bao hàm hết được các trường hợp tấn công và (2) trong tập dữ liệu hiện đang có sự chênh lệch về lượng dữ liệu giữa các kiểu tấn công (kiểu tấn công SQLi đang chiếm đa số). Để khắc phục các hạn chế trên có thể sử dụng thêm phương pháp sinh dữ liệu, giúp tập dữ liệu cân bằng hơn giữa các loại tấn công.

Nội dung của chương này cũng được công bố tại công trình:

1. Hoàng Xuân Dậu, **Nguyễn Trọng Hưng**, “Phát hiện tấn công web thường gặp dựa trên học máy sử dụng web log”, Hội nghị khoa học quốc gia về "Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin" FAIR 2020.

CHƯƠNG 3. PHÁT HIỆN TẤN CÔNG THAY ĐỔI GIAO DIỆN TRANG WEB

Chương 3 giới thiệu khái quát về tấn công thay đổi giao diện, các phương pháp phát hiện tấn công thay đổi giao diện, so sánh các phương pháp phát hiện thay đổi giao diện sử dụng đặc trưng ảnh chụp màn hình trang web. Phần cuối của chương mô tả việc xây dựng, cài đặt, thử nghiệm và đánh giá mô hình phát hiện tấn công thay đổi giao diện trang web dựa trên học sâu sử dụng kết hợp đặc trưng ảnh chụp màn hình và đặc trưng nội dung văn bản của trang web.

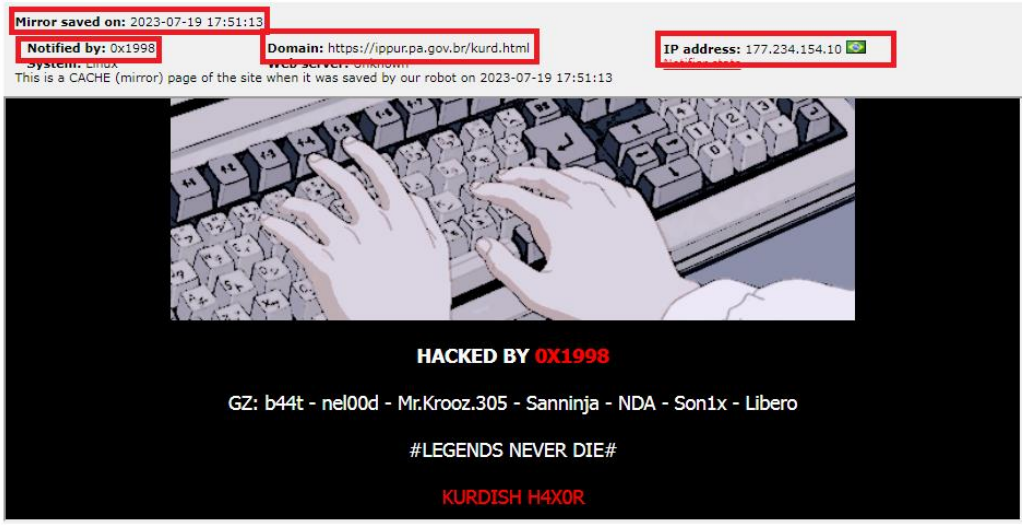
3.1. Khái quát về tấn công thay đổi giao diện và phòng chống

3.1.1. Giới thiệu

Tấn công thay đổi giao diện trang web là việc khai thác các lỗ hổng trên trang web hoặc máy chủ web nhằm thay đổi giao diện hoặc xóa, thay đổi nội dung của trang web thông qua văn bản, hình ảnh hoặc cả hai [7]. Theo thống kê trên trang web Zone-h.org, khoảng 250.000 trang web bị tấn công thay đổi giao diện trên toàn thế giới vào năm 2022 và con số này là hơn 100.000 trang web trong 6 tháng đầu năm 2023[8]. Hình 3. 1 và Hình 3. 2 là ảnh chụp màn hình bị tấn công thay đổi giao diện của các trang web ippur.pa.gov.br và jbail-byblos.gov.lb thuộc Brazil tương ứng vào tháng 7 và tháng 10 năm 2023[8]. Theo các thông báo để lại, trang web ippur.pa.gov.br đã bị nhóm tin tặc "0x1998" tấn công.

Hình 3. 1. Trang web jbail-byblos.gov.lb bị thay đổi giao diện 10/2023.

[8] "Zone-H.org," [Online]. Available: http://www.zone-h.org/stats/ymd.



Hình 3. 2. Trang web có tên miền ippur.gov.br của Brazil bị tấn công thay đổi giao diện vào tháng 7/2023

Có nhiều nguyên nhân khiến các trang web, cổng thông tin điện tử và ứng dụng web bị tấn công thay đổi giao diện [36]. Tuy nhiên, nguyên nhân chính là các lỗ hổng bảo mật nghiêm trọng tồn tại trong các trang web, cổng thông tin điện tử và ứng dụng web hoặc các máy chủ lưu trữ, cho phép tin tặc thực hiện các cuộc tấn công [7] [38] [42] [43] [62]. Theo [83], XSS (Cross-Site Scripting), SQLi (SQL injection), lỗi bao hàm các tệp cục bộ hoặc từ xa, quản lý tài khoản và mật khẩu không đúng cách và phần mềm không cập nhật là những lỗ hổng bảo mật nghiêm trọng và phổ biến nhất tồn tại trong các trang web, cổng thông tin điện tử và ứng dụng web tạo điều kiện cho tin tặc thực hiện các cuộc tấn công thay đổi giao diện.

Các cuộc tấn công thay đổi giao diện trang web có thể gây ra các hậu quả nghiêm trọng cho cá nhân, tổ chức sở hữu của trang web đó. Những cuộc tấn công này có thể ngay lập tức làm gián đoạn các hoạt động bình thường của trang web, gây tổn hại đến danh tiếng của cá nhân, tổ chức sở hữu và có thể gây thất thoát dữ liệu. Thiệt hại về danh tiếng các tổ chức và cá nhân là chủ sở hữu website bị tấn công cũng nghiêm trọng và có thể kéo dài. Có hàng trăm, thậm chí hàng ngàn khách hàng chứng kiến website của công ty bị thay đổi giao diện dẫn đến mất lòng tin và có thể kéo theo nhiều hậu quả khác, như giá cổ phiếu sụt giảm, khó khăn trong bán hàng,… Nếu không được xử lý kịp thời, điều này có thể biến thành cuộc khủng hoảng truyền thông khi sự kiện được quảng bá nhanh chóng trên các mạng xã hội và các phương tiện truyền thông đại chúng.

Một trong các hậu quả nghiêm trọng khác của tấn công thay đổi giao diện là tiềm tàng gây mất mát dữ liệu. Không phải tất cả các cuộc tấn công thay đổi giao diện

đều có mục đích phá hoại trực diện. Do tấn công thay đổi giao diện rất dễ gây chú ý, nên một số tin tặc sử dụng chúng để chuyển hướng sự chú ý để có thêm thời gian thực hiện các hành vi tấn công nguy hiểm khác như đánh cắp thông tin nhạy cảm, cài đặt mã độc, leo thang đặc quyền,… Trong một số trường hợp, việc thay đổi giao diện hay xóa trang web có thể có nghĩa là một sự mất mát dữ liệu đã diễn ra. Riêng việc mất mát dữ liệu có thể có hậu quả rất lớn. Nó có thể dẫn đến các vụ kiện và phạt nặng, đặc biệt nếu các cuộc điều tra cho thấy rằng công ty chủ sở hữu đã không tuân thủ các yêu cầu pháp lý về bảo mật dữ liệu và đảm bảo tính riêng tư cho người dùng. Nó thậm chí có thể buộc giám đốc điều hành công ty phải từ chức, hoặc bị đuổi việc do hậu quả của việc mất mát dữ liệu được công bố rộng rãi.

3.1.2. Phòng chống tấn công thay đổi giao diện trang web

Do mức độ lan rộng và hậu quả nghiêm trọng của các cuộc tấn công làm thay đổi giao diện với các trang web, cổng thông tin điện tử và ứng dụng web, nhiều biện pháp và công cụ đã được nghiên cứu, phát triển và triển khai trên thực tế để chống lại các cuộc tấn công này [38] [43] [44]. Các giải pháp phòng chống tấn công thay đổi giao diện website có thể được chia thành 3 nhóm chính sau:

- Nhóm (A) bao gồm các giải pháp và công cụ để rà quét các lỗ hổng bảo mật trong máy chủ lưu trữ và ứng dụng web, như Acunetix Vulnerability Scanner, App Scanner [6] và Abbey Scan[9] đã được trình bày tại mục *1.3.2. Các giải pháp và công cụ phát hiện tấn công web.*

- Nhóm (B) sử dụng các công cụ giám sát và phát hiện tấn công thay đổi giao diện, như VNCS Web Monitoring [61], Nagios Web Application Monitoring Software [103], Site24x7 Website Defacement Monitoring [78] và WebOrion Defacement Monitor[10], Visualping[11]; Imperva Application Security [109] và Fluxguard [77], đã được trình bày tại mục *1.3.2. Các giải pháp và công cụ phát hiện tấn công web*.

- Nhóm (C) gồm các kỹ thuật phát hiện tấn công thay đổi giao diện dựa trên chữ ký và dựa trên bất thường, bao gồm các giải pháp phát hiện đơn giản, như: so sánh checksum, so sánh DIFF và phân tích DOM tree trên các trang web [44]; hoặc các kỹ thuật, giải pháp phát hiện tấn công thay đổi giao diện trang web sử dụng thống kê, học máy. Các nghiên cứu của luận án tập trung theo hướng đề xuất các mô hình

[9] "Abbey Scan," Misterscanner, [Online]. Available: https://misterscanner.com. [Accessed 5 2021].
[10] "WebOrion Defacement Monitor," Banff Cyber Technologies, [Online]. Available: https://www.weborion.io/website-defacement-monitor/. [Accessed 5 2021].
[11] "What is Visualping?," Visualping, [Online]. Available: https://visualping.io/. [Accessed 7 2023].

phát hiện tấn công thay đổi giao diện dựa trên học máy nhằm nâng cao hiệu suất phát hiện và giảm tỷ lệ cảnh báo sai.

3.1.3. Phát hiện tấn công thay đổi giao diện

Từ các nghiên cứu đánh giá được khảo sát tại mục *1.3.3.2. Phát hiện dựa trên bất thường* có thể thấy các giải pháp đã đề xuất cho phát hiện tấn công thay đổi giao diện trang web đang tồn tại những vấn đề sau:

- Các giải pháp phát hiện tấn công thay đổi giao diện trang web dựa trên kỹ thuật đơn giản, như kiểm tra checksum, so sánh DIFF và phân tích cây DOM chỉ có thể hoạt động tốt với các trang web tĩnh.

- Một số đề xuất yêu cầu sử dụng nhiều tài nguyên tính toán do chúng sử dụng các mô hình phát hiện có độ phức tạp cao, điển hình như trong [13] [24].

- Một số đề xuất khác có mức độ cảnh báo sai cao, trong khi hiệu suất phát hiện lại phụ thuộc vào việc lựa chọn các ngưỡng phát hiện, điển hình là công trình nghiên cứu [54].

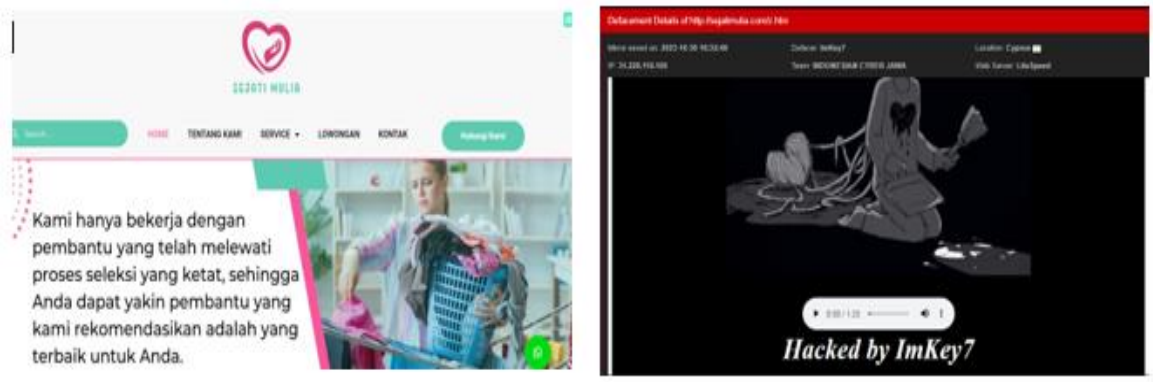
- Nhiều đề xuất chỉ có thể xử lý nội dung văn bản của các trang web. Các thành phần trang web quan trọng khác, như mã JavaScript, CSS, các tệp ảnh nhúng không được xử lý hoặc chỉ xử lý bằng kỹ thuật đơn giản, chẳng hạn như kiểm tra tính toàn vẹn dựa trên hàm băm, điển hình là các nghiên cứu [38] [43] [44].

- Các nghiên cứu [13] [38] [43] [44] sử dụng tập dữ liệu nhỏ hoặc rất nhỏ với khoảng 300 đến hơn 1000 dữ liệu trang web bị tấn công và bình thường. Tập dữ liệu thử nghiệm nhỏ có thể ảnh hưởng đến độ tin cậy của kết quả phát hiện.

- Hầu hết các nghiên cứu chỉ tập trung nghiên cứu các đặc trưng tệp tin HTML và kiểm tra mã hàm băm của các ảnh nhúng trong nội dung trang web, chưa có nghiên cứu nào tập trung vào nội dung văn bản thuần trong tệp HTML kết hợp sử dụng ảnh chụp màn hình của trang web.

Theo nghiên cứu [45] các trang web bị tấn công thay đổi giao diện thường chứa văn bản và hình ảnh trong nội dung trang web thể hiện cho mục đích của kẻ tấn công. Cũng theo nghiên cứu này, việc kết hợp dữ liệu các cuộc tấn công và dữ liệu văn bản, dữ liệu hình ảnh là nguồn dữ liệu phong phú cho phát hiện rộng quy mô hoạt động tấn công thay đổi giao diện. Mặt khác, theo Mao và cộng sự [66] và thống kê các cuộc tấn công thay đổi giao diện trang web từ [12] cho thấy sau khi bị thay đổi giao diện thì trang web có giao diện mới chỉ thuần một dải màu *(màu đen – trắng, đỏ - đen,..)* hoặc chứa tin nhắn, hình ảnh nhúng hoặc logo và video không liên quan tới

tiêu đề của trang web hoặc là các đoạn văn bản thông báo ngắn gọn “owned” hoặc “hacked by” hoặc “hacked by anonymous”. Ví dụ: Hình 3. 3 biểu diễn (a) giao diện trang web trước và (b) sau khi thay đổi về toàn màu đen chủ đạo và thông điệp “Haker by Imkey7”.



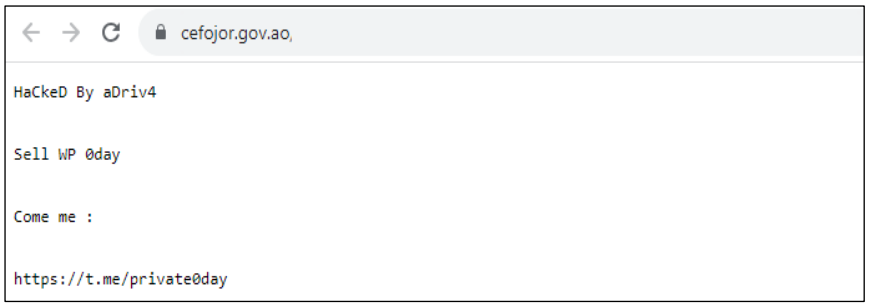
(a) (b)

Hình 3. 3. Giao diện trang sejatimulia.com trước và sau khi bị thay đổi giao diện

Với việc giao diện trang web bị tấn công thay đổi toàn bộ hoặc một phần như Hình 3. 3 (b) thì việc sử dụng đặc trưng là hình ảnh chụp màn hình trang web ngay sau khi bị tấn công sẽ có giá trị trong phát hiện tấn công thay đổi giao diện. Tuy nhiên, với những trường hợp bị thay đổi giao diện với màu sắc không nổi bật, chỉ có một vài thông điệp ngắn, như trang web cefojor.gov.ao khi bình thường trên Hình 3.4 và bị thay đổi giao diện trên Hình 3. 5, thì việc áp dụng đặc trưng ảnh chụp màn hình không còn hiệu quả. Do đó, với trường hợp này, việc sử dụng đặc trưng văn bản như các thông điệp trên giao diện bị tấn công sẽ hiệu quả hơn.

Từ những đánh giá trên, luận án đề xuất mô hình phát hiện tấn công thay đổi giao diện trang web dựa trên việc kết hợp hai đặc trưng hình ảnh chụp màn hình trang web và nội dung văn bản trích xuất từ trang web.

Hình 3. 4. Trang web cefojor.gov.ao trước khi bị tấn công thay đổi giao diện



Hình 3. 5. Trang web cefojor.gov.ao bị tấn công thay đổi giao diện

Cụ thể, luận án đề xuất mô hình phát hiện tấn công thay đổi giao diện trang web dựa trên kết hợp 2 đặc trưng hình ảnh chụp màn hình trang web và nội dung văn bản trang web theo hai nhánh: Nhánh (1) là mô hình phát hiện dựa trên đặc trưng hình ảnh chụp màn hình trang web, tập trung phát hiện hình ảnh những trang web bị tấn công thay đổi toàn bộ giao diện, như Hình 3. 3 (b); và Nhánh (2) là mô hình phát hiện tấn công thay đổi giao diện trang web dựa trên đặc trưng văn bản của trang web, tập trung phát hiện những trang web bị tấn công thay đổi giao diện với màu sắc không nổi bật, chỉ có một vài thông điệp ngắn như Hình 3. 5. Kết quả phát hiện của 2 nhánh được kết hợp để cho kết quả cuối cùng của mô hình phát hiện tổng thể.

Trong mô hình đề xuất, đặc trưng văn bản được trích xuất từ nội dung văn bản thuần của các trang web. Đặc trưng hình ảnh được trích xuất từ hình ảnh chụp màn hình của trang web. Ưu điểm của ảnh chụp màn hình của trang web là cung cấp hình ảnh chính xác về giao diện của trang web, gồm cả cấu trúc và nội dung của trang.

Mục tiếp theo của luận án sẽ trình bày về tập dữ liệu cho quá trình huấn luyện và thử nghiệm mô hình phát hiện của Nhánh (1) với đặc trưng hình ảnh chụp màn hình trang web, và Nhánh (2) với đặc trưng văn bản trích xuất từ nội dung trang web.

3.2. Thu thập bộ dữ liệu thử nghiệm

Các nghiên cứu [13] [38] [43] [44] [106] đều sử dụng tập dữ liệu có kích thước tương đối nhỏ từ 100 đến khoảng 4000 mẫu trang web bình thường và bị tấn công. Các bộ dữ liệu nhỏ có kích thước như vậy không thực sự phù hợp với các thuật toán học sâu được sử dụng trong các mô hình huấn luyện và phát hiện. Đồng thời, các nghiên cứu này tự thu thập tập dữ liệu dán nhãn tấn công từ các nguồn mở, cơ sở lưu trữ các trang web bị tấn công thay đổi giao diện, như zone-h.org, zone-xsec.com và đều không công bố đầy đủ tập dữ liệu sử dụng. Do đó, luận án sẽ tiến hành thu thập dữ liệu tấn công với nhãn "Defaced" từ nguồn zone-h.org - đây là một kho lưu trữ dữ liệu về tấn công thay đổi giao diện trang web được thành lập từ năm 2002 với nhiều thông số được lưu trữ, trong đó có hình ảnh chụp màn hình trang web tại thời điểm

bị tấn công thay đổi giao diện. Dữ liệu các trang web bình thường được dán nhãn "Normal" được thu thập từ tập 1 triệu trang web hàng đầu do Alexa cung cấp[12].

Quá trình thu thập và xây dựng bộ dữ liệu như sau:

- Thu thập các trang web bình thường: Được thu thập trực tiếp từ các trang web và các địa chỉ web uy tín trên thế giới trích xuất từ danh sách 1 triệu tên miền web xếp hạng hàng đầu theo Alexa. Tiếp tục tiến hành sàng lọc, loại bỏ các tên miền trùng lặp, sau đó thực hiện kiểm tra lại các tên miền trong danh sách và chỉ sử dụng những tên miền còn hoạt động. Tiếp theo, tải và chụp ảnh màn hình giao diện trang web, dán nhãn "Normal" và lưu trữ trong thư mục "normal\image"; Sau đó trích xuất nội dung văn bản từ mã HTML của trang web và dán nhãn "Normal", lưu trữ trong thư mục "normal\text". Bộ dữ liệu thu thập được gồm 57.220 trang bình thường. Các trang web được thu thập đảm bảo tính chính xác của bộ dữ liệu khi được thu thập từ các trang web uy tín, đồng thời có tính đa dạng khi bao gồm cả các trang web tĩnh và các trang web động với nội dung được thay đổi liên tục. Bên cạnh đó, bộ dữ liệu bao gồm cả các trang web với các ngôn ngữ khác nhau để đảm bảo mô hình huấn luyện hoạt động tốt với nhiều dạng trang web.

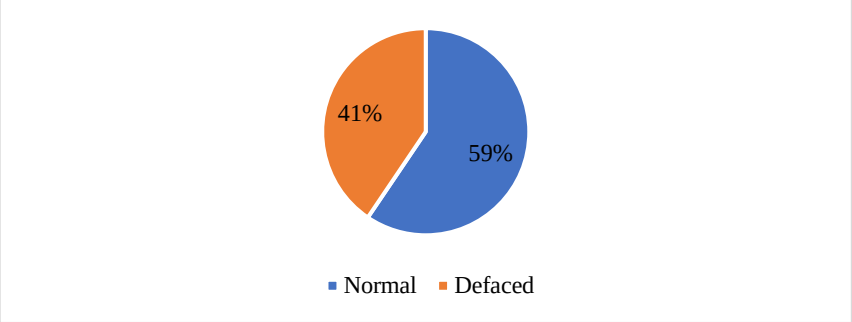
- Thu thập các trang web bị tấn công thay đổi giao diện: Đối với dữ liệu trang web bị tấn công thay đổi giao diện, tiến hành thu thập từ cơ sở dữ liệu của zone-h.org [12] với các trang web được lưu trữ từ năm 2008 đến 2019. Dữ liệu thu được sẽ là ảnh màn hình tại thời điểm trang web bị tấn công thay đổi giao diện, được dán nhãn "Defaced" và lưu trữ trong thư mục "defaced\image". Cùng với ảnh chụp màn hình, thu thập mã HTML tương ứng của trang web tại thời điểm bị tấn công thay đổi giao diện, sau đó từ mã này trích xuất các đặc trưng văn bản, được dán nhán "Defaced" và lưu trữ trong thư mục "defaced\text". Luận án chỉ thu thập các trang có cả hình ảnh chụp màn hình và mã HTML tại thời điểm bị tấn công. Bộ dữ liệu thu thập được gồm 39.014 trang web bị tấn công thay đổi giao diện. Các trang web được thống kê ở website Zone-H.org đã được xác nhận từ các quản trị viên của trang web nên đảm bảo được tính xác thực của bộ dữ liệu. Các trang web bị tấn công thay đổi giao diện được thu thập đảm bảo tính đa dạng, gồm các trang web quan trọng của các quốc gia bị tấn công thay đổi giao diện, như cổng thông tin điện tử và các trang web của các công ty, doanh nghiệp lớn cũng như của chính phủ các nước.

[12] Top Alexa one million domains.," DN Pedia, [Online]. Available: https://dnpedia.com/tlds/topm.php. [Accessed 10 2020].

Cụ thể, NCS đã thu thập 5.75GB dữ liệu ảnh chụp màn hình và nội dung văn bản của các trang web, phân bố như cho trên bảng Bảng 3. 1 và Hình 3. 6.

Bảng 3. 1. Tập dữ liệu thực nghiệm

Tập dữ liệu	Nhãn		Tổng
	Normal - 0	Defaced-1	
Image	57220	39014	96234
Text	57220	39014	96234
Image + Text			192468



Hình 3. 6. Tỷ lệ dữ liệu Normal và Defaced

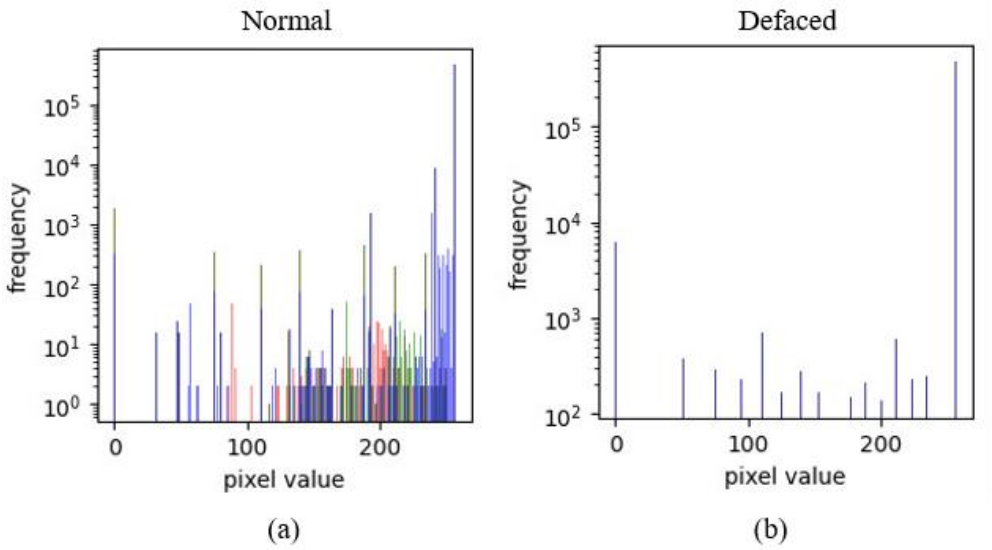
3.3. Phát hiện thay đổi giao diện sử dụng ảnh chụp màn hình trang web

3.3.1. Giới thiệu mô hình

Để hiểu rõ hơn về đặc trưng ảnh chụp màn hình các trang web được sử dụng cho mô hình đề xuất, trong luận án sử dụng kỹ thuật Histogram để tính giá trị của điểm ảnh sau đó thể hiện trên biểu đồ với 2 ảnh chụp màn hình của trang web bình thường và trang web bị tấn công. Hình 3. 7 (a) là ảnh chụp màn hình trang web bình thường có nhãn "Normal" trong tập dữ liệu và Hình 3. 7 (b) là ảnh chụp màn hình trang web bị tấn công thay đổi giao diện có nhãn "Defaced" trong tập dữ liệu.

(a) (b)

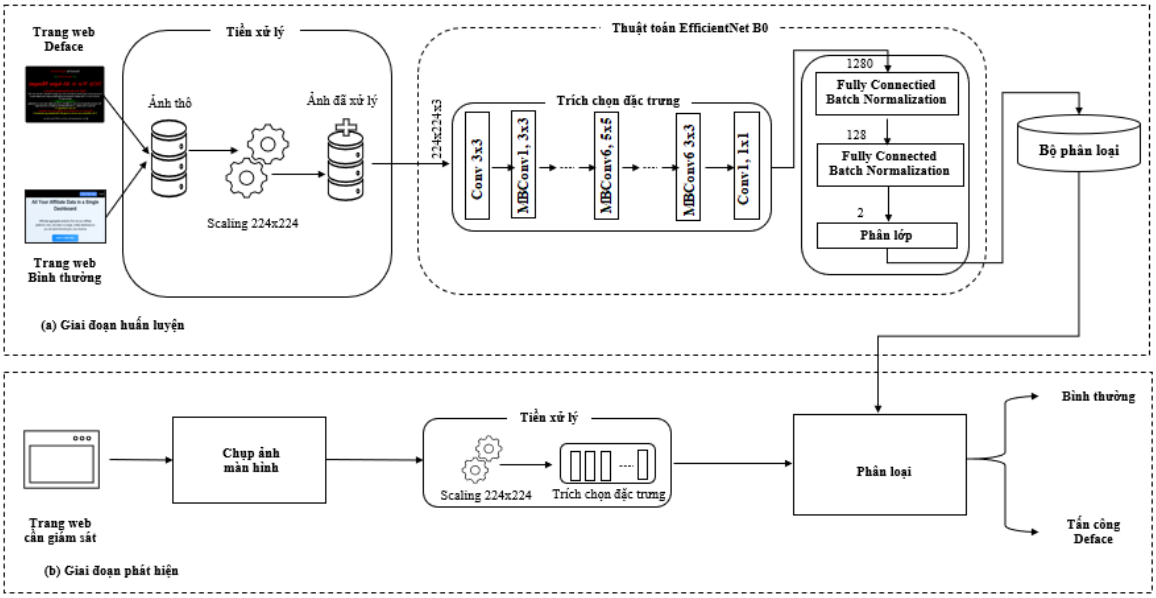
Hình 3. 7. Dữ liệu ảnh chụp trang web bình thường và khi bị tấn công



Hình 3. 8. Histogram của ảnh chụp màn hình trang khi bình thường và trang khi bị tấn công

Hình 3. 8 (a) và Hình 3. 8 (b) tương ứng là Histogram của ảnh chụp màn hình trên Hình 3. 7 (a) và Hình 3. 7 (b). Từ giá trị Histogram có thể thấy với ảnh chụp màn hình bình thường các giá trị của điểm ảnh xuất hiện đều và đa dạng hơn, còn giá trị điểm ảnh của ảnh chụp màn hình bị tấn công thay đổi giao diện có tính đơn điệu hơn và giá trị các điểm ảnh rải rác, không đồng đều. Do đó, có thể khẳng định đặc trưng ảnh chụp màn hình có khả năng phân loại tấn công thay đổi giao diện trang web.

Hình 3. 9 biểu diễn mô hình phát hiện tấn công thay đổi giao diện trang web dựa trên dữ liệu ảnh chụp màn hình trang web. Mô hình sử dụng dữ liệu đầu vào là ảnh chụp màn hình của trang web bình thường và trang web bị tấn công thay đổi toàn bộ giao diện. Mô hình được xây dựng dựa trên cơ sở phân tích đặc điểm nhận dạng của tấn công thay đổi giao diện trang web, khi trang web bị tấn công thay đổi giao diện thì toàn bộ nội dung trang web bị thay đổi và giao diện trang web cũng bị thay đổi. Mô hình đề xuất phát hiện tấn công thay đổi giao diện trang web được triển khai trong 2 giai đoạn: (a) giai đoạn huấn luyện và (b) giai đoạn phát hiện. Trong giai đoạn huấn luyện tập dữ liệu huấn luyện được thu thập từ ảnh chụp các trang web thường và trang web bị thay đổi giao diện, tiếp theo ảnh được chuẩn hóa về cùng kích thước 224x224, thực hiện tiền xử lý, sau đó trích chọn đặc trưng và huấn luyện với thuật toán học sâu EfficientNet(B0) để cho ra bộ phân loại. Trong giai đoạn phát hiện, trang web giám sát được chụp ảnh màn hình, qua quá trình tiền xử lý dữ liệu như giai đoạn huấn luyện và đến bước phân loại sử dụng bộ phân loại từ giai đoạn Huấn luyện để xác định bình thường hay tấn công.



Hình 3. 9. Mô hình phát hiện tấn công thay đổi giao diện trang web sử dụng ảnh chụp màn hình trang web

3.3.2. Tiền xử lý dữ liệu và huấn luyện mô hình phát hiện

Từ dữ liệu thu thập được của các trang web bình thường và trang web bị tấn công thay đổi giao diện, chương trình thực hiện chụp ảnh màn hình của từng trang web chuẩn bị tiền xử lý, chuẩn hóa làm dữ liệu đầu vào cho thuật toán EfficientNet(B0) thực hiện trích chọn đặc trưng và huấn luyện. Cụ thể quá trình thực hiện như sau:

Bước 1: Sử dụng kỹ thuật Scaling để chuẩn hóa ảnh mẫu thô ban đầu về đúng kích thước 224x224 - kích thước chuẩn đầu vào của thuật toán EfficientNet(B0) [98].

Tiếp đến với mỗi điểm ảnh, giá trị của chúng được chuyển về giá trị trong khoảng [0,1] với mục đích làm giảm mất mát dữ liệu khi tính toán với giá trị của các điểm ảnh ban đầu từ [0, 255] và giúp tăng tốc độ hội tụ của mô hình.

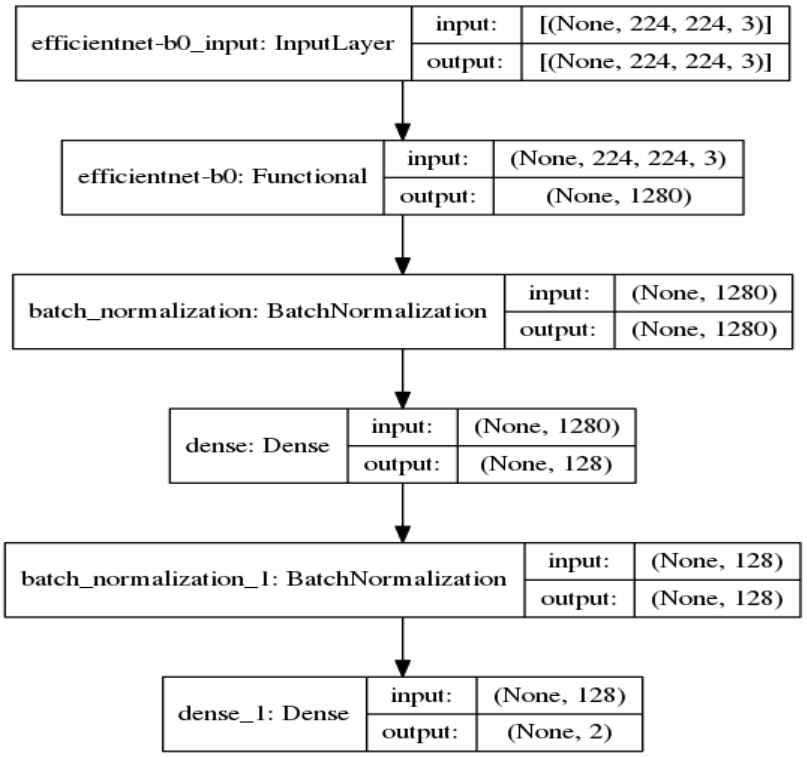
Bước 2: Sử dụng mô hình EfficientNet(B0) với cấu trúc cơ bản như Bảng 3. 2, qua từng lớp của thuật toán thu được vector đặc trưng có kích 1280.

Bước 3: Sau khi thu được tập đặc trưng 1280 từ mô hình EfficientNet(B0), tiếp đến sử dụng một lớp BatchNormalization giúp chuẩn hóa dữ liệu, tránh nhiễu ở các đặc trưng. Sau đó là 1 lớp Dense với 128 node sử dụng hàm kích hoạt là softmax kết hợp với 1 lớp BatchNormalization để chuẩn hóa ngay sau đó, giá trị đầu ra của lớp này sẽ đưa vào làm giá trị đầu vào để tính ra kết quả cuối cùng ở lớp đầu ra với 2 node là 2 giá trị xác suất cho việc hình ảnh đầu vào là bị tấn công thay đổi giao diện hay là trang web bình thường. Mạng EfficientNet được bỏ đi lớp fully-connected cuối

cùng và thay thế bằng các lớp fully-connected phân loại ảnh chụp màn hình của trang web là bình thường hay bị tấn công. Kỹ thuật BatchNormalization được sử dụng để tăng tốc quá trình hội tụ của mô hình đồng thời cũng ngăn chặn việc quá khớp trong quá trình huấn luyện. Quá trình sử dụng thuật toán EfficientNet(B0) để trích chọn đặc trưng và huấn luyện được thể hiện tại Hình 3. 10.

Bảng 3. 2. Kiến trúc cơ bản của mạng EfficientNet(B0) [98]

Stage (i)	**Operator** $(\widehat{F}_i)$	**Resolution** $(\widehat{H}_i)$ x $(\widehat{W}_i)$	**Channels** $(\widehat{C}_i)$	**Layers** $(\widehat{L}_i)$
1	Conv3x3	224 x 224	32	1
2	MBConv1, k3x3	112 x 112	16	1
3	MBConv6, k3x3	112 x 112	24	2
4	MBConv6, k5x5	56 x 56	40	2
5	MBConv6, k3x3	28 x 28	80	3
6	MBConv6, k5x5	14 x 14	112	3
7	MBConv6, k5x5	14 x 14	192	4
8	MBConv6, k3x3	7 x 7	320	1
9	Conv1x1 & Pooling & FC	7 x 7	1280	1



Hình 3. 10. Kiến trúc mạng EfficientNet(B0) cho trích chọn đặc trưng và huấn luyện

Luận án sử dụng thuật toán học sâu EfficientNet(B0) để xây dựng mô hình phát hiện từ ảnh chụp màn hình và sử dụng mô hình để phát hiện tấn công thay đổi giao diện. EfficientNet hiện được coi là một trong những kiến trúc CNN mạnh mẽ nhất trong lĩnh vực xử lý và phân loại ảnh. Dựa trên kỹ thuật phóng to mô hình, EfficientNet có khả năng đạt độ chính xác cao trong việc phân loại ảnh trong khi yêu cầu tài nguyên tính toán thấp hơn đáng kể so với các kiến trúc mạng nơ ron trước đây [98]. Ví dụ, EfficientNet nhỏ nhất (B0) chỉ với 5 triệu tham số đã có hiệu suất phân loại tốt hơn so với mô hình nổi tiếng ResNet50 với 23 triệu tham số. EfficientNet có thể giảm đáng kể số lượng tham số huấn luyện để đạt hiệu suất cao bằng cách sử dụng các khối MBConv được giới thiệu trong mạng MobileNetV2. Hơn nữa, EfficientNet có khả năng phóng to hiệu quả bằng cách cân bằng các yếu tố tạo nên mô hình: độ sâu, độ rộng và độ phân giải của mạng.

Ngoài ra, luận án lựa chọn mạng EfficientNet nhỏ nhất (B0) dựa trên 2 tiêu chí chính: (i) EfficientNet(B0) là mạng cơ sở là tiền đề cho việc xây dựng và phát triển các mạng EfficientNet từ B1 đến B7 nên có kích thước nhỏ và ít tham số hơn đáng kể so với các phiên bản sau, nên tốc độ xử lý nhanh hơn. Chẳng hạn, Mingxing Tan và cộng sự [81] đã sử dụng EfficientNet(B0) trên tập dữ liệu ImageNet với số lượng tham số nhỏ hơn 1,5 lần so với EfficientNet(B1) và nhỏ hơn 12 lần so với EfficientNet(B7); (ii) EfficientNet(B0) có khả năng hoạt động tốt trên các tập dữ liệu tương đối nhỏ. Cụ thể, EfficientNet(B0) cho hiệu suất phân loại tốt trên hai bộ dữ liệu tương đối nhỏ là CIFAR-10 và CIFAR-100 (chỉ thấp hơn khoảng 2% so với EfficientNet(B7)), nhưng EfficientNet(B0) lại nhanh hơn 12 lần so với EfficientNet(B7) [106]. Các phiên bản EfficientNet(B1) đến EfficientNet(B7) thường phù hợp cho các tập dữ liệu lớn hơn và yêu cầu sức mạnh tính toán cao hơn. Trong trường hợp dữ liệu không quá lớn, EfficientNet(B0) đạt được hiệu suất tốt với tốc độ tính toán nhanh hơn so với các phiên bản B1 đến B7 của EfficientNet. Do tập dữ liệu sử dụng trong bài toán phát hiện tấn công thay đổi giao diện trang web không quá lớn và yêu cầu thời gian phát hiện ngắn, nên việc lựa chọn EfficientNet(B0) để xây dựng mô hình phát hiện là phù hợp.

3.3.3. Tập dữ liệu thử nghiệm

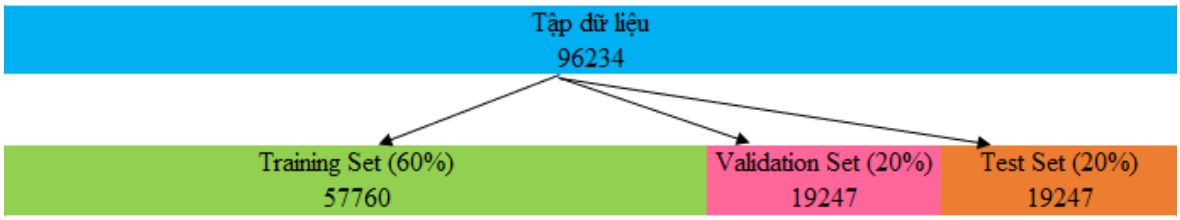
Dữ liệu trong quá trình thử nghiệm lấy từ bộ dữ liệu mô tả tại mục *3.2. Thu thập bộ dữ liệu thử nghiệm*. Tập dữ liệu ảnh chụp màn hình được sử dụng làm đầu vào cho mô hình huấn luyện. Tập dữ liệu được chia ngẫu nhiên thành 3 phần là tập huấn luyện (Training set), tập xác thực (Validation set) và tập kiểm tra (Test set) theo tỷ lệ như sau:

- Tập huấn luyện chiếm 60% được sử dụng để làm đầu vào mô hình và tinh chỉnh tham số mô hình;

- Tập xác thực chiếm 20% được dùng để kiểm tra độ chính xác của mô hình trong quá trình huấn luyện mô hình nhằm điều chỉnh tham số mô hình tránh việc quá khớp trong quá trình huấn luyện;

- Tập kiểm tra chiếm 20% dùng để đánh giá mô hình sau khi mô hình đã được huấn luyện xong.

Hình 3. 11 biểu diễn tỷ lệ dữ liệu của các Tập huấn luyện, Tập xác thực và Tập kiểm tra. Tỷ lệ giữa số trang web bình thường và số trang web bị tấn công trong mỗi tập con tương đương với tỷ lệ trong cả bộ dữ liệu thử nghiệm.



Hình 3. 11. Tỷ lệ dữ liệu ảnh chụp màn hình của các tập huấn luyện, xác thực và kiểm tra

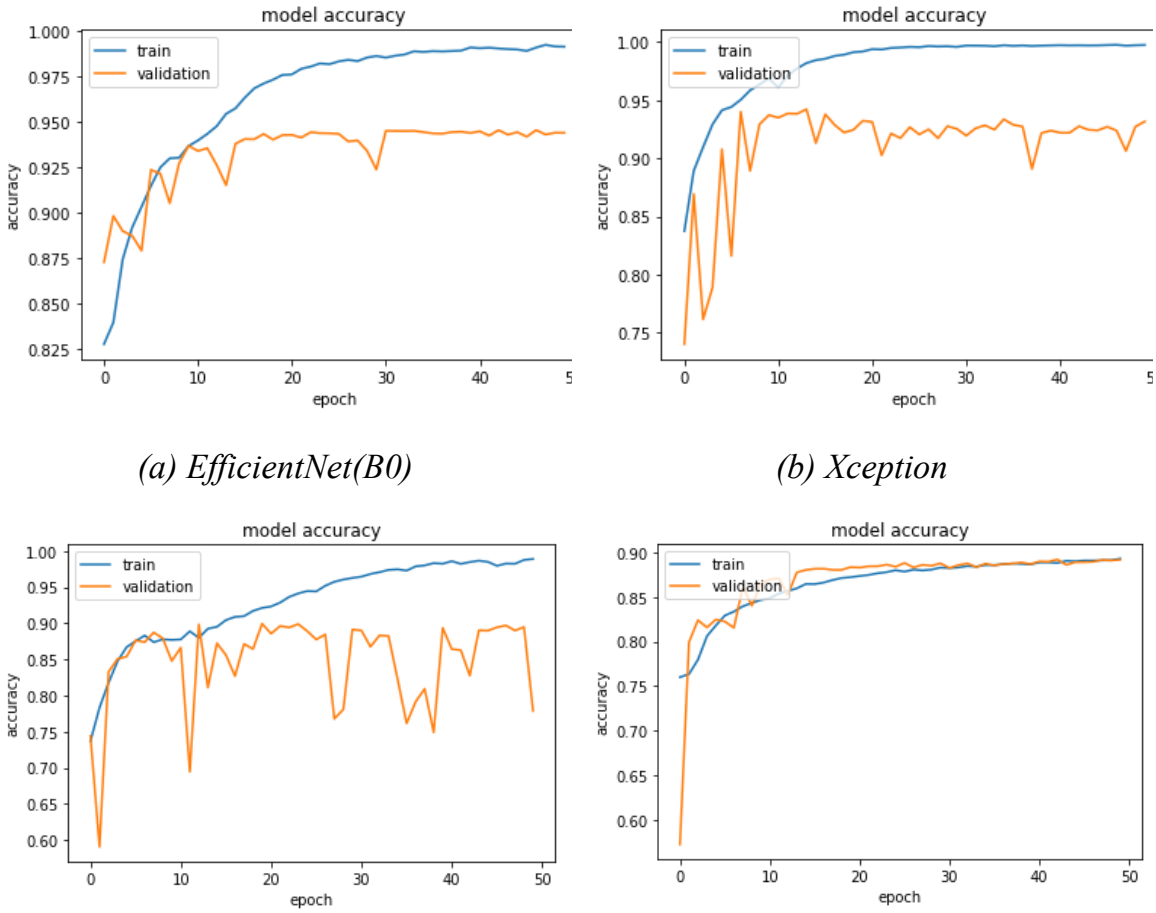
3.3.4. Thử nghiệm và kết quả

3.3.4.1. Kịch bản thử nghiệm

Để huấn luyện mô hình, luận án sử dụng tập huấn luyện với 57.740 file ảnh, tập xác thực với 19.247 file ảnh và tập kiểm tra với 19.247 file ảnh để huấn luyện và đánh giá hiệu suất của mô hình. Thuật toán học sâu được sử dụng là EfficientNet(B0). Ngoài ra, tập dữ liệu trên cũng được sử dụng để xây dựng mô hình phát hiện dựa trên các thuật toán học sâu khác, như Xception, Inception V3 và BiLSTM để so sánh với mô hình đề xuất dựa trên EfficientNet(B0).

3.3.4.2. Kết quả thử nghiệm

Hình 3. 12 là biểu đồ thay đổi accuracy (ACC - độ chính xác) trong quá trình huấn luyện với các thuật toán học sâu, gồm (a) cho EfficientNet(B0), (b) cho Xception, (c) cho Inception V3 và (d) cho BiLSTM với tập huấn luyện (train) và tập xác thực (validation). Có thể thấy kết quả huấn luyện với tập huấn luyện và tập xác thực ổn định và tốt nhất là thuật toán EfficientNet(B0), theo sau là thuật toán Xception, BiLSTM và Inception.



(a) EfficientNet(B0) *(b) Xception*

(c) Inception *(d) BiLSTM*

Hình 3. 12. Biểu đồ thay đổi accuracy (độ chính xác) trong quá trình huấn luyện với các thuật toán học sâu

Bảng 3. 3 cho thấy ma trận nhầm lẫn của mô hình đề xuất và

Bảng 3. 4 cung cấp hiệu suất của mô hình đề xuất dựa trên EfficientNet(B0) và các mô hình dựa trên Xception, BiLSTM và Inception. Kết quả cho thấy mô hình đề xuất dựa trên EfficientNet(B0) cho độ đo ACC và F1 tốt nhất, tương ứng là 94,12% và 92,62%, tiếp sau là mô hình dựa trên Xception với ACC và F1 tương ứng là 94,01% và 92,58%. Các mô hình dựa trên các thuật toán Inception và BiLSTM cho kết quả độ đo ACC và F1 đều dưới 90%.

Bảng 3. 3. Ma trận nhầm lẫn mô hình đề xuất sử dụng đặc trưng ảnh

Mô hình EfficientNet – Sử dụng đặc trưng hình ảnh chụp màn hình trang web		Actual Class	
		Attacked	Normal
Predicted Class	Attacked	7099	405
	Normal	727	11016

Bảng 3. 4. Hiệu suất của mô hình phát hiện với các thuật toán học sâu

Kỹ thuật học sâu	ACC(%)	PPV(%)	TPR(%)	F1(%)	FPR(%)	FNR(%)
EfficientNet(B0)	94.12	94.60	90.71	92.62	3.55	9.29
Xception	94.01	93.98	91.21	92.58	4.05	8.79
Inception	89.91	89.37	84.78	87.02	6.69	15.22
BiLSTM	89.18	87.73	85.22	86.46	8.13	14.78

Để có so sánh toàn diện với các nghiên cứu đã có, NCS đã cài đặt và thử nghiệm lại các mô hình phát hiện dựa trên các thuật toán học máy có giám sát đề xuất trong Hoang [38] và Hoang [44] trên cùng tập dữ liệu thử nghiệm đã thu thập mô tả trong mục *3.2. Thu thập bộ dữ liệu thử nghiệm*. Bảng 3. 5 cho thấy, hiệu suất của mô hình đề xuất so sánh với mô hình của Hoang [44] sử dụng thuật toán RandomForest khi sử dụng cùng tập huấn luyện với độ đo F1 và ACC lần lượt là 92,62% và 94,12% so với 92,26% và 93,88%. Tỷ lệ âm tính giả không quá chênh lệch trong khi tỷ lệ dương tính giả thấp hơn. Cụ thể tỷ lệ âm tính giả và dương tính giả lần lượt là 9,29% và 3,55% của mô hình đề xuất trong luận án so với 9,24% và 4,03% của mô hình trong Hoang [44] sử dụng thuật toán Rừng ngẫu nhiên. Như vậy, có thể khẳng định tập dữ liệu ảnh chụp màn hình trong mô hình đề xuất có khả năng phân loại tấn cống thay đổi giao diện trang web tốt hơn so với tập đặc trưng văn bản trong Hoang [44].

Bảng 3. 5. Hiệu suất mô hình đề xuất so với Hoang [44]

Mô hình phát hiện	ACC (%)	PPV (%)	TPR (%)	F1 (%)	FPR (%)	FNR (%)
Rừng ngẫu nhiên Hoang [44]	93,88	93,81	90,76	92,26	4,03	9,24
EfficientNet(B0)	94,12	94,60	90,71	92,62	3,55	9,29

Bảng 3. 6. cung cấp số liệu tổng hợp so sánh hiệu suất của mô hình đề xuất với các thuật toán học sâu và các mô hình phát hiện dựa trên các thuật toán học máy Naïve Bayes, cây quyết định và Rừng ngẫu nhiên đề xuất trong Hoang [38] và Hoang [44]. Có thể thấy mô hình đề xuất cho hiệu suất tốt hơn so với các đề xuất của Hoang và cộng sự [44] và tốt hơn nhiều so với Hoang và cộng sự [38].

Bảng 3. 6. Hiệu suất mô hình đề suất với các thuật toán học sâu và mô hình trước đó

Mô hình phát hiện	Đặc trưng	ACC (%)	PPV (%)	TPR (%)	F1 (%)	FPR (%)	FNR (%)
Naïve Bayes Hoang [38]	Văn bản	82,54	78,12	79,26	78,69	15,21	20,74

Mô hình phát hiện	Đặc trưng	ACC (%)	PPV (%)	TPR (%)	F1 (%)	FPR (%)	FNR (%)
Cây quyết định Hoang [38]	Văn bản	87,33	84,4	84,4	84,4	10,67	15,6
Rừng ngẫu nhiên Hoang [44]	Văn bản	93,88	93,81	90,76	92,26	4,03	9,24
Xception	Ảnh	94,01	93,98	91,21	92,58	4,05	8,79
Inception	Ảnh	89,91	89,37	84,78	87,02	6,69	15,22
BiLSTM	Ảnh	89,18	87,73	85,22	86,46	8,13	14,78
EfficientNet(B0)	**Ảnh**	**94,12**	**94,60**	**90,71**	**92,62**	**3,55**	**9,29**

3.3.5. Nhận xét

Dựa trên kết quả cho trong Bảng 3. 6, có thể rút ra nhận xét: Mô hình đề xuất hoạt động tốt hơn các đề xuất trước đó trên hầu hết các độ đo đánh giá, trong đó mô hình đề xuất cho độ chính xác chung (ACC) và độ do F1 cao hơn đáng kể so với các mô hình trước đó. Cụ thể, độ do F1 của Hoang và cộng sự [44] dựa trên thuật toán Rừng ngẫu nhiên , Hoang [38] dựa trên thuật toán Naïve Bayes và Cây quyết định và các thuật toán học sâu Xception, Inception, BiLSTM và mô hình phát hiện đề xuất tương ứng là 92,26%, 78,69%, 84,4%, 92,58%, 87,02%, 86,46% và 92,62%.

Có thể nhận thấy tỷ lệ dương tính giả (FPR) của mô hình phát hiện đề xuất thấp hơn đáng kể so với các mô hình học sâu khác sử dụng chung tập đặc trưng ảnh, và thấp hơn nhiều so với các mô hình đã có của Hoang [38] dựa trên thuật toán Cây quyết định và Naïve Bayes và mô hình Hoang và cộng sự [44] dựa trên thuật toán Rừng ngẫu nhiên. Cụ thể, FPR của mô hình đề xuất, Hoang và cộng sự [38] dựa trên thuật toán Cây quyết định và Naïve Bayes, Hoang và cộng sự [44] dựa trên thuật toán Rừng ngẫu nhiên lần lượt là 3,55%, 10,67%, 15,21% và 4,03%. Đồng thời, FNR của mô hình đề xuất cũng thấp hơn nhiều so với Hoang và cộng sự [38] dựa trên thuật toán Cây quyết định và Naïve Bayes và gần như tương đương với Hoang và cộng sự [38] dựa trên thuật toán Rừng ngẫu nhiên lần lượt là 9,29%, 20,74%, 15,6% và 9,24%.

Kết quả thực nghiệm cho thấy mô hình đề xuất cho kết quả tốt hơn các nghiên cứu khác với hai lý do: (1) ảnh chụp màn hình trang web tại thời điểm bị tấn công thay đổi giao diện được sử dụng làm đặc trưng trong mô hình đề xuất chứa đầy các thông tin và giao diện chính của trang web đã thay đổi toàn bộ nên cho khả năng phân loại hiệu quả giữa các trang web bình thường và các trang bị thay đổi giao diện; (2) với tập dữ liệu ảnh đầu vào, việc luận án lựa chọn một thuật toán học sâu được đánh giá là mạnh trong việc xử lý ảnh như EfficientNet [98] là phù hợp và cho hiệu suất phân loại tốt.

Hạn chế của mô hình đề xuất là mặc dù mô hình đề xuất có độ chính xác chung cao hơn đáng kể so với các mô hình đã có, nhưng tỷ lệ cảnh báo sai, gồm FPR và FNR vẫn trên 10% là tương đối cao. Ngoài ra, với đặc trưng ảnh, mô hình phát hiện kém với những trang bị tấn công có ít thay đổi về màu sắc như Hình 3. 5. Do đó, phần tiếp theo của Luận án sẽ đưa ra mô hình phát hiện tấn công thay đổi giao diện trang web có thể giải quyết được các vấn đề còn tồn tại nêu trên bằng cách sử dụng mô hình phát hiện sử dụng đặc trưng văn bản trong các tệp HTML.

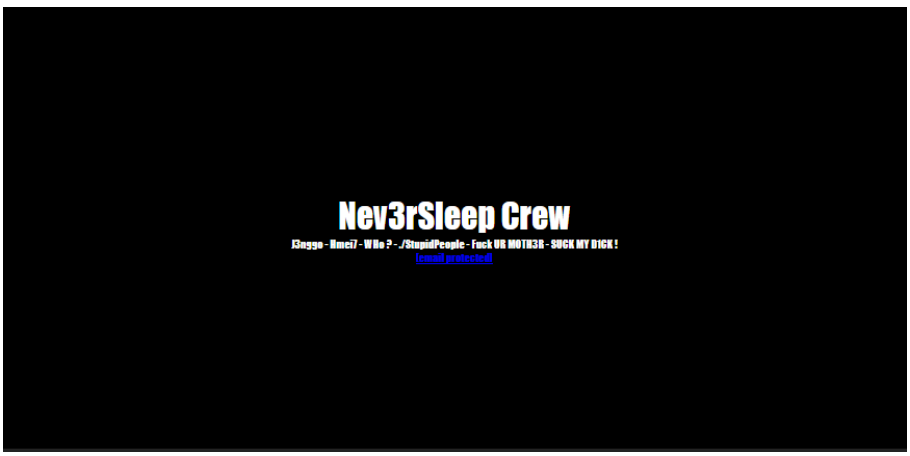
3.4. Phát hiện tấn công thay đổi giao diện sử dụng nội dung văn bản

3.4.1. Giới thiệu mô hình

Cùng với dữ liệu ảnh chụp màn hình là một đặc trưng cho phát hiện tấn công thay đổi giao diện đã được nghiên cứu đánh giá tại mục *3.3. Phát hiện thay đổi giao diện sử dụng ảnh chụp màn hình trang web*, thì việc phân tích đặc trưng văn bản trong nội dung trang web cũng đem tới giá trị phân loại tấn công thay đổi giao diện [66]. Đồng thời kế thừa từ các nghiên cứu [38] [43] [44] [106] về việc sử dụng dữ liệu ký tự trong văn bản HTML, trong phần này luận án đề xuất sử dụng đặc trưng văn bản trên các tệp HTML.

Ví dụ như tại Hình 3. 13, đặc trưng văn bản trong tệp HTML được thu thập là: Nev3rSleep Crew J3nggo - Hmei7 - WHo ? - ./StupidPeople - Fuck UR MOTH3R - SUCK MY D1CK ! .

Để trích chọn các đặc trưng văn bản thuần của các trang web, NCS thực hiện tính tần suất xuất hiện của các từ trong nội dung văn bản tệp HTML là các thông điệp hiển thị trên các trang web. Hình 3. 14 biểu diễn 1000 từ xuất hiện nhiều nhất trong tập dữ liệu trang web bị thay đổi giao diện (defaced) và Hình 3. 15 biểu diễn 1000 từ xuất hiện nhiều nhất trong tập dữ liệu trang web bình thường (normal).



Hình 3. 13. Đặc trưng văn bản trong trang web bị tấn công thay đổi giao diện

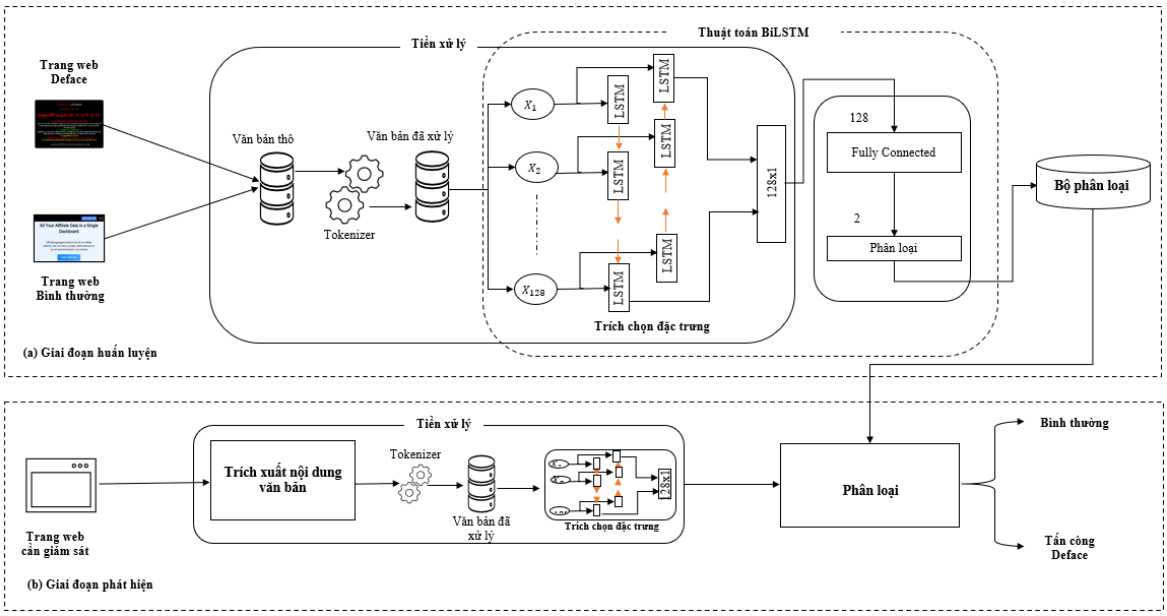
Hình 3. 14. 1000 từ xuất hiện nhiều nhất trong tập dữ liệu defaced

Hình 3. 15. 1000 từ xuất hiện nhiều nhất trong tập dữ liệu normal

Từ Hình 3. 14 và Hình 3. 15 có thể nhận thấy trong các trang web bị tấn công thay đổi giao diện, các từ như “Hacker”, “Hacked”,.. xuất hiện nổi bật với tần suất xuất hiện cao nhất, trong khi các từ trong dữ liệu bình thường xuất hiện với tần suất khá tương đồng nhau và không có từ nào thực sự nổi bật. Do đó việc phân tích các đặc trưng văn bản sẽ có ý nghĩa trong phân loại tấn công thay đổi giao diện và bình thường.

Hình 3. 16 biểu diễn mô hình đề xuất cho phát hiện tấn công thay đổi giao diện trang web sử dụng đặc trưng văn bản được triển khai trong 2 giai đoạn: (a) giai đoạn huấn luyện và (b) giai đoạn phát hiện. Trong giai đoạn huấn luyện, tập dữ liệu huấn luyện được thu thập từ trích xuất nội dung văn bản trong các tệp HTML từ các trang web bị tấn công và trang web bình thường, sau đó tệp văn bản được tách thành các từ với kỹ thuật Tokenizer, sau đó huấn luyện với thuật toán học sâu BiLSTM để tạo ra

bộ phân loại. Trong giai đoạn phát hiện, trang web giám sát được trích xuất nội dung văn bản, qua quá trình tiền xử lý dữ liệu như giai đoạn huấn luyện và đến bước phân loại sử dụng bộ phân loại từ giai đoạn Huấn luyện để xác định trạng thái là bình thường hay bị thay đổi giao diện.



Hình 3. 16. Mô hình huấn luyện, phát hiện tấn công thay đổi giao diện với đặc trưng văn bản

3.4.2. Tiền xử lý dữ liệu và huấn luyện mô hình phát hiện

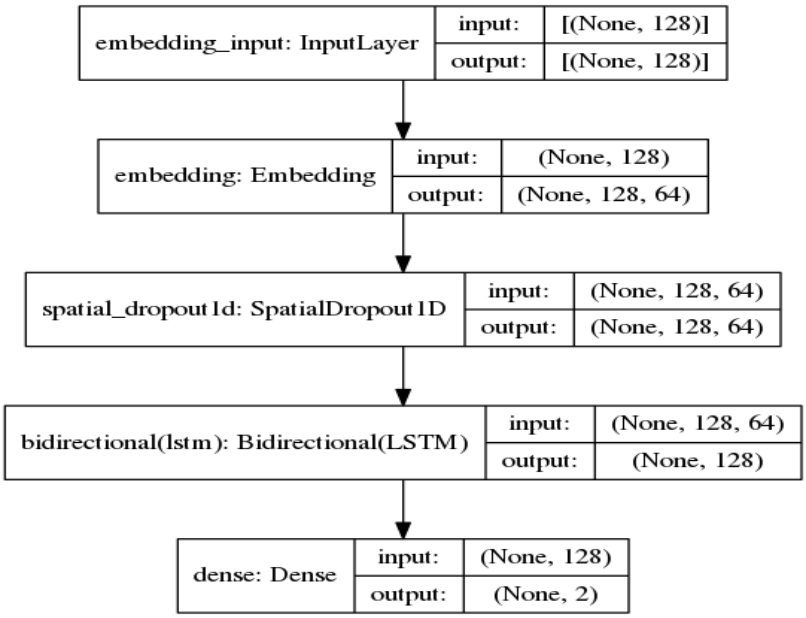
Dữ liệu văn bản thuần trích xuất từ trang web được tiền xử lý để trích chọn các đặc trưng tạo thành một vector đại diện cho trang web đó. Tiếp theo dữ liệu được huấn luyện để xây dựng mô hình phát hiện. Quá trình xử lý của mô hình đề xuất được thực hiện theo các bước như sau:

Bước 1: Từ các trang web bình thường và trang web bị thay đổi giao diện, sử dụng một chương trình tự viết bằng python trích xuất các nội dung văn bản làm dữ liệu cho quá trình huấn luyện.

Bước 2: Từ tập dữ liệu văn bản thu được sử dụng sử dụng kỹ thuật Tokenizer [95] để tách các từ trong văn bản và mỗi từ này được ánh xạ thành một số nguyên dương (khi tokenizer phân tách văn bản, nó xây dựng một từ điển từ vựng, gán cho mỗi từ một chỉ số duy nhất. Từ điển này giúp ánh xạ mỗi từ trong văn bản thành một số nguyên, tạo ra một biểu diễn số cho các chuỗi văn bản).

Tiếp đó lựa chọn 128 từ đầu tiên liên tiếp nhau làm đầu vào cho thuật toán BiLSTM, 128 từ được chọn do lượng thông tin thu được là vừa đủ cho tính toán giúp mô hình hội tụ nhanh và giảm yêu cầu tài nguyên tính toán. Ngoài ra, việc chọn dãy

từ liên tiếp nhau bảo đảm cho mô hình BiLSTM khi xử lý dữ liệu sẽ không bỏ sót thông tin nhờ mối liên hệ giữa các từ liền kề nhau. Hình 3. 17 biểu diễn thuật toán BiLSTM xử lý trích chọn đặc trưng và huấn luyện mô hình.



Hình 3. 17. Cấu trúc thuật toán BiLSTM sử dụng trong mô hình đề xuất

Bước 3: Sử dụng lớp Embedding để giúp mô hình hiểu được mối quan hệ ngữ nghĩa của các từ thông qua vector đầu vào của mô hình. Kết quả là một vector 128x128 thể hiện đặc trưng của các từ và mối quan hệ giữa các từ với nhau trong tập dữ liệu, giúp tăng khả năng hiểu nội dung văn bản của mô hình.

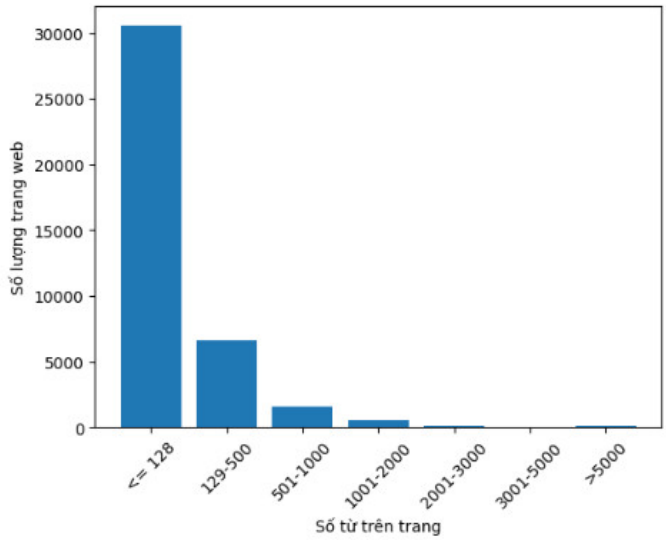
Bước 4: Sử dụng lớp GlobalMaxPooling để giảm chiều dữ liệu còn 128. Vì kích thước của dữ liệu đầu vào lớn là 128x128 việc duỗi chiều dữ liệu vector này sẽ tốn thời gian mà không đạt hiệu quả.

Bước 5: Lớp kết nối đầy đủ cuối cùng chuyển hóa 128 đặc trưng về giá trị phân loại của mô hình, sử dụng hàm kích hoạt softmax để tính xác suất phát hiện tấn công hay bình thường.

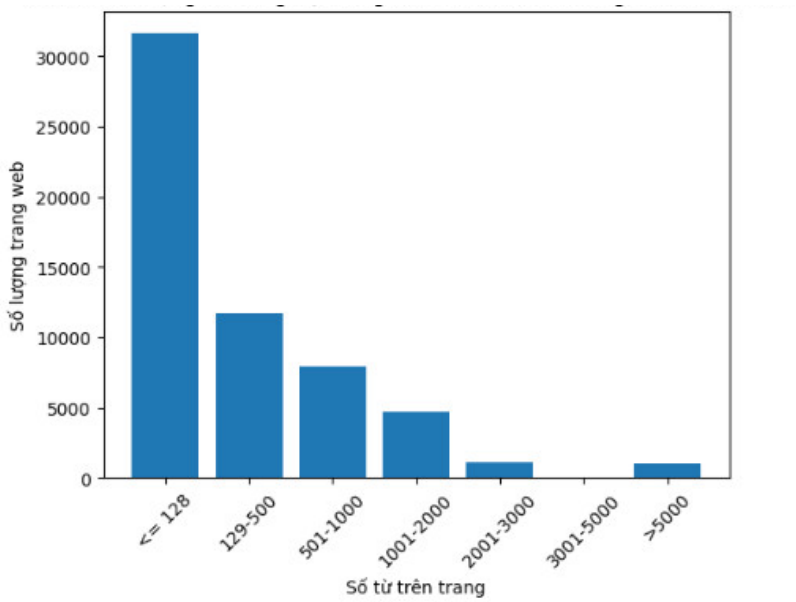
Luận án lựa chọn thuật toán BiLSTM cho mô hình phát hiện tấn công thay đổi giao diện dựa trên nội dung văn bản với một số lý do: (i) BiLSTM là thuật toán học sâu được sử dụng nhiều trong xử lý ngôn ngữ tự nhiên với các bài toán phân loại văn bản, phân loại cảm xúc,… [18] [63] [67] [107]. Về cấu trúc, mạng BiLSTM bao gồm các đơn vị LSTM hoạt động theo cả hai hướng để kết hợp thông tin từ bối cảnh quá khứ và tương lai. BiLSTM có thể tìm hiểu các phụ thuộc dài hạn mà không giữ lại thông tin trùng lặp. Do đó, nó đã chứng tỏ hiệu suất cao cho các vấn đề mô hình hóa tuần tự và được sử dụng rộng rãi để phân loại văn bản. Không giống như mạng LSTM,

mạng BiLSTM có hai lớp song song truyền theo hai hướng với các đường truyền thuận và nghịch để nắm bắt các phụ thuộc trong hai ngữ cảnh [46]; (ii) với đặc điểm văn bản của các trang web bị tấn công thay đổi giao diện như Hình 3. 13 có tính đặc trưng, số lượng từ ít nên việc sử dụng mô hình BiLSTM với tính chất là tổng hợp thông tin từ hai chiều của nội dung văn bản do đó phù hợp với bài toán đề xuất.

Luận án chọn số lượng 128 từ liên tiếp cho đầu vào của thuật toán BiLSTM với những lý do sau: (i) với tính chất của thuật toán BiLSTM đã được trình bày tại mục *1.5.7. BiLSTM* nên số từ liên tiếp nhau sẽ phù hợp với việc lựa chọn các tần suất xuất hiện của từ trong nội dung văn trên các trang web và nếu không gian từ càng lớn sẽ làm tăng yêu cầu tài nguyên tính toán, do đó cần lựa chọn không gian từ phù hợp; (ii) với đặc điểm của trang web bị tấn công thay đổi giao diện thì nội dung trang web thường được thay thế bằng hình ảnh và phần văn bản trên trang web thường không nhiều, điển hình như ví dụ tại hình Hình 3. 1 và Hình 3. 2. Đồng thời, qua đánh giá tập dữ liệu với yếu tố số từ trên các trang web bị tấn công và trên trang web bình thường thì nhận thấy số lượng trang web có số từ nhỏ hơn hoặc bằng 128 chiếm tỷ lệ vượt trội. Cụ thể, như trên Hình 3. 18, số lượng trang web bị tấn công có số lượng từ <=128 là hơn 30000 trang web (chiếm 76,90% tập dữ liệu), trong khi con số này ở trang web bình thường là khoảng gần 35000 trang web (chiếm 61,17% tập dữ liệu), như thể hiện trên Hình 3. 19. Do đó, trong bài toán này việc lựa chọn không gian từ là 128 từ làm đầu vào cho thuật toán BiLSTM là phù hợp.



Hình 3. 18. Số lượng từ trên một trang web bị tấn công thay đổi giao diện



Hình 3. 19. Số lượng từ trên một trang web bình thường

3.4.3. Tập dữ liệu thử nghiệm

Dữ liệu trong quá trình thử nghiệm được sử dụng như trong mô tả tại mục *3.2. Thu thập bộ dữ liệu* thử nghiệm. Tập dữ liệu văn bản được làm đầu vào cho mô hình huấn luyện. Tập dữ liệu được chia ngẫu nhiên thành 3 phần là tập huấn luyện (Training set), tập xác thực (Validation set) và tập kiểm tra (Testing set) như sau:

- Tập huấn luyện chiếm 60% được sử dụng để làm đầu vào mô hình và tinh chỉnh tham số mô hình;

- Tập xác thực chiếm 20% được dùng để kiểm tra độ chính xác của mô hình trong quá trình huấn luyện mô hình nhằm điều chỉnh tham số mô hình tránh việc quá khớp trong quá trình huấn luyện;

- Tập kiểm tra chiếm 20% dùng để đánh giá mô hình sau khi mô hình đã được huấn luyện xong.

3.4.4. Thử nghiệm và kết quả

Luận án lựa chọn thử nghiệm mô hình phát hiện đề xuất dựa trên thuật toán BiLSTM và các mô hình phát hiện đề xuất bởi [38] (Naive Bayes, Decision Tree) và [44] (Rừng ngẫu nhiên) chỉ sử dụng dữ liệu văn bản trích xuất từ trang web để so sánh, đánh giá.

Kết quả thử nghiệm cho trên Bảng 3. 7 và Bảng 3. 8. Có thể thấy độ chính xác ACC và độ đo F1 của mô hình phát hiện dựa trên BiLSTM cao nhất so với các mô hình đề xuất bởi [38] (Naive Bayes, Decision Tree) và [44] (Rừng ngẫu nhiên). Cụ thể, ACC và độ đo F1 của các mô hình kể trên lần lượt là 96,54% và 95,66%, 82,54%

và 78,69%, 87,33% và 84,4%, 93,88% và 92,26%. Đồng thời độ đo dương tính giả và âm tính giả của mô hình đề xuất với thuật toán BiLSTM cũng thấp hơn so với [44] [38] lần lượt là 2,03%, 5,57% so với 4,04%, 9,24% và 10,67%, 15,6%.

Bảng 3. 7. Ma trận nhầm lẫn mô hình đề xuất sử dụng đặc trưng văn bản

Mô hình BiLSTM – Sử dụng đặc trưng văn bản		Actual Class	
		Attacked	Normal
Predicted Class	Attacked	7347	233
	Normal	433	11234

Bảng 3. 8. Kết quả thử nghiệm các mô hình phát hiện dựa trên các thuật toán học máy chỉ sử dụng đặc trưng văn bản

Mô hình phát hiện	**Đặc trưng**	**ACC (%)**	**PPV (%)**	**TPR (%)**	**F1 (%)**	**FPR (%)**	**FNR (%)**
Naïve Bayes Hoang [38]	Văn bản	82,54	78,12	79,26	78,69	15,21	20,74
Cây quyết định Hoang [38]	Văn bản	87,33	84,4	84,4	84,4	10,67	15,6
Rừng ngẫu nhiên Hoang [44]	Văn bản	93,88	93,81	90,76	92,26	4,03	9,24
BiLSTM	**Văn bản**	**96,54**	**96,93**	**94,43**	**95,66**	**2,03**	**5,57**

3.4.5. Nhận xét

Từ kết quả thử nghiệm cho trên Bảng 3. 8 có thể thấy, mô hình phát hiện đề xuất dựa trên BiLSTM sử dụng đặc trưng văn bản cho các độ đo phát hiện tốt hơn đáng kể so với kết quả của các mô hình đề xuất bởi Hoang [44] và Hoang [38], cụ thể độ đo ACC, F1 lần lượt là 96,54% và 95,66% so với 93,88% và 92,26%, 87,33% và 84,4%, 82,54% và 78,69%, đồng thời độ âm tính giả và dương tính giả cũng thấp hơn nhiều so với các nghiên cứu trước đó. Kết quả thực nghiệm cho kết quả tốt hơn các nghiên cứu liên quan với hai lý do sau: (i) luận án đã lựa chọn tập dữ liệu thuần văn bản phù hợp với việc phát hiện tấn công thay đổi giao diện do các văn bản này thường thể hiện các thông tin của các nhóm tấn công và (ii) luận án lựa chọn thuật toán học sâu BiLSTM phù hợp với xử lý dữ liệu là văn bản, thuật toán BiLSTM được sử dụng nhiều trong xử lý ngôn ngữ tự nhiên và thuật toán BiLSTM bao gồm các đơn vị LSTM hoạt động theo cả hai hướng để kết hợp thông tin từ bối cảnh quá khứ và tương lai phù hợp với nội dung tập dữ liệu của bài toán đề xuất.

Như vậy có thể kết luận, các đặc trưng văn bản trích xuất từ trang web có thể được sử dụng hiệu quả để phân loại các trang web bình thường và các trang web bị

thay đổi giao diện, đặc biệt là các trang web bị thay đổi giao diện không có hoặc có ít hình ảnh và nhiều nội dung văn bản.

Tuy vậy, do dữ liệu sử dụng trong mô hình đề xuất là văn bản thuần nên mô hình đề xuất không thể phát hiện, hoặc phát hiện với độ chính xác thấp với những trường hợp tấn công chỉ thay đổi giao diện bằng hình ảnh hoặc những nội dung từ trang web nhưng mang tính chất đưa tin sai lệch. Do đó, việc kết hợp thêm đặc trưng hình ảnh trong mô hình phát hiện là cần thiết. Phần tiếp theo của luận án sẽ sử dụng mô hình kết hợp dựa trên đặc trưng hình ảnh chụp màn hình và văn bản trích xuất từ trang web để nâng cao hiệu quả phát hiện tấn công thay đổi giao diện.

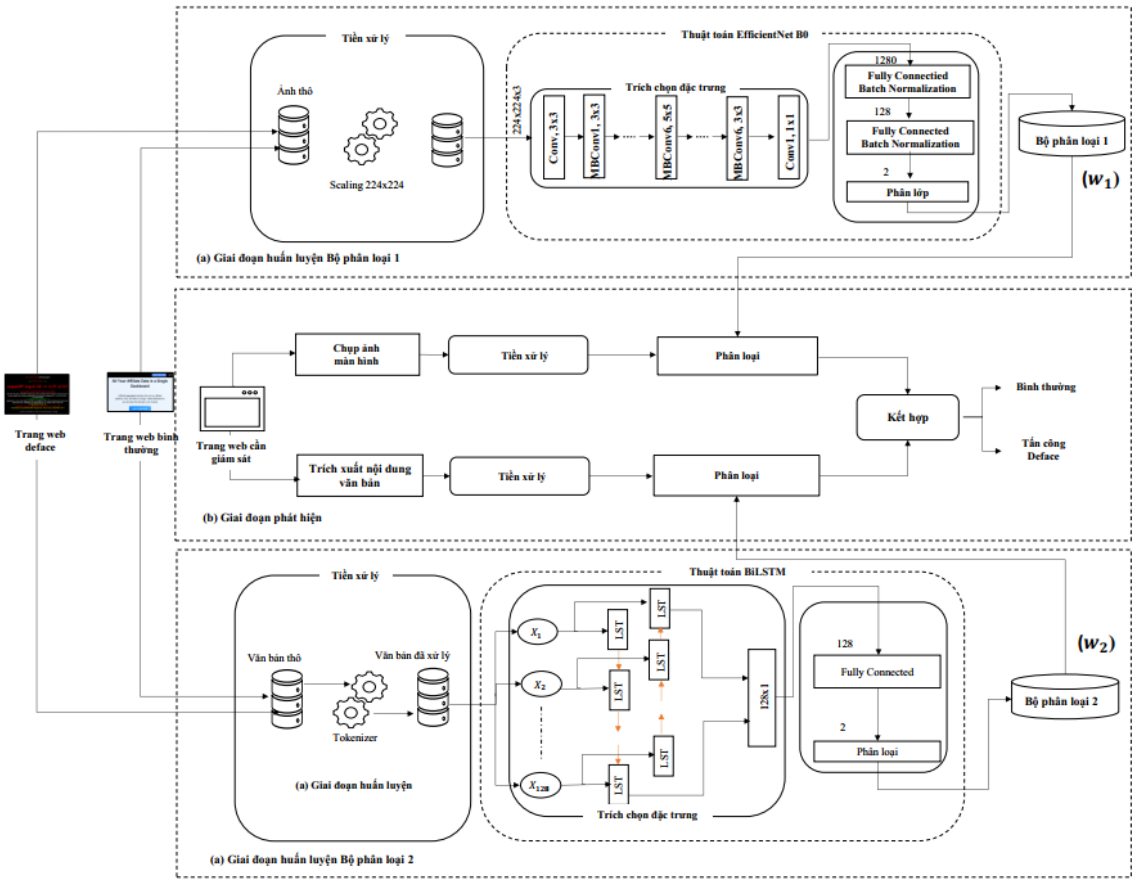
3.5. Phát hiện thay đổi giao diện sử dụng kết hợp nội dung văn bản và ảnh chụp màn hình trang web

3.5.1. Mô tả mô hình phát hiện

Mô hình phát hiện đề xuất nhằm cải thiện khả năng phát hiện chính xác và giảm tỷ lệ cảnh báo sai bằng cách kết hợp các đặc trưng văn bản và đặc trưng hình ảnh từ chụp màn hình trang web. Lý do sử dụng kết hợp đặc trưng văn bản và hình ảnh chụp màn hình là do các đặc trưng này chứa hầu hết các thông tin quan trọng nhất của mỗi trang web, đặc biệt giúp khắc phục các tồn tại của các mô hình nhánh được trình bày tại mục *3.3. Phát hiện thay đổi giao diện sử dụng ảnh chụp màn hình trang web* và *3.4. Phát hiện tấn công thay đổi giao diện sử dụng nội dung văn bản*. Các kết quả phát hiện từ các mô hình nhánh được kết hợp bằng phương pháp tính trung bình có trọng số để tạo ra kết quả cuối cùng. Trong mô hình kết hợp NCS lựa chọn trọng số cho hai mô hình thành phần dựa trên đặc trưng hình ảnh và văn bản là bằng nhau và là 0,5. Lý giải cho việc lựa chọn trọng số bằng nhau là do với mỗi dữ liệu thu thập bao gồm cả dữ liệu deface và dữ liệu sạch, NCS đều trích chọn đặc trưng ảnh và đặc trưng văn bản theo cặp nhất định. Cụ thể, với mỗi trang web bị deface hoặc trang web bình thường, cặp đặc trưng ảnh chụp màn hình và văn bản thuần trích xuất từ mã HTML của trang được thu thập và sử dụng.

Hình 3. 20 biểu diễn mô hình phát hiện tấn công thay đổi giao diện trang web sử dụng kết hợp đặc trưng văn bản và đặc trưng hình ảnh. Mô hình đề xuất phát hiện tấn công thay đổi giao diện trang web sử dụng kết hợp đặc trưng hình ảnh và văn bản bao gồm hai giai đoạn: (a) giai đoạn huấn luyện và (b) giai đoạn phát hiện. Giai đoạn huấn luyện đã được thực hiện tại các mô hình nhánh tại mục *3.3. Phát hiện thay đổi giao diện sử dụng ảnh chụp màn hình trang web* với kết quả thu được là Bộ phân loại 1 và mục *3.4. Phát hiện tấn công thay đổi giao diện sử dụng nội dung văn bản* với

kết quả thu được là Bộ phân loại 2. Cả hai bộ phân loại đều được sử dụng cho giai đoạn phát hiện như mô tả trên Hình 3. 20.



Hình 3. 20. Mô hình phát hiện tấn công thay đổi giao diện kết hợp đặc trưng văn bản và hình ảnh trang web

Trong giai đoạn phát hiện, từ trang web cần giám sát được tách hai đặc trưng là hình ảnh chụp màn hình và dữ liệu văn bản, lần lượt thực hiện với các mô hình nhánh tương ứng, kết quả cuối cùng của mỗi nhánh sau khi được phân loại cùng bộ phân loại tương ứng tại quá trình huấn luyện sẽ được kết hợp bằng phương pháp học kết hợp (tính trung bình có trọng số kết quả từ các mô hình nhánh) từ đó có bộ phân loại cuối cùng giúp phát hiện trạng thái bình thường hoặc bị tấn công thay đổi giao diện của trang web cần giám sát.

Học kết hợp (Ensemble learning) cho phép hợp nhất các giá trị quyết định đã học với các cơ chế sau: lấy trung bình, bỏ phiếu, hoặc một mô hình đã học,… Ưu điểm của phương pháp này là cho phép sử dụng các mô hình khác nhau trên các phương thức khác nhau do đó sẽ linh hoạt hơn. Luận án sử dụng phương pháp lấy trung bình có trọng số của các xác xuất thu được từ các mô dự đoán độc lập tức là lấy trung bình cộng xác suất dự đoán của các mô hình thành phần dựa trên BiLSTM và

ExfficientNet. Dự đoán tổng thể được thực hiện theo công thức $\sum_{i=1}^{2} w_i p_i / \sum_{i=1}^{2} w_i$, trong đó p_i là xác xuất dự đoán của mô hình nhánh thứ i và w_i là trọng số của mô hình nhánh thứ i. Trọng số cho mô hình phân loại ảnh là w_1 và cho mô hình phân loại văn bản là w_2.

Bảng 3. 9. Thuật toán cho mô hình kết hợp

Thuật toán: Phát hiện sử dụng kết hợp

Đầu vào: Trang web cần giám sát
Đầu ra: Trạng thái trang web là bình thường hay bị thay đổi giao diện
Khởi tạo tham số và dữ liệu:

- Đầu vào: A dữ liệu đầu vào là ảnh chụp màn hình trang web giám sát, B dữ liệu đầu vào là văn bản được trích xuất từ trang web giám sát
- M1 mô hình phân loại dựa trên ảnh có trọng số tương ứng là w_1
- M2 mô hình phân loại dựa trên nội dung văn bản với trọng số tương ứng là w_2
- Lựa chọn trọng số (w_1 ; w_2) = (0,5; 0,5) tương ứng với mô hình M1 và M2

1: **Thực hiện phát hiện sử dụng đặc trưng ảnh với mô hình M1:**
2: Trích chọn đặc trưng ảnh, với mô hình EfficientNet(B0) với dữ liệu ảnh A
3: Áp dụng phân loại nhị phân kết hợp với Bộ phân loại 1 từ mô hình M1 có trọng số w_1 và tính xác suất dự đoán $p1$
4: **Thực hiện phát hiện sử dụng đặc trưng văn bản với mô hình M2:**
5: Trích chọn đặc trưng văn bản, sử dụng mô hình BiLSTM với dữ liệu văn bản B
6: Áp dụng phân loại nhị phân kết hợp với Bộ phân loại 2 từ mô hình M2 có trọng số w_2 và tính xác suất dự đoán $p2$
7: **Thực hiện kết hợp kết quả 2 mô hình phát hiện**
8: Thực hiện dự đoán cuối bằng phương pháp kết hợp $\sum_{i=1}^{2} w_i p_i / \sum_{i=1}^{2} w_i$
9: **Phát hiện**
Trạng thái trang web là bình thường hay bị tấn công thay đổi giao diện

3.5.2. Tiền xử lý dữ liệu, huấn luyện và phát hiện

Các tập dữ liệu sau tiền xử lý được huấn luyện để xây dựng các mô hình phát hiện thành phần. Dữ liệu văn bản thuần được huấn luyện sử dụng thuật toán học sâu BiLSTM và dữ liệu ảnh chụp màn hình được huấn luyện sử dụng thuật toán học sâu EfficientNet. Thuật toán BiLSTM là một cải tiến của LSTM phù hợp với bài toán xử lý dữ liệu dạng văn bản và dự đoán được mối liên hệ của các từ xa hơn nên hạn chế được việc bỏ sót thông tin [34] [60]. Hình 3. 17 mô tả cấu trúc của BiLSTM sử dụng trong mô hình đề xuất. Cấu trúc của BiLSTM bao gồm một lớp Embedding, một lớp SpatialDropout, và một lớp Bidirectinal (LSTM), cuối cùng là là lớp Dense sử dụng hàm Softmax để tính xác suất và dự đoán trang web bình thường hoặc bị tấn công.

Lý do lựa chọn thuật toán học sâu EfficientNet(B0) để xây dựng bộ phân loại thành phần sử dụng dữ liệu ảnh chụp màn hình đã được phân tích tại mục *3.3.2. Tiền xử lý dữ liệu và huấn luyện mô hình phát hiện.*

Như đã đề cập trong mục *3.5.1. Mô tả mô hình phát hiện*, quá trình phát hiện bao gồm 2 lớp phát hiện dựa trên 2 mô hình nhánh hoặc thành phần sử dụng dữ liệu văn bản thuần và ảnh chụp màn hình. Việc kết hợp kết quả phát hiện của 2 mô hình thành phần sử dụng phương pháp kết hợp.

3.5.3. Tập dữ liệu thử nghiệm

Dữ liệu trong quá trình thử nghiệm được sử dụng như trong mô tả tại mục *3.2. Thu thập bộ dữ liệu thử nghiệm*. Tập dữ liệu thử nghiệm cũng được chia ngẫu nhiên thành 3 tập con: 60% cho tập huấn luyện (Training Set), 20% tập xác thực (Validation Set) và 20% cho tập kiểm tra (Testing Set). Tỷ lệ giữa số trang web bình thường và số trang web bị tấn công trong tập huấn luyện, tập xác thực và tập kiểm tra tương đương với tỷ lệ trong tập dữ liệu thử nghiệm.

3.5.4. Thử nghiệm và kết quả

3.5.4.1. Các kịch bản thử nghiệm

Để đánh giá toàn diện mô hình kết hợp đề xuất và so sánh với các mô hình đề xuất bởi các nghiên cứu đã có, luận án xây dựng và thực hiện các kịch bản thử nghiệm như sau:

- Thử nghiệm các mô hình phát hiện đề xuất dựa trên kết hợp dữ liệu văn bản trích xuất từ trang web và dữ liệu ảnh chụp màn hình trang web sử dụng phương pháp kết hợp.

- So sánh đánh giá với các mô hình phát hiện đơn (chỉ sử dụng duy nhất một dạng dữ liệu) và so sánh đánh giá với các nghiên cứu liên quan. Trong quá trình thực nghiệm NCS có đánh giá các công trình [38] [44] [106] trên cùng tập dữ liệu với mô hình đề xuất, trên môi trường thử nghiệm Kaggle với cấu hình RAM 13G, GPU P100.

3.5.4.2. Kết quả thử nghiệm

Các kết quả thử nghiệm mô hình kết hợp được cho trên Bảng 3. 10 và Bảng 3. 11. Theo Bảng 3. 11, kết quả của mô hình kết hợp là tốt hơn kết quả cho bởi các nhánh độc lập, cụ thể là phương pháp kết hợp có độ đo ACC và F1 tốt hơn từ mô hình nhánh (EfficientNet(B0)) với đặc trưng ảnh và mô hình nhánh (BiLSTM) với đặc trưng văn bản lần lượt là là 98,12% và 97,65% so với 94,12% và 92,62%, 96,54% và 95,66%. Đồng thời, tỷ lệ âm tính giả và dương tính giả cũng thấp hơn đáng kể.

Cũng theo Bảng 3. 11, kết quả của mô hình kết hợp tốt hơn các mô hình đã có, gồm Naïve Bayes của Hoang [38], Cây quyết định của Hoang [38], Rừng ngẫu nhiên của Hoang [44], SVM của Siyan Wu [106] với độ đo ACC và F1 lần lượt là 98,12% và 97,65% so với 82,54% và 78,69%, 87,33% và 84,40%, 93,88% và 92,26%, 91,28% và 88,94%.

Bảng 3. 10. Ma trận nhầm lẫn mô hình kết hợp sử dụng đặc trưng văn bản và hình ảnh chụp màn hình trang web

Mô hình kết hợp (Văn bản và hình ảnh)		Actual Class	
		Attacked	Normal
Predicted Class	Attacked	7507	89
	Normal	273	11378

Bảng 3. 11. Kết quả thực nghiệm mô hình kết hợp

Kỹ thuật học sâu và kết hợp	**Đặc trưng**	**ACC (%)**	**PPV (%)**	**TPR (%)**	**F1 (%)**	**FPR (%)**	**FNR (%)**	**Time (phát hiện)**
Naïve Bayes Hoang [38]	Văn bản	82,54	78,12	79,26	78,69	15,21	20,74	140,48
Cây quyết định Hoang [38]	Văn bản	87,33	84,4	84,4	84,40	10,67	15,6	132,30
Rừng ngẫu nhiên Hoang [44]	Văn bản	93,88	93,81	90,76	92,26	4,03	9,24	159,29
SVM Siyan Wu [106]	Văn bản	91,28	90,10	87,81	88,94	6,42	12,19	5,91
EfficientNet(B0)	Ảnh	94,12	94,60	90,71	92,62	3,55	9,29	93,55
BiLSTM	Văn bản	96,54	96,93	94,43	95,66	2,03	5,57	133,77
BiLSTM+ EfficientNet (Kết hợp)	**Văn bản và Ảnh**	**98,12**	**98,83**	**96,49**	**97,65**	**0,78**	**3,51**	**212,59**

Trong quá trình thực nghiệm, nghiên cứu sinh đánh giá mô hình phát hiện đề xuất sử dụng EfficientNet kết hợp với BiLSTM với các tham số phù hợp khác nhau tương ứng với tập dữ liệu thu thập được, nhằm đánh giá hiệu suất của mô hình dựa trên số lượng dữ liệu, số lượng lớp ẩn, số lượng đơn vị ẩn trong lớp, số lượng epoch,… Các tham số chính sử dụng trong mô hình học sâu khi xây dựng mô hình phát hiện bao gồm:

- lrfn là một hàm để thiết lập lịch trình học (learning rate schedule). Hàm này xác định learning rate cho mỗi epoch dựa trên các tham số đã cung cấp.

- epoch=50: Số lượng epoch hiện tại.

- lr_start=0.0001: Learning rate bắt đầu.

- lr_max=0.0004: Learning rate lớn nhất.

- lr_min=0.0001: Learning rate nhỏ nhất.

- lr_rampup_epochs=5: Số lượng epoch để tăng learning rate từ lr_start đến lr_max.

- lr_sustain_epochs=5: Số lượng epoch để duy trì lr_max.

- lr_exp_decay=0.8: Tỷ lệ giảm theo cấp số nhân sau giai đoạn sustain.

Biên dịch mô hình với các siêu tham số tham số:

- optimizer='adam': Sử dụng Adam optimizer, một thuật toán tối ưu hóa phổ biến.

- loss='categorical_crossentropy': Hàm mất mát cho bài toán phân loại đa lớp.

- metrics=['accuracy']: Sử dụng độ chính xác (accuracy) như một thước đo hiệu suất.

- epochs=50: Sử dụng số lượng epoch huấn luyện là 50

- loss_weights=None: Áp dụng trọng số cho các hàm mất mát nếu mô hình có nhiều đầu ra. Trọng số này giúp điều chỉnh mức độ quan trọng của từng hàm mất mát trong tổng hợp hàm mất mát chung.

- weighted_metrics=None: Tương tự như metrics, nhưng cho phép áp dụng trọng số cho các chỉ số.

- run_eagerly=False: Quyết định có thực hiện mô hình theo chế độ eager execution hay không. False có nghĩa là sử dụng chế độ đồ thị (graph mode), giúp tăng hiệu suất và tốc độ huấn luyện bằng cách biên dịch mô hình thành đồ thị tĩnh.

- steps_per_execution=1: Xác định số bước (batches) thực hiện trước khi cập nhật trọng số của mô hình. 1 có nghĩa là cập nhật sau mỗi bước. Việc tăng giá trị này có thể cải thiện hiệu suất bằng cách giảm tần suất cập nhật trọng số.

- jit_compile='auto': Điều chỉnh việc biên dịch Just-In-Time (JIT) cho mô hình. Khi đặt là 'auto', Keras sẽ tự động quyết định có nên sử dụng biên dịch JIT để tối ưu hóa hiệu suất mô hình hay không.

- auto_scale_loss=True: Cho phép tự động điều chỉnh hàm mất mát để cải thiện độ ổn định trong quá trình huấn luyện. True có nghĩa là tự động điều chỉnh hàm mất mát nếu cần thiết để đảm bảo việc huấn luyện diễn ra trơn tru.

3.5.5. Nhận xét

Bảng 3. 11 trình bày tổng hợp hiệu suất của các mô hình đã có và các mô hình đề xuất trong luận án. Có thể thấy rằng, mô hình phát hiện kết hợp dựa trên BiLSTM+EfficientNet sử dụng các đặc trưng văn bản và đặc trưng ảnh chụp màn hình trích xuất từ trang web cho kết quả phát hiện vượt trội so với kết quả của các đề xuất đã có trên hầu hết các độ đo. Mô hình phát hiện dựa trên EfficientNet chỉ sử dụng các đặc trưng ảnh chụp màn hình trích xuất từ trang web cho kết quả phát hiện khá tốt, dù tỷ lệ âm tính giả (FNR) còn tương đối cao. Cụ thể, độ đo F1 của các mô hình phát hiện đề xuất bởi [38] (Naive Bayes, Decision Tree) và [44] (Rừng ngẫu nhiên), mô hình phát hiện đề xuất bởi luận án dựa trên EfficientNet và BiLSTM+EfficientNet lần lượt là 78,69%, 84,4%, 92,26%, 92,62% và 97,65%.

Mô hình phát hiện kết hợp cũng giúp giảm đáng kể tỷ lệ cảnh báo sai, gồm cả tỷ lệ FPR và tỷ lệ FNR. Cụ thể, FPR và FNR của mô hình kết hợp dựa trên BiLSTM+EfficientNet và mô hình thành phần dựa trên EfficientNet lần lượt là 0,78% và 3,51%; và 3,55% và 9,29%.

Kết quả thực nghiệm trên mô hình kết hợp tốt hơn các nghiên cứu khác với lý do sau: (i) luận án lựa chọn tập dữ liệu kết hợp các đặc trưng ảnh chụp màn hình và văn bản từ trang web thể hiện đầy đủ những đặc điểm về hình thức và nội dung tấn công thay đổi giao diện; (ii) luận án cũng lựa chọn được thuật toán phù hợp với những đặc trưng (thuật toán EfficientNet, BiLSTM lần lượt với đặc trưng ảnh và văn bản); (iii) luận án sử dụng phương pháp kết hợp lấy trung bình trọng số từ các mô hình thành phần sẽ hỗ trợ nhau trong mô hình phát hiện kết hợp.

Hạn chế của mô hình kết hợp là yêu cầu tài nguyên tính toán cao hơn cho quá trình huấn luyện và phát hiện do mô hình dựa trên các thuật toán học sâu và xử lý hình ảnh.

3.6. Kết luận chương

Chương 3 đã giới thiệu khái quát về phát hiện tấn công thay đổi giao diện và phòng chống, các phương pháp phát hiện tấn công thay đổi giao diện và so sánh đánh giá các giải pháp và một số kỹ thuật phức tạp được sử dụng để phát hiện tấn công thay đổi giao diện trang web.

Phần tiếp theo của chương đề xuất ba mô hình phát hiện tấn công thay đổi giao diện trang web sử dụng đặc trưng ảnh chụp màn hình trang web, đặc trưng văn bản trang web và kết hợp đặc trưng văn bản trang web và ảnh chụp màn hình trang web.

Các mô hình nhánh và mô hình kết hợp đề xuất đều cho các độ đo hiệu suất tốt hơn so với các công bố trước đó.

Nội dung của chương này cũng được công bố tại các công trình:

1. **Trong Hung Nguyen**, Xuan Dau Hoang, Duc Dung Nguyen, “Detecting Website Defacement Attacks using Web-page Text and Image Features”, Article Published in International Journal of Advanced Computer Science and Applications(IJACSA), Volume 12 Issue 7, 2021, Scopus Q3. Truy vấn từ https://thesai.org/Publications/ViewPaper?Volume=12&Issue=7&Code=IJACSA&SerialNo=25.

2. **Nguyễn Trọng Hưng**, Hoàng Xuân Dậu, Nguyễn Đức Dũng, Vũ Xuân Hạnh, “Phát hiện tấn công thay đổi giao diện trang web sử dụng đặc trưng văn bản” Hội nghị khoa học quốc gia về "Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin" FAIR 2024.

3. Xuan Dau Hoang, **Trong Hung Nguyen**, Hoang Duy Pham, “A Novel Model for Detecting Web Defacement Attacks Using Plain Text Features” Indonesian Journal of Electrical Engineering and Computer Science (IJEECS), 2024, Scopus Q3.

KẾT LUẬN

Tấn công ứng dụng web ngày càng trở nên phổ biến và nguy hiểm, gây mất an ninh an toàn thông tin cho cá nhân, cơ quan, tổ chức, với các hình thức liên tục thay đổi đòi hỏi phải có những giải pháp, kỹ thuật mới, cập nhật thường xuyên để có thể phòng chống và ngăn chặn hiệu quả. Các dạng tấn công web thường gặp, như SQLi, XSS, CMDi, duyệt đường dẫn thường để lại lịch sử truy cập tại các tệp web log của máy chủ web, do đó việc khai thác các tệp web log kết hợp với các thuật toán học máy là một hướng đi có triển vọng. Đối với dạng tấn công thay đổi giao diện trang web, việc phân tích nội dung trang web trích xuất từ mã HTML và ảnh chụp màn hình của trang web kết hợp các thuật toán học sâu đem lại hiệu quả tốt. Luận án này tập trung giải quyết hai vấn đề: (1) nghiên cứu, đề xuất mô hình phát hiện tấn công web dựa trên học máy có giám sát sử dụng dữ liệu web log, nhằm tăng tỷ lệ phát hiện đúng và giảm tỷ lệ cảnh báo sai, đồng thời mô hình có khả năng phát hiện 4 kiểu tấn công web nguy hiểm bao gồm SQLi, XSS, CMDi và duyệt đường dẫn; và (2) nghiên cứu, đề xuất các đặc trưng và lựa chọn sử dụng phương pháp học sâu phù hợp với các đặc trưng cụ thể cho xây dựng mô hình phát hiện tấn công thay đổi giao diện trang web, nhằm xây dựng mô hình phát hiện cho phép phát hiện hiệu quả tấn công thay đổi giao diện trang web. Vấn đề (1) được giải quyết bởi đóng góp thứ nhất của luận án, còn vấn đề thứ (2) được giải quyết bởi đóng góp thứ hai của luận án.

Đóng góp thứ nhất của luận án là đề xuất mô hình phát hiện các dạng tấn công web dựa trên học máy sử dụng các đặc trưng ký tự trong dữ liệu truy vấn URI trích xuất từ web log. Các thuật toán học máy có giám sát được sử dụng, gồm Rừng ngẫu nhiên, Cây quyết định, Navie Bayes và SVM. Mô hình đề xuất có khả năng phát hiện hiệu quả bốn dạng tấn công web thường gặp nguy hiểm nhất, bao gồm SQLi, XSS, CMDi và duyệt đường dẫn. Các thử nghiệm trên tập dữ liệu mẫu được dán nhãn và tập dữ liệu web log thực khẳng định mô hình đề xuất dựa trên thuật toán Rừng ngẫu nhiên cho hiệu suất tốt hơn các mô hình phát hiện dựa trên học sâu [58] [76], với độ chính xác phát hiện chung và độ đo F1 lần lượt là 99.68% và 99.57%. Ngoài hiệu suất phát hiện cao, mô hình đề xuất còn có một số ưu điểm so với các đề xuất trước đó: (i) mô hình đề xuất được xây dựng sử dụng các thuật toán học máy có giám sát truyền thống với chi phí tính toán thấp nhưng vẫn đạt được kết quả cao, điều này rất quan trọng khi triển khai thực tế vì hệ thống phát hiện tấn công web thường phải xử lý một lượng web log rất lớn và (ii) mô hình đề xuất có thể được xây dựng tự động từ dữ liệu huấn luyện và không yêu cầu cập nhật thường xuyên.

Đóng góp thứ hai của luận án là đề xuất ba mô hình phát hiện tấn công thay đổi giao diện trang web dựa trên học sâu sử dụng đặc trưng hình ảnh chụp màn hình của trang web, đặc trưng văn bản trích xuất từ trang web và kết hợp các đặc trưng văn bản trích xuất từ trang web kết hợp với các đặc trưng hình ảnh chụp màn hình của trang web. Các thuật toán học sâu được sử dụng là BiLSTM và EfficientNet. Các kết quả thử nghiệm cho thấy mô hình phát hiện nhánh dựa trên EfficientNet, mô hình phát hiện nhánh dựa trên BiLSTM và mô hình kết hợp đều cho hiệu suất phát hiện cao hơn các mô hình đề xuất bởi các nghiên cứu đi trước và mô hình dựa trên các thuật toán học sâu khác. Đặc biệt, mô hình phát hiện dựa trên kết hợp hai đặc trưng hình ảnh và văn bản của trang web có hiệu suất phát hiện vượt trội so với kết quả đề xuất bởi Hoang và cộng sự [44] và Hoang [38], cũng như các mô hình dựa trên các thuật toán học sâu Xception, Inception và BiLSTM và EfficientNet chỉ các đặc trưng hình ảnh chụp màn hình của trang web.

Các vấn đề hay tồn tại của các đề xuất trong luận án cũng chính là các hướng mở cho tiếp tục nghiên cứu, bổ sung. Cụ thể:

- Vấn đề thứ nhất là các truy vấn URI chỉ có thể trích xuất từ web log nếu phương thức HTTP sử dụng là GET. Nếu phương thức sử dụng là POST, dữ liệu gửi từ máy khách đến máy chủ không được lưu trong log. Một hướng giải quyết vấn đề này là triển khai mô hình phát hiện dưới dạng 1 tường lửa ứng dụng web (WAF) để bắt và xử lý tất cả các yêu cầu truy cập từ người dùng.

- Mô hình phát hiện tấn công thay đổi giao diện dựa trên BiLSTM và EfficientNet nhìn chung đòi hỏi chi phí tính toán lớn cho huấn luyện do phải xử lý một lượng lớn ảnh chụp màn hình và nội dung văn bản trang web sử dụng các thuật toán học sâu. Một hướng khắc phục vấn đề này là việc huấn luyện mô hình có thể thực hiện offline nên không ảnh hưởng nhiều đến thời gian phát hiện. Ngoài ra, mô hình có thể kết hợp sử dụng phát hiện dựa trên chữ ký với các dạng tấn công đã biết nhằm giảm đáng kể thời gian phát hiện với những mẫu tấn công đã biết.

DANH MỤC CÁC CÔNG TRÌNH CÔNG BỐ

TẠP CHÍ KHOA HỌC

[CT1] Hoang Xuan Dau, Ninh Thi Thu Trang, **Nguyen Trong Hung**. "A Survey of Tools and Techniques for Web Attack Detection". *Journal of Science and Technology on Information security*, Special Issue CS (15) 2022, pp. 109-118.

[CT2] **Trong Hung Nguyen**, Xuan Dau Hoang, Duc Dung Nguyen, "Detecting Website Defacement Attacks using Web-page Text and Image Features", *Article Published in International Journal of Advanced Computer Science and Applications(IJACSA),* Volume 12 Issue 7, 2021, Scopus Q3.

[CT3] Xuan Dau Hoang, **Trong Hung Nguyen**, Hoang Duy Pham, "A Novel Model for Detecting Web Defacement Attacks Using Plain Text Features" *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, 2024, Scopus Q3.

HỘI THẢO KHOA HỌC

[CT4] Hoàng Xuân Dậu, **Nguyễn Trọng Hưng**, "Phát hiện tấn công web thường gặp dựa trên học máy sử dụng web log", *Hội nghị khoa học quốc gia về "Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin"* FAIR 2020.

[CT5] **Nguyễn Trọng Hưng**, Hoàng Xuân Dậu, Nguyễn Đức Dũng, Vũ Xuân Hạnh, "Phát hiện tấn công thay đổi giao diện trang web sử dụng đặc trưng văn bản", *Hội nghị khoa học quốc gia về "Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin"* FAIR 2024.

TÀI LIỆU THAM KHẢO

[1] Axelsson S, "Research in intrusion-detection systems: a survey," Technical report 98–17,Department of Computer Engineering, Chalmers University of Technology, 1998, 1998.

[2] A05, "Báo cáo tình hình an ninh mạng năm 2020," Hà Nội, 2020.

[3] Abdullayev V., Chauhan A. S, "SQL injection attack: Quick view," *Mesopotamian Journal of CyberSecurity,* 2023.

[4] Alahmad M., Alkandari A., Alawadhi N, "Survey Of Os Command Injection Web Application Vulnerability Attack," *Journal of Engineering Science and Technology,* vol. 17, pp. 75 - 84, 2022.

[5] Alaoui R. L, "Web attacks detection using stacked generalization ensemble for LSTMs and word embedding," in *Procedia Computer Science 215*, 2022.

[6] Albahar M., Alansari D., Jurcut A, "An empirical comparison of pen-testing tools for detecting web app vulnerabilities.," in *Electronics*, 2022.

[7] Albalawi M., Aloufi R., Alamrani N., Albalawi N., Aljaedi A., Alharbi A. R, "Website Defacement Detection and Monitoring Methods: A Review," *Electronics,* vol. https://doi.org/10.3390/electronics11213573, pp. 11, 3573, 2022.

[8] Baranwal A. K, "Approaches to detect SQL injection and XSS in web applications," EECE 571B, Term Survey Paper, British Columbia, Canada, 2012.

[9] Baranwal A. K, "Approaches to detect SQL injection and XSS in web applications," EECE 571B, Term Survey Paper, University of British Columbia, Canada, 2012.

[10] Biau G, "Analysis of a random forests model," *The Journal of Machine Learning Research,* pp. 063-1095., 2012.

[11] Babiker M., Karaarslan E., Hoscan Y, "Web application attack detection and forensics: A survey," in *International Symposium on Digital Forensic and Security (ISDFS)*, Antalya, Turkey, 2018.

[12] Banerjee S., Swearingen T., Shillair R., Bauer J. M., Holt T., Ross A, "Using machine learning to examine cyberattack motivations on web defacement data," *Social Science Computer Review,* vol. 40.4, pp. 914-932, 2022.

[13] Bartoli A., Davanzo G., Medvet E, "A Framework for Large-Scale Detection of Web Site Defacements.," in *ACM Transactions on Internet Technology,*, 2010.

[14] Betarte G., Giménez E., Martínez R., Pardo Á, "Machine learning-assisted virtual patching of web applications," in *https://arxiv.org/abs/1803.05529*, Mar 2018..

[15] Betarte G., Pardo Á., Martínez R. “Web application attacks detection using machine learning techniques” In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, (2018) 1065–1072.

[16] Bisht P., Madhusudan P., Venkatakrishnan V. N, "CANDID: Dynamic Candidate Evaluations for Automatic Prevention of SQL Injection Attacks," *ACM Transactions on Information and System Security,* vol. 13, 2010.

[17] Buehrer G., Weide B. W., Sivilotti P. A. G, "Using Parse Tree Validation to Prevent SQL Injection Attack," *Proceedings of the 5th international workshop on Software engineering and middleware,* p. 106–113., 2005.

[18] Ceraj T., Kliman I., Kutnjak M, "Redefining cancer treatment: comparison of Word2vec embeddings using deep BiLSTM classification model," in *ext Analysis and Retrieval 2019 Course Project Reports*, 2019.

[19] Chandola V., Banerjee A., Kumar V, "Anomaly detection for discrete sequences: A survey," *IEEE transactions on knowledge and data engineering,* vol. 24, no. 5, pp. 823-839, 2012.

[20] Chandola V., Banerjee A., Kumar V, "Anomaly detection: A survey. . Jul 30;41(3):1-58.," *ACM computing surveys (CSUR),* vol. 41, p. 71–97, 2009 .

[21] Cheah C. S., Selvarajah V, "A review of common web application breaching techniques (SQLi, XSS, CSRF)," in *International Conference on Integrated Intelligent Computing Communication & Security*, 2021.

[22] Cova M., Balzarotti D., Felmetsger V., Vigna G, "Swaddler: An Approach for the Anomaly-based Detection of State Violations in Web Applications," *International Symposium on Recent Advances in Intrusion Detection,* vol. 4637, pp. 63-86, 2007.

[23] Davanzo G., Medvet E., Bartoli A, "A Comparative Study of Anomaly Detection Techniques in Web Site Defacement Detection," in *International Information Security Conference*, 2008.

[24] Davanzo G., Medvet E., Bartoli A, "Anomaly detection techniques for a web defacement monitoring service," in *Journal of Expert Systems with Applications*, 2011.

[25] Díaz-Verdejo J., Muñoz-Calle J., Estepa Alonso A., Estepa Alonso R., Madinabeitia G, "On the detection capabilities of signature-based intrusion detection systems in the context of web attacks.," in *Applied Sciences 12.2 (2022)*, 2022.

[26] Díaz-Verdejo J, Muñoz-Calle J, Estepa Alonso A, Estepa Alonso R, Madinabeitia G, "On the Detection Capabilities of Signature-Based Intrusion," in *Applied Sciences*, 2022, 2022.

[27] Sisodia D. S., Verma S, "Web Usage Pattern Analysis Through Web Logs: A Review," in *International Joint Conference on Computer Science and Software Engineering*, Bangkok, Thailand., 2012.

[28] Do Xuan C., Thanh H., Lam N. T, "Optimization of network traffic anomaly detection using machine learning," *International Journal of Electrical and Computer Engineering (IJECE),* vol. 11, p. 2360~2370, June 2021.

[29] Enaw E. E., Prosper D. P, "A conceptual approach to detect webdefacement through artificial intelligence," *International Journal of Advanced Computer Technology (IJACT),* vol. 3, pp. 77-83, 2014.

[30] Fang W., Chen Y., Xue Q, "Survey on research of RNN-based spatio-temporal sequence prediction algorithms," *Journal on Big Data,* vol. 3, p. 97, 2021.

[31] Fiore U., Palmieri F., Castiglione A., De Santis A, "Network anomaly detection with the restricted Boltzmann machine," *Neurocomputing,* vol. 122, p. 13–23, 2013.

[32] Garcia-Teodoro P., Diaz-Verdejo J., Maciá-Fernández G., Vázquez E, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & Security,* vol. 28, pp. 18-28, 2009.

[33] Gottschalk K., Graham S., Kreger H., Snell J, "Introduction to Web services architecture," *IBM Systems Journal,* vol. 41, pp. 170-177, 2002.

[34] Guillod T., Papamanolis P., Kolar J. W, "Artificial Neural Network (ANN) Based Fast and Accurate Inductor Modeling and Design," *IEEE Open Journal of Power Electronics,* vol. 1, pp. 284-299, 2020.

[35] Gupta S., Gupta B. B, "Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art," *International Journal of System Assurance Engineering and Management,* vol. 8, p. 512–530, 2017.

[36] Hoffman A, Web application security, O'Reilly Media, Inc, 2024.

[37] Hoàng X. D, "An toàn ứng dụng web và cơ sở dữ liệu," Học viện Công nghệ Bưu Chính Viễn thông, 2021.

[38] Hoang X. D, "A Website Defacement Detection Method Based on Machine Learning Techniques.," in *International Symposium on Information and Communication Technology*, Da Nang, 2018.

[39] Halfond W. G. J., Orso A, "AMNESIA: analysis and monitoring for neutralizing SQL-injection attacks," *IEEE and ACM International Conference on Automated Software Engineering,* pp. 174-183, 2005.

[40] Hao S., Long J., Yang Y, "BL-IDS: Detecting Web Attacks Using Bi-LSTM Model Based on Deep Learning," in *Security and Privacy in New Computing Environments*, 2019.

[41] Helmiawan M. A., Firmansyah E., Fadil I., Sofivan Y., Mahardika F., Guntara A, "Analysis of web security using open web application security project 10," in *2020 8th International Conference on Cyber and IT Service Management (CITSM). IEEE*, 2020.

[42] Hoang D. X., Nguyen H. T, "A CNN-based model for detecting website defacements," *Journal of Science and Technology on Information and Communications,* vol. 1, pp. 4-9, 2021.

[43] Hoang X. D., Nguyen N. T, "A Multi-layer Model for Website Defacement Detection," in *Tenth International Symposium on Information and Communication Technology*, Ha Long, 2019.

[44] Hoang X. D., Nguyen N. T, "Detecting Website Defacements Based on Machine Learning Techniques and Attack Signatures," *Computers,* 2019.

[45] Holt T. J., Stonhouse M., Freilich J., Chermak S. M, "Examining Ideologically Motivated Cyberattacks Performed by Far-Left Groups," *Terrorism and Political Violence,* 2021.

[46] Jang B., Kim M., Harerimana G., Kang S., Kim J. W, "Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism," *Applied Sciences,* vol. 5841, p. 10, 2020.

[47] Jupin J. A., Sutikno T., Ismail M. A., Mohamad M. S., Kasim S., Stiawan D, "Review of the machine learning methods in the classification of phishing attack," *Bulletin of Electrical Engineering and Informatics,* vol. 8, pp. 1545-1555, 2019.

[48] Liu M, Zhang B, Chen W, Zhang X, "A Survey of Exploitation and Detection Methods of XSS Vulnerabilities," *IEEE access,* 7, pp.182004-182016, 2019.

[49] Kokkonen T, Anomaly-based online intrusion detection system as a sensor for cyber security situational awareness system, PH.D, University of Jyväskylä, 2016.

[50] Kaur J., Garg U., Bathla G, "Detection of cross-site scripting (XSS) attacks using machine learning techniques: a review.," *Artificial Intelligence Review,* pp. 12725-12769, 2023.

[51] Kemalis K., Tzouramanis T, "SQL-IDS: A Specification-based Approach for SQLinjection Detection," in *Proceedings of the 2008 ACM symposium on Applied computing*, 2008.

[52] Khan S., Saxena A, "Detecting Input Validation Attacks in Web Application," *International Journal of Computer Applications,* vol. 109, 2015.

[53] Khan S., Saxena A, "Detecting Input Validation Attacks in Web Application," *International Journal of Computer Applications,* vol. 109, 2015.

[54] Kim W., Lee J., Park E., Kim S, "Advanced Mechanism for Reducing False Alarm Rate in Web Page Defacement Detection," in *National Security Research Institute*, Korea, 2006.

[55] Krishnan S., Zolkipli M. F, "Survey on SQL Injection and Cross-Site Scripting Malware Injection Attacks," *International Journal of Advances in Engineering and Management,* vol. 5, pp. 822-833, 2023.

[56] Deng L., Yu D, "Deep Learning: Methods and Applications," in *Foundations and Trends in Signal Processing*, 2014.

[57] Langhnoja S., Barot M., Mehta D, "Pre-processing: procedure on web log file for web usage mining," *International Journal of Emerging Technology and Advanced Engineering,* vol. 12, no. 2, pp. 419-422, 2012.

[58] Liang J., Zhao W., Ye W, "Anomaly-Based Web Attack Detection: A Deep Learning Approach," in *International Conference on Network, Communication and Computing*, Kunming, China, 2017.

[59] Liciotti D., Bernardini M., Romeo L., Frontoni E, "A sequential deep learning application for recognising human activities in smart homes," *Neurocomputing,* vol. vol. 396, pp. pp.501-513, 2020.

[60] Liu G., Guo J, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," *Neurocomputing,* vol. 337, pp. 325-338, 2019.

[61] Lộc P. D., Dậu H. X, "Khảo sát các nền tảng và kỹ thuật xử lý log truy cập dịch vụ mạng cho phát hiện nguy cơ mất an ninh an toàn thông tin," in *Dalat University Journal of Science (2018)*, 2018.

[62] Lu D., Fei J., Liu L, "A semantic learning-based SQL injection attack detection technology," *Electronics,* p. 1344, 2023.

[63] Luan Y., Lin S, "Research on text classification based on CNN and LSTM," in *international conference on artificial intelligence and computer applications* , 2019.

[64] Mahesh B, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR),* pp. 381-386, 2020.

[65] Mitchell T. M., Mitchell T. M, Machine Learning, McGraw-Hill Science, 1997.

[66] Mao B.-M., Bagolibe K. D, "A Contribution to Detect and Prevent a Website Defacement," in *International Conference on Cyberworlds (CW)*, Kyoto, Japan, 2019.

[67] Melamud O., Goldberger J., Dagan I, "context2vec: Learning Generic Context Embedding with Bidirectional LSTM," in *Conference on Computational Natural Language Learning*, 2016.

[68] Mitropoulos D., Louridas P., Polychronakis M., Keromytis A. D, "Defending Against Web Application Attacks: Approaches, Challenges and Implications," *IEEE Transactions on Dependable and Secure Computing,* vol. 16, pp. 188-203, 5 2017.

[69] Moh M., Pininti S., Doddapaneni S., Moh T.-S, "Detecting Web Attacks Using Multi-stage Log Analysis," in *International Conference on Advanced Computing*, 2016.

[70] Mohammed M., Khan M. B., Bashier E. B. M, "Machine Learning - Algorithms and Applications," Taylor & Francis, 2017.

[71] Sangani N. K., Zarger H, "Machine Learning in Application Security," Book chapter in Advances in Security in Computing and Communications, 2017.

[72] Nasereddin M., ALKhamaiseh A., Qasaimeh M., Al-Qassas R, "A systematic review of detection and prevention techniques of SQL injection attacks," *Information Security Journal: A Global Perspective,* pp. 252-265, 2023.

[73] Nassif A. B., Talib M. A., Nasir Q., Dakalbab F. M, "Machine learning for anomaly detection: A systematic review," *Ieee Access,* vol. 9, pp. 78658-78700, 2021 .

[74] OWASP, "The Ten Most Critical Web Application Security Risk - OWASP Top 10 - 2013," [Online]. Available: https://owasp.org/www-pdf-archive/OWASP_Top_10_-_2013.pdf. [Accessed 5 2023].

[75] Bisht P., Venkatakrishnan V. N, "XSSGUARD: Precise dynamic prevention of Cross-Site," in *Conference on Detection of Intrusions and Malware*, 2008.

[76] Pan Y., Sun F., Teng Z., et al, "Detecting web attacks with end-to-end deep learning,” Journal of Internet Services and Applications," *Journal of Internet Services and Applications,* vol. 10, pp. 1--22, 2019..

[77] Patel V, "Real-Time Threat Detection with JavaScript: Monitoring and Response Mechanisms.," in *International Journal of Computer Trends and Technology* , 2023.

[78] Paulikas G., Sandonavičius D., Stasiukaitis E., Vilutis G., Vaitkunas M, "Survey of Cloud Traffic Anomaly Detection Algorithms.," in *International Conference on Information and Software Technologies*, 2022.

[79] Priyanka A. K., Smruthi S. S, "WebApplication Vulnerabilities: Exploitation and Prevention," in *International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2020.

[80] Kozik R., Choraś M., Renk R., Hołubowicz W, "Patterns Extraction Method for Anomaly Detection in HTTP Traffic," in *International Joint Conference*, 2015.

[81] Rafaiani G., Battaglioni M., Compagnoni S., Senigagliesi L., Chiaraluce F., Baldi M, "A Machine Learning-based Method for Cyber Risk Assessment," in *International Symposium on Computer-Based Medical Systems*, 2023.

[82] Romagna M., van den Hout N. J, "Hacktivism and website defacement: Motivations, capabilities and potential threats," in *Proceedings of the 27th Virus Bulletin International Conference*, 2017.

[83] Romagna M, van den Hout NJ, "Hacktivism and website defacement: Motivations, capabilities and potential threats," in *In Proceedings of the 27th Virus Bulletin International Conference*, Madrid, Spain, 2017.

[84] Shalev-Shwartz S., Ben-David S, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2018.

[85] Saleem S., Sheeraz M., Hanif M., Farooq U, "Web Server Attack Detection using Machine Learning," in *International Conference on Cyber Warfare and Security*, 2020.

[86] Salama R., Al-Turjman F., Bhatla S., Yadav S. P, "Social engineering attack types and prevention techniques-A survey," in *International Conference on Computational Intelligence, Communication Technology and Networking (CICTN)*, 2023.

[87] Sangani N. K., Zarger H, "Machine Learning in Application Security," Advances in Security in Computing and Communications, 2017.

[88] Saxena A., Arora A., Saxena S., Kumar A, "Detection of web attacks using machine learning based URL classification techniques," in *International Conference on Intelligent Technologies (CONIT). IEEE*, 2022.

[89] Sentamilselvan K., Pandian S. L., Sathiyamurthy D. K, "Survey on Cross Site Request Forgery," in *International Conference on Research and Development Prospects on Engineering and Technology*, 2013.

[90] Seyyar M. B., Çatak F. Ö., Gül E, "Detection of attack-targeted scans from the Apache HTTP Server access logs," *Applied computing and informatics,* vol. 14, no. 1, pp. 28-36, 2018.

[91] Sharma S., Zavarsky P., Butakov S, "Machine learning based intrusion detection system for web-based attacks," in *Conference on Big Data Security on Cloud (BigDataSecurity)*, 2020.

[92] Shiri F. M., Perumal T., Mustapha N., Mohamed R, "A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU," arXiv:2305.17473, 2023.

[93] Shiri F. M., Perumal T., Mustapha N., Mohamed R., Ahmadon M. A. B., Yamaguchi S, "A Survey on Multi-Resident Activity Recognition in Smart Environments," 2023.

[94] Siami-Namini S., Tavakoli N., Namin A. S, "The Performance of LSTM and BiLSTM in Forecasting Time Series," in *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, 2019.

[95] Singh P., Manure A, "Natural Language Processing with TensorFlow 2.0," in *Learn TensorFlow 2.0*, Apress, Berkeley, CA, 2020, p. 107–129.

[96] Su Z., Wassermann G, "The essence of command injection attacks in web applications," *ACM SIGPLAN NOTICES,* vol. Volume 41, pp. 372-382, 2006.

[97] Suneetha K. R., Krishnamoorthi R, "Identifying user behavior by analyzing web server access log file," *IJCSNS International Journal of Computer Science and Network Security,* vol. 4, no. 9, pp. 327-332, 2009.

[98] Tan M., Le Q, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *InInternational conference on machine learning,* pp. 6105-6114, 2019.

[99] Tian Z., Luo C., Qiu J., Du X., Guizani M, "A distributed deep learning system for web attack detection on edge devices," *IEEE Transactions on Industrial Informatics,* vol. 16, pp. 1963-1971, 2019.

[100] Torrano-Giménez C., Perez-Villegas A., Alvarez Maranón G, "An anomaly-based approach for intrusion detection in web traffic," *The Allen Institute for Artificial Intelligence,* 2009.

[101] Tiệp V. H, Machine Learning cơ bản, Hà Nội: Nhà xuất bản khoa học và kỹ thuật, 2018.

[102] Vogels W, "Web services are not distributed objects," *IEEE Internet Computing,* vol. 7, pp. 59-66, 2003.

[103] Verginadis Y, "A Review of Monitoring Probes for Cloud Computing Continuum.," in *International Conference on Advanced Information Networking and Applications*, 2023.

[104] Waleed A., Jamali A. F., Masood A, "Which open-source ids? snort, suricata or zeek.," in *Computer Networks 213*, 2022.

[105] Wu H. C., Luk R. W. P., Wong K. F., Kwok K. L, "Interpreting TF-IDF term weights as making relevance decisions," in *ACM Transactions on Information Systems vol. 26, no.3*, 2008..

[106] Wu S., Tong X., Wang W., Xin G., Wang B., Zhou Q, "Website defacements detection based on support vector machine classification method," in *In Proceedings of the 2018 International Conference on Computing and Data Engineering*, Shanghai, China, 2018.

[107] Xiao L., Wang G., Zuo Y., "Research on patent text classification based on word2vec and LSTM," in *International Symposium on Computational Intelligence and Design*, 2018.

[108] Kumari Y, "Survey on Web Application Vulnerabilities," *International Research Journal of Engineering and Technology (IRJET),* vol. 06, pp. 922-925, 2019.

[109] ari I. A., Abdullahi B., Adeshina S. A, "Towards a framework of configuring and evaluating modsecurity WAF on tomcat and apache web servers," in *International Conference on Electronics, Computer and Computation (ICECCO)*, 2019.

[110] Yu Y., Si X., Hu C., Zhang J, "A review of recurrent neural networks: LSTM cells and network architectures.," *Neural computation,* pp. 1235-1270., 2019.

[111] Zhang M., Xu B., Bai S., Lu S., Lin Z, "A Deep Learning Method to Detect Web Attacks Using a Specially Designed CNN," in *International Conference on Neural Information Processing*, 2017.

[112] Zhu K. Q., Fisher K., Walker D, "Incremental learning of system log formats," *ACM SIGOPS Operating Systems Review,* vol. 44, no. 1, p. 85–90, 2010.