HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Vũ Thị Quý

NGHIÊN CỨU PHÁT HIỆN TẤN CÔNG WEB CƠ BẢN DỰA TRÊN HỌC MÁY SỬ DỤNG WEB LOG

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI - 2020

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Vũ Thị Quý

NGHIÊN CỨU PHÁT HIỆN TẤN CÔNG WEB CƠ BẢN DỰA TRÊN HỌC MÁY SỬ DỤNG WEB LOG

CHUYÊN NGÀNH: KHOA HỌC MÁY TÍNH

MÃ SỐ: 8.48.01.01

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC

TS. HOÀNG XUÂN DẬU

HÀ NỘI - 2020

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi. Nội dung của luận văn có tham khảo và sử dụng các tài liệu, thông tin được đăng tải trên những tạp chí và các trang web theo danh mục tài liệu tham khảo. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Tôi xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

Hà nội, ngày tháng năm 2020

Người cam đoan

Vũ Thị Quý

LỜI CẢM ƠN

Đầu tiên em xin gửi lời biết ơn sâu sắc nhất tới Thầy giáo, Tiến sĩ Hoàng Xuân Dậu, người Thầy đã tận tình chỉ bảo, dành nhiều thời gian trong việc hướng dẫn học viên cách đọc tài liệu, thu thập và đánh giá thông tin đồng thời giúp em tiếp cận được nhiều phương pháp tư duy và nghiên cứu khoa học mới để hoàn thành một luận văn cao học.

Em xin gửi lời cảm ơn chân thành tới tất cả các thầy, cô giáo khoa Sau đại học cùng các thầy cô giáo trường – Học viện Công nghệ Bưu chính Viễn thông đã luôn nhiệt tình giúp đỡ và tạo điều kiện tốt nhất cho em trong suốt quá trình học tập và nghiên cứu tại trường.

Xin chân thành cảm ơn các anh, các chị và các bạn học viên lớp Cao học – trong Học viện đã luôn động viên, giúp đỡ và nhiệt tình chia sẻ với em những kinh nghiệm học tập, công tác trong suốt khoá học.

Học viên cũng xin chân thành cảm ơn các vị lãnh đạo và các bạn đồng nghiệp tại cơ quan đã luôn tạo mọi điều kiện tốt nhất để em có thể hoàn thành tốt đẹp khoá học Cao học này.

Mặc dù đã rất cố gắng hoàn thành luận văn này, xong luận văn sẽ khó tránh khỏi những thiếu sót. Em rất mong nhận được sự nhận xét, góp ý, tận tình chỉ bảo từ các thầy, cô.

Em xin chân thành cảm ơn!

Hà Nội, tháng năm 2020

Vũ Thị Quý

MỤC LỤC

Trang

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

TỪ VIẾT TẮT	TIẾNG ANH	TIẾNG VIỆT/GIẢI THÍCH
AES	Advanced Encryption Standard	Tiêu chuẩn mã hóa nâng cao
API	Application Programming Interface	Giao diện lập trình ứng dụng
CGI	Common Gateway Interface	Giao diện cổng chung giữa Server và chương trình
CLF	Common log format	Định dạng nhật ký chung
CMDi	Command Injection	Lỗ hổng chèn mã dòng lệnh trên web
CSDL		Cơ sở dữ liệu
CSRF	Cross Site Request Forgery	Một dạng tấn công web
DHTML	Dynamic HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản động
DR	Detection Rate	Tỉ lệ phát hiện chính xác
HTTP	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản
HTTPS	HyperText Transfer Protocol Secure	Giao thức bảo mật HTTP an toàn
IDS	Intrucsion Detection System	Hệ thống phát hiện xâm phạm
IIS	Internet Information Services	Các dịch vụ dành cho máy chủ

IPS	Intrusion prevention System	Hệ thống ngăn chặn xâm phạm
LDAP	Lightweight Directory Access Protocol	Giao thức ứng dụng truy cập các cấu trúc thư mục.
OS	Operating system	Phần mềm hệ thống
OWASP	Open Web Application Security Project	Dự án mở về bảo mật ứng dụng web
PCA	Principal Component Analysis	Phân tích thành phần chính
RSA	Rivest Shamir Adleman	Hệ thống mật mã khóa công khai
SQLi	SQL Injection	Lỗ hổng chèn mã sql trên web
SSL	Secure Sockets Layer	Giao thức bảo mật SSL
TCP/IP	Transmission Control Protocol/ Internet Protocol	Giao thức điều khiển truyền nhận/ Giao thức liên mạng
TF/IDF	Term frequency/ inverse document frequency	Tần suất từ /tần suất tài liệu nghịch
TLS	Transport Layer Security	Bảo mật tầng truyền tải
URI	Uniform Resource Identifier	Nhận diện địa chỉ web
URL	Uniform Resource Locator	Địa chỉ web
WAF	Web Application Firewall	Tường lửa ứng dụng web
XSS	Cross-Site Scripting	Là một loại lỗ hổng web

DANH SÁCH HÌNH VẼ

DANH SÁCH BẢNG

PHẦN MỞ ĐẦU

1. Lý do chọn đề tài

Trong thế giới hiện đại ngày nay, ứng dụng web ngày một trở nên quan trọng và là một phần không thể thiếu trên mạng Internet. Các ứng dụng web, website chiếm tỷ lệ áp đảo trong số các ứng dụng trên nền Internet. Cũng chính vì vậy mà vấn đề về bảo mật web ngày càng trở thành một vấn đề được quan tâm.

Theo số liệu thống kê của BKAV [11], năm 2019, thiệt hại do virus máy tính gây ra đối với người dùng Việt Nam đã lên tới 20.892 tỷ đồng (902 triệu USD), vượt xa con số 14.900 tỷ đồng của năm 2018. Tổng số lượt máy tính bị nhiễm mã độc được ghi nhận trong năm 2019 lên tới 85,2 triệu lượt, tăng 3,5% so với năm 2018. Năm này cũng tiếp tục chứng kiến sự hoành hành của các loại mã độc mã hóa dữ liệu tống tiền (ransomware). Số lượng máy tính bị mất dữ liệu trong năm 2019 lên tới 1,8 triệu lượt, tăng 12% so với năm 2018. Nghiêm trọng hơn, trong số này có rất nhiều máy chủ (server) chứa dữ liệu của các cơ quan. Không chỉ gây thiệt hại lớn, việc các máy chủ bị xóa dữ liệu cũng gây đình trệ hoạt động của cơ quan, doanh nghiệp trong nhiều ngày sau đó, thậm chí đến cả tháng.

Đối với các công ty lớn, nguy cơ bị tấn công vào hệ thống đồng nghĩa với việc họ sẽ bị thiệt hại hàng tỷ USD, uy tín trước khách hàng bị giảm sút. Với các cơ quan y tế và quốc phòng thì thiệt hại còn có thể thảm khốc hơn gấp nhiều lần.

Qua số liệu trên cho thấy tấn công web cơ bản là các dạng tấn công thường gặp lên các website, web portal và các ứng dụng trên nền web. Các dạng tấn công này có thể bao gồm: tấn công chèn mã SQL (SQLi hay SQL Injection), tấn công chèn mã XSS (Cross-Site Scripting), tấn công duyệt đường dẫn (Path traversal) và tấn công chèn dòng lệnh hệ điều hành (CMDi hay Command Injection). Trong số này, tấn công chèn mã SQL là một trong các dạng tấn công phổ biến và nguy hiểm nhất. 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 [1]. Tấn công XSS có thể cho phép tin tặc đánh cắp dữ liệu người dùng lưu trong cookie của trình duyệt, từ đó kiểm soát tài khoản của người dùng trên máy chủ. Theo một hướng khác, tấn công duyệt đường dẫn cho phép tin tặc tải hoặc truy nhập vào các file chứa dữ liệu nhạy cảm trên các máy chủ và thông qua đó có thể xâm nhập sâu vào hệ thống. Tấn công chèn dòng lệnh hệ điều hành có thể cho phép tin tặc thực hiện các lệnh nguy hiểm cho phép xóa file, dữ liệu trên hệ thống nạn nhân.

Mặc dù các dạng tấn công thường gặp lên các website và các ứng dụng trên nền web đã được biết đến từ lâu và đã có nhiều biện pháp phòng chống được nghiên cứu, triển khai, như sử dụng các bộ lọc, tường lửa, các cơ chế kiểm soát truy nhập… Tuy nhiên, các dạng tấn công web cơ bản vẫn khá phổ biến và gây nhiều thiệt hại cho các trang web, các cổng thông tin điện tử, các trang thương mại điện tử của các cơ quan tổ chức. Nguyên nhân của điều này là do vẫn có nhiều website và các ứng dụng trên nền web không có, hoặc thiếu cơ chế lọc dữ liệu đầu vào thực sự hiệu quả, và/hoặc sử dụng các mã chương trình trộn lẫn với dữ liệu, tạo điều kiện cho tin tặc chèn mã độc tấn công hệ thống [1]. Việc xây dựng các bộ lọc dựa trên các mẫu cố định thực sự gặp khó khăn, khi các mẫu tấn công liên tục thay đổi và ngày càng tinh vi hơn. Việc xây dựng các bộ lọc phát hiện các dạng tấn công web cơ bản dựa trên học máy là một hướng giải quyết hiệu quả thay thế cho các bộ lọc mẫu truyền thống. Theo hướng nghiên cứu này, đề tài luận văn thạc sĩ của học viên có tên **“Nghiên cứu phát hiện tấn công web cơ bản dựa trên học máy sử dụng web log”** tập trung nghiên cứu vấn đề phát hiện tấn công web cơ bản dựa trên học máy sử dụng web log.

Do còn nhiều hạn chế về thời gian và tài liệu nên đề tài còn nhiều thiếu sót. Rất mong nhận được sự đóng góp của các thầy cô và các bạn để đề tài được hoàn thiện hơn.

Tôi xin chân thành cảm ơn!

2. Tổng quan về vấn đề nghiên cứu

Đã có nhiều giải pháp phòng chống các dạng tấn công web cơ bản được nghiên cứu và ứng dụng [1][3]. Các giải pháp thực tế có thể kể đến gồm:

- Sử dụng các bộ lọc để kiểm tra và lọc dữ liệu đầu vào. Các bộ lọc có thể sử dụng bao gồm, lọc dựa trên từ khóa, lọc dựa trên mẫu và lọc dựa trên biểu thức chính quy.
- Sử dụng các dạng tường lửa, hoặc proxy ở mức ứng dụng, như tường lửa ứng dụng web (WAF – Web Application Firewall). WAF được sử dụng để lọc tất cả truy vấn của người dùng. WAF có ưu điểm là có thể bảo vệ đồng thời nhiều website và không đòi hỏi chỉnh sửa mã nguồn của website.
- Kết hợp sử dụng các biện pháp kiểm soát truy nhập, phân quyền người dùng để giảm thiểu khả năng bị tấn công, khai thác.
- Sử dụng các công cụ theo dõi, giám sát website, ứng dụng web, như các bộ phát hiện xâm nhập (IDS).

Trên phương diện nghiên cứu học thuật, có thể chia các đề xuất nghiên cứu phát hiện tấn công, xâm nhập nói chung và tấn công web cơ bản nói riêng thành 2 nhóm dựa trên kỹ thuật phát hiện: (1) nhóm phát hiện dựa trên chữ ký, mẫu, hoặc luật và (2) nhóm phát hiện dựa trên bất thường.

Phát hiện dựa trên chữ ký (signature), mẫu (pattern), hoặc luật (rule) là phương pháp phát hiện tấn công dựa trên việc tìm hay so khớp tập chữ ký của các tấn công đã biết với các dữ liệu giám sát thu thập được. Một tấn công được phát hiện khi có ít nhất một so khớp chữ ký thành công. 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 mới, hay tấn công khai thác lỗ hổng zero-day do

chữ ký của chúng chưa tồn tại trong 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.

Phát hiện tấn công, xâm nhập dựa trên bất thường dựa trên giả thiết: *các hành vi tấn công, xâm nhập thường có quan hệ chặt chẽ 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 [3]. 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 này, 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. 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.

Phương pháp phát hiện tấn công web cơ bản dựa trên học máy sử dụng web log thực hiện trong luận văn thuộc nhóm kỹ thuật phát hiện dựa trên bất thường. Theo đó, các URI truy nhập được tách ra từ web log và được phân loại bởi một bộ phân loại đã được huấn luyện sử dụng tập dữ liệu đã được gán nhãn. Luận văn dự kiến sử dụng các thuật toán học máy có giám sát nên có thể giảm thời gian huấn luyện và phát hiện.

3. Mục đích nghiên cứu

Luận văn nghiên cứu một số thuật toán học máy có giám sát và ứng dụng cho việc phát hiện tấn công web cơ bản sử dụng web log.

Trên cơ sở đó tiến hành thực nghiệm để đánh giá hiệu quả trong việc phát hiện tấn công web cơ bản của một số thuật toán học máy.

4. Đối tượng và phạm vi nghiên cứu

Đối tượng và phạm vi nghiên cứu của luận văn bao gồm:

- Một số dạng tấn công web cơ bản

- Một số thuật toán học máy để phát hiện tấn công

- Web log

- Một số công cụ, phần mềm để thử nghiệm và đánh giá hiệu quả các thuật toán học máy trong phát hiện dựa trên Web log.

5. Phương pháp nghiên cứu

- Phương pháp lý thuyết: Khảo sát, phân tích các tài liệu khoa học liên quan đến các dạng tấn công và một số thuật toán học máy.

- Phương pháp thực nghiệm: Sử dụng các công cụ, phần mềm để thử nghiệm và đánh giá hiệu quả các thuật toán học máy trong phát hiện dựa trên web log đối với bộ dữ liệu được lựa chọn.

Trên cơ sở đó đánh giá được các ưu, nhược điểm và từ đó định hướng xây dựng định hướng nghiên cứu, cải tiến mở rộng quy trình, phương pháp.

CHƯƠNG 1: TỔNG QUAN VỀ CÁC DẠNG TẤN CÔNG VÀO WEBSITE, ỨNG DỤNG WEB VÀ CÁC GIẢI PHÁP PHÒNG CHỐNG

Chương này trình bày về kiến trúc ứng dụng web cũng như mô hình, các thành phần, cách thức hoạt động, cách thức liên kết giữa các thành phần đặc trưng thông thường trong một ứng dụng web và các yêu cầu về bảo mật. Ngoài ra, chương còn đề cập đến các hình thức tấn công vào ứng dụng web cũng như cách phòng chống bị tấn công của các hình thức tấn công phổ biến trong các năm gần đây dựa theo OWASP. Phần cuối của chương trình bày các biện pháp bảo mật ứng dụng web, bao gồm nguyên tắc chung và một số biện pháp bảo mật cụ thể cho ứng dụng web.

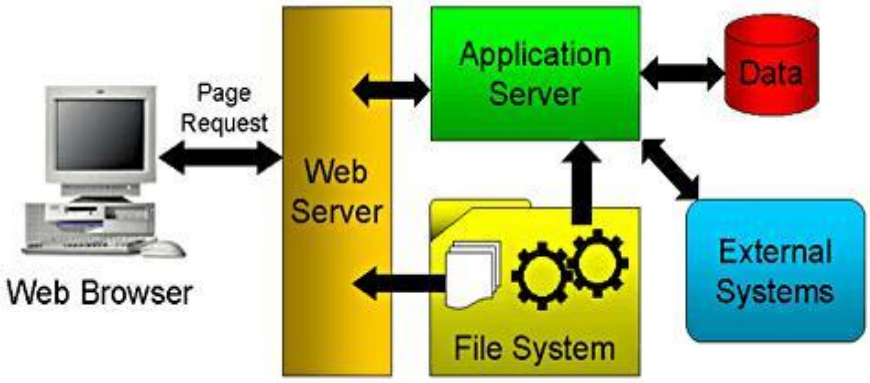
1.1. Kiến Trúc Ứng Dụng Web và Các Yêu Cầu Bảo Mật

1.1.1 Kiến trúc ứng dụng web

Dưới góc độ chức năng, ứng dụng Web là các chương trình máy tính cho phép người dùng website đăng nhập, truy vấn vào ra dữ liệu qua mạng Internet trên trình duyệt Web yêu thích của họ, mang tính kỹ thuật nhiều hơn có thể giải thích các ứng dụng Web truy vấn máy chủ chứa nội dung và tạo tài liệu Web động để phục vụ yêu cầu của máy khách. Giao diện web đặt ra rất ít giới hạn khả năng người dùng. Thông qua JavaScript, DHTML, Flash và những công nghệ khác, những phương pháp chỉ ứng dụng mới có như vẽ trên màn hình, chơi nhạc, và dùng được bàn phím và chuột tất cả đều có thể thực hiện được. Ứng dụng web phổ biến nhờ vào sự có mặt vào bất cứ nơi đâu của trình duyệt web và kết nối Internet. Khả năng cập nhật và bảo trì ứng dụng Web mà không phải phân phối và cài đặt phần mềm trên hàng ngàn máy tính của người dùng cũng là một lý do cho sự phổ biến của nó.

Một ứng dụng web (Web application) 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), Máy chủ cơ sở dữ liệu và Hạ tầng mạng TCP/IP kết nối giữa máy khách và máy chủ web.



Hình 1.1: Kiến trúc chuẩn của ứng dụng web [1]

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 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.

Các máy chủ web phổ biến hiện này có thể kể tới là Apache, Nginx, IIS, Tomcat… Các ứng dụng web thì tùy thuộc vào yêu cầu triển khai mà có thể được tạo nên bởi các ngôn ngữ lập trình khác nhau như: C#, Java, Python, PHP, Ruby… Cơ sở dữ liệu (Data) sẽ đóng vai trò lưu trữ, cung cấp thông tin cho ứng dụng web trong quá trình xử lý request. Một số hệ quản trị cơ sở dữ liệu thường được sử dụng

bao gồm: SQL Server, MySQL, MongoDB, Oracle. Ngoài ra, tùy thuộc vào độ phức tạp, quy mô, yêu cầu trong việc phát triển mà website có thể có thêm nhiều thành phần khác như Message Queue, Proxy, Cache.

- Giao thức HTTP

HTTP là viết tắt của Hypertext Transfer Protocol hay còn gọi là giao thức truyền tải siêu văn bản, là một trong năm giao thức chuẩn của mạng internet, được dùng để liên hệ thông tin giữa máy cung cấp dịch vụ (Web server) và máy khách web (Web client) trong mô hình Client/Server. HTTP sử dụng cổng chuẩn 80 là một giao thức ứng dụng của bộ giao thức TCP/IP.

- Giao thức HTTPS

HTTPS là viết tắt của Hypertext Transfer Protocol Secure, là một giao thức kết hợp giữa giao thức HTTP và giao thức bảo mật SSL hoặc TLS nhằm mục đích gia tăng tính an toàn cho việc truyền dữ liệu giữa Web server và trình duyệt Web. Giao thức HTTPS thường được dùng trong các giao dịch nhạy cảm cần tính bảo mật cao sử dụng các kỹ thuật mật mã trước khi gửi giúp bảo mật thông tin trao đổi tránh bị nghe lén và lạm dụng. Theo mặc định giao thức HTTPS sử dụng cổng 443 để truyền dữ liệu.

Máy khách web và máy chủ web giao tiếp với nhau bằng giao thức HTTP hoặc HTTPS thông qua phương thức yêu cầu/đáp ứng (Request/Response), trong đó yêu cầu là http request gửi từ máy khách web lên máy chủ web và đáp ứng hay phản hồi là http response gửi từ máy chủ web tới máy khách web.

Cấu trúc của một http request, như mô tả trên Hình 1.2 gồm các thành phần sau:

  - Method: là phương thức mà HTTP Request này sử dụng, thường là GET,POST, ngoài ra còn một số phương thức khác như HEAD, PUT, DELETE, OPTION, CONNECT.

- URI: là địa chỉ định danh của tài nguyên. URI có thể là dấu *. – HTTP version: là phiên bản HTTP đang sử dụng
- Request headers: cho phép client gửi thêm các thông tin bổ sung về thông điệp HTTP request và về chính web browser. Một số trường request header thông dụng như: Accept, Cookie, User-Agent, Content-Type, Connection…
- Request body: nội dung web browser gửi cho web server (file, nội dung dạng json, parameter post…)

```
POST
/gen_204?atyp=i&ei=4BTKXYUJ7NfPuw-pk6oY&ct=slh&v=2&m=HV&t=C&s=1&pv=0.348789283644
0,R,1,7,20,28,92,33:0,R,1,CAcQAA,166,172,600,315:0,R,1,CAcQAQ,166,172,600,95:6,B,
Host: www.google.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.google.com/
Cookie: 1P_JAR=2019-11-12-02;
NID=191=VR0UnwUg5UXmMB8wNm-9wCgxlFOelFZY5rXJcDkFgt_1NwvXxr6REZ7oRfYLC72lVoPBY1aST
kc5ALiNhsRtkfq60f1VL2PGNdU; ANID=AHWqTUlwgbWb26DpDUzd_xlWeZM67Wpa4SHR1j_lHUQAAXWR
Connection: close
Content-Length: 0
```

Hình 1.2: Cấu trúc của http request

```
HTTP/1.1 204 No Content
Content-Type: text/html; charset=UTF-8
Date: Tue, 12 Nov 2019 02:11:54 GMT
Server: gws
Content-Length: 0
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Set-Cookie: 1P_JAR=2019-11-12-02; expires=Thu, 12-Dec-2019 02:11:54 GMT; path=/;
Alt-Svc: quic=":443"; ma=2592000; v="46,43",h3-Q050=":443"; ma=2592000,h3-Q049="
ma=2592000,h3-Q043=":443"; ma=2592000
Connection: close
```

Hình 1.3: Cấu trúc của http reponse

Cấu trúc của một http response, như mô tả trên Hình 1.3 gồm các thành phần sau:

- HTTP-version: phiên bản HTTP cao nhất mà server hỗ trợ.
- Status-Code: mã kết quả trả về.
- Reason-Phrase: mô tả về Status-Code.
- Response header: phép web server gửi thêm các thông tin bổ sung về thông điệp HTTP response
- Response body: nội dung ứng dụng web trả về cho web browser

1.1.2 Các yêu cầu bảo mật ứng dụng web, website

Trước những nguy cơ tiềm ẩn về an ninh mạng như hiện nay để đảm bảo tính bảo mật và giảm thiểu các rủi ro liên quan đến ứng dụng web cần có những chính sách, tiêu chuẩn an toàn thông tin cho ứng dụng web, website. Dưới đây liệt kê một số yêu cầu bảo mật ứng dụng web, website.

1.1.2.1. Yêu cầu về cài đặt

- Yêu cầu cài đặt trên hệ điều hành an toàn, đã được thiết lập cấu hình chính sách bảo mật để đảm bảo ATTT mức hệ điều hành cho web server.
- Phiên bản cài đặt phải cập nhật các bản vá của nhà sản xuất để tránh bị tấn công qua các lỗ hổng đã biết.

1.1.2.2. Tắt/disable các thành phần mặc định

- Gỡ bỏ các thư mục/trang mặc định như: các trang ví dụ, hướng dẫn, các trang quản trị web server từ xa, các trang phục vụ development, debug để tránh bị khai thác lỗ hổng qua các nội dung web mặc định.
- Tắt các Module/Extension nguy hiểm không sử dụng như: các module hiển thị thông tin server (module info, status, version), hiển thị nội dung thư mục, các module xử lý CGI, xử lý webdav để tránh bị khai khác các module extension không sử dụng.

- Tắt chế độ tự động triển khai hoặc gỡ lỗi trên web server (nếu có). Chế độ tự động triển khai hoặc gỡ lỗi trên web server có thể đưa ra nhiều thông tin nhạy cảm hoặc gây mất kiểm soát về security.

1.1.2.3. Thay đổi các thành phần mặc định

Thay đổi thông báo lỗi mặc định của web server. Tránh để lộ các thông tin nhạy cảm của hệ thống khi xảy ra lỗi như 500, 503. Thay đổi thông tin banner của dịch vụ HTTP, không để lộ thông tin phiên bản hệ điều hành, web server để tránh việc hacker có thể tìm ra phiên bản web server. Từ đó, họ tấn công vào các lỗ hổng 1-days liên quan, đã được công bố [12].

1.1.2.4. Giới hạn truy cập

- Chỉ truy cập quản trị từ xa trong mạng nội bộ của doanh nghiệp và có phương thức xác thực người dùng. Hacker có thể chiếm quyền kiểm soát dịch vụ web qua kênh quản trị từ xa từ ngoài internet. Do đó cần cô lập các kết nối quản trị từ xa và có các phương thức xác thực người dùng như: sử dụng tài khoản/mật khẩu, OTP, private key…
- Không cho phép liệt kê file, thư mục để tránh bị hacker phát hiện ra các file, thư mục nhạy cảm của hệ thống.

1.2. Các Nguy Cơ và Các Dạng Tấn Công Lên Ứng Dụng Web

1.2.1 Các nguy cơ và các lỗ hổng bảo mật trong website, ứng dụng web (TOP 10 OWASP 2017)

Open Web Application Security Project (OWASP) [8][9][10] là một tổ chức bao gồm các chuyên gia bảo mật hàng đầu thế giới, chuyên cung cấp các thông tin về những ứng dụng và rủi ro đặt ra một cách trực tiếp, khách quan và thực tế nhất. Từ năm 2013 đến nay, cứ 4 năm 1 lần, OWASP lại công bố danh sách Top 10 các rủi ro bảo mật ứng dụng lớn nhất, được gọi là OWASP Top 10.

Danh sách này được coi là chuẩn AppSec và được cộng đồng an ninh mạng tin tưởng. Danh sách bao gồm thông tin mới nhất về các lỗ hổng, các mối đe dọa và cuộc tấn công cũng như những thủ thuật để phát hiện và khắc phục. Các thành viên dự án lập ra danh sách này dựa trên việc phân tích tỉ lệ xuất hiện và mức độ nghiêm trọng của từng mối đe dọa.

OWASP Top 10 năm 2017 được phát hành công khai, dựa trên cuộc thăm dò, kiểm tra hơn 2,3 triệu lỗ hổng tác động đến 50000 ứng dụng, bao gồm 2 bản cập nhật lỗ hổng quy mô lớn và cập nhật các kịch bản tấn công mới. Phần tiếp theo mô tả danh sách Top 10 của năm 2017.

A1 – Injection (Lỗi nhúng mã)

Nếu ứng dụng của bạn có thể nhận dữ liệu đầu vào người dùng đến cơ sở dữ liệu back-end thì ứng dụng của bạn có thể sẽ phải đối mặt với cuộc tấn công bằng mã nhúng. Lỗi nhúng mã (Injection) là tập hợp các lỗ hổng bảo mật xảy ra khi dữ liệu đáng ngờ được chèn vào ứng dụng dưới dạng lệnh hay truy vấn. Injection xảy ra do sự thiếu sót trong việc lọc các dữ liệu đầu vào không đáng tin cậy. Khi bạn truyền các dữ liệu chưa được lọc tới CSDL (Phổ biến như lỗ hổng SQL Injection), tới trình duyệt (lỗ hổng XSS), tới máy chủ LDAP (lỗ hổng LDAP Injection hoặc tới bất cứ vị trí nào khác. Vấn đề là kẻ tấn công có thể chèn các đoạn mã độc để gây ra lộ lọt dữ liệu và chiếm quyền kiểm soát trình duyệt của khách hàng.

A2 - Broken Authentication and Session Management (Lỗi xác thực và quản phiên yếu)

Đây là nhóm các vấn đề có thể xảy ra trong quá trình xác thực và quản lý phiên. Có một lời khuyên là không nên tự phát triển các giải pháp mã hóa vì rất khó có thể làm được chính xác. Vì khi các chức năng của ứng dụng được thực hiện không chính xác, tin tặc có thể dễ dàng xâm nhập, ăn cắp thông tin tài khoản, mật khẩu và khai thác các lỗ hổng khác bằng cách sử dụng các chứng chỉ đã đánh cắp. Tài khoản mỗi người dùng cá nhân nên là duy nhất, không bị trùng lặp dưới bất kì hình thức nào. Nếu không có bất kì sự quản lý cần thiết nào thì tin tặc có thể lẻn

vào, ngụy trang thành người dùng để ăn cắp thông tin tài khoản, mật khẩu và sử dụng cho những lần truy cập sau này.

A3 - Cross-Site Scripting (XSS)

Lỗ hổng XSS là một lỗ hổng thường thấy trong các ứng dụng web. XSS cho phép tin tặc đưa các mã script như là một phần của đầu vào các trang web từ phía máy khách. Khi đầu vào này không được lọc, tin tặc có thể sử dụng các công cụ kiểm soát truy cập của họ. Chúng thực hiện bằng cách đánh lừa trình duyệt chấp nhận dữ liệu từ một nguồn không đáng tin cậy để thực thi mã độc trên trình duyệt của người dùng.

Các ứng dụng cho phép người dùng nhập dữ liệu vào mà không có toàn quyền kiểm soát dữ liệu ra có nguy cơ bị tấn công XSS rất cao. Một cuộc tấn công XSS thành công có thể gây thiệt hại nghiêm trọng cho các trang web và kẻ tấn công có thể lấy được cookie của người dùng trên hệ thống hoặc lừa người dùng đến các trang web độc hại. Một số kiểu tấn công XSS khác như: Stored XSS, DOM Based XSS và Reflected XSS.

A4 - Broken Access Control (Điều khiển truy nhập yếu)

Đây là trường hợp điển hình của việc cho rằng đầu vào của người dùng là tin cậy từ đó dẫn đến lỗ hổng bảo mật. Dạng lỗi này liên quan đến việc các kiểm soát truy nhập đối với người dùng không được thực hiện chặt chẽ. Kẻ tấn công có thể khai thác các lỗi dạng này để truy nhập trái phép vào các tính năng, dữ liệu, như truy nhập, hoặc sửa đổi dữ liệu của người dùng khác, xem các file nhạy cảm, thay đổi quyền truy nhập.

Nguyên nhân lỗi kiểm soát truy cập xảy ra có thể là do các nhà phát triển thường bị bế tắc trong việc kiểm soát truy cập phù hợp với các quy tắc đặt ra.

A5 - Security Misconfiguration (Cấu hình thiếu an toàn)

Trong thực tế, máy chủ web và các máy chủ ứng dụng thường bị cấu hình sai. Một lỗi cấu hình máy chủ web và các máy chủ ứng dụng thường gặp gồm:

- Chạy ứng dụng khi chế độ debug được bật.
- Cho phép liệt kê thư mục
- Sử dụng phần mềm lỗi thời (WordPress plugin, PhpMyAdmin cũ)
- Cài đặt các dịch vụ không cần thiết.

Với cấu hình an ninh lỏng lẻo tại các tầng kiến trúc của web như nền tảng, framework, máy chủ, cơ sở dữ liệu và mã tùy chỉnh nên tin tặc có thể khai thác tấn công và có quyền truy cập dữ liệu. Vì thế, tất cả các tầng kiến trúc của web phải được cập nhật thường xuyên.

A6 - Sensitive data exposure (Rò rỉ dữ liệu nhạy cảm)

Nhiều ứng dụng web không có các cơ chế 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. Vì vậy các dữ liệu nhạy cảm phải được mã hóa mọi lúc, bao gồm cả khi gửi đi và khi lưu trữ, không được phép có ngoại lệ như thông tin thẻ tín dụng và mật khẩu người dùng không bao giờ được gửi đi hoặc được lưu trữ không được mã hóa.

Do vậy, các cơ chế bổ sung 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,…) là rất cần thiết, nếu không sẽ có thể dẫn đến việc vi phạm quyền riêng tư ở quy mô lớn và hậu quả để lại cũng sẽ rất nghiêm trọng.

A7 - Missing function level access control (Lỗi phân quyền)

Nhiều ứng dụng web kiểm tra quyền truy nhập vào một tính năng trước khi hiển thị tính năng đó trên giao diện người dùng. Trong khi ứng dụng cần thực hiện các phép kiểm tra quyền truy nhập trên mỗi tính năng trên máy chủ khi tính năng đó được truy nhập. Nếu các yêu cầu không được kiểm tra đầy đủ, kẻ tấn công có thể làm giả các yêu cầu để truy nhập vào các tính năng mà không qua khâu kiểm tra quyền truy nhập.

Lỗ hổng này là sai sót trong vấn đề phân quyền, khi một hàm được gọi trên máy chủ, quá trình phân quyền không chính xác. Các nhà phát triển dựa vào thực tế là phía máy chủ tạo ra giao diện người dùng và họ nghĩ rằng khách hàng không thể truy cập các chức năng nếu không được cung cấp bởi máy chủ. Tuy nhiên, kẻ tấn công luôn có thể yêu cầu các chức năng "ẩn" và sẽ không bị cản trở bởi việc giao diện người dùng không cho phép thực hiện các chức năng này. Hãy tưởng tượng trong giao diện người dùng chỉ có bảng điều khiển/admin và nút nếu người dùng thực sự là quản trị viên. Không có gì ngăn cản kẻ tấn công phát hiện ra những tính năng này và lạm dụng nó nếu không phân quyền.

A8 - Cross Site Request Forgery (CSRF)

CSRF là dạng tấn công người dùng web, lợi dụng cơ chế tự động đăng nhập của một số website. Kẻ tấn công lừa người dùng thực hiện các đoạn mã độc, nhúng trong các trang web bình thường trong ngữ cảnh người dùng đang ở trong phiên làm việc với website. Mã độc chạy trên trình duyệt của người dùng đang ở trong phiên làm việc có thể giúp hacker thực hiện các giao dịch hoặc đánh cắp thông tin.

A9 - Using component with known vulnerabilities (Sử dụng những thư viện, ứng dụng tồn tại lỗ hổng trước đó)

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.

Việc dùng những components, libraries, frameworks không an toàn sẽ làm cho ứng dụng của bạn dễ bị khai thác hơn. 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. Việc sử dụng những ứng dụng chứa lỗ hổng đã biết có thể làm

suy giảm khả năng phòng vệ của ứng dụng và cho phép tin tặc thực hiện nhiều loại tấn công lên hệ thống.

A10- Underprotected APIs (Các API không được bảo vệ)

API ngày càng trở nên phổ biến trong thế giới ứng dụng ngày nay bởi các ứng dụng thường được viết bằng JavaScript và sử dụng API để lấy dữ liệu. Do đó, API đóng vai trò như một liên kết giữa các nền tảng máy khách phức tạp và một loạt các ứng dụng hay dịch vụ web. Tuy nhiên, bản thân các API này thường không được bảo vệ và chứa đựng nhiều lỗ hổng bảo mật khiến ứng dụng của chúng ta rất dễ bị tấn công. API cũng chứa nhiều giao thức phức tạp như SOAP/XML, REST/JSON, RPC và GWT mà kiểm thử bảo mật không thể kiểm tra thành công, khiến các API trở thành điểm mù quan trọng trong các tổ chức đang sử dụng chúng

1.2.2 Một số dạng tấn công web cơ bản

1.2.2.1. Tấn công chèn mã SQLi

Tấn công chèn mã SQL (SQL Injection - SQLi) 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 [1][5][18]. 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.

Có 2 nguyên nhân của lỗ hổng trong ứng dụng cho phép thực hiện tấn công chèn mã SQL:

- Dữ liệu đầu vào từ người dùng hoặc từ các nguồn khác không được kiểm tra hoặc kiểm tra không kỹ lưỡng;
- 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.

Để minh họa kỹ thuật tấn công SQLi, ta giả thiết có 1 form đăng nhập (Log in) và đoạn mã xử lý xác thực người dùng lưu trong bảng cơ sở dữ liệu tbl_accounts(username, password) cho như sau:

```
<!— Form đăng nhập -->
<form method="post" action="/log_in.asp">
  Tên đăng nhập: <input type=text name="username"><br \>
  Mật khẩu: <input type=password name="password"><br \>
  <input type=submit name="login" value="Log In">
</form>
<%
' Mã xử lý đăng nhập trong file log_in.asp:
' giả thiết đã kết nối với CSDL SQL qua đối tượng conn và bảng tbl_accounts lưu thông tin người dùng
Dim username, password, sqlString, rsLogin
' lấy dữ liệu từ form
username = Request.Form("username")
password = Request.Form("password")
' tạo và thực hiện câu truy vấn sql
sqlString = "SELECT * FROM tbl_accounts WHERE username='" & username & "' AND password = '" & password & "'"
set rsLogin = conn.execute(sqlString)
if (NOT rsLogin.eof()) then
  ' cho phép đăng nhập, bắt đầu phiên làm việc
else
  ' từ chối đăng nhập, báo lỗi
end if
     %>
```

Nếu người dùng nhập 'admin' vào trường *username* và 'abc123' vào trường *password* của form, mã xử lý hoạt động đúng:

- Nếu tồn tại người dùng với *username* và *password* kể trên, hệ thống sẽ cho phép đăng nhập với thông báo đăng nhập thành công;

- Nếu không tồn tại người dùng với *username* và *password* đã cung cấp, hệ thống sẽ từ chối đăng nhập và trả lại thông báo lỗi.

Tuy nhiên, nếu người dùng nhập *aaaa' OR 1=1--* vào trường *username* và một chuỗi bất kỳ, chẳng hạn 'aaaa' vào trường *password* của form, mã xử lý hoạt động sai và chuỗi chứa câu truy vấn SQL sẽ trở thành:

```
SELECT * FROM tbl_accounts WHERE username='aaaa' OR 1=1--
' AND password='aaaa'
```

Câu truy vấn sẽ trả về mọi bản ghi trong bảng do thành phần OR 1=1 làm cho điều kiện trong mệnh đề WHERE trở lên luôn đúng và phần kiểm tra mật khẩu đã bị loại bỏ bởi ký hiệu (--). Phần lệnh sau ký hiệu (--) được coi là ghi chú và không được thực hiện. Nếu trong bảng tbl_accounts có chứa ít nhất một bản ghi, kẻ tấn công sẽ luôn đăng nhập thành công vào hệ thống.

1.2.2.2. Tấn công Cross-Site Scriting (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 (ban đầu được gọi là LiveScript – trên trình duyệt Netscape). 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. 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 [6].

XSS có thể được xem là một dạng của chèn mã HTML (HTML Injection). Trên thực tế, có thể thực hiện tấn công bằng chèn mã HTML mà không cần mã JavaScript và cũng không cần liên site, hoặc liên miền. Kẻ tấn công khai thác các lỗ hổng bảo mật để chèn mã XSS vào trang web, trong đó dữ liệu web (như tên và địa

chỉ email) và mã (cú pháp và các phần tử như <script>) của XSS được trộn lẫn vào mã gốc của trang web.

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. 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.

Nhìn chung, mã tấn công HTML/XSS có thể được chèn vào mọi vị trí trong địa chỉ (URI) và nội dung trang web. Các vị trí cụ thể có thể chèn mã:

- Các thành phần của URI (URI Components)
- Các trường nhập liệu (Form Fields)
- HTTP Request Header & Cookie
- JavaScript Object Notation (JSON)
- Các thuộc tính của DOM (Document Object Model)
- CSS (Cascade Style Sheet)

- Các nội dung do người dùng tạo ra.

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)

1.2.2.3. Duyệt đường dẫn (Directory traversal)

Directory Traversal là một dạng tấn công cho phép tin tặc truy cập đến những chỉ mục bị giới hạn, thực thi lệnh bên ngoài chỉ mục gốc của máy chủ web [15]. Hình thức tấn công này không cần sử dụng một công cụ nào mà chỉ đơn thuần là thao tác với các biến ../ (dot-dot-slash) để truy cập đến các file, thư mục, bao gồm cả source code, những file hệ thống…

Với một hệ thống tồn tại lỗ hổng directory traversal, tin tặc có thể lợi dụng nó để tìm ra chỉ mục gốc và truy cập một phần của file hệ thống. Điều này cho phép tin tặc xem những file bị giới hạn hay nguy hiểm hơn là tin tặc có thể thực thi những lệnh mạnh trên máy chủ, dẫn đến xâm hại hoàn toàn hệ thống. Chẳng hạn, tin tặc có thể sử dụng truy vấn có dạng như sau để truy cập trái phép vào file “sam” sao lưu tài khoản người dùng trong hệ điều hành Windows:

https://wahh-app.com/scripts/GetImage.aspx?file=..\..\windows\repair\sam

1.2.2.4. Tấn công CMDi

OS Command Injection (CMDi) là một lỗ hổng bảo mật web cho phép kẻ tấn công có thể thực thi các lệnh của hệ điều hành (OS) tùy ý trên máy chủ đang chạy service nào đó. Các lỗ hổng CMDi xảy ra khi phần mềm tích hợp dữ liệu do người dùng quản lý trong một lệnh, các dữ liệu này được xử lý trong trình thông dịch lệnh. Nếu dữ liệu không được kiểm tra, một hacker có thể sử dụng các ký tự đặc biệt để thay đổi lệnh đang được thực thi từ đó kẻ tấn công có thể khai thác, truy xuất thông tin, tấn công sang các hệ thống máy chủ khác trong cùng vùng mạng [16].

Ví dụ tại một trang đọc báo http://readnews.com.vn?news=monday.txt. Hệ thống trên phía server sẽ thực hiện câu lệnh: cat Monday.txt lấy dữ liệu file Monday.txt để truyền tải nội dung cho người dùng cần đọc.

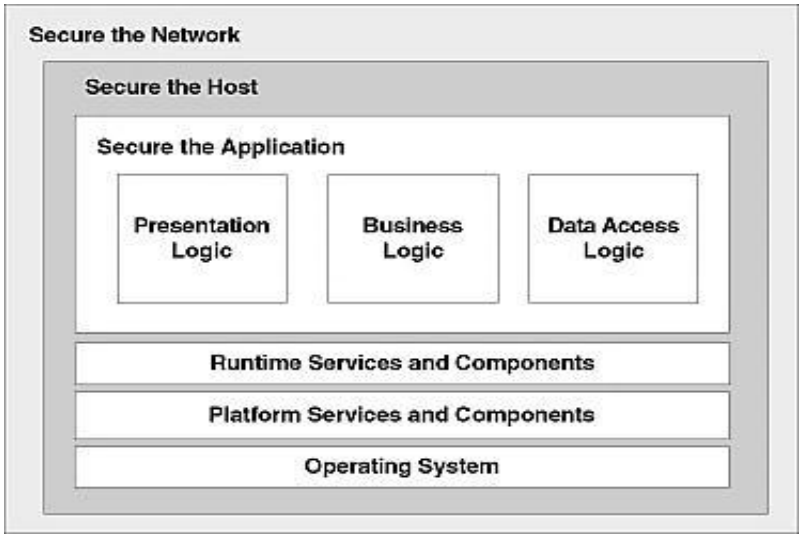
Kẻ tấn công có thể sử dụng các ký tự nối lệnh thực thi trên từng hệ điều hành để thực thi các câu lệnh mo ng muốn như là: *| ls –al; cat /etc/passwd*

1.2.3 Các biện pháp bảo mật ứng dụng web, website

Để đảm bảo tính bảo mật và giảm thiểu tối đa các rủi ro liên quan đến ứng dụng web vấn đề về an ninh mạng được quan tâm hàng đầu vì vậy cần có những chính sách, tiêu chuẩn an toàn thông tin cho ứng dụng web, website.

1.2.3.1. Nguyên tắc chung

Nguyên tắc bảo mật ứng dụng web tuân theo nguyên tắc chung của bảo mật an toàn hệ thống thông tin là phòng vệ nhiều lớp theo chiều sâu (Defense in depth). Hình 1.4 biểu diễn 3 lớp bảo mật ứng dụng web: Lớp bảo mật mạng (Network), Lớp bảo mật máy chủ (Host) và Lớp bảo mật ứng dụng (Application). Trong đó, lớp bảo mật mạng thực hiện bảo vệ ở vòng ngoài, lớp bảo mật máy chủ thực hiện bảo vệ nền tảng và lớp bảo mật ứng dụng thực hiện bảo vệ dữ liệu thông qua kiểm soát quyền truy nhập.



Hình 1.4: Các lớp bảo mật ứng dụng web [1]

Tiếp theo từng lớp bảo mật cũng phải đảm bảo từng nhiệm vụ cụ thể:

❖ Lớp bảo mật mạng đảm bảo cung cấp hạ tầng mạng an toàn cho giao tiếp giữa máy chủ và máy khách. Theo đó, các thiết bị mạng cần được cài đặt và cấu hình theo chuẩn, đảm bảo an toàn. Các thiết bị mạng thường được sử dụng trong lớp bảo mật mạng bao gồm:

✓ Switch: bộ chuyển mạch

✓ Router: bộ định tuyến

✓ Firewall: tường lửa

✓ IPS/IDS: hệ thống ngăn chặn/phát hiện đột nhập.

❖ Lớp bảo mật máy chủ (Host) có nhiệm vụ đảm bảo an toàn cho các thành phần nền tảng trong hệ thống. Cụ thể, lớp bảo mật máy chủ bao gồm:

✓ Bảo mật hệ điều hành

✓ Bảo mật cơ sở dữ liệu

✓ Bảo mật các phần mềm, dịch vụ trong hệ thống.

❖ Lớp bảo mật ứng dụng có trách nhiệm đảm bảo an toàn cho người dùng và dữ liệu của người dùng lưu trong hệ thống ứng dụng web. Các vấn đề có liên quan đến bảo mật ứng dụng bao gồm:

✓ Xác thực, trao quyền cho người dùng

✓ Quản lý cấu hình

✓ Kiểm tra dữ liệu đầu vào

✓ Quản lý phiên làm việc

✓ Mã hóa dữ liệu

✓ Quản lý các ngoại lệ

✓ Ghi và quản lý logs.

1.2.3.2. Một số biện pháp bảo mật cụ thể

1.2.3.2.1. Kiểm tra dữ liệu đầu vào

Kiểm tra dữ liệu đầu vào là một phần việc bắt buộc thực hiện với mọi loại dữ liệu cung cấp từ người dùng, đặc biệt với các dữ liệu từ mạng, hoặc các nguồn không tin cậy. Có thể nói, đây là một trong các phương pháp tiếp cận bảo mật hiệu quả nhất cho các ứng dụng web. Với ứng dụng web, việc kiểm tra dữ liệu đầu vào cần được thực hiện cả trên máy khách và máy chủ. Việc chỉ kiểm tra dữ liệu đầu vào trên máy khách (như sử dụng JavaSript) không thể đảm bảo chắc chắn các dữ liệu là hợp lệ khi được xử lý trên máy chủ do kẻ tấn công có thể sử dụng các kỹ thuật vô hiệu hóa bước kiểm tra trên máy khách như tắt JavaSript, hoặc tự tạo ra các form nhập liệu riêng. Các khâu cần thực hiện trong kiểm tra dữ liệu đầu vào, bao gồm: kiểm tra kích thước, định dạng và trong một số trường hợp kiểm tra cả nội dung và sự hợp lý của dữ liệu. Có thể sử dụng các bộ lọc dữ liệu để lọc bỏ các dữ liệu sai, dữ liệu chứa mã tấn công, hoặc lọc chỉ chấp nhận dữ liệu đúng. Nhìn chung, nên sử dụng các bộ lọc của các hãng, hoặc các tổ chức lớn, như bộ lọc XSS của dự án OWASP, hoặc Microsoft, do các bộ lọc này đã được kiểm thử kỹ và được cộng đồng đánh giá có hiệu quả trong một thời gian dài.

1.2.3.2.2. Giảm thiểu các giao diện có thể bị tấn công

Giảm thiểu các giao diện có thể bị tấn công là một phương pháp tiếp cận bảo mật hiệu quả khác cho các ứng dụng web. Nguyên tắc chung là sử dụng các biện pháp kiểm soát truy nhập để hạn chế đến tối thiểu việc người dùng truy nhập trực tiếp các ứng dụng, dịch vụ và hệ thống, nếu không thực sự cần thiết. Chẳng hạn, với các website, người dùng Internet chỉ được cấp quyền để truy nhập các trang web và bị cấm truy nhập trực tiếp vào hệ thống cơ sở dữ liệu của website. Mỗi người dùng, hoặc nhóm người dùng chỉ được cấp các quyền truy nhập "vừa đủ" để họ có thể thực hiện nhiệm vụ được giao. Ngoài ra, có thể sử dụng hợp lý các kỹ thuật mã để bảo mật các dữ liệu nhạy cảm cũng như dữ liệu truyền giữa máy chủ và máy khách, như sử dụng giao thức HTTPS thay cho HTTP.

1.2.3.2.3. Phòng vệ theo chiều sâu

Phòng vệ nhiều lớp theo chiều sâu (Defense in depth) là phương pháp tiếp cận bảo mật hiệu quả cho ứng dụng web nói riêng và các hệ thống thông tin nói chung, các lớp bảo mật thường được sử dụng cho ứng dụng web bao gồm: lớp bảo mật mạng, lớp bảo mật máy chủ và lớp bảo mật ứng dụng. Mỗi lớp bảo mật có tính năng tác dụng riêng và hỗ trợ cho nhau trong vấn đề đảm bảo an toàn tối đa cho ứng dụng web.

1.3. Kết luận Chương 1

Chương 1 đã giới thiệu tổng quan về kiến trúc ứng dụng web, các yêu bảo mật đối với ứng dụng web, web server. Chương này cũng đã giới thiệu các lỗ hổng nằm trong TOP 10 OWASP 2017 và một số lỗ hổng tấn công web điển hình hiện nay như là SQLi, XSS, Duyệt đường dẫn *(Directory traversal)*, CMDi cũng như cách phòng chống tương ứng đối với mỗi loại lỗ hổng cụ thể và đối với hệ thống web nói chung.

Trong chương 2, với nội dung là **PHÁT HIỆN TẤN CÔNG WEB DỰA TRÊN HỌC MÁY SỬ DỤNG WEB LOG,** luận văn sẽ tiếp tục đi tìm hiểu về WEBLOG, khái quát và các dạng, đồng thời đi sâu vào việc giới thiệu học máy và các thuật toán học máy, đưa ra mô hình phát hiện tấn công website và chi tiết các khâu xử lý dữ liệu.

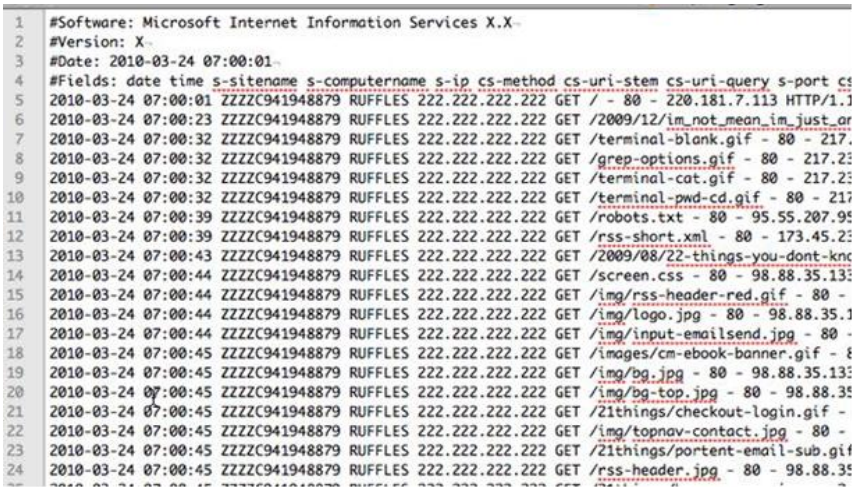
CHƯƠNG 2: PHÁT HIỆN TẤN CÔNG WEB DỰA TRÊN HỌC MÁY SỬ DỤNG WEB LOG

Chương 2 của luận văn sẽ trình bày khái quát WEB LOG, các dạng WEB LOG, hoạt động cũng như định dạng của từng loại WEB LOG, những hiểu biết cơ bản. Tiếp theo là tìm hiểu về học máy, một số thuật toán học máy sử dụng để phát hiện các loại tấn công web. Phần quan trọng trong chương 2 chính là đưa ra mô hình phát hiện tấn công web dựa trên học máy bao gồm sơ đồ thành phần cũng như nguyên lý hoạt động.

2.1. Tìm hiểu về Web log

2.1.1. Khái quát về Web log

Web log hay nhật ký web là tệp nhật ký tự động được tạo và duy trì bởi một máy chủ web. Mỗi lần người dùng truy cập vào trang Web, bao gồm từng chế độ xem tài liệu HTML, hình ảnh hoặc đối tượng khác đều được máy chủ web ghi lại [17][20]. Định dạng một bản ghi nhật ký web về cơ bản là một dòng văn bản cho mỗi lần truy cập vào trang web. Tài liệu này chứa thông tin về những người đã truy cập trang web, nơi họ đến và chính xác những gì họ đang làm trên trang web bao gồm một loạt các mục được sắp xếp theo thứ tự thời gian đảo ngược, thường được cập nhật thường xuyên với thông tin mới về các chủ đề cụ thể.



Hình 2.1:Các bản ghi web log sinh bởi máy chủ web Microsoft IIS [20]

Các web server chuẩn như Apache và Microsoft IIS tạo thông điệp ghi nhật ký theo một chuẩn chung (CLF – common log format). Tệp nhật ký CLF chứa các dòng thông điệp cho mỗi một gói HTTP request theo định dạng như sau:

Host Ident Authuser Date Request Status Bytes

Trong đó:

- Host: Tên miền đầy đủ của client hoặc IP
- Ident: Nếu chỉ thị IdentityCheck được kích hoạt và client chạy identd, thì đây là thông tin nhận dạng được client báo cáo
- Authuser: Nếu URL yêu cầu xác thực HTTP thì tên người dùng là giá trị của mã thông báo này
- Date: Ngày và giờ yêu cầu
- Request: Dòng yêu cầu của client, được đặt trong dấu ngoặc kép (“ ”)
- Status: Mã trạng thái (gồm ba chữ số)
- Bytes: số bytes trong đối tượng trả về cho client, ngoại trừ các HTTP header

Mỗi yêu cầu có thể chứa các các dữ liệu bổ sung như đường liên kết hoặc chuỗi ký tự của người dùng.

Nếu mã thông báo không có giá trị, thì mã thông báo được biểu thị bằng một dấu gạch ngang (-).

Ví dụ:

192.168.40.131 - - [08/May/2018:08:43:52 -0400] "GET /dvwa/login.php HTTP/1.1" 200 1289 "-"

"Mozilla/5.0 (X11; Linux x86_64; rv:31.0) Gecko/20100101 Firefox/31.0 Iceweasel/31.8.0"

2.1.2. Các dạng web log

- ✓ Tệp nhật ký truy cập
- ✓ Tệp nhật ký đối tượng
- ✓ Tệp nhật ký lỗi
- ✓ Tệp nhật ký giới thiệu

Bảng 2.1: Các loại định dạng của tệp nhật ký máy chủ Web

Các loại tệp nhật ký	Hoạt động	Định dạng	Trích xuất kiến thức
Nhật ký Truy cập	1. Ghi lại tất cả các yêu cầu người dùng xử lý bởi máy chủ. 2. Ghi thông tin về người dùng.	[Wed Oct 11 14:32:52 2000] [error] [Client 127.0.0.1] máy khách bị từ chối bởi máy chủ cấu hình: /export/home/live/ap/htdocs/test	Người dùng hồ sơ cá nhân Các mẫu thường xuyên. Sử dụng băng thông.
Nhật ký đối tượng	1. Trình duyệt người dùng 2. Phiên bản trình duyệt	"Mozilla/4.0 (compatible; MSIE 4.01; Windows NT)"	Phiên bản đại lý Hệ điều hành được sử dụng.
Tệp nhật ký lỗi	Danh sách lỗi cho người dùng yêu cầu được thực hiện bởi máy chủ.	[Wed Oct 11 14:32:52 2000] [error] [client 127.0.0.1] máy khách bị từ chối bởi máy chủ cấu hình: /export/home/live/ap/htdocs/test	Các loại lỗi Tạo địa chỉ lỗi IP Ngày và thời gian xảy ra lỗi.
Nhật ký giới thiệu	1.Thông tin về liên kết. 2. Chuyển hướng khách truy cập vào trang web.	"http://www.google.com/search?q=keyword", "/page.html"	Trình duyệt đã sử dụng Từ khóa. Chuyển hướng nội dung liên kết.

Định dạng tệp nhật ký: Có ba loại định dạng tệp nhật ký

Định dạng tệp nhật ký chung:

Được sử dụng bởi hầu hết các máy chủ web. Định dạng của tệp nhật ký này được chuẩn hóa và có thể được phân tích bởi chương trình phân tích web, định dạng mẫu của loại này được hiển thị sau đây:

127.0.0.1 user-identifier frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif

HTTP/1.0" 200 2326

Định dạng tệp nhật ký kết hợp:

Giống như định dạng tệp nhật ký chung nhưng có bổ sung thông tin hiện tại, những thông tin này là "phần giới thiệu, phần đối tượng người dùng và cookie prt ".

Nhiều nhật ký truy cập:

Hãy xem xét sự kết hợp của hai loại trước

Định dạng tệp (nhật ký chung và nhật ký kết hợp), trong loại định dạng tệp này có thể tạo nhiều thư mục cho Nhật ký truy cập, định dạng mẫu của loại này như dưới đây:

Logformat "%h %l %u %t \"%r\" %>s %b"

common

CustomLog logs/access_log common

CustomLog logs/referer_log "%{Referer}i -> %U"

CustomLog logs/agent_log "%{User-agent}i"

Thông số tệp nhật ký máy chủ:

Các tệp nhật ký chứa các tham số khác nhau và có thể rất hữu ích đối với người dùng được công nhận thuộc tính duyệt, nhiều thuộc tính có thể được thêm hoặc bật tùy thuộc vào cấu hình máy chủ và người dùng thỏa thuận bảo mật, một số cookie và thông tin cá nhân có thể được sử dụng nhưng nói chung có các tham số phổ biến tìm thấy trong các tập tin nhật ký.

Dưới đây sẽ minh họa trong Bảng 2.2, danh sách một số thông số hữu ích cho các quá trình phân tích.

Bảng 2.2: Một số trường của Web log

TT	Tên trường	Sự miêu tả
1	DATE	Ngày xử lý yêu cầu theo định dạng yyyy-mm-dd (năm-tháng-ngày)
2	TIME	Giờ xử lý yêu cầu theo định dạng hh:mm:ss (giờ:phút:giây)
3	CLIENT_IP	Địa chỉ IP của máy khách
4	HTTP_METHOD	Phương thức HTTP máy khách gửi yêu cầu
5	URI_STEM	Địa chỉ tương đối của trang, ví dụ /products/search.aspx
6	URI_QUERY	Chuỗi truy vấn của trang (HTTP query string). Ví dụ: category_id=100& category_desc=Science Fiction Books
7	HTTP_STATUS	Mã trạng thái xử lý yêu cầu. Ví dụ 200 là mã xử lý yêu cầu thành công
8	BYTE_RECEIVED	Số lượng Byte của yêu cầu (request) máy chủ nhận được từ máy khách
9	BYTE_SENT	Số lượng Byte của trả lời (response) máy chủ gửi được từ máy khách
10	TIME_TAKEN	Thời gian xử lý yêu cầu tính bằng giây

2.2. Khái quát về Học Máy và các thuật toán Học Máy

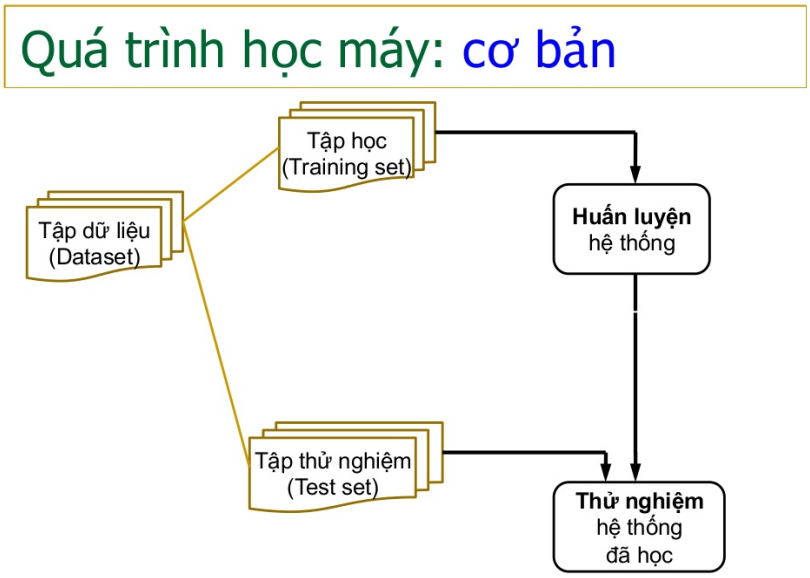
2.2.1. Khái quát về học máy

2.2.1.1. Khái niệm

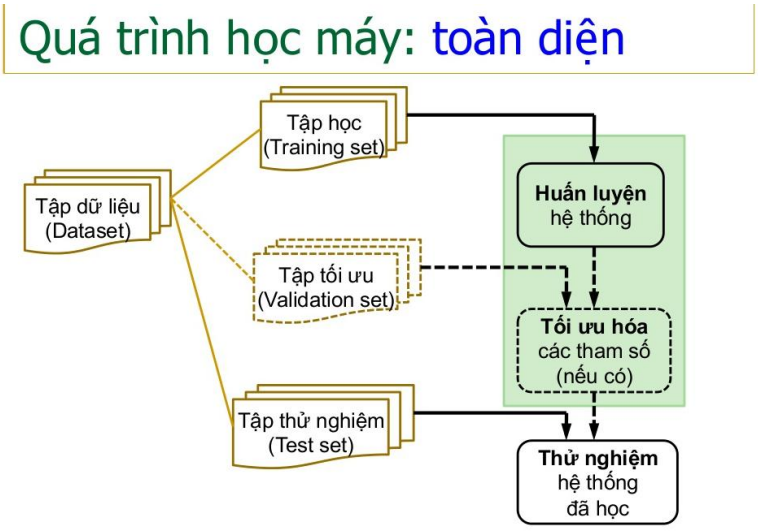
Học máy (machine learning) là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động

từ dữ liệu để giải quyết những vấn đề cụ thể. Cụ thể hơn, học máy là một phương pháp để tạo ra các chương trình máy tính bằng việc phân tích các tập dữ liệu (*là khả năng của chương trình máy tính sử dụng kinh nghiệm, quan sát, hoặc dữ liệu trong quá khứ để cải thiện công việc của mình trong tương lai thay vì chỉ thực hiện theo đúng các quy tắc đã được lập trình sẵn).* Học máy có liên quan lớn đến thống kê, vì cả hai lĩnh vực đều nghiên cứu việc phân tích dữ liệu, nhưng khác với thống kê, học máy tập trung vào sự phức tạp của các giải thuật trong việc thực thi tính toán. Nhiều bài toán suy luận được xếp vào loại bài toán NP-khó, vì thế một phần của học máy là nghiên cứu sự phát triển các giải thuật suy luận xấp xỉ mà có thể xử lý được [3][13].

Quá trình học máy đơn giản được hiểu là ta cung cấp tập dữ liệu để cho thuật toán có thể tự học mà không cần phải cài đặt các luật quyết định. Để từ đó ta sẽ đưa các dữ liệu kiểm thử vào để hệ thống đưa ra các kết quả nhận định dựa vào quá trình học trên tập dữ liệu huấn luyện (quá trình học thông thường, một hệ thống học máy cần có khả năng ghi nhớ, thích nghi, và đặc biệt là tổng quát hóa. Tổng quát hóa là khả năng của hệ thống học máy ra quyết định chính xác trong các trường hợp mới, chưa gặp, dựa trên kinh nghiệm học được từ dữ liệu hoặc các quan sát trước đó).



Hình 2.2: Quá trình học máy cơ bản [14]



Hình 2.3: Quá trình học máy toàn diện [14]

Học máy có hiện nay được áp dụng rộng rãi bao gồm máy truy tìm dữ liệu, chẩn đoán y khoa, phát hiện thẻ tín dụng giả, phân tích thị trường chứng khoán, phân loại các chuỗi DNA, nhận dạng tiếng nói và chữ viết, dịch tự động, chơi trò chơi và cử động rô-bốt (robot locomotion).

2.2.1.2. Phân loại kỹ thuật học máy

Xét theo phương thức học, các thuật toán ML được chia làm bốn nhóm, bao gồm "Học có giám sát" (Supervised Learning), "Học không giám sát" (Unsupervised Learning), "Học bán giám sát" (hay học kết hợp - Semi-supervised Learning) và "Học tăng cường" (Reinforcement Learning).

- Học có giám sát: Là phương pháp sử dụng những dữ liệu đã được gán nhãn từ trước để suy luận ra quan hệ giữa đầu vào và đầu ra. Các dữ liệu này được gọi là dữ liệu huấn luyện và chúng là cặp các đầu vào-đầu ra. Học có giám sát sẽ xem xét các tập huấn luyện này để từ đó có thể đưa ra dự đoán đầu ra cho 1 đầu vào mới chưa gặp bao giờ. Ví dụ dự đoán giá nhà, phân loại email.
- Học không giám sát: học phi giám sát sử dụng những dữ liệu chưa được gán nhãn từ trước để suy luận. Phương pháp này thường được sử dụng để

tìm cấu trúc của tập dữ liệu. Tuy nhiên lại không có phương pháp đánh giá được cấu trúc tìm ra được là đúng hay sai. Ví dụ như phân cụm dữ liệu, triết xuất thành phần chính của một chất nào đó.

- Học bán giám sát: sử dụng cả dữ liệu đã gán nhãn và chưa gán nhãn để huấn luyện, điển hình là một lượng nhỏ dữ liệu có gán nhãn cùng với lượng lớn dữ liệu chưa gán nhãn. Đây là phương pháp kết hợp giữa học có giám sát và học không giám sát.
- Học tăng cường: Phương pháp học tăng cường tập trung vào việc làm sao để cho 1 tác tử trong môi trường có thể hành động sao cho lấy được phần thưởng nhiều nhất có thể. Khác với học có giám sát nó không có cặp dữ liệu gán nhãn trước làm đầu vào và cũng không có đánh giá các hành động là đúng hay sai.

2.2.2. Một số thuật toán học máy

2.2.2.1. Naive Bayes

Naive Bayes Classification (NBC) [14] 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. Naive Bayes Classification là một trong những thuật toán được ứng dụng rất nhiều trong các lĩnh vực Machine learning dùng để đưa các dự đoán chính xác nhất dựa trên một tập dữ liệu đã được thu thập, vì nó khá dễ hiểu và độ chính xác cao. Nó thuộc vào nhóm Supervised Machine Learning Algorithms (thuật toán học có giám sát), tức là máy học từ các ví dụ từ các mẫu dữ liệu đã có.

Định luật Bayes được phát biểu như sau:

Định lý Bayes cho phép tính xác suất xảy ra của một sự kiện ngẫu nhiên A khi biết sự kiện liên quan B đã xảy ra. Xác suất này được ký hiệu là **P(A|B)**, và đọc là "xác suất của A nếu có B". Đại lượng này được gọi xác suất có điều kiện hay xác

suất hậu nghiệm vì nó được rút ra từ giá trị được cho của B hoặc phụ thuộc vào giá trị đó.

Theo định lí Bayes, xác suất xảy ra A khi biết B sẽ phụ thuộc vào 3 yếu tố:

* Xác suất xảy ra A của riêng nó, không quan tâm đến B. Kí hiệu là **P(A)** và đọc là xác suất của A. Đây được gọi là xác suất biên duyên hay xác suất tiên nghiệm, nó là "tiên nghiệm" theo nghĩa rằng nó không quan tâm đến bất kỳ thông tin nào về B.

* Xác suất xảy ra B của riêng nó, không quan tâm đến A. Kí hiệu là **P(B)** và đọc là "xác suất của B". Đại lượng này còn gọi là hằng số chuẩn hóa (normalising constant), vì nó luôn giống nhau, không phụ thuộc vào sự kiện A đang muốn biết.

* Xác suất xảy ra B khi biết A xảy ra. Kí hiệu là **P(B|A)** và đọc là "xác suất của B nếu có A". Đại lượng này gọi là khả năng (likelihood) xảy ra B khi biết A đã xảy ra. Chú ý không nhầm lẫn giữa khả năng xảy ra B khi biết A và xác suất xảy ra A khi biết B.

Tóm lại định lý Bayes sẽ giúp ta tính ra xác suất xảy ra của một giả thuyết bằng cách thu thập các bằng chứng nhất quán hoặc không nhất quán với một giả thuyết nào đó. Khi các bằng chứng tích lũy, mức độ tin tưởng vào một giả thuyết thay đổi. Khi có đủ bằng chứng, mức độ tin tưởng này thường trở nên rất cao hoặc rất thấp, tức là xác xuất sảy ra giả thuyết sẽ thay đổi thì các bằng chứng liên quan đến nó thay đổi.

Công thức của định luật Bayes [14] được phát biểu như sau:

$$P(H|x) = P(H) * P(x|H) / P(x)$$

Với:

- $P(H|x)$ là xác suất để xảy ra giả thuyết H với đầu vào là tập dữ liệu ngẫu nhiên cần dự đoán x.

- $P(H)$ là xác suất xảy ra của bản thân giả thuyết H mà không quan tâm đến x.
- $P(x|H)$ là xác suất xảy ra x khi biết H xảy ra, gọi là "xác suất của x nếu có H".
- $P(x)$ là xác suất xảy ra của riêng tập dữ liệu dự đoán x.

Tổng quát:

$$P(H|x1 \dots xn) = P(H)P(x1|H) \dots P(xn|H)/P(x1) \dots P(xn)$$

Điểm mạnh Naive Bayes:

- Dễ dàng và nhanh chóng để dự đoán và phân lớp dữ liệu thành các nhóm khác nhau.
- Dự đoán đa lớp với độ chính xác cao
- Naive Bayes thực hiện tốt hơn so với các mô hình khác như hồi quy và cần ít dữ liệu training.

Một số hạn chế:

- Khi tiến hành dự đoán một tập dữ liệu mà không hề có trong bộ đã được training thì xác suất này được xác định là 0.
- Việc bộ dữ liệu về các lớp phân loại không đồng đều sẽ dẫn đến dự đoán sai. Giả sử trong một câu có hai từ thuộc "khen" và "chê" thì hệ thống sẽ dự đoán nó nghiêng hẳn một hướng nếu số lượng tập dữ liệu đã được train trong hệ thống chúng ta có lượng từ tích cực (khen, tốt) vượt trội hơn lượng từ không tích cực (phê bình, không tốt) hoặc ngược lại.

Ứng dụng:

1. **Real time Prediction:** Tốc độ của thuật toán phân loại có thể giúp nó sử dụng trong việc ra quyết định trong thời gian thực.

2. **Multi class Prediction**: Bản chất thuật toán là phân loại và dự đoán rồi chia thành nhiều lớp.

3. **Sentiment Analysis**: Naive Bayes được sử dụng trong phân loại ngôn ngữ tự nhiên và cho kết quả tốt hơn so với một số thuật toán khác. Bên cạnh đó còn phân loại được spam-mail và nhận định được bình luận tích cực hay không tích cực trong mạng xã hội.

4. **Recommendation System:** các hệ thống gợi ý hoạt động dựa trên dự đoán.

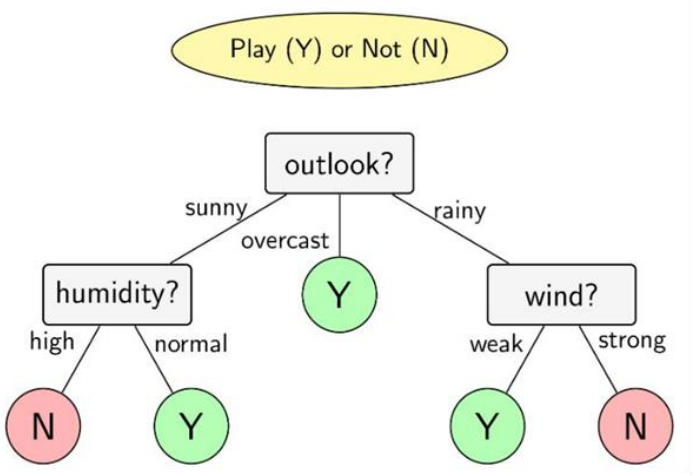
2.2.2.2. Cây quyết định

Cây quyết định (Decision Tree) là một đồ thị của các quyết định và các hậu quả có thể của nó (bao gồm rủi ro và hao phí tài nguyên). Cây quyết định được sử dụng để xây dựng một kế hoạch nhằm đạt được mục tiêu mong muốn. Các cây quyết định được dùng để hỗ trợ quá trình ra quyết định. Cây quyết định là một dạng đặc biệt của cấu trúc cây [3].

Trong lĩnh vực học máy, cây quyết định là một kiểu mô hình dự báo (predictive model), nghĩa là một ánh xạ từ các quan sát về một sự vật/hiện tượng tới các kết luận về giá trị mục tiêu của sự vật/hiện tượng. Mỗi một nút trong (internal node) tương ứng với một biến; đường nối giữa nó với nút con của nó thể hiện một giá trị cụ thể cho biến đó. Mỗi nút lá đại diện cho giá trị dự đoán của biến mục tiêu, cho trước các giá trị của các biến được biểu diễn bởi đường đi từ nút gốc tới nút lá đó. Kỹ thuật học máy dùng trong cây quyết định được gọi là học bằng cây quyết định, hay chỉ gọi với cái tên ngắn gọn là cây quyết định. Mô hình được biểu diễn trong hình 2.3 là một ví dụ đặc trưng của thuật toán cây quyết định.

Học bằng cây quyết định cũng là một phương pháp thông dụng trong khai phá dữ liệu. Khi đó, cây quyết định mô tả một cấu trúc cây, trong đó, các lá đại diện cho các phân loại còn cành đại diện cho các kết hợp của các thuộc tính dẫn tới phân loại đó. Một cây quyết định có thể được học bằng cách chia tập hợp nguồn thành

các tập con dựa theo một kiểm tra giá trị thuộc tính. Quá trình này được lặp lại một cách đệ qui cho mỗi tập con dẫn xuất. Quá trình đệ qui hoàn thành khi không thể tiếp tục thực hiện việc chia tách được nữa, hay khi một phân loại đơn có thể áp dụng cho từng phần tử của tập con dẫn xuất.



Hình 2.4: Mô hình thuật toán cây quyết định [13]

Cây quyết định có 2 loại:

Cây hồi quy (Regression tree): ước lượng các hàm có giá trị là số thực thay vì được sử dụng cho các nhiệm vụ phân loại (định giá, ước lượng giá trị của một căn nhà cần giao bán, khoảng thời gian nằm viện của 1 bệnh nhân).

Cây phân loại (Classification tree): được dùng trong các bài toán phân loại kết quả (phân biệt giới tính, kết quả trận đấu, …).

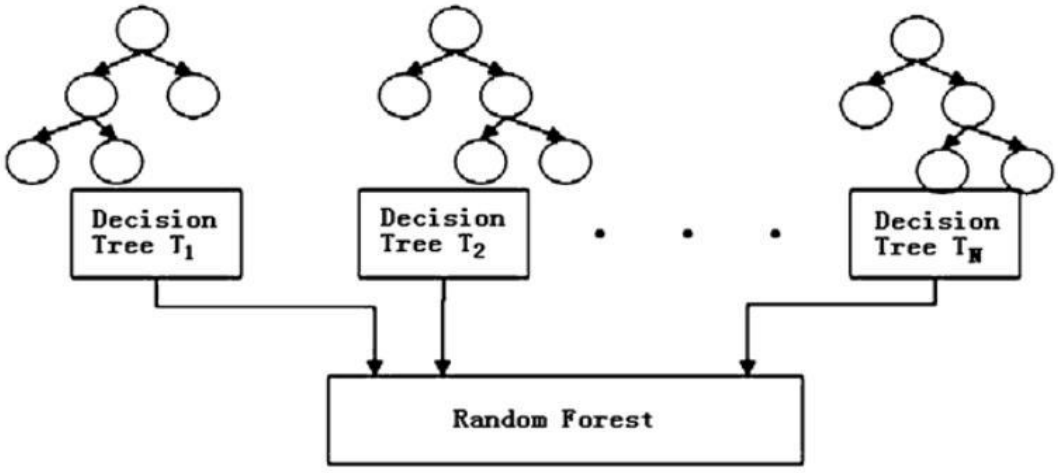
Ưu điểm của thuật toán cây quyết định là đơn giản và phổ biến. Mô hình sinh ra các quy tắc dễ hiểu cho người đọc, tạo ra bộ luật với mỗi nhánh là một luật của cây. Dữ liệu đầu vào không cần chuẩn hóa, có thể làm việc với các dữ liệu số và dữ liệu phân loại và có thể làm việc với dữ liệu lớn. Ngoài ra còn có thể xác thực mô hình bằng cách sử dụng các kiểm tra thống kê.

Bên cạnh ưu điểm, thuật toán còn một số nhược điểm đi kèm như là mô hình phụ thuộc rất lớn vào dữ liệu ta có, với một sự thay đổi nhỏ trong bộ dữ liệu thì cấu

trúc mô hình cây quyết định có thể thay đổi hoàn toàn. Ngoài ra cây quyết định hay gặp vấn đề overfitting.

2.2.2.3. Rừng ngẫu nhiên

Rừng ngẫu nhiên là một thuật toán học có giám sát. Rừng ngẫu nhiên sử dụng các cây để làm nền tảng. Rừng ngẫu nhiên là một tập hợp của các cây quyết định, mà mỗi cây được chọn theo một thuật toán dựa vào ngẫu nhiên. Rừng ngẫu nhiên hoạt động bằng cách đánh giá nhiều cây quyết định ngẫu nhiên, và lấy ra kết quả được đánh giá tốt nhất trong số kết quả trả về. Mô hình rừng ngẫu nhiên rất hiệu quả cho các bài toán phân loại vì nó huy động cùng lúc hàng trăm mô hình nhỏ hơn bên trong với quy luật khác nhau để đưa ra quyết định cuối cùng. Mỗi mô hình con có thể mạnh yếu khác nhau, nhưng theo nguyên tắc "wisdom of the crowd", ta sẽ có cơ hội phân loại chính xác hơn so với khi sử dụng bất kì một mô hình đơn lẻ nào. Mô hình tiêu biểu cơ bản của thuật toán Random Forest được biểu diễn như hình sau:



Hình 2.5: Mô hình thuật toán rừng ngẫu nhiên [13]

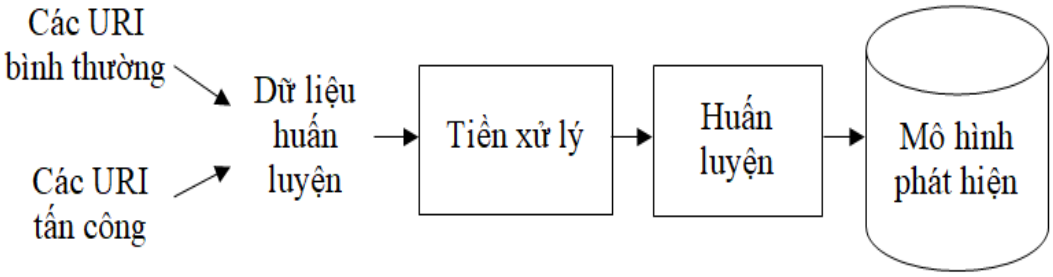
Ưu điểm của thuật toán Random Forest là có thể sử dụng cho cả bài toán Classification và Regression, làm việc được với dữ liệu thiếu giá trị. Khi rừng có nhiều cây hơn thì mô hình sẽ tránh được việc bị overfitting so với mô hình cây quyết định.

Nhược điểm của giải thuật này là tốn nhiều thời gian thực hiện do phải duyệt nhiều cây để tìm được kết quả tốt nhất.

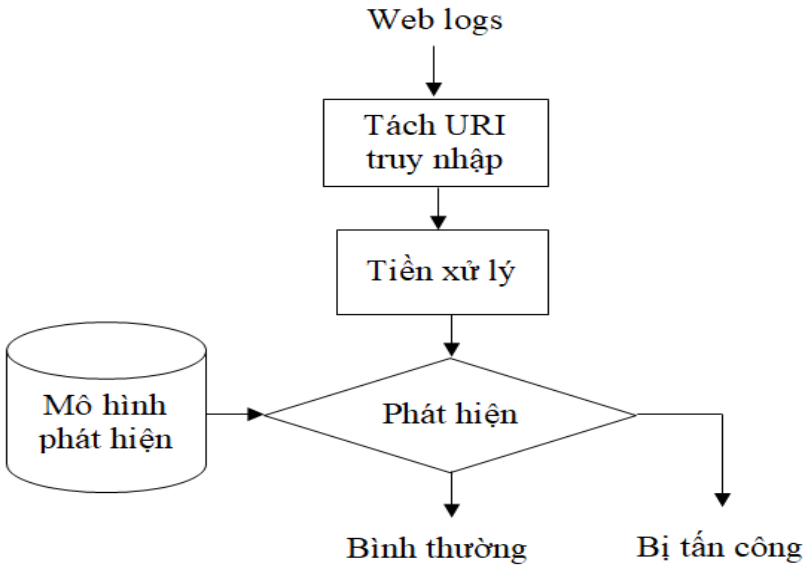
2.3. Phát hiện tấn công web dựa trên học máy sử dụng web log

2.3.1. Mô hình phát hiện

Mô hình phát hiện tấn công web cơ bản dựa trên học máy sử dụng web log trong luận văn này được triển khai theo 2 giai đoạn: (1) giai đoạn huấn luyện như biểu diễn trên Hình 2.5 và (2) giai đoạn phát hiện như biểu diễn trên Hình 2.6. Trong đó, các URI (Uniform Resource Indicator) là các chuỗi truy nhập được bóc tách từ web log. Các URI bình thường và URI tấn công dùng cho giai đoạn huấn luyện được lấy từ tập dữ liệu mẫu đã được gán nhãn [2].



Hình 2.6: Mô hình phát hiện tấn công web cơ bản: Giai đoạn huấn luyện [2]



Hình 2.7: Mô hình phát hiện tấn công web cơ bản: Giai đoạn phát hiện [2][20]

2.3.2. Các giai đoạn huấn luyện và phát hiện

2.3.2.1. Giai đoạn huấn luyện

Giai đoạn này thực hiện xây dựng mô hình phát hiện từ dữ liệu huấn luyện và gồm các bước sau:

- Thu thập tập dữ liệu huấn luyện: Dữ liệu huấn luyện được lấy từ tập Http Params Dataset [19]. Bộ dữ liệu này gồm 31.067 URI payload của các truy vấn web, kèm theo độ dài, loại tấn công (norm-bình thường, sqli-tấn công SQLi, xss-tấn công XSS, cmdi-tấn công chèn mã lệnh hệ điều hành, path-traversal-tấn công duyệt đường dẫn) và nhãn (norm - bình thường, anom-bất thường). Tập dữ liệu được chia thành 2 phần: phần dữ liệu cho huấn luyện gồm hơn 20.000 URI payload, phần còn lại để kiểm thử.
- Tiền xử lý: Khâu này thực hiện vector hoá các URI payload sử dụng phương pháp 3-gram và số hoá sử dụng phương pháp TF-IDF (tần suất từ - tần suất tài liệu nghịch). Tiếp theo sử dụng phương pháp PCA (Phân tích thành phần chính) để giảm số chiều vector xuống 256 (lựa chọn qua thực nghiệm).
- Huấn luyện: thực hiện huấn luyện dữ liệu sau tiền xử lý sử dụng thuật toán cây quyết định CART (hỗ trợ bởi thư viện Python) để sinh mô hình phát hiện. Mô hình phát hiện được lưu vào file để sử dụng cho giai đoạn phát hiện.
- Kiểm thử mô hình phát hiện: Sử dụng tập dữ liệu kiểm thử trong bộ dữ liệu Http Params Dataset [19] để kiểm thử độ chính xác phát hiện.

2.3.2.2. Giai đoạn phát hiện

Giai đoạn này thực hiện phân tích các dòng web log nhằm phát hiện các dấu hiệu tấn công SQLi, XSS, duyệt đường dẫn và chèn dòng lệnh hệ điều hành và gồm các bước sau:

- Tách URI truy nhập: từ dòng web log, tách lấy phần địa chỉ trang (URI stem) và truy vấn (URI query) phục vụ phát hiện. Đây là các thành phần tin tặc

thường sử dụng để nhúng các đoạn mã tấn công vào địa chỉ URL của trang web.

- Tiền xử lý: Khâu này thực hiện vector hoá URI truy nhập sử dụng phương pháp 3-gram, TF-IDF và PCA tương tự như trong giai đoạn huấn luyện.
- Phát hiện: vector của URI truy nhập được xác định là bình thường hay bị tấn công bởi mô hình phát hiện.

2.4. Kết luận Chương 2

Chương 2 đã giới thiệu những hiểu biết cơ bản về WEB LOG, các dạng WEB LOG, hoạt động cũng như định dạng của từng loại WEB LOG. Ngoài ra chương 2 cũng giới thiệu khái quát về học máy và đưa ra một số thuật toán học máy được sử dụng để phát hiện tấn công web như Naive Bayes, Cây quyết định (Decision Tree), Rừng ngẫu nhiên. Tuy nhiên mục đích của chương 2 chủ yếu là đi sâu vào việc trình bày mô hình phát hiện tấn công được sử dụng, các pha xử lý trong mô hình như là tiền xử lý, huấn luyện và phát hiện.

Trong chương 3, nội dung chủ yếu là giới thiệu tập dữ liệu được sử dụng để huấn luyện cho học máy, cách xử lý tiền dữ liệu, các bước làm trong pha huấn luyện và phân loại các dữ liệu đầu vào. Trình bày một số phương pháp để sử dụng huấn luyện và phát hiện, các kết quả sẽ được dùng để đánh giá mức độ hiệu quả khi sử dụng các phương pháp học máy khác nhau.

CHƯƠNG 3: CÀI ĐẶT VÀ THỬ NGHIỆM

Trong chương 3 của luận văn sẽ đi trọng tâm vào phần cài đặt và thử nghiệm để đưa ra các kết quả khả năng phát hiện tấn công web dựa trên web log của mô hình học máy. Chi tiết hơn vào từng phần sẽ là giới thiệu về tập dữ liệu được sử dụng để huấn luyện, các kịch bản huấn luyện và đánh giá thuật toán để đưa ra kết quả chính xác. Từ những kết quả thu được sẽ rút ra được những nhược điểm của mô hình học máy sử dụng để cải thiện thêm kết quả và hướng đi trong tương lai.

3.1. Giới thiệu tập dữ liệu

3.1.1. Tập dữ liệu mẫu

Tập dữ liệu mẫu dùng cho thử nghiệm đánh giá mô hình phát hiện là HttpParamsDataset [19]. Tập này gồm các tham số truy vấn HTTP với 19.304 truy vấn bình thường được gán nhãn *norm* và 11.763 truy vấn bất thường được gán nhãn *anom*. Bảng 3.1 cung cấp một số bản ghi mẫu trong tập dữ liệu này. Theo đó, dữ liệu được lưu trữ trong các file theo định dạng CSV và mỗi file có 4 cột ứng với 4 thuộc tính: payload (tải hay chuỗi truy vấn), length (độ dài payload), attack type (loại tấn công, gồm norm-bình thường, sqli, xss, cmdi và path-traversal) và label (nhãn, gồm norm-bình thường và anom-bất thường). Các truy vấn bất thường gồm 4 loại với số lượng như sau:

- 10.852 truy vấn tấn công chèn mã SQL được gán nhãn *sqli*
- 532 truy vấn tấn công XSS được gán nhãn *xss*
- 89 truy vấn tấn công chèn mã lệnh hệ điều hành được gán nhãn *cmdi*
- 290 truy vấn tấn công duyệt đường dẫn được gán nhãn *path-traversal*.

Tập dữ liệu HttpParamsDataset được chia thành 2 phần sử dụng cho huấn luyện và kiểm thử:

- Tập cho huấn luyện gồm 20.712 truy vấn, trong đó có 7.842 truy vấn bất thường;

- Tập cho kiểm thử gồm 10.355 truy vấn, trong đó có 3.921 truy vấn bất thường.

Bảng 3.1: Một số bản ghi của tập dữ liệu HttpParamsDataset [19]

payload (tải)	length	attack type	label
castellanos de zapardiel	24	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")) and 4386=utl_inaddr.get_host_address(chr(113)||chr(113)||chr(112)||chr(106)||chr(113)||(select (case when (4386=4386) then 1 else 0 end) from dual)||chr(113)||chr(122)||chr(118)||chr(122)||chr(113)) and (("smnw" like "smnw	227	sqli	anom
-2604)) as sekb where 6897=6897 or 1000=7683	44	sqli	anom
1');begin dbms_lock.sleep(5); end and ('jzlr'='jzlr	51	sqli	anom
1%")););create or replace function sleep(int) returns int as '/lib/libc.so.6','sleep' language 'c' strict; select sleep(5) and ((("%"="	134	sqli	anom
-1638' or 2724 in ((char(113)+char(113)+char(112)+char(106)+char(113)+(select (case when (2724=2724) then char(49) else char(48) end))+char(113)+char(122)+char(118)+char(122)+char(113))) and 'xkne'='xkne	203	sqli	anom
1%";call regexp_substring(repeat(left(crypt_key(char(65)||char(69)||char(83),null),0),500000000),null)--	104	sqli	anom
tweddle	7	norm	norm
sirevici	8	norm	norm

3.1.2. Dữ liệu web log thực

Dữ liệu web log thực là dữ liệu thu thập thực tế từ các máy chủ web. Luận văn sử dụng một phần dữ liệu web log thu thập bởi đề tài khoa học công nghệ cấp nhà nước, mã số KC.01.05/16-20 [20] thực hiện tại Học viện Công nghệ Bưu chính Viễn thông. Web log được thu thập và chuẩn hóa theo định dạng W3C Extended phục vụ cho phân tích, xử lý. Bảng 3.2 mô tả một số trường của web log thực [20], trong đó luận văn sử dụng thông tin từ trường URI_QUERY để phân tích phát hiện tấn công web.

Bảng 3.2: Một số trường của web log thực [20]

Tên trường	Mô tả
DATE	Ngày xử lý yêu cầu theo định dạng yyyy-mm-dd (năm-tháng-ngày).
TIME	Giờ xử lý yêu cầu theo định dạng hh:mm:ss (giờ:phút:giây)
CLIENT_IP	Địa chỉ IP của máy khách.
HTTP_METHOD	Phương thức HTTP máy khách gửi yêu cầu.
URI_STEM	Địa chỉ tương đối của trang, ví dụ /products/search.aspx.
URI_QUERY	Chuỗi truy vấn của trang (HTTP query string). Ví dụ: category_id=100& category_desc=Science Fiction Books.
HTTP_STATUS	Mã trạng thái xử lý yêu cầu. Ví dụ mã 200 là xử lý yêu cầu thành công.
BYTE_RECEIVED	Số lượng byte của yêu cầu (Request) máy chủ nhận được từ máy khách.
BYTE_SENT	Số lượng byte của trả lời (Response) máy chủ gửi được từ máy khách.
TIME_TAKEN	Thời gian xử lý yêu cầu tính bằng giây.

3.2. Tiền xử lý dữ liệu

Khâu tiền xử lý dữ liệu nhằm trích chọn và số hóa các đặc trưng cho mỗi truy vấn HTTP được thực hiện theo các bước được mô tả trong phần giới thiệu mô hình phát hiện ở Chương 2.

Do ta sử dụng một bộ 3-gram chuẩn được xây dựng từ việc lấy tất cả các phần tử 3-gram khác nhau trong quá trình phân tách 3-gram của các bản ghi của tập huấn luyện gồm 20.712 truy vấn, chính vì vậy độ dài của bộ 3-gram chuẩn này rất lớn có thể lên tới vài chục nghìn phần tử. Nếu thực hiện lưu trữ, ánh xạ các phần tử trên tập chuẩn này thì sẽ gặp khó khăn trong quá trình cả về lưu trữ và tốc độ xử lý tính toán. Vì vậy, để quá trình huấn luyện được diễn ra nhanh hơn mà không mất đi tính chính xác thì mô hình sẽ sử dụng một phương pháp để giảm chiều dữ liệu bộ 3-gram chuẩn đó là Principal Component Analysis (PCA).

Phương pháp PCA này dựa trên quan sát rằng dữ liệu thường không phân bố ngẫu nhiên trong không gian mà thường phân bố gần các đường/mặt đặc biệt nào đó. PCA sẽ dựa vào danh sách tập dữ liệu 3-gram đã phân tách ban đầu, sau đó dựa vào cách phân bố dữ liệu của tập dữ liệu 3-gram, PCA sẽ lựa chọn ra k phần tử có tầm quan trọng trong việc quyết định phân loại, và bỏ qua những phần tử ít quan trọng, không ảnh hưởng trong việc quyết định phân loại kết quả request. Kết quả của khâu tiền xử lý là vector với 256 chiều đại diện cho mỗi truy vấn URI.

3.3. Huấn luyện và kiểm thử mô hình phát hiện

Tập dữ liệu huấn luyện sau tiền xử lý được sử dụng để huấn luyện sử dụng thuật toán cây quyết định để sinh mô hình phân loại (cụ thể là thuật toán cây quyết định CART được hỗ trợ trong thư viện sk-learn của Python). Mô hình được lưu vào file cho khâu kiểm thử. Trong khâu kiểm thử, tập dữ liệu kiểm thử sau tiền xử lý được sử dụng để đánh giá độ chính xác phân loại.

Độ đo được sử dụng là độ chính xác được tính bằng số truy vấn phân loại đúng chia cho tổng số truy vấn thuộc mỗi loại. Độ đo được tính cho truy vấn bình thường, 4 loại bất thường và trung bình.

3.4. Thử nghiệm, kết quả và nhận xét

3.4.1. Lựa chọn công cụ thử nghiệm

Việc cài đặt và phát triển các chức năng của mô đun được thực hiện sử dụng các nền tảng và công cụ sau:

- ❖ Hệ điều hành Ubuntu 16.04, 64 bít
- ❖ Ngôn ngữ lập trình Python 3.5 và các thư viện kèm theo

3.4.2. Kết quả thử nghiệm

Các thử nghiệm được thực hiện với mô đun phát hiện tấn công SQLi, XSS, duyệt đường dẫn, CMDi bao gồm:

- ❖ Thử nghiệm độ chính xác phát hiện với tập dữ liệu kiểm thử, kết quả cho như trên Bảng 3.2
- ❖ Thử nghiệm phát hiện trên dữ liệu web log thực, kết quả cho như trên Bảng 3.3

Bảng 3.2: Kết quả kiểm thử mô hình phát hiện tấn công web cơ bản sử dụng tập dữ liệu kiểm thử [19]

```
$ python3 payload_training.py

Overall detection rate =  98.56
Normal correct:  6344 / 6434 = 98.6
SQLi correct:  3593 / 3617 = 99.34
XSS correct:  152 / 177 = 85.88
CMDi correct:  22 / 30 = 73.33
Path travel correct:  95 / 97 = 97.94
```

Bảng 3.3: Một số kết quả chi tiết phát hiện tấn công web sử dụng web log thực

Loại tấn công phát hiện	Chuỗi tấn công sử dụng
Tấn công SQLi	fpw=(select%20convert(int%2cCHAR(65)))
Tấn công XSS	type=vh01i%27%3e%3cscript%3ealert%281%29%3c%2fscript%3eooq5g
Duyệt đường dẫn	type=../../../../../../../../../../etc/passwd%00
Tấn công CMDi	fpw=WEB-INF/web.xml%3f

3.4.3. Nhận xét

Từ kết quả thử nghiệm mô hình phát hiện, có thể rút ra một số nhận xét sau:

- ✓ Mô hình phát hiện tấn công web cơ bản đạt độ chính xác phát hiện trung bình khá cao, đạt 98.51%. Hầu hết các dạng tấn công và trạng thái bình thường đều có độ chính xác phát hiện cao, riêng độ chính xác phát hiện tấn công CMDi chỉ đạt 66.67% do lượng dữ liệu huấn luyện cho dạng tấn công này khá ít. Trên thực tế, tấn công CMDi ít gặp trên dịch vụ web hơn các dạng SQLi, XSS và duyệt đường dẫn.
- ✓ Kết quả phát hiện thử trên web log thực cho thấy mô hình phát hiện khá chính xác từng loại tấn công. Mô hình có khả năng phát hiện 4 dạng tấn công web cơ bản bao gồm SQLi và XSS, tấn công duyệt đường dẫn và CMDi.

3.5. Kết luận chương 3

Trong chương 3 của luận văn đã mô tả chi tiết dữ liệu được sử dụng cho mô hình phát hiện tấn công web sử dụng học máy, mô tả chi tiết các phương pháp huấn luyện và phát hiện, thống kê chi tiết các kết quả đạt được bằng nhiều kịch bản thử nghiệm khác nhau từ đó rút ra được những nhận xét ưu điểm và những hạn chế của phương pháp học máy sử dụng.

KẾT LUẬN

Kết quả đạt được:

Từ nội dung của 3 chương, luận văn đã đạt được những kết quả sau:

- Trình bày khái quát về ứng dụng web, các yêu cầu bảo mật đối với ứng dụng web, web server, các loại tấn công web cũng như đặc điểm cách khai thác của loại tấn công web phổ biến và các biện pháp bảo mật, cách phòng chống.
- Trình bày các phương pháp phát hiện tấn công web sử dụng học máy, các thuật toán học máy được áp dụng cho bài toán phát hiện tấn công web. Đưa ra mô hình phát hiện tấn công web và nguyên lý hoạt động của mô hình phát hiện tấn công. Trình bày quá trình xử lý dữ liệu, đưa dữ liệu vào huấn luyện và phát hiện kiểm tra.
- Thử nghiệm mô hình phát hiện tấn công web cơ bản dựa trên học máy với các kịch bản cụ thể.

Hướng phát triển trong tương lai

- Do hạn chế về thời gian và khả năng, luận văn mới chỉ thử nghiệm mô hình trên 1 thuật toán học máy là Cây quyết định. Trong tương lai sẽ sử dụng các thuật toán khác trong quá trình huấn luyện và phát hiện, như Naive Bayes, Rừng ngẫu nhiên và SVM, từ đó tìm ra thuật toán tối ưu.
- Cập nhật thêm dữ liệu để phát hiện được các loại tấn công mới hiện nay cũng như cập nhật được cách thức tấn công mới trên các lỗ hổng cũ.

DANH MỤC CÁC TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Hoàng Xuân Dậu, Bài giảng An toàn ứng dụng web và cơ sở dữ liệu, Học viện Công nghệ bưu chính viễn thông, 2017.

[2] Hoàng Xuân Dậu, "Nghiên cứu, phát triển hệ thống phân tích vết truy cập dịch vụ cho phép phát hiện, cảnh báo hành vi bất thường và nguy cơ mất an toàn thông tin trong Chính phủ điện tử", Báo cáo đề tài KC.01.05/16-20, 2019.

[3] Từ Minh Phương, Giáo trình Nhập môn Trí Tuệ Nhân Tạo, Học viện Công nghệ bưu chính viễn thông, 2015.

Tiếng Anh

[4] Abhishek Kumar Baranwal (2012), Approaches to detect SQL injection and XSS in web applications, EECE 571B, Term Survey Paper, University of British Columbia, Canada, April 2012.

[5] Kemalis, K. and T. Tzouramanis (2008). SQL-IDS: A Specification-based Approach for SQLinjection Detection. SAC'08. Fortaleza, Ceará, Brazil, ACM (2008), pp. 2153-2158.

[6] P. Bisht, and V.N. Venkatakrishnan (2008), "XSS-GUARD: Precise dynamic prevention of Cross-Site Scripting Attacks," In Proceeding of 5th Conference on Detection of Intrusions and Malware & Vulnerability Assessment, LNCS 5137, 2008, pp. 23-43.

[7] Doyen Sahoo, Chenghao Liu, and Steven C.H. Hoi (2017), Malicious URL Detection using Machine Learning: A Survey, [Online] https://arxiv.org/abs/1701.07179, Mar 2017.

Trang WEB

[8] OWASP, Open Web Application Security Project, http://www.owasp.org, truy nhập tháng 1.2018.

[9] OWASP ModSecurity Core Rule Set, https://www.owasp.org/index.php/Category:

OWASP_ModSecurity_Core_Rule_Set_Project, truy nhập tháng 1.2018.

[10] https://ereka.vn/post/owasp-cong-bo-top-10-rui-ro-bao-mat-ung-dung-nam-2017

[11] https://m.bkav.com.vn/tin_tuc_noi_bat/-/chi_tiet/669034/tong-ket-an-ninh-mang-nam-2019-va-du-bao-

[12] https://quantrimang.com/lo-hong-bao-mat-nhung-hieu-biet-can-ban-93098

[13] https://machinelearningcoban.com/2017/08/08/nbc/

[14] https://ereka.vn/post/cach-xay-dung-mot-mo-hinh-hoc-may-machine-learning-model-5298271823815628220

[15] https://securitydaily.net/tan-cong-directory-traversal-la-gi/

[16] https://www.owasp.org/index.php/Command_Injection

[17] https://www.loganalyzer.net/log-analysis-tutorial/what-is-log-file.html

[18] https://quantrimang.com/tan-cong-kieu-sql-injection-va-cac-phong-chong-trong-asp-net-34905

[19] HTTP Param Dataset, https://github.com/Morzeux/HttpParamsDataset, truy nhập 12.2018.

[20] Hoàng Xuân Dậu và nhóm thực hiện tại Lab ATTT – Học viện Công nghệ BCVT. Đề tài "Nghiên cứu, phát triển hệ thống phân tích vết truy cập dịch vụ cho phép phát hiện, cảnh báo hành vi bất thường và nguy cơ mất an toàn thông tin trong Chính phủ điện tử", mã số KC.01.05/16-20. 2020.