ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

----------*----------

ĐỒ ÁN

TỐT NGHIỆP ĐẠI HỌC

NGÀNH CÔNG NGHỆ THÔNG TIN

ÁP DỤNG PHƯƠNG PHÁP HỌC MÁY TRONG PHÁT HIỆN VÀ NGĂN CHẶN TẤN CÔNG WEB

Sinh viên thực hiện: **Hoàng Phú Hoan**

Lớp CNTT1.02 – K58

Giảng viên hướng dẫn: TS **Nguyễn Hồng Quang**

HÀ NỘI, 05 – 2018

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Hoàng Phú Hoan

Điện thoại liên lạc: 0968756614 Email:hoangphuhoan2012@gmail.com

Lớp: CNTT1.02 Hệ đào tạo: Đại trà

Đồ án tốt nghiệp được thực hiện tại: Viện Công nghệ thông tin, trường Đại học Bách Khoa Hà Nội

Thời gian làm ĐATN: Từ ngày 1/3/2018 đến 30/5/2018

2. Mục đích nội dung của ĐATN

Trong đồ án này, tôi muốn thực hiện việc áp dụng công nghệ học máy trong việc phát hiện tấn công bảo mật nhằm vào ứng dụng web đồng thời xây dựng một hệ thống để thực hiện việc ngăn chặn tấn công web

3. Các nhiệm vụ cụ thể của ĐATN

Nhiệm vụ cụ thể của đồ án là:

- Tìm ra phương pháp học máy cho độ chính xác cao nhất đối với việc phát hiện tấn công web
- Áp dụng phương pháp học máy tìm được để xây dựng một hệ thống web application firewall cơ bản, bảo vệ ứng dụng web.

4. Lời cam đoan của sinh viên:

Tôi *Hoàng Phú Hoan* cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *Tiến sỹ Nguyễn Hồng Quang*

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

Hà Nội, ngày 29 tháng 5 năm 2018

Hoàng Phú Hoan

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo S

Hà Nội, ngày 29 tháng 5 năm 2018

Tiến sỹ Nguyễn Hồng Quang

TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Trong thế giới hiện đại ngày nay, ứng dụng web ngày một trở nên quan trọng và là một phần không thể thiếu của mạng internet. Hâu hết các thông tin trên internet đều được lưu trữ trên các website. Cũng chính vì vậy mà vấn đề về bảo mật web ngày càng trở thành một vấn đề được quan tâm. Hầu hết các ứng dụng web được bảo vệ hiện này thì đều được được sử dụng công nghệ dựa trên signature-based. Phương pháp này đã sớm bộc lộ nhiều hạn chế trong việc phát triển và duy trì cũng như phù hợp với đa dạng các tấn công ngay nay. Nhận thấy yêu cầu đó thì trong đồ án này tôi muốn tìm kiếm một phương pháp phát hiện tấn công web sử dụng học máy. Nó sẽ đem lại khả năng phát hiện tấn công vượt trội, dễ duy trì, phát triển và có khả năng đạt độ chính xác cao. Cùng với đó, tôi cũng muốn áp dụng phương pháp mà mình nghiên cứu được vào việc xây dựng một hệ thống Web Application Firewall với chức năng cơ bản. Để đảm bảo khả năng phát hiện với độ chính xác cao thì trong nghiên cứu của mình tôi sẽ thực hiện việc phát hiện tấn công dựa trên một tập dữ liệu đã được sử dụng nhiều trong các nghiên cứu trước cũng như chưa một lượng dữ liệu đa dạng. Các công việc được thực hiện trong đồ án này:

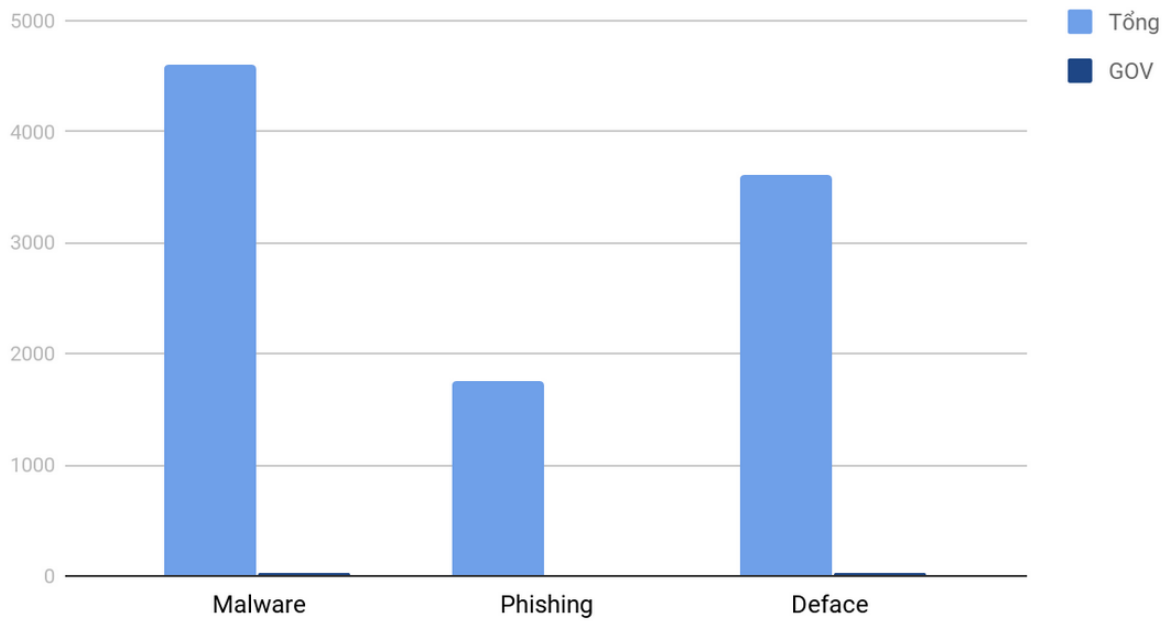
- Nghiên cứu về bảo mật ứng dụng web
- Nghiên cứu về các phương pháp đã được sử dụng để phát hiện tấn công web
- Đề xuất phương pháp phát hiện tấn công web dựa trên học máy của bản thân
- Thử nghiệm phương pháp của mình trên tập dữ liệu CSIC 2010
- Xây dựng hệ thống Web Application Firewall áp dụng phương pháp do mình đề xuất để bảo vệ hệ thống web thực

MỤC LỤC

I. Giới thiệu

1. Tổng quan và mục tiêu của đồ án

Ngành công nghệ thông tin càng phát triển thì đồng thời với nó, khả năng mất an toàn thông tin càng cao, đặc biệt là đối với các ứng dụng public rộng rãi đối với người dùng như một website thì khả năng trở thành mục tiêu tấn công lại càng lớn. Theo thống kê từ những công ty bảo mật lớn như Bkav, CMC thì hàng năm ở Việt Nam có hàng ngàn website bị tấn công, và trong năm 2017 có tới 40% số website ở Việt Nam tồn tại những lổ hổng nghiệm trọng, cho phép hacker có khả năng ăn cắp cơ sở dữ liệu thậm chí leo quyền, thực thi lệnh tùy ý đối với server. Nguy hiểm hơn, trong số những website tồn tại các lỗ hổng nghiêm trọng này thì có không ít là các website trọng yếu của các cơ quan nhà nước cũng như là website của các tập đoàn kinh tế lớn, quan trọng. Tiêu biểu có thể kể tới một số sự vụ xảy ra trong vài năm gần đây liên quan trực tiếp tới việc đảm bảo an toàn thông tin cho website như vụ website của Tổng công ty hàng không Việt Nam năm 2016 bị hack khiến thông tin của 411.000 hành khách bị phát tán trên mạng hay trên vụ rò rỉ thông tin của 163 triệu tài khoản ZingID của VNG, một trong những công ty internet lớn nhất Việt Nam diễn ra vào đầu năm 2018. Thực tế này cho thấy rằng rõ ràng vấn đề an ninh, bảo mật đối với website, bộ mặt của một tổ chức và cũng là nơi chứa nhiều thông tin quan trọng chưa được xem xét và đầu tư đúng mức, cũng như đặt ra một yêu cầu cấp thiết phải nâng cấp khả năng phòng chống tấn công cho các website.



Hình 1.1 Thống kê số lượng website bị tấn công trong 9 tháng đầu năm 2017 (http://securitybox.vn)

Để giúp các website có khả năng tự bảo vệ mình ở một mức nào đó mà không đòi hỏi chủ nhân website cần có kiến thức về bảo mật web, các công ty về an ninh mạng ở Việt Nam nói riêng và trên thế giới nói chung đều đưa ra các ứng dụng bảo vệ website tự động được gọi chung với tên WAF (Web Application Firewall). Tại Việt Nam, một số nhà cung cấp các sản phẩm dòng WAF có thể kể tới là Bkav, Viettel, Cystack… Tuy nhiên tại các doanh nghiệp này, công nghệ đang được sử dụng chủ yếu là chặn tấn công dựa trên việc phân tích http traffic và phát hiện tấn công dựa trên phương pháp signature-based. Điểm yếu của phương pháp này đó là nó chủ yếu tập trung vào các kiểu tấn công, từ đó các chuyên gia về bảo mật sẽ dựa vào các đặc điểm như từ khóa, pattern để tạo lên các bộ luật mới nhằm lọc ra và ngăn chặn các traffic thỏa mãn các điều kiện hay nói cách khác là các traffic giống với các đặc điểm của một tấn công sẽ bị ngăn chặn. Điều đó đòi hỏi cần có một đội kĩ sử bảo mật luôn luôn túc trực để liên tục cập nhật bộ luật để chống lại các tấn công mới được phát hiện, mà trong bối cảnh có đến cả trăm ngàn kiểu traffic tấn công khác nhau, trong đó lại có các hình thức tấn công phức tạp, rất khó viết luật thì phương pháp này sau một thời gian sẽ dẫn tới vấn đề là khó dùy trì, quản lý, tập luật càng lớn theo thời gian thì thời gian xử lý sẽ càng lớn gây ra một loạt các vấn đề hệ thống khác. Hơn thế nữa, phương pháp tiếp cận signature-based lại không có khả năng phát hiện đối với các lỗ hổng chưa được công bố hay còn gọi là zero-day, đây chính là phương pháp mà cách hacker mũ đen chuyên nghiệp thường dùng để tấn công vào các hệ thống quan trọng, được bảo vệ kĩ tuy nhiên lại các phương pháp bảo vệ này hầu hết lại chỉ có tác dụng đối với các dạng tấn công đã được công bố.

Vì vậy, để giải quyết các khó khăn trên đối với việc phát triển WAF, trong đồ án này, tôi muốn đề xuất ra một mô hình học máy đóng vai trò quyết định trong việc phát hiện tấn công web. Mô hình học máy này thay vì việc chạy theo các phương thức tấn công mới thì sẽ chỉ cần tập trung vào ứng dụng web mà nó đang bảo vệ, dựa trên các thuộc tính, đặc điểm của chính ứng dụng web đó để phát hiện ra các traffic độc hại. Qua đó dễ thấy được sự vượt trội của phương pháp phát hiện và ngăn chặn tấn công nhằm vào ứng dụng web dựa trên học máy ở việc dễ dàng bảo trì, mở rộng, không cần có đội ngũ theo dõi và cải tiến ngày đêm để cập nhật khả năng đánh chặn đối với các kiểu tấn công mới. Thay vào đó, ta chỉ việc cập nhật lại tập dữ liệu học trong trường hợp có sự thay đổi về nội dụng, cấu trúc của website, và việc này sẽ diễn ra nhanh chóng, dễ dàng vì hầu hết mọi công đoạn đều được thực hiện một cách tự động và không cần nhiều đến sự can thiệp của kĩ sư chuyên

trách. Ngoài ra, đối với cách tiếp cận này thì việc khai thác các lỗ hổng zero-day sẽ dễ dàng bị phát hiện, không khác gì so với các lỗ hổng thông thường.

2. Bố cục của đồ án

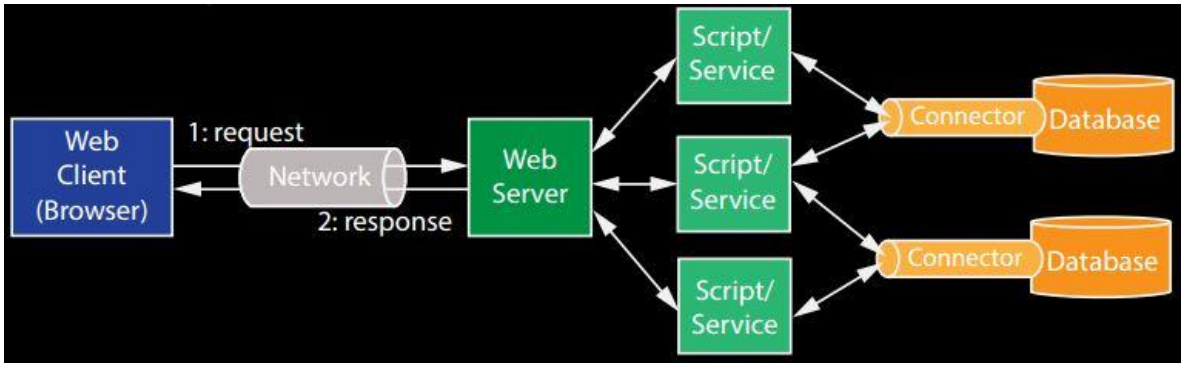
Trong đồ án này, tôi xin được trình bày các vấn đề chính như sau:

- **Tổng quan về ứng dụng và vấn đề bảo mật cho ứng dụng web**: Chương này đem đến cái nhìn tổng quan về ứng dụng web như mô hình, các thành phần, các thức hoạt động, cách thức liên kết giữa các thành phần đặc trưng thông thường trong một ứng dụng web. Ngoài ra, tôi cũng đề cập đến các vấn đề chính trong bảo mật web cũng như giới thiệu về WAF (Web Application Firewall) như một giải pháp được ưa chuộng trong việc triển khai hệ thống phát hiện và chống tấn công cho hệ thống website
- **Các nghiên cứu liên quan:** Trong chương này, ta sẽ cùng điểm qua các hướng tiếp cận cả theo phương pháp truyền thống lẫn các phương pháp sử dụng học máy đã được nghiên cứu và công bố trong các báo cáo trong những năm gần đây để thấy được những kết quả và tiến bộ của những nghiên cứu đã được báo công bố trong lĩnh vực phát hiện tấn công bảo mật nhằm vào website.
- **Phương pháp đề xuất và kết quả thu được:** Đây là chương quan nhất trong bản báo cáo đồ án tốt nghiệp này, nó sẽ tập trung phân tích một cách chi tiết nhất vào mô hình học máy do tôi đề xuất, kết quả của mô hình đó và những cải tiến của nó so với các nghiên cứu đã được công bố trước đó khi thực hiện trên cùng tập dữ liệu, cũng như định hướng phát triển và cải tiến trong tương lai.
- **Mô hình triển khai thực tế:** Để chứng minh cho khả năng ứng dụng phương pháp học máy được đề xuất vào việc bảo vệ các website trong thực tế, trong chương này tôi xin đề xuất một hệ thống phát hiện và bảo vệ website theo thời gian thực cũng như thử áp dụng hệ thống này trong thực tế để bảo vệ một website thật trước các request tấn công. Do đây chỉ là một hệ thống ở mức thử nghiệm đơn giản, thực hiện các chức năng cốt lõi nên tôi cũng xin đưa ra một số định hướng phát triển trong tương lai để thực sự biến hệ thống này trở thành một lá chắn hiệu quả đối với các website phức tạp và chịu tải lớn.

II. Tổng quan về ứng dụng web và vấn đề bảo mật cho ứng dụng web

1. Ứng dụng web

Ứng dụng web từ lâu đã trở thành một thành phần cốt lõi trong mạng internet và là đối tượng tương tác chính của người dùng internet. Các ứng dụng web dù có quy mô khác nhau nhưng về cơ bản thì đều tuân theo một kiến trúc cơ bản.

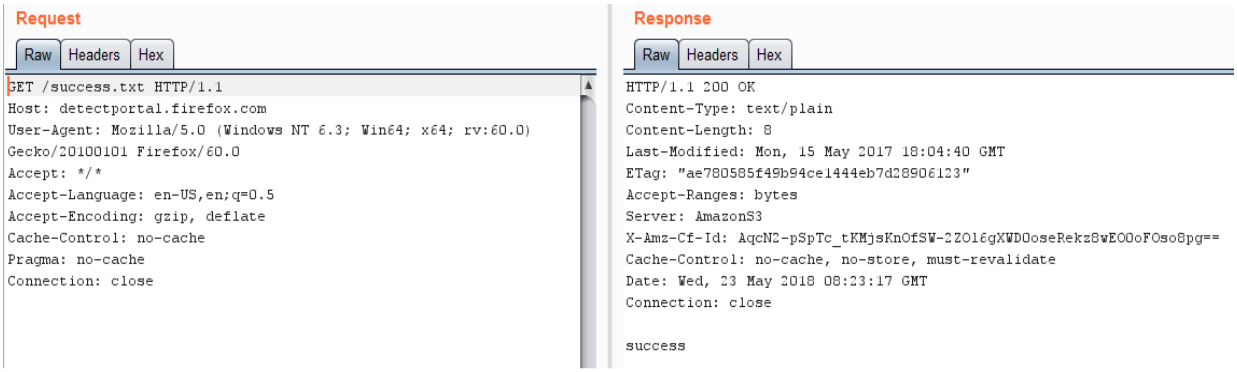


Hình 2.1 Kiến trúc của một ứng dụng web cơ bản (nguồn internet)

Hình 2 cho ta thấy kiến trúc cơ bản của một ứng dụng web bao gồm: webserver, các file service/script (ứng dụng web), cơ sở dữ liệu (database). Trong đó webserver đóng vai trò tiếp nhận và điều phối mỗi khí có request từ phía client. Các yêu cầu này sau khi được webserver tiếp nhận sẽ được chuyển đến service (thuộc ứng dụng web) tương ứng, tại đây ứng dụng web sẽ làm nhiệm vụ phân tích yêu cầu của request và trả về tài nguyên tương ứng cho client thông qua webserver. Cơ sở dữ liệu sẽ đóng vai trò lưu trữ, cung cấp thông tin cho ứng dụng web trong quá trình xử lý request. Các webserver 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… Còn cơ sở dữ liệu thì có một số cái tên quen thuộc đó là SQLServer, 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.

Client và Webserver liên lạc với nhau bằng giao thức HTTP hoặc HTTPS (một dạng mở rộng của HTTP với khả năng mã hóa dữ liệu trên đường truyền). HTTP là viết tắt của Hypertext Transfer Protocol, cho phép giao tiếp giữu Client và Server, là một stateless protocol, hoạt động dựa trên giao thức TCP/IP, mặc định thông qua

cổng 80 đối với HTTP và 443 đối với HTTPS. Giao thức này bao gồm hai thành phần chính là http request (gửi từ client lên server) và http response (phản hồi từ server cho client).



Hình 2.2 Cấu trúc cơ bản của HTTP request và HTTP response

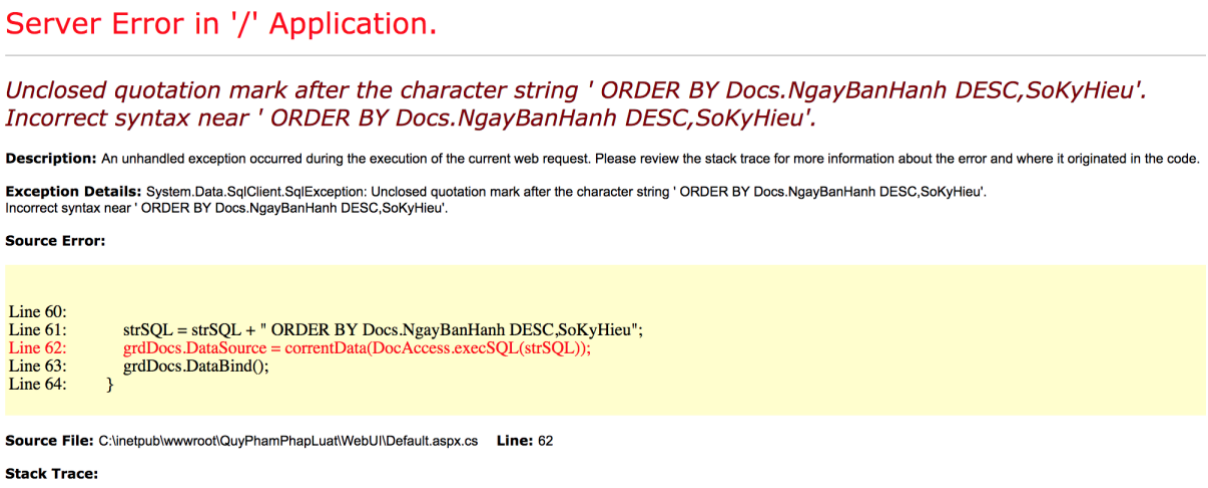
2. Các vấn đề trong bảo mật web

Vấn đề bảo mật website đã được cân nhắc và cảnh báo từ những ngày đầu khi mà những ứng dụng web đầu tiên có mặt trên internet. Có rất nhiều dạng tấn công web khác nhau, dưới đây là danh sách các loại lỗ hổng web được đánh giá là có mức độ nguy hiểm cao nhất đối với một ứng dụng web được thống kê bởi OWASP [16] - tổ chức phi lợi nhuận nổi tiếng thế giới hoạt động với mục đích phổ biến những kiến thức về bảo mật web, cũng như phương pháp phòng chống cho người dùng, tổ chức, doanh nghiệp trên khắp thế giới:

- **Tồn tại các file cũ:** Các files cũ không được sử dụng nữa nhưng vẫn còn lưu lại trên server có thể vô tình tiết lộ nhiều thông tin nhạy cảm nếu để hacker truy cập được
- **File mặc đinh:** Các file mặc định trên server là nới hacker có thể dễ dàng truy cập và tìm kiếm các thông tin nhạy cảm
- **Lộ mã nguồn website:** Việc lộ lọt này cho phép hacker nắm được mã nguồn website và thực hiện tìm kiếm các lỗ hổng nghiêm trọng dựa trên đó
- **Các HTTP method nguy hiểm:** Một số method http có thể cho phép thực hiện việc chỉnh sửa, xóa file trên server như PUT, DELETE
- **Chèn CRLF:** Lỗ hổng cho phép chèn các kí tự đặc biệt làm thay đổi cấu trúc gói tin hoặc thậm chí thực thi câu lệnh tùy ý
- **Không hạn chế truy cập:** Việc không hạn chế truy cập vào các tài nguyên quan trọng như trang quản trị sẽ là một kẽ hở lớn tạo cơ hội cho hacker truy cập sâu vào hệ thống.

- **Không kiểm tra dữ liệu người dùng:** Đây là một lỗi nghiệm trọng cho phép hacker tùy ý thay đổi dữ liệu truyền lên server, tạo cơ hội cho hacker khai thác nhiều loại lỗ hổng nghiêm trọng.
- **Chèn câu lệnh tùy ý:** Ứng dụng web truyền trực tiếp dữ liệu người dùng cung cấp và các lời gọi lệnh, lợi dụng điều này hacker có thể thực hiện lệnh tùy ý với quyền của ứng dụng web
- **Cross site scripting:** Đây là dạng lỗ hổng xuất hiện khi ứng dụng web không kiểm tra dữ liệu mà người dùng truyền lên, sau đó lại dùng chính những dữ liệu này để hiển thị lên website. Lợi dụng điều đó, hacker có thể thay dữ liệu thường bằng những script độc hại, chạy script này trên website từ đó có thể ăn cắp phiên làm việc của người dùng, hay deface website
- **SQL injection:** Lổ hổng này xảy ra khi dữ liệu của người dùng được truyền vào các câu truy vấn cơ sở dữ liệu mà không được kiểm tra, lợi dụng điều này hacker có thể thay đổi dữ liệu truyền lên bằng những từ khóa SQL để thực hiện các thao tác tùy ý đối với cơ sở dữ liệu của website.
- **Buffer overflow:** Đây là lỗ hổng nhằm vào các thành phần của webserver có cách xử lý đặc biệt đối với các dữ liệu có độ dài vượt quá giới nào đó, lỗ hổng có thể cho phép hacker làm gián đoạn hoạt động của thành phần tương ứng, cao hơn là có thể thực hiện câu lệnh tùy ý trên server
- **Broken authentication and session management:** Đây là lỗ hổng xảy ra khi những dữ liệu quan trọng như định danh người dùng, session token không được bảo vệ một cách đúng đắn, hacker có thể dễ dàng ăn cắp các thông tin này, qua đó đăng nhập vào ứng dụng web với quyền của người dùng.
- **Broken access control:** Đây là lỗ hổng khai thác việc ứng dụng web không có cơ chế phân quyền đúng đắn, khiến cho hacker có thể tùy ý truy cập vào dữ liệu của người dùng khác hay thậm chí là thực hiện các chức năng mà đáng nhẽ ra chỉ được cấp cho quản trị viên.
- **Remote admin flow:** Rất nhiều website có cung cấp các trang quản trị dưới dạng một thành phần của website, nếu các chức năng cửa trang này không được bảo vệ chu đáo thì rất có thể nó sẽ tạo điều kiện thuận lợi cho hoạt động tấn công của hacker
- **Web application and server misconfiguration:** Việc rà soát và đảm bảo cấu hình chuẩn cho một ứng dụng là rất quan trọng, nhất là khi một ứng dụng web có thể gồm nhiều thành phần, ứng dụng, nền tảng khác nhau. Chỉ cần một trong số đó có cấu hình yếu hay lỗi thì sẽ ngay lập tức tạo điều kiện cho hacker xâm nhập vào toàn bộ hệ thống.

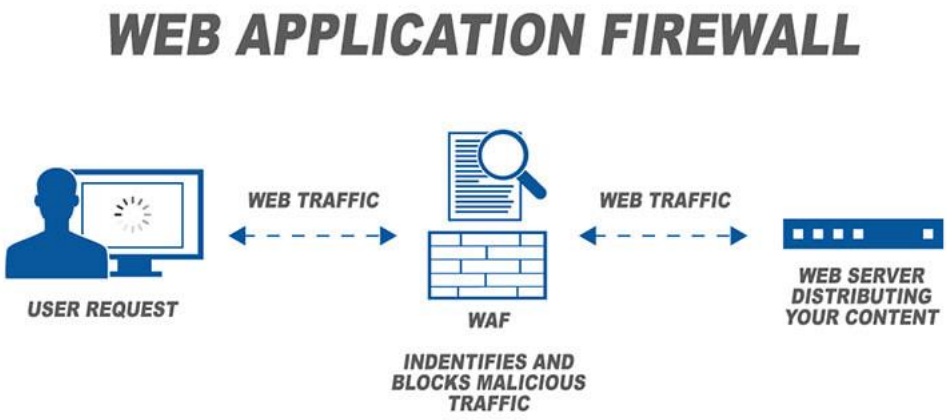
- **Malicious File Execution:** Đây là lỗ hổng cho phép hacker tùy ý thực thi một file chứa mã độc trên server. Lỗ hổng này thường được thực hiện khi hacker có khả năng tải một file tùy ý lên server.
- **Insecure Direct Object Reference:** Đây là dạng lỗ hổng khi mà việc tham chiếu đến một đối tượng trên website bị phô bày với người thông qua các tham số được truyền lên ví dụ như id, user, file … Hacker sẽ thay đổi các tham số này nhờ đó mà thực hiện việc tham chiếu đến một đối tượng khác tùy ý, cao hơn là có thể thực hiện các thao tác chỉnh sửa trên đối tượng đó, thậm chí là thực thi mã.
- **Information Leakage and Improper Error Handling:** Thông tin về ứng dụng web có thể bị lộ lọt thông qua nhiều con đường như response header, mã nguồn, lỗi cấu hình. Ngoài ra việc xử lý đối với các request lỗi cũng rất quan trọng vì các thông báo lỗi này nếu không được xử lý đúng có thể sẽ dẫn tới lộ lọt các thông tin về cấu trúc thư mục trên server, hệ điều hành, nền tảng lập trình, ngôn ngữ, phiên bản webserver, thậm chí là làm lộ một phần của mã nguồn cho hacker.



Hình 2.3 Lộ lọt dữ liệu khi ứng dụng web không xử lý lỗi đúng cách

3. Giới thiệu về Web Application Firewall

Những năm gần đây, WAF trở thành một thành phần không thể thiếu đối với các website quan trọng. Một cách đơn giản nếu như firewall bảo vệ website thông qua việc phân tách gói tin ở tầng 3 và tầng 4 trong mô hình mạng thì WAF lại làm việc trên tầng ứng dụng, bảo vệ website dựa vào việc phân tích http request để xác định request tấn công và có cách xử lý phù hợp.



Hình 2.4 Mô hình cơ bản của WAF (nguồn: internet)

WAF thường hoạt động theo một trong hai cơ chế: chủ động hoặc bị động. Mô hình chủ động chỉ cho phép traffic hợp lệ được phép đi qua và chặn tất cả các traffic còn lại. Trong khi đó, mô hình bị động sẽ cho phép tất cả traffic đi qua nó và chỉ chặn đối với các trường hợp được xác định là traffic độc hại. Về mô hình triển khai trong mạng thì WAF thường được triển khai theo một số cách sau: Reverse Proxy, Transparent Proxy, Host-Based. Trong đó mô hình Reverse Proxy là phổ biến hơn cả nhờ các khả năng đa dạng mà nó đem lại.

III. Các nghiên cứu liên quan

1. Sử dụng các phương pháp phân loại dựa trên học máy cơ bản

Bài toán phát hiện tấn công web đã có một một số nghiên cứu trong những năm gần đây. Phần lớn trong số chúng áp dụng các phương pháp phân lớp nổi tiếng trong học máy như Logistic Regression, Support Vector Machine, Decision Tree, XGBoost, Naïve Bayes…

(Rafal Kozik et al., 2016) [1] tiến hành thử nghiêm sử dụng nhiều phương pháp phân loại khác nhau bao gồm Naïve Bayes, AdaBoost, PART, J48 để phân loại request độc hại trên cùng tập dữ liệu và so sánh kết quả giữa các phương pháp. Tập dữ liệu được sử dụng là tập CSIC 2010, tập dữ liệu bao gồm các request bình thường cùng với các request tấn công đã được gán nhãn sẵn, các request này được tạo lên bằng cách mô phỏng request thường và request tấn công đến một trang web thương mai điện thử thật sự. Nhóm tác giả tập trung chủ yếu vào việc phân tích dữ liệu có trong câu truy vấn, địa chỉ truy cập và phương thức truy cập. Các dữ liệu này sẽ được vector hóa sử dụng thuật toán dựa trên phương thức nén Lempel-Zic-Welch (LZW). Sau đó để đánh giá hiệu năng của từng mô hình học máy, tác giả sử dụng phương pháp 10-folds để tính toán độ chính xác trung bình của mỗi mô hình. Kết quả cho thấy J48, một phương pháp cải tiến của mô hình Decision Tree đạt hiệu quả cao nhất qua hai thông số đánh giá là Detection Rate (TP rate) và False Positive Rate (FP rate). Kết quả khá hứa hẹn với độ chính xác còn cao hơn hơn cả một báo cáo do chính tác giả của bộ dứ liệu CSIC công bố vào năm 2016 theo như so sánh của tác giả.

(Sara Althubiti et al., 2017) [2] lại có một cách tiếp cận độc đáo khác, thay vì tự mình trích rút các thuộc tính thì nhóm tác giả sẽ sử dụng lại các thuộc tính của một bài báo trước đó của (Nguyen et al., 2011) [3], tuy nhiên trong nghiên cứu của mình nhóm tác giả sẽ sử dụng phương pháp Feature Selection, một quá trình nhằm mục đích chọn ra các thuộc tính liên quan nhất và sử dụng những thuộc tính đó để tiến hành việc phân loại. Nhóm tác giả sử dụng phần mềm Weka để thực hiện quá trình này và thu gọn lại còn năm thuộc tính được sử dụng. Năm thuộc tính này sẽ được sử dụng bởi nhiều mô hình học máy khác nhau gồm Random Forest, Logistic Regression, J48, Ada Boost, SGDc, Naïve Bayes và cùng tiến hành phân tích trên tập dữ liệu CSIC 2010. Dựa trên các thông số Precison, Recall, F-Measure, TP rate, FP rate, bảng so sánh cho thấy các kết quả mà tác giả đạt được hoàn toàn vượt trội

so với các bài báo trước đó với độ chính xác gần như tuyệt đối. Bài báo cho thấy được sự hiệu quả của các phương pháp lựa chọn thuộc tính trong việc giúp cho mô hình học máy hoạt động một cách hiệu quả hơn.

Để đưa ra một một cái nhìn tổng quan cho việc áp dụng Generic Feature Selection (GeFS) trong bài toán phát hiện tấn công, (Nguyen et al., 2011) [3] đã sử dụng hai ví dụ trong phương pháp GeFS là The Correlation Feature Selection (CFS) và The minimal- Redundancy- Maximal- Relevance (mRMR). Trong đó CFS được sử dụng để định lượng mức liên hệ giữa các thuộc tính theo tương quan tuyến tính, còn mRMR sẽ đánh giá mối liên hệ phi tuyến tính giữa các thuộc tính. Hai phương pháp này sẽ được dùng để lựa chọn một tập các thuộc tính rút gọn từ tập 30 thuộc tính cho trước, đều là các thuộc tính được tác giả đánh giá là liên quan tới quá trình phát hiện tấn công. Tập dữ liệu được sử dụng trong thí nghiệm này bao gồm hai tập dữ liệu là CSIC 2010 và ECML/PKDD 2007. Sau quá trình chọn thuộc tính theo hai phương pháp kể trên tác giả tiếp tục đưa bộ dữ liệu vào nhiều mô hình học máy khác nhau để có được đánh giá tốt nhất. Kết quả thí nghiệm cho thấy cả hai phương pháp trích chọn thuộc tính đều không tạo được sự khác biệt lớn, thậm chí còn cho kết quả thấp hơn so với việc dùng bộ dữ liệu với 30 thuộc tính ban đầu. Tuy kết quả phân loại là khá tốt đối với tập dữ liệu ECML/PKDD 2007, nhưng tập dữ liệu này lại chỉ chứa các kiểu tấn công cơ bản dễ nhận diện như SQL injection, Cross-Site Scripting, LDAP Injection, XPATH Injection, Path traversal, Command Injection, SSI chứ chưa thể dùng để đánh giá khả năng phát hiện tấn công web nói chung.

(EIEI HAN et al., 2015) [4] đã thử phân loại tấn công web bằng cách kết hợp nhiều mô hình học máy với nhau. Trong đó, tác giả đã thử kết hợp giữa phương pháp K-means và ID3 decision tree, trong quá trình training K-means sẽ được áp dụng đầu tiên tiếp đó là ID3 decision tree, sự kết hợp này được tác giả kì vọng là sẽ bù đắp cho nhưng hạn chế của mỗi phương pháp khi được sử dụng riêng biệt. Để chứng minh cho giả thiết của mình thì tác giả đã so sánh phương pháp trên với một mô hình học máy khác được đánh giá là một bước tiến so với những hướng tiếp cận kiểu Decision Tree cũ đó là Random Forest, ngoài ra tác giả cũng tiến hành áp dụng hai mô hình là ID3-decision tree và K-means một cách tách biệt. Các phương pháp này sẽ được tiến hành đánh giá, so sánh khi cùng tiến hành việc phân tích trên tập dữ liệu CSIC 2010. Kết quả sau quá trình thử nghiệm cho thấy phương pháp Random Forest cho kết quả ấn tượng nhất, tuy nhiên phương pháp kết hợp hai mô hình do tác giả đề xuất cũng cho thấy sự hiệu quả so với việc áp dụng từng phương pháp riêng rẽ.

Ngoài việc kết hợp nhiều phương pháp học máy khác nhau để nâng cao hiệu quả trong quá trình phát hiện tấn công web, thì cũng có các nghiên cứu kết hợp giữa phương pháp học máy và phương pháp phát hiện tấn công dựa trên signature-based, đó là trường hợp nghiên cứu của (Melody Moh et al., 2016) [5], trong nghiên cứu này tác giả thư so sánh việc phát hiện tấn công theo phương pháp truyền thống signature-based hay còn gọi là pattern-matching với việc áp dụng những phương pháp theo hướng tiếp cận học máy như Naïve Bayes hay Bayes Net, tiếp theo đó là việc thử nghiệm kết hợp giữa hai cách làm này, cụ thể là áp dụng một phương pháp trước để phân loại tấn công trước và phương pháp còn lại được sử dụng sau để phân loại các trường hợp bị bỏ sót hay không bị phát hiện ở giai đoạn đầu. Nghiên cứu được tiến hành trên tập dữ liệu 12000 mẫu do tác giả tạo tự động dùng Log4j Framework. Kết quả thí nghiệm cho thấy việc kết hợp giữa hai phương pháp cho kết quả tốt nhất. Điểm hạn chế là trong thí nghiệm này, dữ liệu tấn công đều được mô phỏng theo tấn công SQL Injection, nếu áp dụng cho đa dạng các loại tấn công web thì chắc chắn việc phân loại dựa trên pattern-matching sẽ gặp nhiều khó khăn.

Gần đây, (Shailendra et al, 2017) [6] cũng đã công bố một nghiên cứu về phát hiện tấn công XSS trên mạng xã hội dựa trên các phương pháp phân lớp của học máy. Theo báo cáo này thì tác giả sẽ chủ yếu khai thác dữ liệu từ ba nhóm thuộc tính đó là thuộc tính của url, thuộc tính của thẻ HTML, thuộc tính của nền tảng mạng xã hội. Mỗi nhóm sẽ bao gồm nhiều thuộc tính khác nhau, chúng sẽ được sử dụng trong quá trình phân loại tấn công được tiến hành bằng mười phương pháp học máy khác nhau bao gồm RandomForest, ADTree, RandomSubspace, Decorate, AdaBoost, JRip, NaiveBayes, Support Vector Machine, Logistic Regression, k-Nearest Neighbors. Tập dữ liệu dùng để thử nghiệm sẽ bao gồm các website thường trong cơ sở dữ liệu Alexa và các website đã có dấu hiệu bị tấn công XSS trong cơ sở dữ liệu XSSed. Sử dụng phương pháp 10-folds để đánh giá đối với từng mô hình học máy cho thấy phương pháp Random Forest cho kết quả tốt nhất ở cả bốn thông số là Precision, Recall, Fmeasure, False positive rate. Cũng giống như một số báo cáo khác đã được đề cập trước đó, tuy rằng cho một kết quả khá tốt trong việc phát hiện các website đang bị dính lỗi XSS tuy nhiên phương pháp trong báo cáo này vẫn chưa thể dùng để áp dụng rộng rãi cho nhiều chủng loại tấn công khác nhau trong thực tế.

2. Sử dụng deep learning

Ngoài các phương pháp học máy phân lớp thông thường vẫn được ưu chuộng sử dụng trong các bài toán phân loại như Logistic Regression, Support Vector

Machine, Decision Tree, Random Forest… thì các nhà nghiên cứu cũng đã từng tìm kiếm giải pháp cho việc phát hiện tấn công web theo hướng tiếp cận deep learning, sử dụng mô hình mạng neural sâu cho việc phân loại request.

(D. Atienza et al., 2015) [7] từng thử áp dụng một số phương pháp mô tả dữ liệu trực quan như Principal Component Analysis (PCA), Cooperative Maximum Likelihood Hebbian Learning (CMLHL), Self-Organizinf Maps cho bài toán phát hiên tấn công nhằm vào ứng dụng web. Phương pháp PCA được sử dụng cho những bài toán mà số lượng biến số hay là thuộc tính lớn, gây cản trở cho quá trình tính toán cũng như khó có thể có đánh giá trực quan dữ liệu ví dụ, phương pháp này sẽ chiếu dữ liệu đa chiều nên một không gian có cơ sở trực giao, tức là nếu ta xem mỗi cơ sở trong không gian mới là một biến thì hình ảnh của dữ liệu gốc trong không gian mới này sẽ được biểu diễn thông qua các biến độc lập mà không làm mất đi các thông tin giá trị. Nhờ phương pháp này mà kích thước của dữ liệu được giảm, số chiều của dữ liệu giảm dồng nghĩa với việc ta có khả năng dụng các đồ thị trực quan để mô tả dữ liệu. Phương pháp CMLHL là một trường hợp mở rộng của mô hình Maximum Likehood Hebbian Learning dựa trên phương pháp Exploration Projection Pursuit. Phương pháp thông kê của EPP được thiết kế để giải quyết các bài toán phức tạp trong việc xác định cấu trúc của dữ liệu nhiều chiều và chiếu nó vào một không gian với số lượng chiều thấp hơn nhằm mục địch có thể đánh giá được dữ liệu một cách trực quan. Cuối cùng là phương pháp SOM, tương tự như hai phương pháp trên, với mục đích là dễ dàng "visualization" dữ liệu trong một không gian tương ứng với số chiều ít hơn, nhưng lại vẫn phản ánh được các thông tin và tính chất của tập dữ liệu ban đầu. Cả ba phương pháp trên đều được các tác giả thử nghiệm trên tập dữ liệu CSIC 2010, tập dữ liệu bao gồm các request bình thường cũng với các request tấn công đã được gán nhãn sẵn, các request này được tạo lên bằng cách mô phỏng request thường và request tấn công đến một trang web thương mai điện thử thật sự. Trước khi tiến hành phân tích thì các tác giả cũng đã có các bước tiền xử lý đối với dữ liệu ban đầu nhằm mục đích loại bỏ đi các trường dữ liệu được coi là không cần thiết, các bản ghi trùng nhau, các dữ liệu cuối cùng được dữ lại để phân tích bao gồm method, host, content-type, content-length, payload. Tất cả đều được biểu diễn dưới dạng số tương ứng với đặc điểm của từng dạng dữ liệu. Tuy nhiên sau quá trình thử nghiệm với từng phương pháp đươc nêu ở trên thì không có phương pháp nào có khả năng phân biệt rõ ràng giữa request thông thường và các request độc hại, điều này được tác giả giải thích là do quá trình tiền xử lý dữ liệu đã bỏ đi một phần các thông tin hữu ích, cần thiết cho quá trình phân loại.

Khác với cách tiếp cận sử dụng deep-learning theo hướng trực quan hóa dữ liệu của D. Atienza, (Yao Pan et al.,2017) [8] trong đã tiến hành đánh giá khả năng ứng dụng của mô hình End-to-End Deep Learning với đại diện là hai phương pháp PCA và autoencoder so với các phương pháp phân loại truyền thống như Logistic Regression (LR), Support Vector Machine (SVM) trong việc phát hiện tấn công web. Việc đánh giá độ chính xác của các phương pháp trên được thực hiện dựa trên dữ liệu của hai ứng dụng web về quản lý video và dich vụ nén dữ liệu. Trong đó dữ liệu tấn công được mô phỏng bằng một số hình thức tấn công cơ bản và phổ biến là SQL injection, XSS và Object deserialization. Các thuộc tính được trích chọn bao gồm thời gian xử lý request, tên của người dùng hệ thống, số lượng kí tự trong các tham số, số lượng domain có mặt trong truy vấn, số lượng kí tự trùng lặp, vector thuộc tính được tính dựa trên N-gram của câu truy vấn. Kết quả cho thấy phương pháp End-to-End Deep Learning sử dụng autoencoder cho kết quả tốt nhất dựa trên thông số F-score. Tuy nhiên kết quả này mới chỉ được kiểm chứng đối với ba dạng tấn công cơ bản ở trên cũng như chỉ một số dạng dữ liệu tấn công đặc trưng, trong khi payload tấn công trong thực tế thì lại không theo bất kì khuôn dạng cụ thể nào.

Cùng ý tưởng so sánh giữa phương pháp sử dụng deep learning và các phương pháp truyền thống như SVM trong việc phát hiện tấn công web, tuy nhiên (Farhan Douksieh et al.,2017) [9] lại chỉ tập trung vào việc phân tích các url độc hại. Trong nghiên cứu của mình, nhóm tác giả đã sử dụng mạng Convolutional Neural Networks (CNNs) thử nghiệm cùng với các thuộc tính được trích chọn từ hai phương pháp Word2Vec và Term Frequency-inverse Document Frequency Features (TFIDF). Mạng CNN hay còn được gọi là mạng neural tích chập là một mô hình mạng học sâu nổi tiếng được ứng dụng vào nhiều bài toán nhận dạng và cho độ chính xác cao. Nghiên cứu được thực hiện trên tập dữ liệu do nhóm tác giả tự thu thập và đã được gán nhãn thành hai lớp tương ứng độc hại và bình thường. Kết quả sau quá trình thử nghiệm cho thấy SVM và CNN là hai phương pháp đem lại kết quả phân loại chính xác nhất dựa trên precision, recall, f1-score. Đây là một cách làm đơn giản nhưng lại cho thấy độ chính xác cao, tuy nhiên do chủ yếu dựa trên việc phân tích thuộc tính của chuỗi nên phương pháp này không cho ta thấy được các đặc tính để phân biệt giữa một request tấn công và một request thông thường. Hơn nữa, dữ liệu mà nhóm tác giả thu thập là từ nhiều nguồn khác nhau, chưa phản ánh hết được sự đa dạng các kiểu tấn công web cũng như mới chỉ tập trung vào phân tích dữ liệu trên url.

Cuối cùng, để phát hiện một loại tấn công nguy hiểm và nổi tiếng bậc nhất đối với các ứng dụng web là SQL injection, (Asaad Moosa et al., 2010) [10] cũng đã tiến

hành thử nghiệm bằng việc sử dụng Artificial Neural Networks (ANNs) . Phương pháp này dựa trên cơ sở những quan sát cho thấy rằng đối với kiểu tấn công SQL injection thì thường một số kí tự đặc trưng sẽ có được sử dụng nhiều hơn các kí tự khác. Do vậy tác giải chia nhóm các kí tự và sử dụng những thuộc tính được trích rút từ những nhóm này làm đầu vào cho mạng neural. Với lý do là tại thời điểm làm báo cáo vẫn chưa có tập dữ liệu chuẩn nào dành cho việc thử nghiệm các phương pháp phát hiện SQL Injection nên tác giả đã tự tổng hợp các payload cho loại tấn công này để dùng làm tập thử nghiệm. Kết quả của nghiên cứu là khá khả quan khi khả năng phát hiện tấn công SQL injection đối của model rất cao. Một điểm hạn chế của phương pháp này theo như tác giả đã thừa nhận đó là tập dữ liệu dùng để học và đánh giá là nhỏ và chưa bao quát được các dạng tấn công SQL injection. Ngoài ra thì phương pháp này vẫn chỉ được thiết kế riêng cho SQL injection và chắc chắn không thể áp dụng mô hình tương tự đối với các dạng tấn công còn lại nếu chúng không có đặc điểm về phân bố kí tự giống như SQL injection. Tiếp đó, tác giả còn giới thiệu một phương pháp tiếp cận khác đó là dựa vào đặc điểm về "từ khóa" hay gặp trong các cuộc tấn công SQL injection, kết hợp với mô hình ANNs, phương pháp cho thấy một kết quả tốt tuy nhiên cũng giống như cách tiếp cận trước đó dựa vào phân bố kí tự, nó vẫn còn rất nhiều hạn chế, cũng như không phải là một giải pháp cho phát hiện tấn công web nói chung.

3. Các phương pháp khác

Song song với các nghiên cứu dựa trên học máy, nhiều phương pháp tiếp cận khác cũng được cộng đồng nghiên cứu an ninh mạng tiến hành thử nghiệm, một vài bản báo cáo sau đây sẽ cho ta thấy những phương pháp khá thú vị trong việc phát hiện tấn công web.

(Carmen Torrano-Gimenez et al.,2016) [11] đã thử sử dụng phương pháp phát hiện tấn công web dựa trên một file XML mô tả cho toàn bộ website. Theo đó, dựa vào dữ liệu mà chính website cần được bảo vệ cung cấp, tác giả sẽ xây dựng một file XML trong file này định nghĩa các thông tin về phương thức được sử dụng, giá trị các trường trong header được chấp nhận, các tham số được chấp nhận đối với url mà request muốn truy cập, các đặc điểm về giá trị mà các tham số đó phải tuân thủ như độ dài tối đa, số lượng chữ số tối đa, tập các kí tự được phép sử dụng… Các mô tả này sau đó sẽ được sử dụng như một chuỗi các điều kiện mà một request khi truy cập vào website tương ứng sẽ phải thỏa mãn, chỉ cần vi phạm một trong số những điều kiện này thì request đó sẽ được coi như request độc hại. Quá trình thử nghiệm của phương pháp này trên một website thương mại điện tử đã cho thấy kết quả phát

hiện tấn công khá tốt. Thử nghiệm cũng cho thấy càng nhiều dữ liệu thông thường được sử dụng để xây dựng file XML thì độ chính xác trong việc phát hiện tấn công càng cao. Tuy nhiên, như chính tác giả đã thừa nhận, hạn chế của phương pháp này nằm ở việc làm sao để xây dựng được file XML một cách chính xác và tự động. Vì bản thân việc lọc ra các traffic thường dã không phải là một việc dễ dàng, hơn thế nữa sẽ rất khó để xây dựng một file XML mô tả những đặc điểm thông thường cho một ứng dụng web khi ứng dụng web đó phức tạp hoặc ứng dụng web cho phép người dùng tạo ra những trang nhỏ cũng như tạo thêm những url để truy cập vào những trang đó.

Một cách tiếp cận khác sử dụng những dấu hiệu bất thường gắn liền với kiểu tấn công đã được (Vigna et al,.2005) [12] giới thiệu để chống lại tấn công SQL Injection. Trong báo cáo của mình tác giả đã vạch ra các dấu hiệu nhận biết tấn công SQL Injection và từ đó xây dựng nên các mô hình riêng biệt để phát hiện ra tấn công dựa trên các đặc điểm này. Các đặc điểm này dựa trên độ dài chuỗi, phân bố kí tự trong chuỗi, tiền tố, hậu tố đặc biệt, cấu trúc chuỗi, những giá trị đặc biệt, chưa từng được nhìn thấy. Tác giả sử dụng những request thông thường để giúp các mô hình của mình nhận diện được các dấu hiệu khác thường. Tuy nhiên kết quả thí nghiệm cho thấy rằng chỉ số False Alarm Rate khá cao đối với các trường hợp request mà hệ thống chưa từng nhận diện trong quá trình học.

Xa hơn nữa, bản báo cáo (Kruegel et al., 2003) cũng đã từng sử dụng phương pháp dựa trên những dấu hiệu bất thường để triển khai thử nghiệm mô hình phát hiện tấn công web tổng quát. Phương pháp này chủ yếu được xây dựng dựa trên việc phân tích các dấu hiệu bất thường trên thành phần url của request. Tác giả xây dựng một loạt các mô hình nhỏ, mỗi mô hình dựa trên một dấu hiệu nhận biết như độ dài của câu truy vấn, phân bố các kí tự trong câu truy vấn, cấu trúc câu truy vấn, các giá trị đặc biệt của các tham số, sự toàn vẹn các tham số trong câu truy vấn, thứ tự các tham số trong câu truy vấn. Để đánh giá một request có phải tấn công hay không thì request đó sẽ được đi qua lần lượt các mô hình này, ở mỗi một mô hình thì request này sẽ nhận được một mức điểm, mức điểm này sau đó được tổng hợp và so sánh với một ngưỡng để xác định xem request đó phải là request bất thường hay không. Ngưỡng này có thể điều chỉnh được để mô hình đạt kết quả tốt nhất. Nhóm tác giả đã áp dụng mô hình của mình lên một tập dữ liệu tự thu thập được với giả thiết rằng các dữ liệu đó đều là dữ liệu thường và đánh giá dựa tên False Positive Rate. Kết quả đạt được là khá tốt khi FP rate rất thấp tuy nhiên với tập dữ liệu lớn và được thu thập từ thực tế thì việc giả sử tập dữ liệu đó đều là dữ liệu sạch là không thỏa đáng.

IV. Phương pháp đề xuất và kết quả thu được

1. Tập dữ liệu

Với mục đích có một đánh giá và so sánh khách quan về độ chính xác của phương pháp đề xuất, trong đồ án của mình, tôi sẽ sử dụng một tập dữ liệu chuẩn đã được sử dụng trong nhiều nghiên cứu về phát hiện tấn công web trước đó cũng như một tập dữ liệu được công bố sau quá trình chuẩn hóa kĩ lưỡng bởi một viện nghiên cứu quốc tế. Đó là tập dữ liệu CSIC 2010 [14].

Tập dữ liệu CSIC 2010 được công bố bởi hội đồng nghiên cứu quốc gia Tây Ban Nha và cập nhật lần cuối vào năm 2012. Tập dữ liệu bao gồm hàng chục ngàn request được tạo một cách tự động bằng cách mô phỏng những request thật với mục tiêu là một trang web thương mai điện tử với các chức năng cơ bản như đăng kí, giỏ hàng…

Tập dữ liệu bao gồm 36000 request thông thường và hơn 25000 request độc hại đã được phân loại sẵn vào các file tương ứng. Vì là một ứng dụng web Tây Ban Nha nên tập kí tự được sử dụng là các kí tự Latin. Các request độc hại bao gồm đa dạng các kiểu tấn công như SQL injection, XSS, Local File Inclusion, CRLF injection, lộ lọt thông tin, buffer overflow… và được phân chia theo theo ba mục chính:

- **Tấn công tĩnh**: Gồm các request tới những file cũ, những file cấu hình, những file mặc định trên server
- **Tấn công động**: Gồm các request mô phỏng các kiểu tấn công nổi tiếng như SQL injection, CRLF, XSS …
- **Các request không hợp lệ**: Đây là các request không có ý định tấn công thực sự nhưng lại được sử dụng như một cách để thu thập dữ liệu từ phía website bằng cách cung cấp các kiểu dữ liệu khác với định dạng hợp lệ lên server (ví dụ cung cấp địa chỉ email không tuân theo định dạng chuẩn, hay là cung cấp username nhưng lại toàn các kí tự đặc biệt)

Cấu trúc của mỗi request trong tập dữ liệu bao gồm đầy đủ các thành phần của một http request như method, host, url, query, phiên bản http, User-Agent, Cookie, Connection…

```
POST http://localhost:8080/tienda1/publico/anadir.jsp HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; Konqueror/3.5; Linux) KHTML/3.5.8 (like Gecko)
Pragma: no-cache
Cache-control: no-cache
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Encoding: x-gzip, x-deflate, gzip, deflate
Accept-Charset: utf-8, utf-8;q=0.5, *;q=0.5
Accept-Language: en
Host: localhost:8080
Cookie: JSESSIONID=933185092E0B668B90676E0A2B0767AF
Content-Type: application/x-www-form-urlencoded
Connection: close
Content-Length: 68

id=3&nombre=Vino+Rioja&precio=100&cantidad=55&B1=A%F1adir+al+carrito

GET http://localhost:8080/tienda1/publico/autenticar.jsp?modo=entrar&login=choong&pwd=d1se3ci%F3n&remember=off&B1=Entra
User-Agent: Mozilla/5.0 (compatible; Konqueror/3.5; Linux) KHTML/3.5.8 (like Gecko)
Pragma: no-cache
Cache-control: no-cache
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Encoding: x-gzip, x-deflate, gzip, deflate
Accept-Charset: utf-8, utf-8;q=0.5, *;q=0.5
Accept-Language: en
Host: localhost:8080
Cookie: JSESSIONID=8FA18BA82C5336D03D3A8AFA3E68CBB0
Connection: close
```

Hình 4.1 Ví dụ về các request trong tập dữ liệu CSIC 2010

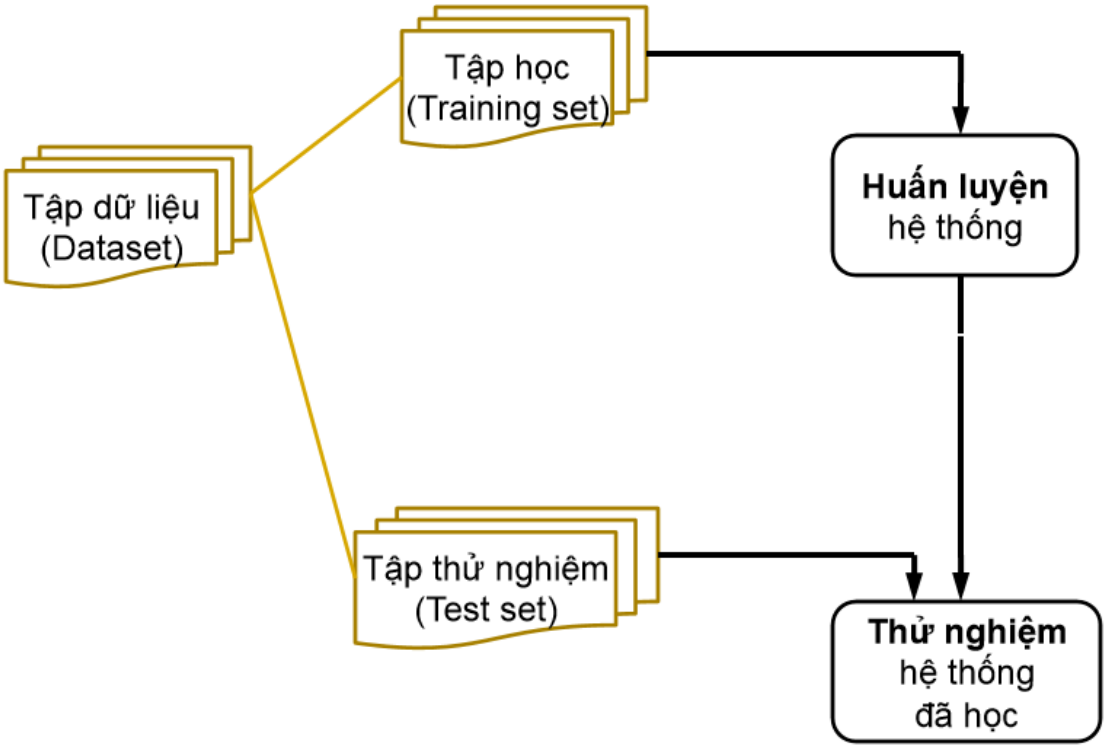
2. Một số khái niệm

a. Học máy

Học máy được định nghĩa là một lĩnh vực nghiên cứu trong trí tuệ nhân tạo, nhằm cung cấp cho máy tính khả năng học mà không cần được lập trình để hướng dẫn.

Máy tính được gọi là học từ kinh nghiệm E với tác vụ T và được đánh giá bởi độ đo P nếu máy tính khiến tác vụ T này cải thiện được độ chính xác P thông qua dữ liệu E cho trước.

Ví dụ: Trong bài toán phân loại email spam thì kinh nghiệm E là các email đã được phân loại trước đó, tác vụ T là việc phân loại email vào hai nhóm spam và không spam, còn độ chính xác P là độ chính xác trong việc phân loại email.



Hình 4.2 Quá trình học máy cơ bản [15]

Học máy được chia làm 2 dạng bài toán học cơ bản [15]:

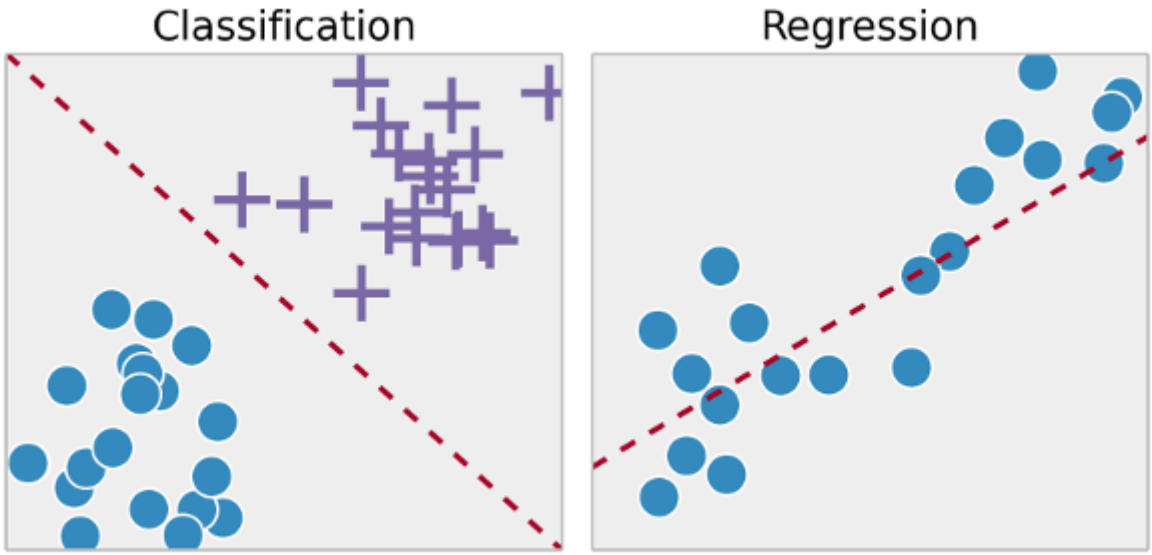
- Học có giám sát: là khi dữ liệu dùng để học có gắn liền với một đầu ra tương ứng. Học có giám sát được chia là hai dạng nhỏ hơn: Phân lớp và Hồi quy. Trong đó bài toán thuộc dạng phân lớp khi mà đầu ra thuộc một tập hữu hạn các giá trị, còn nếu đầu ra là một giá trị số thực thì bài toán thuộc dạng hồi quy. Ví dụ: bài toán phân loại email spam, bài toán phát hiện tấn công mạng …
- Học không giám sát: là khi dữ liệu đầu ra có thể là một cụm dữ liệu hoặc một cấu trúc ẩn. Ví dụ: Xác định biến động giá cổ phiếu, phát hiện các cộng đồng trên mạng xã hội..

b. Một số phương pháp phân lớp thường dùng trong học máy

- Phương pháp logistic: là một phương pháp hồi quy nhằm dự đoán giá trị đầu ra có giá trị rời rạc nằm trong khoảng từ 0 tới 1. Nó đo lường mối quan hệ giữa biến phụ thuộc phân loại và một hoặc nhiều biến độc lập bằng cách ước tính xác suất sử dụng một hàm logistic, là sự phân bố tích lũy logistic.
- Phương pháp decision tree: Sử dụng cấu trúc cây quyết định để xấp xỉ một hàm cần học. Mỗi nút trong cây quyết định biểu diễn một thuộc tính cần được kiểm tra giá trị đối với các ví dụ, mỗi nhánh của mỗi nút sẽ tương ứng

với một giá trị có thể của thuộc tính gắn với nút đó, mỗi nút lá sẽ tương ứng với một lớp. Một cây quyết định học được sẽ phân lớp đối với một ví dụ bằng cách duyệt cây từ nút gốc đến nút lá.

- Support Vector Machine: là phương pháp phân lớp tuyến tính, mục tiêu là xác định một siêu phẳng để phân tách hai lớp dữ liệu. Đây là một phương pháp tốt đối với những bài toán phân lớp trong không gian nhiều chiều, đặc biệt phù hợp với các bài toán phân lớp văn bản
- Naïve Bayes: là phương pháp phân lớp dựa trên phân loại xác suất đơn giản, áp dụng định lý Bayes với các giả định độc lập giữa các thuộc tính



Hình 4.3 Minh họa bài toán phân lớp trong học máy (nguồn: internet)

c. *Cách đánh giá mô hình học máy*

- Một số phương pháp đánh giá mô hình học máy [15]:
  - Hold-out: Chia toàn bộ tập dữ liệu làm hai tập con không giao nhau. Một tập dùng làm tập huấn luyện, một tập làm tập để đánh giá hiệu năng của hệ thống. Phù hợp khi tập dữ liệu lớn
  - Repeated hold-out: Sử dụng phương pháp hold-out nhiều lần một cách ngẫu nhiên
  - Cross-validation: Toàn bộ tập dữ liệu được chia làm k phần không giao nhau có kích thước xấp xỉ nhau. Tiến hành k lần lặp, mỗi lần lặp có một tập con được dùng làm tập kiểm thử và k-1 tập còn lại được dùng làm tập huấn luyện. Các giá trị lỗi được chia trung bình để thu được giá trị cuối cùng. Phương pháp này phù hợp với tập dữ liệu vừa và nhỏ.

- Các chỉ số đánh giá mô hình:
    - Precision: Precision đối với lớp i là tổng số các vụ dụ thuộc lớp i được phân loại chính xác chia cho tổng số các ví dụ được phân loại vào lớp i
    - Recall: Recall đối với lớp i là tổng số các ví dụ thuộc lớp i được phân loại chính xác chia cho tổng số ví dụ thuộc lớp i
    - F1: là tiêu chí kết hợp giữa Precision và Recall và được tính theo công thức sau:

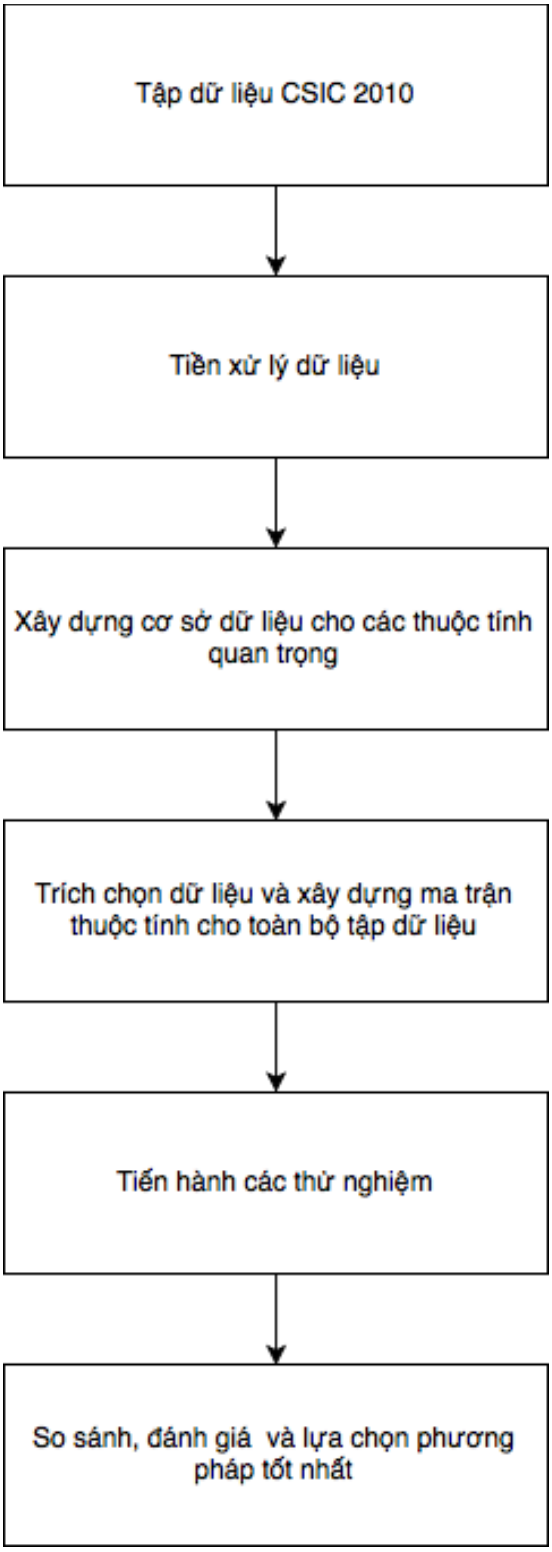
$$F_1 = \frac{2.Precision.Recall}{Precision + Recall} = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

Hình 4.4 Công thức tính F1 [15]

3. Trình bày phương pháp tiến hành và kết quả thu được

a. Tổng quan về phương pháp tiến hành

Sơ đồ sau đây mô tả một cách tổng quan phương pháp tiến hành các công đoạn chính để thực hiện đồ án:



Hình 4.5 Sơ đồ mô tả quá trình tiến hành nghiên cứu

Các bước thực hiện chính bao gồm:

- **Tiền xử lý dữ liệu:** Ở bước này, dữ liệu sẽ được loại bỏ đi các thành phần không cung cấp nhiều thông tin cho quá trình phân loại, cũng như thực hiện các thống kê đánh giá ban đầu về tập dữ liệu
- **Xây dựng cơ sở dữ liệu cho các thuộc tính quan trọng:** Các dữ liệu quan trọng, chứa các thông tin cần thiết cho quá trình phân loại sẽ được lưu vào cơ sở dữ liệu, tương tự như vậy, một số bảng dữ liệu cũng sẽ được tạo ra nhằm làm cở sở cho quá trình tính toán các thuộc tính của dữ liệu sau này.
- **Trích chọn dữ liệu và xây dựng ma trận thuộc tính:** Bước này sẽ tiến hành việc tính toán giá trị các thuộc tính sau đó lưu toàn bộ ma trận cùng với gán nhãn tương ứng ra một file riêng. Điều này sẽ tạo điều kiện thuận lợi cho việc tiến hành các thử nghiệm sau này
- **Tiến hành các thử nghiệm:** Ở bước này, tôi tiến hành rất nhiều các thử nghiệm khác nhau nhằm đánh giá một cách chính xác nhất các phương pháp tiếp cận khác nhau trong học máy đối với bài toán của chúng ta
- **So sánh, đánh giá, kết luận:** dựa vào các kết quả thu được từ các thí nghiệm thực hiên ở bước trước, tôi sẽ đưa ra các kết luận và chọn ra phương pháp phù hợp cũng như cho độ chính xác cao nhất trong việc phát hiện tấn công web.

b. Tiền xử lý đối với dữ liệu thô

Đây là một bước quan trọng và có ảnh hưởng tới toàn bộ quá trình xử lý phía sau này. Dựa vào các kiến thức về bảo mật web đã được nghiên cứu kĩ lưỡng trong quá trình học tập và làm việc, tôi đã trích chọn ra các dữ liêu quan trọng và loại bỏ toàn bộ những dữ liệu không mang lại thông tin cho quá trình đánh giá:

- Dữ liệu được giữ lại bao gồm:
    - **Method**: chứa thông tin về phương thức được sử dụng trong request. Trong trường hợp của tập dữ liệu này thì có ba phương thức được sử dụng đó là GET, POST, PUT. Nó có ý nghĩa quan trọng đối với cách mà server sẽ xử lý request đó.
    - **URI**: đây là dữ liệu cung cấp thông tin về tài nguyên được yêu cầu truy cập trên server. Các dữ liệu này phản ánh nhu cầu truy vấn thực sự của máy khách cũng như nó gắn liền với một tập xác định các dữ liệu sẽ được người dùng cung cấp cho server (nếu có)
    - **Payload:** đây là dữ liệu quan trọng nhất trong bộ dữ liệu. Nó chứa toàn bộ dữ liệu chính mà người dùng gửi lên server cũng như là dữ liệu chủ yếu trong việc phân tích, xử lý của server đối với request tương ứng.

- Các dữ liệu được loại bỏ:
    - Bao gồm: toàn bộ các trường dữ liệu trong phần http headers như Host, Protocol, User-Agent, Cache, Accept, Cookies, Connection...
    - Các dữ liệu trên hoặc là đều chứa một giá trị duy nhất đối với tất cả các request hoặc là chứa dữ liệu khác nhau đối với từng request theo một định dạng hoàn toàn giống nhau nên được coi là không có giá trị đối với quá trình xử lý.

Hình 4.6 Các thành phần dữ liệu khác nhau trong http request

c. Xây dựng cơ sở dữ liệu

Mục tiêu của việc xây dựng cơ sở dữ liệu là để tạo điều kiện thuận tiện cho việc tiến hành nhiều thí nghiệm, khi đó ta không phải chạy lại việc trích rút dữ liệu từ đầu. Ngoài ra việc lưu dữ liệu trong cơ sở dữ liệu rất hữu ích trong trường hợp ta muốn thực hiện các thao tác liên quan tới thống kê, tính toán đơn giản ví dụ như tìm kiếm các đoạn payload cùng liên quan tới một url, hay tính trung bình độ dài của một dạng payload… Các dữ liệu cơ sở được sử dụng trong quá trình tính toán vectơ thuộc tính cũng được tôi lưu trong các bảng riêng biệt trong trường hợp thuộc tính phức tạp cần nhiều bước xử lý.

Dưới đây là các bảng dữ liệu được sử dụng:

- Mal_rqs: lưu các thông tin về request tấn công
    - Id: số thứ tự của request
    - Method: phương thức được sử dụng
    - url: đường dẫn

- payload: dữ liệu được truyền lên server
- type: phân loại request (1 cho request tấn công)

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
method	varchar(20)	NO		NULL	
url	varchar(255)	NO		NULL	
payload	text	YES		NULL	
type	tinyint(4)	YES		NULL	

Hình 4.7 Mô tả bảng dữ liệu mal_rqs

- Normal_rqs: lưu các thông tin request thường
  - Id: số thứ tự của request
  - Method: phương thức được sử dụng
  - url: đường dẫn
  - payload: dữ liệu được truyền lên server
  - type: phân loại request (0 đối với request thường)

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
method	varchar(20)	NO		NULL	
url	varchar(255)	NO		NULL	
payload	text	YES		NULL	
type	tinyint(4)	YES		NULL	

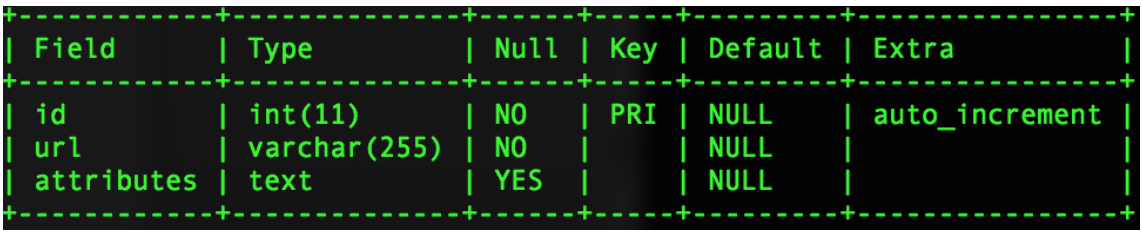
Hình 4.8 Mô tả bảng dữ liệu normal_rqs

- set_values_attribute: lưu trữ các giá trị hữu hạn của các trường dữ liệu có thể được gửi lên tương ứng với đường dẫn (nếu có)
  - id: số thứ tự
  - url: đường dẫn tương ứng
  - key_value: cặp giá trị và trường dữ liệu tương ứng (ví dụ: username=hoan)

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
url	varchar(255)	NO		NULL	
key_value	text	YES		NULL	

Hình 4.9 Mô tả bảng dữ liệu set_values_attribute

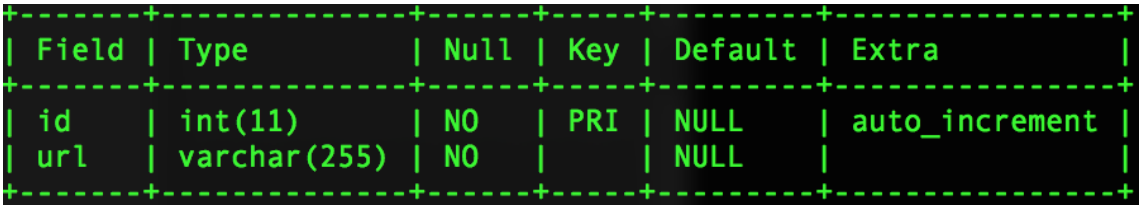
- subset_attributes: lưu trữ các định dạng dữ liệu được gửi lên theo các bộ tham số cụ thể, dùng để đối chiếu với dữ liệu gửi lên sau này nhằm tính toán các thuộc tính liên quan tới payload.
    - Id: số thứ tự
    - url: đường dẫn tương ứng
    - attributes: các bộ tham số tương ứng



Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
url	varchar(255)	NO		NULL	
attributes	text	YES		NULL	

Hình 4.10 Mô tả bảng dữ liệu subset_attributes

- urls: lưu trữ toàn bộ các đường dẫn tồn tại trên website
    - id: số thứ tự
    - url: đường dẫn



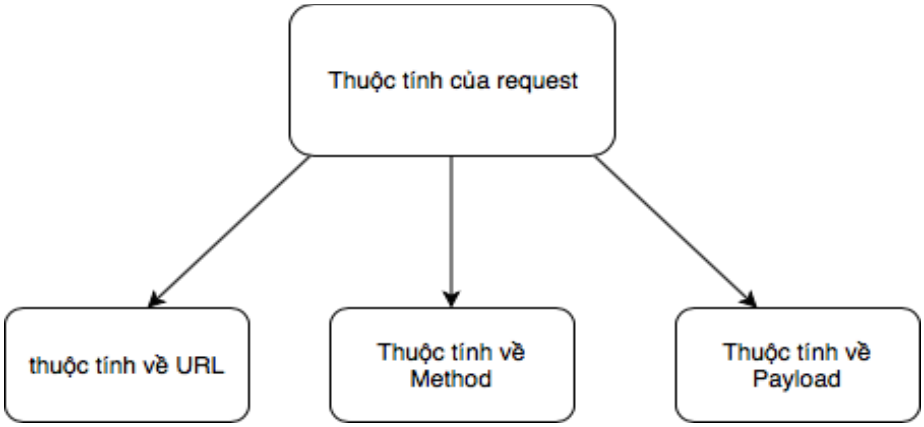
Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
url	varchar(255)	NO		NULL	

Hình 4.11 Mô tả bảng dữ liệu urls

d. Mô tả phương pháp trích chọn dữ liệu

Dựa vào các nghiên cứu về bảo mật web, tôi chia nhóm các dữ liệu cần thiết cho quá trình phát hiện tấn công vào ba nhóm lớn bao gồm:

- URL: Bao gồm các đặc tính liên quan tới URL
- Method: Bao gồm các đặc tính liên quan tới phương thức gửi dữ liệu lên server
- Payload: Bao gồm các đặc tính liên quan tới dữ liệu được gửi lên server



Hình 4.12 Các nhóm thuộc tính của request

Các thuộc tính về URL

- **URL không tồn tại**

Nếu là một người dùng thông thường thì việc truy cập vào một địa chỉ url không tồn tại là một việc rất ít khi có thể xảy ra và nếu có thì cũng là do tính năng đó không còn được hỗ trợ từ phía ứng dụng web (ít có thể xảy ra nếu ứng dụng web đó thực sự được đầu tư). Tuy nhiên, đối với hacker thì việc truy cập vào các url không tồn tại lại là một trong các bước để thu thập thông tin của server. Bằng cách cố gắng truy cập vào các url không tồn tại, các file backup, file default, admin-site, hacker có thể thu thập được các thông tin về hệ điều hành, cấu hình, phiên bản web-server hoặc thậm chí là mã nguồn của website. Để khai thác đặc điểm này, ta sẽ tạo ra một database bao gồm tất cả các url của website dựa vào việc phân tích tập dữ liệu ban đầu và tiến hành so sánh các url trong request đến với tập này để xác định xem url trên có tồn tại hay là không.

Nếu url không tồn tại thì giá trị của thuộc tính là 1, ngược là thì giá trị sẽ là 0.

Ví dụ: *http://localhost:8080/tienda1/publico/pagar.jsp~* => đường dẫn được hacker sử dụng để tìm các file bản sao trên server => không tồn tại thật => trả về 1

- **Kiểm tra kí tự và string bất thường trong url**

Tập kí tự và string bất thường được định nghĩa bao gồm: “"~", "backup", "bak", "zip", "sql", "%"”. Đây là những kí tự và tứ khóa thường được hacker sử dụng trong việc tìm kiếm những tài liệu sao lưu, file cơ sở dữ liệu những file được giấu trên server hoặc đơn giản chỉ là hacker đang thực hiện việc fuzzing trên url để thu thập nhiều thông tin nhất có thể trên website

Nếu có tồn tại kí tự bất thường thì giá trị thuộc tính bằng 1, ngược lại thì bằng 0

Ví dụ: *http://localhost:8080/tienda1/miembros/imagenes.BAK* => đường dẫn được hacker sử dụng để tìm kiếm file backup trên server => có chứa từ khóa bất thường => trả về 1

- **Độ dài của url**

Việc kiểm tra độ dài của url là một thuộc tính mang lại thông tin hữu ích vì trong quá trình hacker thu thập thông tin hay trong quá trình khai thác nhiều loại lỗ hổng phổ biến thì độ dài của Url cũng thay đổi nhiều.

Giá trị trả về là độ dài của url

Ví dụ: *http://localhost:8080/travelnet/home.jsp* => giá trị trả về là độ dài len("/travelnet/home.jsp")

- **Đếm số lượng kí tự đặc biệt (không phải các chữ cái bình thường trong url)**

Thuộc tính này dựa trên việc trong quá trình thu thập thông tin trên server thì hacker thường sử dụng một số các công cụ quét để tìm ra toàn bộ các đường dẫn được ẩn dấu trên server điều này sẽ khiến cho trong đường dẫn xuất hiện thêm nhiều kí tự như "/", ".", "#" … Đây là một trong những đặc điểm để phát hiện tấn công

Giá trị trả về sẽ là số lượng kí tự đặc biệt trong url

Ví dụ: *http://localhost:8080/6909030637832563290.jsp.OLD* => giá trị trả về là 22

Các thuộc tính về method

- **Method mà request sử dụng**

Đây là một thuộc tính quan trọng vì hacker có thể sử dụng nhưng method nguy hiểm như PUT, DELETE … để thực hiện các request lên server thay vì những method thông thường như GET, POST. Trong tập dữ liệu này để mô tả method được sử dụng trong request, em sẽ sử dụng các giá trị số tương ứng: GET -> 1, POST ->2, PUT -> 3

Ví dụ: PUT *http://localhost:8080/tienda1/publico/anadir.jsp HTTP/1.1* => giá trị trả về là 3

Các thuộc tính về payload

- **Xuất hiện các kí tự nằm ngoài bảng mã so với tập học**

Thông thường thì khi sử dụng một ứng dụng web. Một người dùng thông thường sẽ chỉ đơn giản là sử dụng các chức năng có sẵn của website đó, khi đó dữ liệu được gửi lên server của là những dữ liệu đã được chuẩn bị từ trước và luôn bao gồm những kí tự có ý nghĩa và in được. Mặt khác, hacker thường sử dụng những kí tự dưới dạng enconded, không in được khi muốn khai thác các lỗ hổng liên quan tới tràn bộ đệm hay giới hạn độ dài kí tự, padding. Vì vậy, ta có thể coi đây như là một dấu hiệu của việc tấn công.

Nếu dữ liệu trong requests mới có xuất hiện các kí tự nằm ngoài bảng mã thì giá trị của thuộc tính này sẽ là 1, ngược lại là 0.

Ví dụ: *http://localhost:8080/tienda1/publico/entrar.jsp?errorMsg=%2B* => giá trị trả về là 1

- **Đếm các kí tự đặc biệt, thường được sử dụng trong các tấn công web phổ biến**

Dựa vào kinh nghiệm nghiên cứu về bảo mật web cùng với tham khảo các tài liệu liên quan, em đã thống kê ra một số các kí tự thường được sử dụng trong tấn công web như SQL injection, XSS, LFI, command injection … cũng như ko xuất hiện trong tập dữ liệu thường. Ví dụ như: "/>", "*", "/*", "*/", """, "'", "#", "-- -", "[", "]"

Giá trị của thuộc tính này sẽ số lượng kí tự nêu trên xuất hiện trong payload

Ví dụ: *http://localhost:8080/tienda1/publico/autenticar.jsp?modo=1 or 1=1-- -* => giá trị trả về là 1.

- **Đếm số lần xuất hiện các từ khóa mang khả năng tấn công**

OWASP TOP 10 là một bản báo cáo hàng năm để thống kê lại các lỗ hổng được đánh giá là nguy hiểm và phổ biến nhất trong năm. Dựa vào báo cáo này, em đã thống kê ra một loạt các lỗ hổng hay gặp phải ở các hệ thống website, từ đó lập ra danh sách một số các từ khóa thông dụng thường được sử dụng để tiến hành các cuộc tấn công web ví dụ như: “union select”, “order by”. “1=1-- -”, “/etc/passwd”, “onerror=” …

Giá trị của thuộc tính sẽ là số lượng các từ khóa được tìm thấy trong payload

Ví dụ: *http://localhost:8080/tienda1/publico/pagar.jsp?modo=/etc/passwd* => giá trị trả về sẽ là 1

- **Xuất hiện giá trị nằm ngoài tập hữu hạn của một thuộc tính**

Trong một ứng dụng web thì các tham số được truyền lên server do một user thực sự tương tác với website đó thường là xác định và trong một số trường hợp thì giá trị của các tham số đó cũng chỉ nằm trong một tập hữu hạn ví dụ như giá trị của biến "Submit" thường là "submit" đối với Login form hay id của một sản phẩm thì thường cũng chỉ có giá trị trong một dải từ 0->N với N là số lượng của sản phẩm đó… Dựa vào đặc điểm này cộng với việc phân tích tập dữ liệu bình thường ban đầu, em sẽ lọc ra các tham số có số lượng giá trị là hữu hạn và sau đó sẽ so sánh giá trị của các tham số này trong các request mới với tập hữu hạn đó. Nếu giá trị đó không nằm trong tập được xác định từ trước thì đó sẽ là một dấu hiệu cho thấy đó là một request do hacker gửi lên. Bởi vì việc inject dữ liệu không bình thường vào các input của người dùng là một việc làm phổ biến của hacker khi muốn khai thác lỗ hổng của ứng dụng web.

Nếu request gửi lên xuất hiện giá trị nằm ngoài tập hữu hạn thì giá trị của thuộc tính sẽ là 1, ngược lại thì là 0.

Ví dụ: *http://localhost:8080/tienda1/publico/anadir.jsp?id=aaaaaaaaaaaaaaa* => giá trị trả về sẽ là 1

- **Trật tự các thuộc tính đối với từng url cụ thể hoặc xuất hiện thuộc tính nằm ngoài các thuộc tính có thể xuất hiện**

Nếu là một người dùng thông thường, người sử dụng các chức năng của website như đúng những gì được người phát triển thiết kế thì chắc chắn rằng dữ liệu do người dùng gửi lên server sẽ luôn có một form xác định vì nó đã được lập trình từ trước và gửi lên nhờ vào các biểu mẫu html hay javascript. Dựa vào đặc điểm này, từ việc phân tích tập dữ liệu ban đầu ta sẽ lập ra một tập các tham số theo thứ tự xác định cùng với địa chỉ url tương ứng. Sau đó sẽ dùng cơ sở dữ liệu này để đối chiếu tới trật tự các tham sô được gửi lên trong request cần dự đoán.

Nếu trật tự các tham số trong request mới không đúng so với trật tự chuẩn đã được lưu lại gắn liền với url tương ứng thì giá trị của thuộc tính sẽ là 1, ngược lại thì là 0

Ví dụ: *id=1&xxxxx=Jam%F3n+Ib%E9rico&random_hacker_para=85* => giá trị trả về là 1

- **So sánh độ lệch về độ dài của một payload với mức trung bình đối với url nhất định**

Thuộc tính này dựa trên việc khi hacker muốn inject các dữ liệu bất thường vào query để tiến hành khai thác các lỗ hổng như SQLi, buffer overflow, LFI, XSS .. thì dữ liệu được thêm vào query sẽ có độ dài thay đổi nhiều so với độ dài ban đầu. Do đó ta có thể sử dụng sự chênh lệch này như là một thuộc tính. Giá trị thuộc tính được tính toán như sau:

p = (l-u)/u, trong đó p là giá trị thuộc tính, l là độ dài của câu truy vấn trong request hiện tại, u là độ dài trung bình của query đó xét trên tập huấn luyện

Ví dụ: *modo=entrar&login=modestin&pwd=es%27pec%27ia%2Fl* => giá trị trả về là (42-32)/32 = 0,3125 (giả sử độ dài trung bình của payload của url tương ứng là 32)

- **Độ dài của query (payload)**

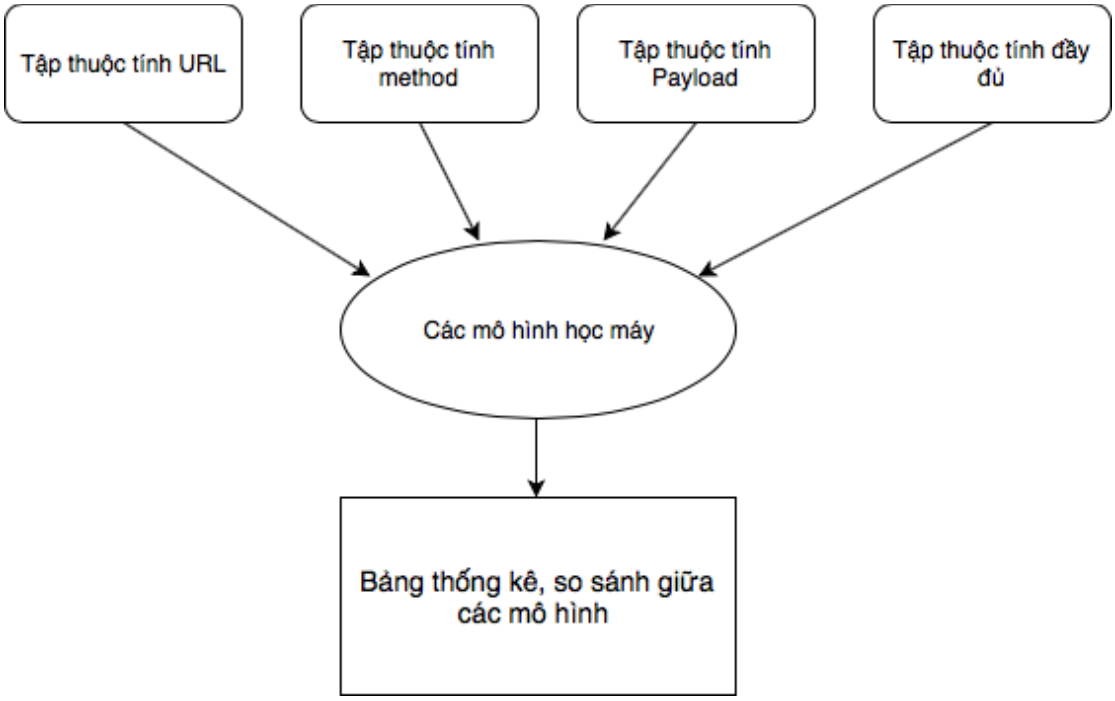
Việc kiểm tra độ dài của câu truy vấn là một thuộc tính mang lại thông tin hữu ích vì trong quá trình hacker thu thập thông tin hay trong quá trình khai thác nhiều loại lổ hổng phổ biến thì độ dài của query cũng thay đổi nhiều

Giá trị trả về là độ dài của payload

Ví dụ: *modo=entrar&login=modestin&pwd=es%27pec%27ia%2Fl* => giá trị trả về là 42

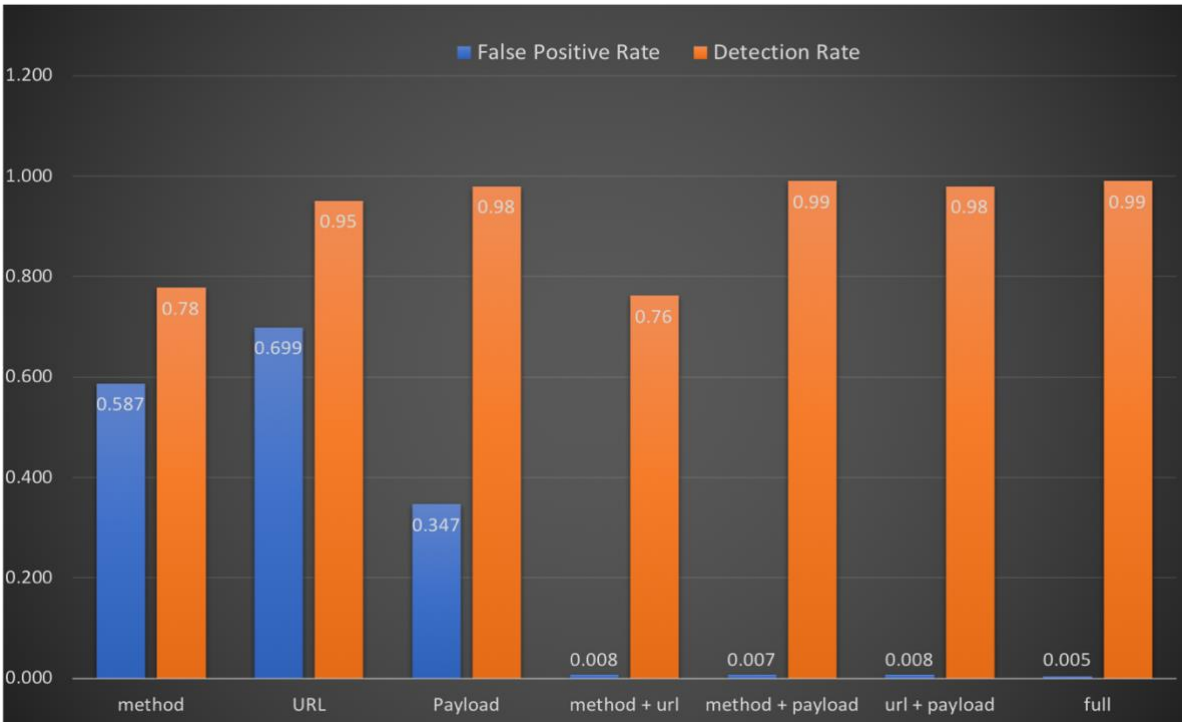
e. Tiến hành thí nghiệm và kết quả thu được

Trong phần thử nghiệm này, tôi sẽ tiến hành áp dụng nhiều mô hình học máy khác nhau, kết hợp cùng với các nhóm thuộc tính đã được giới thiệu thiệu ở phần trước. Các phương pháp học máy sẽ được tiến hành thử nghiệm bao gồm: Logistic Regression, Support Vector Machine, XGBoost, Random Tree. Các phương pháp được đánh giá dựa trên thông số là Detection Rate (DR) và False Positve Rate (FPR). Quá trình đánh giá sẽ áp dụng phương pháp kfolds- validation tức là tập dữ liệu ban đầu sẽ được chia làm 10 phần bằng nhau, sau đó từng phần một sẽ được chỉ định làm tập đánh giá, chín phần còn lại sẽ dùng làm tập học. Lặp lại quá trình này 10 lần ngẫu nhiên sau khi đã thực hiện việc đảo vị trí các bản ghi trong tập dữ liệu, giá trị trung bình của DR và FPR sẽ được dùng để đánh giá hiệu năng của từng mô hình.



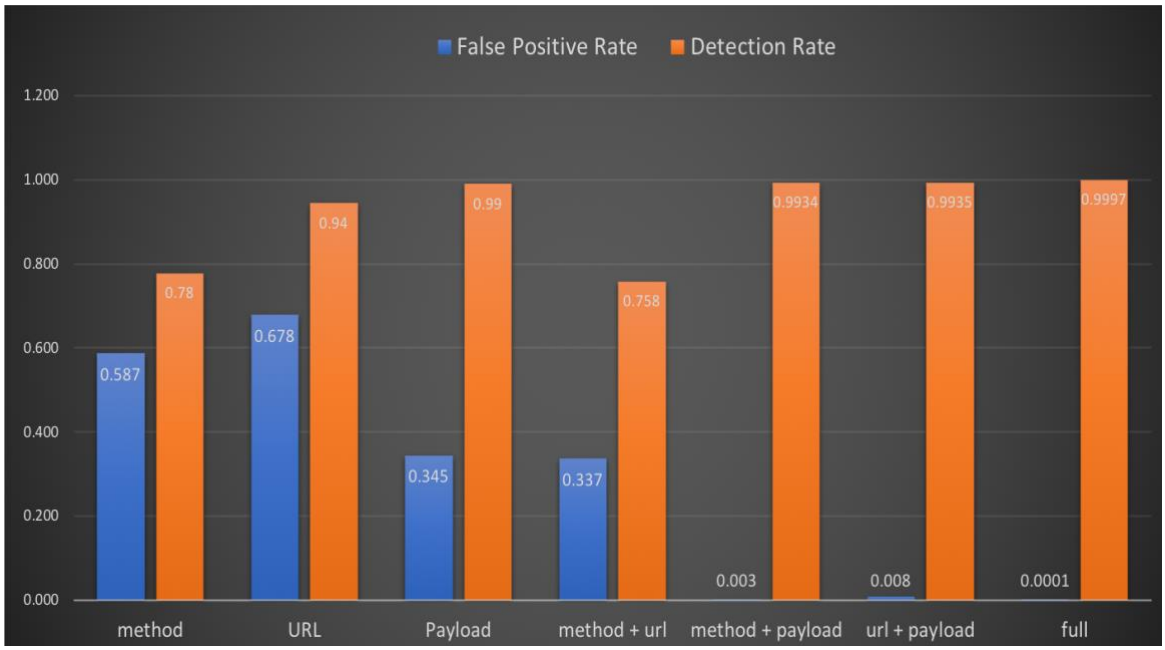
Hình 4.13 Phương pháp đánh giá tổng quát

Áp dụng phương pháp Logistic Regression đối với từng nhóm thuộc tính:



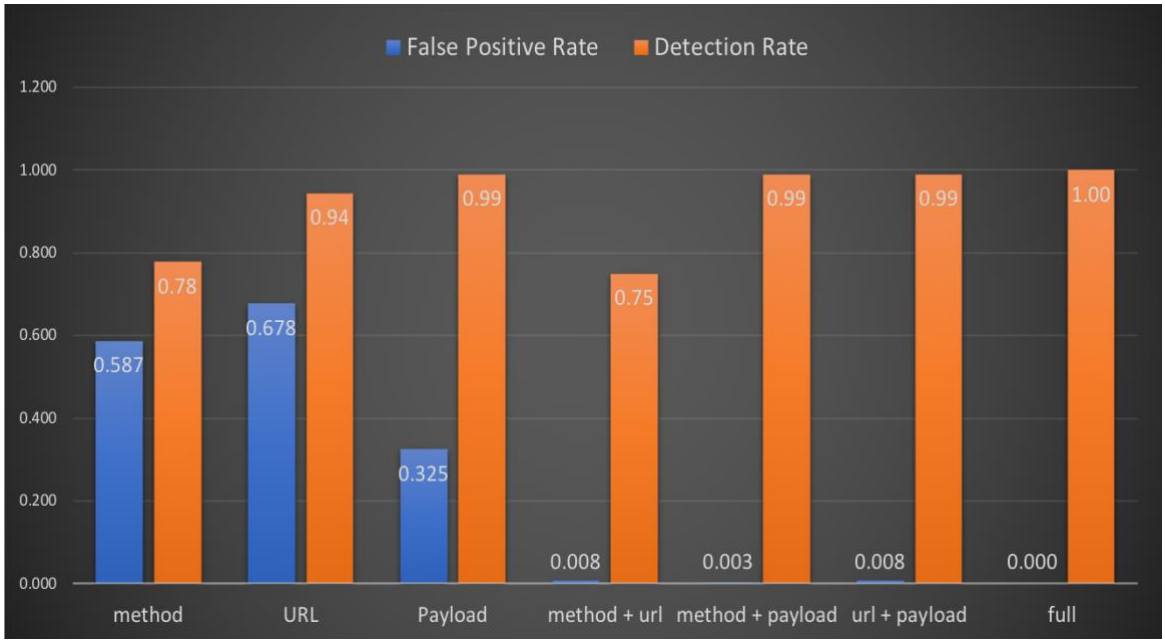
Hình 4.14 Kết quả áp dụng mô hình Logistic Regression trên từng tập thuộc tính

Áp dụng phương pháp Support Vector Machine đối với từng nhóm thuộc tính:



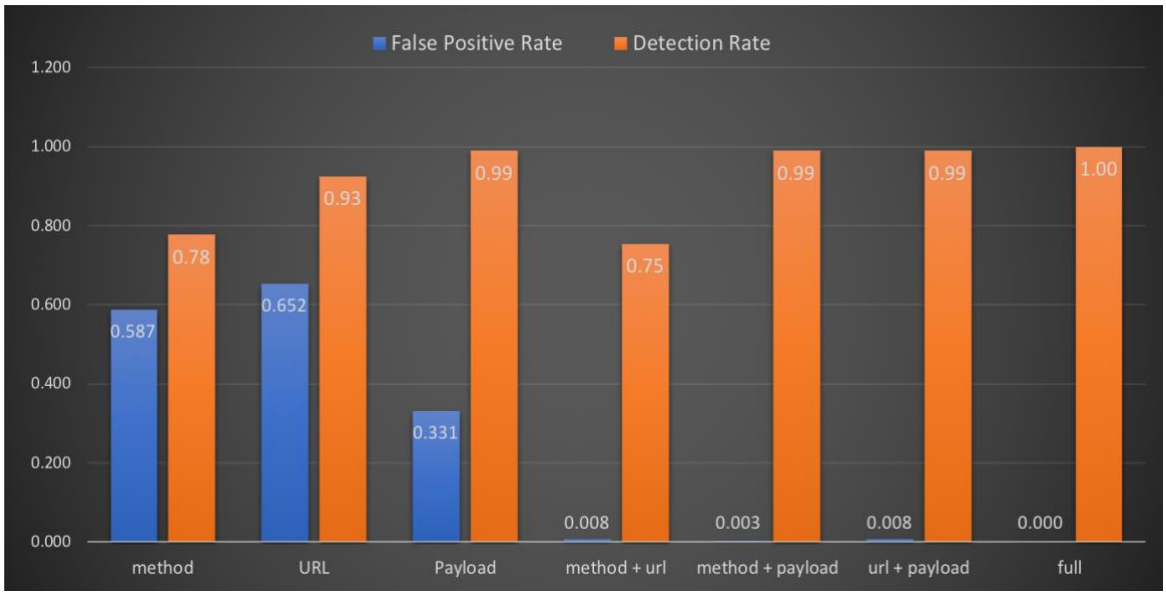
Hình 4.15 Kết quả áp dụng mô hình Support Vector Machine trên từng tập thuộc tính

Áp dụng phương pháp XGBoost đối với từng nhóm thuộc tính:



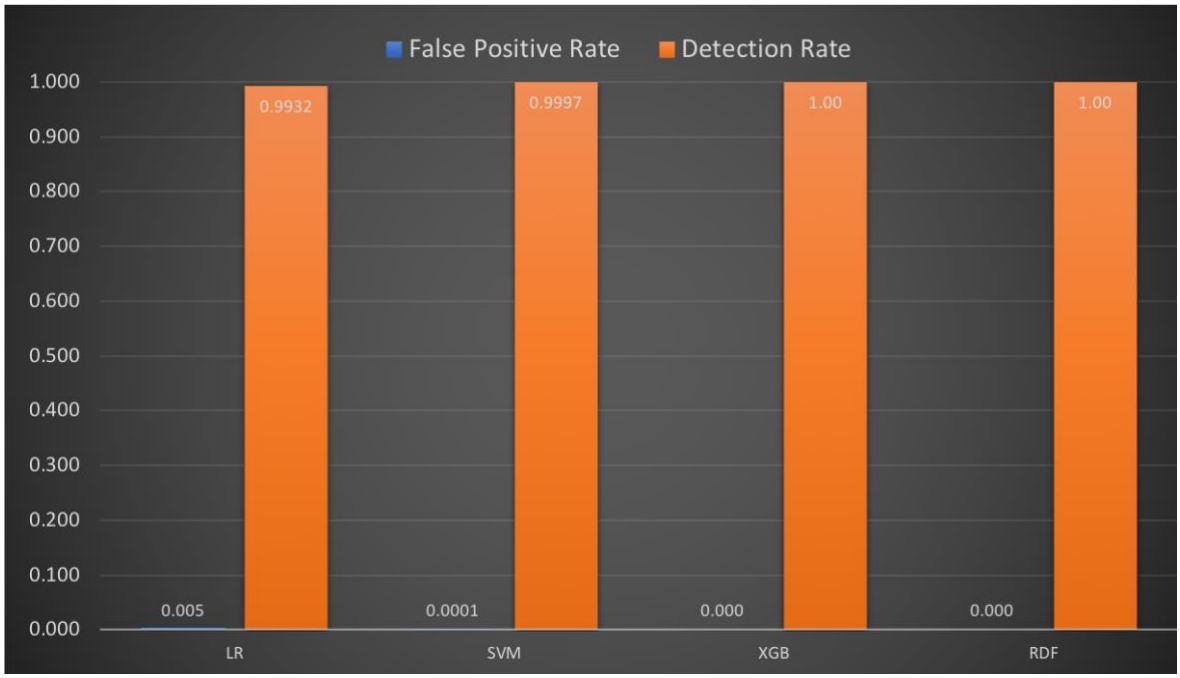
Hình 4.16 Kết quả áp dụng mô hình XGBoost trên từng tập thuộc tính

Áp dụng phương pháp Random Forest đối với từng nhóm thuộc tính:



Hình 4.17 Kết quả áp dụng mô hình Random Forest trên từng tập thuộc tính

So sánh các mô hình với nhau dựa trên tập thuộc tính đầy đủ:



Hình 4.18 So sánh giữa các mô hình học máy

f. Kết luận, các công việc trong tương lai

Qua các kết quả các thí nghiệm và thống kê so sánh ở trên, ta có thể thấy rằng việc phân tích dữ liệu để phát hiện tấn công đạt kết quả tốt nhất khi ta sử dụng tất cả các thuộc tính ở cả ba nhóm là URL, method và payload. Ngoài ra hai phương pháp là XGBoost và Random Forest đều cho kết quả phân loại chính xác tới tuyệt đối và kết quả dự doán không chính xác bằng 0.

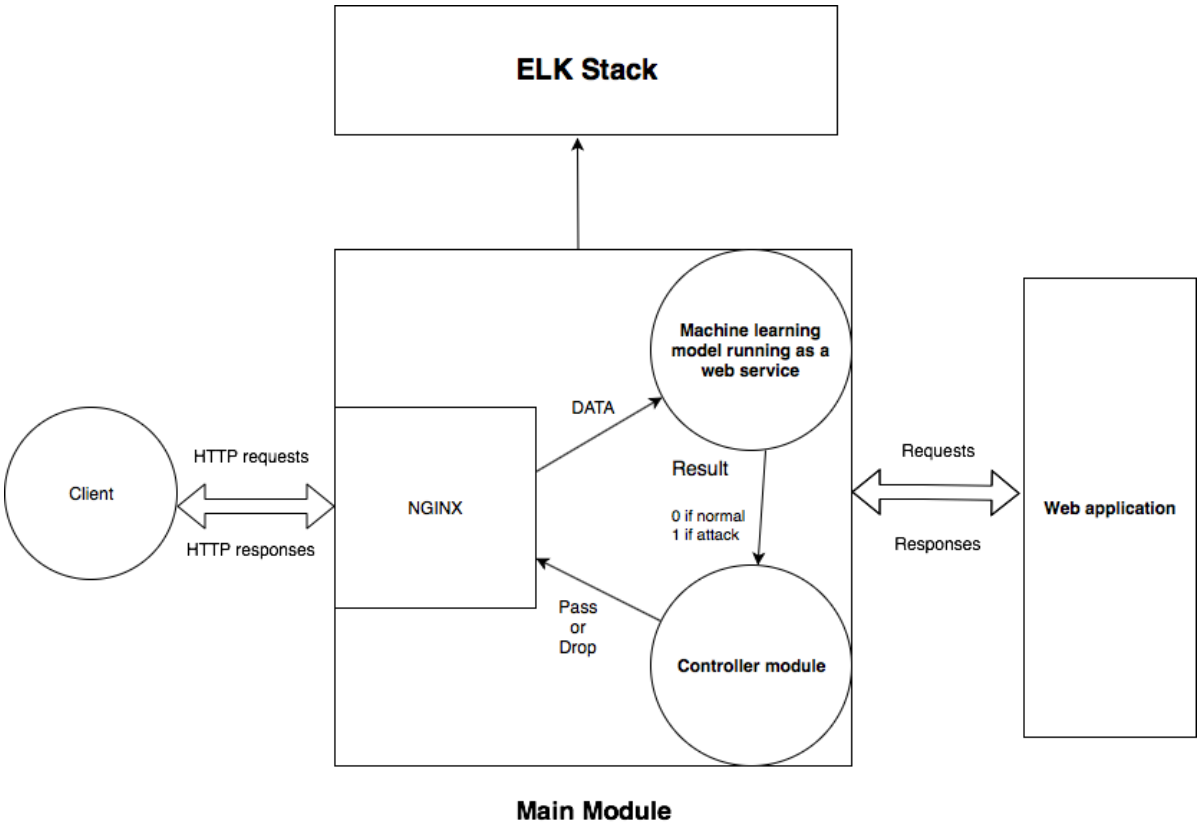
Hơn nữa để so sánh với các nghiên cứu trước đó, thì phương pháp tiếp cận trong đồ án này cho kết quả cao hơn hẳn so với nghiên cứu của chính tác giả của bộ dữ liệu CSIC vào năm 2011 (Hai Thanh Nguyen et al., 2011) **[**3**].** Ở nghiên cứu này độ chính xác tốt nhất mà tác giả đạt được là Detection Rate: 93,65% và False Positive Rate: 6,9%.

Đây là một kết quả rất khả quan tuy nhiên tập dữ liệu CSIC 2010 đã cũ, tuy rằng có chứa nhiều loại tấn công web khác nhau nhưng có thể vẫn còn thiếu nhiều mẫu so với với thời điểm hiện tại. Ngoài ra trong đồ án này do đặc thù của bộ dữ liệu mà tôi mới chỉ tập trung vào việc phát hiện tấn công dựa trên dữ liệu thu thập từ URL, payload, method. Trong khi đó, thực tế hacker có thể thực hiện nhiều loại tấn công web thông qua các trường khác trong header như Cookies, User-agent… Vì vậy trong tương lai, tôi sẽ tiếp tục nghiên cứu hoàn thiện mô hình của mình để nó có khả năng phát hiện tấn công trong mọi trường dữ liệu của http request cũng như

nâng cấp khả năng phân tích xử lý đối với lượng dữ liệu lớn trong quá trình huấn luyện.

V. Mô hình triển khai thực tế

1. Tổng quan mô hình và các thành phần



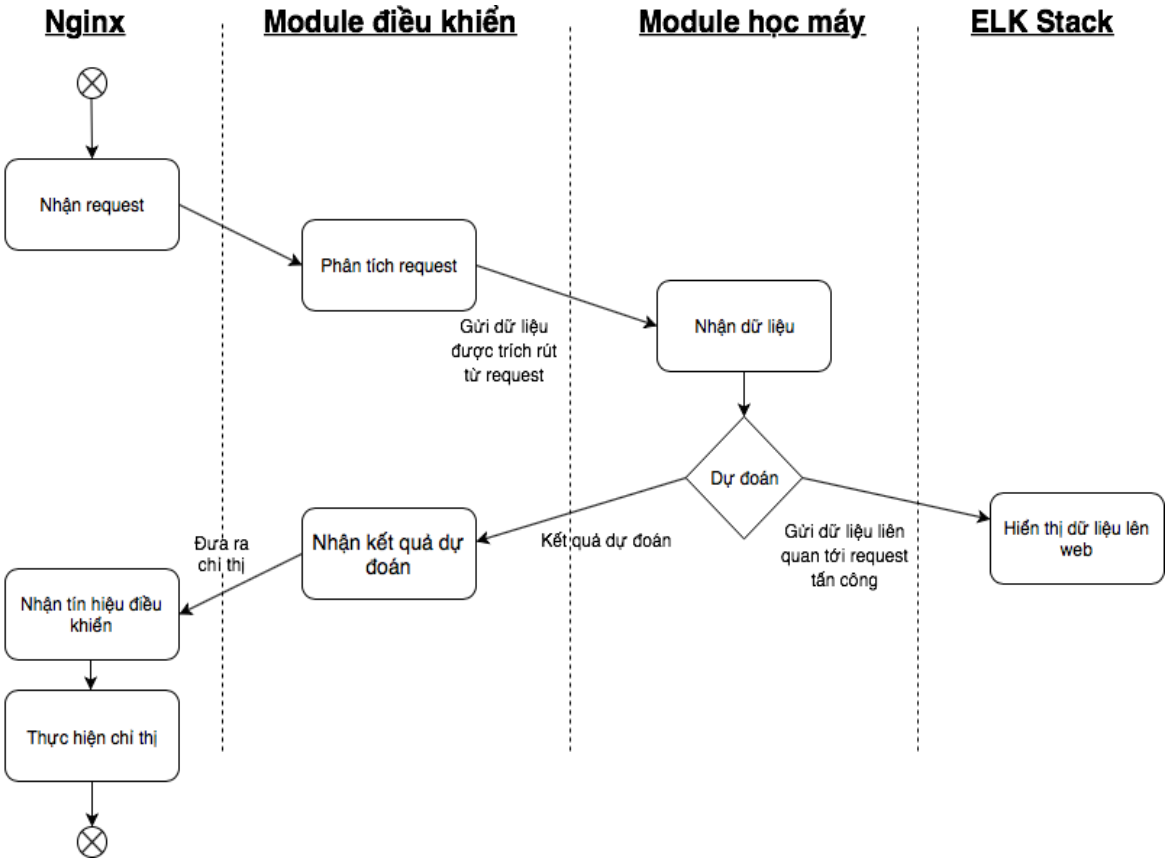
Hình 5.1 Sơ đồ hệ thống WAF áp dụng học máy

Như mô tả ở hình trên, hệ thống WAF do tôi đề xuất sẽ gồm bốn thành phần chính:

- **Nginx server**: Sử dụng trên nền tảnh OpenResty [17] (đây là một phiên bản mở rộng của Nginx, tích hợp nhiều module của các nhà phát triển thứ ba, cho phép lập trình viên có khả năng tuy biến cao hơn so với phiên bản Nginx gốc) chịu trách nhiệm làm reverse proxy, tiếp nhận request từ phía client.
- **Module điều khiển**: viết bằng ngôn ngữ Lua, chịu trách nhiệm phân tích request, gửi dữ liệu đi và quyết định việc xử lý đối với request đó.
- **Module học máy**: Chạy trên nền tảng Flask như một web service, đảm nhận việc dự đoán request có phải tấn công hay không.

- **ELK Stack**: bao gồm ba thành phần cơ bản Elasticsearch, Logstash, Kibana. Đây là một hệ thống hoàn chỉnh và đồng bộ cho việc vận chuyển, lưu trữ và hiển thị dữ liệu log.

2. Nguyên lý hoạt động của hệ thống



Hình 5.2 Sơ đồ hoạt động của hệ thống WAF

1. Đầu tiên, request sẽ được Nginx tiếp nhận, sau đó toàn bộ thông tin về request sẽ được phân tích và xử lý tại module điều khiển.
2. Tại module điều khiển, các thông tin cần cho quá trình dự đoán sẽ được trích rút bao gồm: method, url, payload.
3. Các thông tin này sau đó sẽ được gửi đến cho webservice thông qua giao thức http dưới dạng câu truy vấn sau: http://api.aiservice.local/predict?url=[]&method=[]&payload=[]. Sau đó module điều khiển chờ đợi phản hồi từ phía webservice.
4. Sau khi nhận được đầy đủ dữ liệu, mô hình học máy đã được huấn luyện từ trước đang chạy trên webservice sẽ tiến hành việc trích rút thuộc tính và đưa

ra dự đoán. Giá trị trả về của truy vấn sẽ là 1 nếu đó là request tấn công, là 0 nếu đó là request bình thường. Ngoài ra nếu đó là request tấn công thì thông tin về request cũng sẽ được lưu trữ và hiển thị trên ELK stack.

5. Module điều khiển nhận được kết quả phản hồi từ phía webservice và đưa ra chỉ thị tương ứng cho Nginx. Nếu là request tấn công thì Nginx sẽ phản hồi bằng cách chuyển hướng request sang một trang cảnh báo, nếu là request thường thì sẽ cho đi tới ứng dụng web phía sau.

Hình 5.3 Minh họa trang cảnh báo xuất hiện khi phát hiện request tấn công

3. Đề xuất phát triển thêm cho mô hình này trong tương lai

Do không có điều kiện về hạ tầng nên hệ thống trên mới được tôi triển khai thử nghiệm bằng các máy ảo, tuy rằng các thử nghiệm đều cho thấy khả năng phân loại tức thời của hệ thống nhưng lượng truy cập chưa lớn nên chưa thể đánh giá khả năng của hệ thống trong thực thế. Vì vậy mục tiêu trong tương lai của tôi là tiến hành thêm nhiều thí nghiệm hơn để khảo sát khả năng đáp ứng của hệ thống đối với một website thực tế, có lượng truy cập lớn. Thêm vào đó là việc hoàn thiện các tính năng đã có và thêm các tính năng mở rộng như cân bằng tải, gửi cảnh báo tấn công qua email, xây dựng trang quản trị cho hệ thống.

Tài liệu tham khảo

[1] Rafal Kozik, Michal Choraś, Rafal Renk, Witold Holubowicz. "*A Proposal of Algorithm for Web Appli- cations Cyber Attack Detection*". Khalid Saeed; Václav Snášel. 13th IFIP International Conference on Computer Information Systems and Industrial Management (CISIM), Nov 2014, Ho Chi Minh City, Vietnam. Springer, Lecture Notes in Computer Science, LNCS-8838, pp.680-687, 2014, Computer Information Systems and Industrial Management. <10.1007/978-3-662-45237-0_61>. <hal-01405662>

[2] Althubiti, Sara; Yuan, Xiaohong; and Esterline, Albert, "*Analyzing HTTP requests for web intrusion detection*" (2017). KSU Proceedings on Cybersecurity Education, Research and Practice. 2.

[3] Nguyen, H.T., et al. "*Application of the generic feature selection measure in detection of web attacks*". Computational Intelligence in Security for Information Systems. Berlin: Springer, 2011, 25-32.

[4] EIEI HAN, "*ANALYZING AND CLASSIFYING WEB APPLICATION ATTACKS*", International Journal of Advances in Electronics and Computer Science, ISSN: 2393-2835 Volume-2, Issue-4, April-2015

[5] Melody Moh*, Santhosh Pininti, Sindhusha Doddapaneni, and Teng-Sheng Moh, "*Detecting Web Attacks Using Multi-Stage Log Analysis*", 2016 IEEE 6th International Conference on Advanced Conputing

[6] Shailendra Rathore*, Pradip Kumar Sharma*, and Jong Hyuk Park* , "*XSSClassifier: An Efficient XSS Attack Detection Approach Based on Machine Learning Classifier on SNSs*", J Inf Process Syst, Vol.13, No.4, pp.1014~1028, August 2017 | 1015

[7] David Atienza, Álvaro Herrero and Emilio Corchado , "*Neural Analysis of HTTP Traffic for Web Attack Detection*", © Springer International Publishing Switzerland 2015 201 Á. Herrero et al. (eds.), *International Joint Conference*, Advances in Intelligent Systems and Computing 369, DOI 10.1007/978-3-319-19713-5_18

[8] Yao Pan, Fangzhou Sun, Jules White, Douglas Schmidt, Jacob Staples, Lee Krause, "*Detecting Web Attacks with End-to-End Deep Learning*", IEEE Transactions on Dependable and Secure Computing

[9] Farhan Douksieh Abdi and Lian Wenjuan, "*MALICIOUS URL DETECTION USING CONVOLUTIONAL NEURAL NETWORK* ", International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), Vol.7, No.6, December 2017

[10] Asaad Moosa, "*Artificial Neural Network based Web Application Firewall for SQL Injection*", World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:4, No:4, 2010

[11] C. Torrano-Gimenez, A. Perez-Villegas, G. Alvarez, "*An Anomaly-based Web Application Firewall*". In Proc. of International Conference on Security and Cryptography (SECRYPT 2009), pp. 23-28. INSTICC Press. E. Fernández-Medina, M. Malek, J. Hernando, editores. Milán (Italia), 7-10 Julio (2009)

[12] Fredrik Valeur, Darren Mutz, and Giovanni Vigna , "*A Learning-Based Approach to the Detection of SQL Attacks*", Proceedings of the Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA) Vienna, Austria July 2005

[13] Christopher Kruegel, Giovanni Vigna, "*Anomaly Detection of Web-based Attacks*", proceedings of the ACM Conference on Computer and Communication Security (CCS) Washington, DC October 2003

[14] Giménez, C.T., Villegas, A.P., and Marañón, G.A., ―HTTP Dataset CSIC 2010,‖ CSIC (Spanish Research National Council), 2012, http://www.isi.csic.es/dataset/

[15] Tiến sỹ Thân Quang Khoát, bài giảng Machine Learning, viện Công Nghệ Thông Tin, trường Đại học Bách khoa Hà Nội năm 2016

[16] OWASP, OWASP Top 10, 2017 (https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project)

[17] OpenResty, https://openresty.org/en/ (2018)

Chú giải

Thuật ngữ	Ý nghĩa
URL	Đường dẫn xác đinh tài nguyên tên máy chủ web
Payload	Dữ liệu mà người dùng gửi lên trong quá trình tương tác với website
Request	Là thành phần trong giao thức web dùng để giao tiếp giữa máy khách và máy chủ
Response	Là thành phần trong giao thức web dùng để giao tiếp giữa máy khách và máy chủ
Client	Máy khách
Server	Máy chủ
Web server	Máy chủ web
Hack	Hành động tấn công của kẻ xấu, lợi dụng những lỗ hổng an ninh trong hệ thống mạng
Http	Giao thức mạng được sử dụng để giao tiếp giữa máy chủ và máy khách
Traffic	Dữ liệu được truyền đi trong không gian mạng
Signature-based	Các phương pháp phát hiện tấn công dựa trên những biểu hiện đặc thù tương ứng với từng kiểu tấn công
pattern	Nhưng định dạng dữ liệu cho trước (dùng để phát hiện bất thường)
Zero-day (0-day)	Là những lỗ hổng chưa được công bố
Website	Trang web
Internet	Mạng máy tính toàn cầu
Stateless protocol	Giao thức không lưu trữ trạng thái

Http method	Phương thức gửi dữ liệu từ máy khách tới máy chủ
Request header/ Response header	Thành phần chứa chủ yếu các thông tin diều khiển trong http request, http response
Web Application Firewall (WAF)	Ứng dụng bảo vệ website trước tấn công
Proxy server	Máy tính đứng trung gian giữa máy khách và máy chủ
Fuzzing	Hoạt động của hacker nhằm tìm kiếm thông tin và phát hiện các dấu hiệu bất thường trong cách ứng dụng web xử lý dữ liệu
Module	Một phần. một bộ phận
Log	Lưu trữ dữ liệu về các hoạt động xảy ra trên hệ thống