HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Đỗ Minh Hải

PHÁT HIỆN TẤN CÔNG ỨNG DỤNG WEB DỰA TRÊN LOG TRUY CẬP SỬ DỤNG BỘ PHÂN LỚP RỪNG NGẪU NHIÊN

Chuyên ngành : Hệ thống thông tin

Mã số: 8.48.01.04

TÓM TẮT LUẬN VĂN THẠC SĨ

HÀ NỘI – NĂM 2019

Luận văn được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: TS Nguyễn Ngọc Điệp

Phản biện 1: ……

Phản biện 2: ……

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

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

Có thể tìm hiểu luận văn tại:

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

MỤC LỤC

MỞ ĐẦU

1. Lý do chọn đề tài

Hiện nay, với tốc độ phát triển về công nghệ tin học, truyền thông, thương mại điện tử thì nhu cầu đăng tải, chia sẻ thông tin trên các hệ thống web là rất lớn. các doanh nghiệp đều sở hữu, sử dụng các ứng dụng web như: webmail, bán hàng trực tuyến, đấu giá, mạng xã hội và nhiều chức năng khác để cung cấp dịch vụ trực tuyến, kết nối với khách hàng, đối tác.

Thực tế, mọi ứng dụng web vẫn luôn tiềm ẩn những nguy cơ mất an toàn thông tin do rất nhiều nguyên nhân, cả chủ quan cũng như khách quan gây mất dữ liệu có giá trị, hay làm gián đoạn việc cung cấp dịch vụ. Việc triển khai trực tuyến ứng dụng web sẽ cho phép người dùng quyền truy cập tự do vào ứng dụng thông qua giao thức HTTP/HTTPS, những truy cập này có khả năng vượt qua hệ thống firewall, các lớp bảo vệ hệ thống và các hệ thống phát hiện xâm nhập vì các mã tấn công đều nằm trong các gói giao thức HTTP hợp lệ, kể cả các ứng dụng Web có độ bảo mật cao sử dụng SSL cũng đều cho phép tất cả các dữ liệu đi qua mà không hề kiểm tra tính hợp lệ của dữ liệu. Các ứng dụng web vẫn luôn tiềm ẩn những lỗ hổng bảo mật do mã nguồn, máy chủ…

Bên cạnh đó, việc tấn công xâm nhập các ứng dụng web của harker ngày càng trở nên đa dạng và vô cùng tinh vi. Tuy nhiên, người quản trị có thể phát hiện được những truy cập bất thường dựa vào cơ chế ghi nhận và lưu trữ tất cả truy cập đến máy chủ web thông qua logfile của máy chủ web. Bằng việc thu thập, phân tích tài nguyên này có thể phát hiện được những truy cập bất thường để chủ động phòng ngừa, ngăn chặn những nguy cơ trong tương lai đối với hệ thống.

Trong phạm vi của luận văn này, tác giả lựa chọn đề tài Phát hiện tấn công ứng dụng web dựa trên log truy cập sử dụng bộ phân lớp rừng ngẫu nhiên để nghiên cứu xây dựng, đánh giá mô hình và thử nghiệm kết quả.

2. Tổng quan về vấn đề nghiên cứu

Cho đến nay, nhiều hãng công nghệ của Thế giới cũng như Việt Nam đưa ra các giải pháp hỗ trợ an toàn, bảo mật mạng, đã hạn chế và ngăn chặn rất nhiều các cuộc tấn công

nhằm vào mạng của các đơn vị, doanh nghiệp. Ví dụ như các phần mềm bảo mật, các chương trình diệt virus với cơ sở dữ liệu các mẫu virus liên tục cập nhật hay hệ thống firewall nhằm ngăn chặn những kết nối không tin cậy, thực hiện mã hóa làm tăng an toàn cho dữ liệu được truyền tải trên mạng. Tuy nhiên, các hình thức phá hoại ứng dụng web ngày càng trở nên tinh vi hơn, phức tạp hơn, có thể vượt qua được các công cụ và phần mềm bảo mật có sẵn. Vì vậy, vẫn cần nghiên cứu thêm các giải pháp hỗ trợ để phát hiện được tối đa những tấn công đang diễn ra trong hệ thống mạng để phòng ngừa, hạn chế những thiệt hại cho người dùng, doanh nghiệp.

Với các máy chủ web, việc thu thập, phân tích các log truy cập là cơ chế quan trọng không thể thiếu, nó sẽ giúp tự động ghi nhận tất cả các truy cập gồm bình thường và bất thường đến ứng dụng web. Từ dữ liệu log thô thu thập được, qua quá trình xử lý, phân tích, người quản trị hệ thống có thể trích xuất được các thông tin quan trọng về các hành vi người dùng trực tuyến, các dấu hiệu truy cập bất thường, các dạng mã độc và các dạng tấn công, xâm nhập để giúp người quản trị quyết định áp dụng các phương án phòng ngừa, hoặc đưa ra các cảnh báo về nguy cơ mất an toàn thông tin đối với hệ thống cho người dùng. Đồng thời cũng như là căn cứ giúp cải thiện chất lượng hệ thống và các dịch vụ đáp ứng tốt hơn nhu cầu người dùng.

Có nhiều phương pháp phân tích log đã được nghiên cứu và triển khai, tuy nhiên việc áp dụng bộ phân lớp rừng ngẫu nhiên để phân tích phát hiện tấn công chưa được sử dụng phổ biến. Vì vậy tác giả lựa chọn sử dụng phương pháp học máy có giám sát, áp dụng bộ phân lớp rừng ngẫu nhiên để phân tích các weblog nhằm phát hiện các truy cập bất thường, giúp người quản trị sớm có biện pháp phòng chống, ngăn chặn các nguy cơ có thể mất an toàn thông tin.

3. Mục đích nghiên cứu

Nghiên cứu phương pháp và xây dựng mô hình học máy để phát hiện các tấn công đến ứng dụng web dựa trên log truy cập giúp đánh giá và ngăn ngừa được một số hình thức tấn công phổ biến, có thể đưa ra giải pháp tăng cường các lỗ hổng, nguy cơ tiềm ẩn.

4. Đối tượng và phạm vi nghiên cứu

Đối tượng phân tích là các log file truy cập được tạo ra trên máy chủ web như Apache, Nginx, IIS thông qua luồng mạng pcap.

5. Phương pháp nghiên cứu

Đọc và nghiên cứu tổng quan lý thuyết. Xây dựng mô hình học máy phát hiện tấn công ứng dụng web, đánh giá mô hình, thử nghiệm hệ thống dựa trên dữ liệu đã thu thập.

Cấu trúc của luận văn được tác giả tổ chức thành 4 chương như sau:

Phần 1 – Giới thiệu Cơ sở lý thuyết
1.1. Tổng quan về tấn công Web
1.2. Giới thiệu về Web log
1.3. Phương pháp phát hiện tấn công qua web log sử dụng học máy
Chương 2 – Phương pháp phát hiện tấn công
2.1. Phương pháp phát hiện tấn công
2.2. Tổng quan về thuật toán Random Forest
2.3. Tập dữ liệu huấn luyện (CSIC 2010)
2.4. Phương pháp đánh giá
2.5. Kết quả thử nghiệm
2.6. Kết luận chương
Chương 3: Xây dựng hệ thống thực nghiệm
3.1. Xây dựng hệ thống
3.2. Một số kết quả thử nghiệm hệ thống
Chương 4: Kết luận và kiến nghị
4.1. Những đóng góp của luận văn
4.2. Hướng phát triển luận văn

CHƯƠNG 1 – CƠ SỞ LÝ THUYẾT

1.1. Tổng quan về tấn công Web.

Ngày nay, Web chính là kênh truyền thông cơ bản giúp doanh nghiệp tăng cường hình ảnh trực tuyến của mình trên thế giới mạng, giúp xây dựng, duy trì nhiều mối quan hệ với khách hàng tiềm năng. Với xu hướng phát triển công nghệ CNTT và truyền thông hiện nay, Web đã trở thành kênh bán hàng phổ biến đối với hàng nghìn doanh nghiệp lớn nhỏ. Đặc biệt website hiện nay cho phép đóng gói, xử lý, lưu trữ và truyền tải dữ liệu khách hàng với dữ liệu lớn, quan trọng và có giá trị (như thông tin cá nhân, mã số thẻ tín dụng, thông tin bảo mật xã hội …).

Chính những đặc điểm này, các website thường xuyên là mục tiêu tấn công của tin tặc để khai thác đánh cắp các thông tin quan trọng. Một trong những phương thức tấn công phổ biến là khai thác các lỗi bảo mật liên quan đến ứng dụng web. Nhiều điểm yếu nghiêm trọng hay các lỗ hổng cho phép hacker xâm nhập thẳng và truy cập vào cơ sở dữ liệu để trích xuất các dữ liệu nhạy cảm, quan trọng.

1.1.1. Một số khái niệm cơ bản về ứng dụng web

a. HTTP Request & HTTP Response

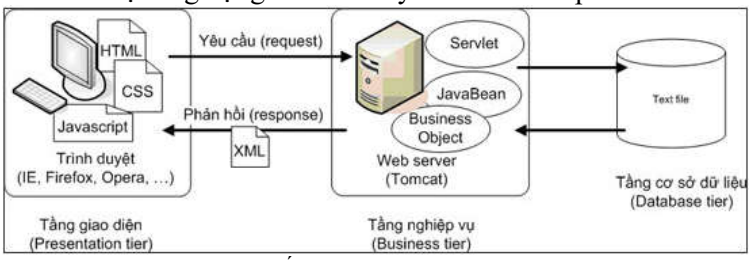
b. Session

c. Cookie

d. Proxy

1.1.2. Kiến trúc của một ứng dụng web

Một ứng dụng Web có đầy đủ các thành phần như sau:



Hình 1. 1 - Kiến trúc của một ứng dụng Web

(Nguồn: https://edu.com.vn)

- Trình khách (hay còn gọi là trình duyệt): Internet Explorer, Firefox, Chrome...
- Trình chủ: Apache, IIS,...
- Hệ quản trị cơ sở dữ liệu: SQL Server, MySQL, DB2, Acess...

1.2. Giới thiệu về Web log file

Web Log là một hoặc nhiều file log được tạo và lưu trữ bởi một Web server, nó chứa tất cả các hành động mà người truy cập tác động lên trang web.

Các web server chuẩn như Apache và 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, cấu tạo 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

Lợi ích lớn nhất của tập tin nhật ký là tính sẵn có tương đối đơn giản. Máy chủ web như Apache mặc định phải cho phép ghi nhật ký. Các ứng dụng thường thực hiện ghi nhật ký để đảm bảo truy xuất nguồn gốc của các hành động của chúng.

1.3. Phương pháp phát hiện tấn công qua web log sử dụng học máy

1.3.1. Tổng quan về học máy

Học máy là các kỹ thuật giúp cho máy tính có thể tự học hỏi dựa trên dữ liệu đưa vào mà không cần phải được lập trình cụ thể.

Một bài toán học máy cần trải qua 3 bước chính:

- Chọn mô hình: Chọn một mô hình thống kê cho tập dữ liệu.
- Tìm tham số: Các mô hình thống kê có các tham số tương ứng, nhiệm vụ lúc này là tìm các tham số này sao cho phù hợp với tập dữ liệu nhất có thể.
- Suy luận: Sau khi có được mô hình và tham số, ta có thể dựa vào chúng để đưa ra suy luận cho một đầu vào mới nào đó.

Một bài toán học máy cần có dữ liệu để huấn luyện, ta có thể coi nó là điều kiện tiên quyết. Dữ liệu sau khi có được cần phải:

- Chuẩn hoá: Tất cả các dữ liệu đầu vào đều cần được chuẩn hoá để máy tính có thể xử lý được. Quá trình chuẩn hoá bao gồm số hoá dữ liệu, co giãn thông số cho phù hợp với bài toán. Việc chuẩn hoá này ảnh hưởng trực tiếp tới tốc độ huấn luyện cũng như cả hiệu quả huấn luyện.
- Phân chia: Việc mô hình được chọn rất khớp với tập dữ liệu đang có không có nghĩa là giả thuyết của ta là đúng mà có thể xảy ra tình huống dữ liệu thật lại không khớp. Vấn đề này trong học máy được gọi là khớp quá (Overfitting). Vì vậy khi huấn luyện người ta phải phân chia dữ liệu ra thành 3 loại để có thể kiểm chứng được phần nào mức độ tổng quát của mô hình. Cụ thể 3 loại đó là:

+ Tập huấn luyện (Training set): Dùng để học khi huấn luyện.
+ Tập kiểm chứng (Cross validation set): Dùng để kiểm chứng mô hình khi huấn luyện.
+ Tập kiểm tra (Test set): Dùng để kiểm tra xem mô hình đã phù hợp chưa sau khi huấn luyện.

1.3.2. Các nhóm giải thuật học máy:

Theo phương thức học, các thuật toán Machine Learning thường được chia làm 4 nhóm:

- Học có giám sát: Máy tính được xem một số mẫu gồm đầu vào và đầu ra tương ứng trước. Sau khi học xong các mẫu này, máy tính quan sát một đầu vào mới và cho ra kết quả.
- Học không giám sát;
- Học nửa giám sát;
- Học tăng cường;

Việc giám sát thu thập, phân tích các log truy cập hệ thống nói chung và các log truy cập các dịch vụ mạng nói riêng là nhiệm vụ không thể thiếu trong các hệ thống giám sát, phân tích hành vi người dùng, phát hiện bất thường, phát hiện tấn công, xâm nhập hệ thống và mạng. Dữ liệu log có thể cung cấp cho người quản trị nhiều thông tin quan trọng về các hành vi người dùng trực tuyến, cũng như các dấu hiệu của các hành vi truy cập bất thường, các dạng tấn công, xâm nhập để đưa ra các cảnh báo nguy cơ mất an toàn thông tin đối với hệ thống.

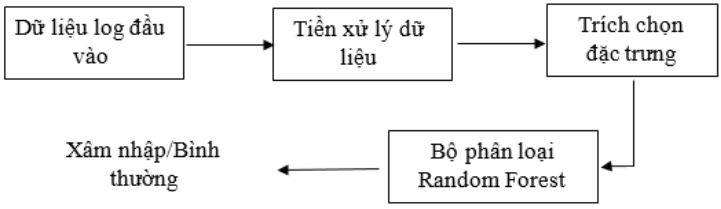
Hiện có nhiều phương pháp phát hiện tấn công từ việc thu thập, xử lý, phân tích log truy cập. Trong nội dung của luận văn này, tác giả đi sâu nghiên cứu ứng dụng phương pháp học máy có giám sát, sử dụng bộ phân lớp rừng ngẫu nhiên để phát hiện các tấn công. Để thuận tiện cho quá trình tiền xử lý dữ liệu, trong phạm vi luận văn này sẽ sử dụng đầu vào dữ liệu là các web log máy chủ web được thu thập từ luồng mạng pcap. Trong chương tiếp theo, tác giả sẽ đi sâu nghiên cứu xây dựng mô hình phát hiện tấn công và đánh giá tính hiệu quả của mô hình.

CHƯƠNG 2: PHƯƠNG PHÁP PHÁT HIỆN TẤN CÔNG

Trong Chương này, tác giả đi sâu nghiên cứu phương pháp phát hiện tấn công dựa vào phương pháp học máy có giám sát, và mô hình cụ thể được sử dụng trong phát hiện tấn công là Rừng ngẫu nhiên (Random Forest).

2.1. Phương pháp phát hiện tấn công

2.1.1. Mô hình phát hiện tấn công



Hình 2. 1- Mô hình hệ thống phát hiện xâm nhập

Các thành phần trong mô hình Phát hiện mã độc tấn công có chủ đích gồm 4 thành phần chính:

- ***Khối Dữ liệu đầu vào:*** Do cấu trúc dữ liệu weblog rất đa dạng ở các hệ thống khác nhau, trong phạm vi luận văn này chỉ tập trung thu thập dữ liệu weblog thu thập từ luồng pcap tại các máy chủ Apache, Nginx, IIS và cho vào khối tiền xử lý dữ liệu.
- ***Khối tiền xử lý dữ liệu***: Tiền xử lý dữ liệu là bước rất quan trọng trong việc giải quyết bất kỳ vấn đề nào trong lĩnh vực Học Máy. Hầu hết các bộ dữ liệu được sử dụng trong Học Máy đều cần được xử lý, làm sạch và biến đổi trước khi một thuật toán Học Máy có thể được huấn luyện trên những bộ dữ liệu này. Các kỹ thuật tiền xử lý dữ liệu phổ biến hiện nay bao gồm xử lý dữ liệu bị khuyết (missing data), mã hóa các biến nhóm (encoding categorical variables), chuẩn hóa dữ liệu (standardizing data), co giãn dữ liệu (scaling data),…

Trong mô hình hệ thống phát hiện xâm nhập trên, chức năng của khối tiền xử lý dữ liệu là trích xuất lấy các thông tin từ log, các truy cập đến hệ thống máy chủ web. Tất cả các tập tin log sẽ được chuyển về một hệ thống chung để phân tích, chuyển đổi cấu trúc, phân tách các trường đặc trưng.

- ***Khối trích chọn đặc trưng***: Chọn ra những đặt trưng tốt (good feature) của dữ liệu, lược bỏ những đặc trưng không tốt của dữ liệu, gây nhiễu (noise), quyết định chọn bao nhiêu đặc trưng để cho vào mô hình.

Để xử lý bộ dữ liệu phù hợp cho mô hình thuật toán Random Forest, quá trình trích chọn đặc trưng sẽ trích chọn các trường sau: Host, Method, Content Type (lưu vị trí của các giá trị đó trong mảng), URL, Payload và Content Length (lưu độ dài chuỗi), vì các trường này đặc trưng nhất của weblog, nhiều nghiên cứu cũng đã sử dụng các trường này.

- ***Khối bộ phân loại Random Forest:*** Chức năng của khối phân lớp Random Forest được mô tả chi tiết trong mục 2.2.

Có nhiều bộ phân lớp có thể áp dụng để xây dựng mô hình phát hiện tấn công này, như SVM, Decision Tree, Navie Bayers, Random forests…, Random forests được coi là một phương pháp chính xác và mạnh mẽ, có thể làm việc được với dữ liệu thiếu giá trị, và khi Forest có nhiều cây hơn, chúng ta có thể tránh được việc Overfitting với tập dữ liệu vì vậy trong nội dung

luận văn, tác giả lựa chọn bộ phân lớp Random Forest để xây dựng mô hình giải quyết bài toán phát hiện tấn công.

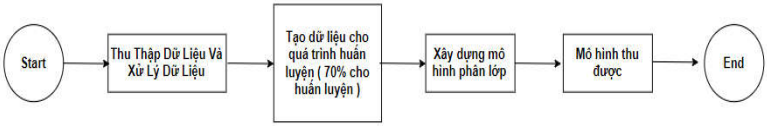
- ***Thông báo kết quả***: Sẽ thông báo cho người dùng kết quả phát hiện Xâm nhập/Bình thường.

2.1.2. Các giai đoạn thực hiện

Thông thường một ứng dụng của một mô hình học máy thường được chia làm hai giai đoạn đó là : Huấn luyện và kiểm tra mô hình . Tỷ lệ độ chính xác phát hiện ra mã độc của mô hình phụ thuộc rất nhiều vào chất lượng của bộ tập mẫu.

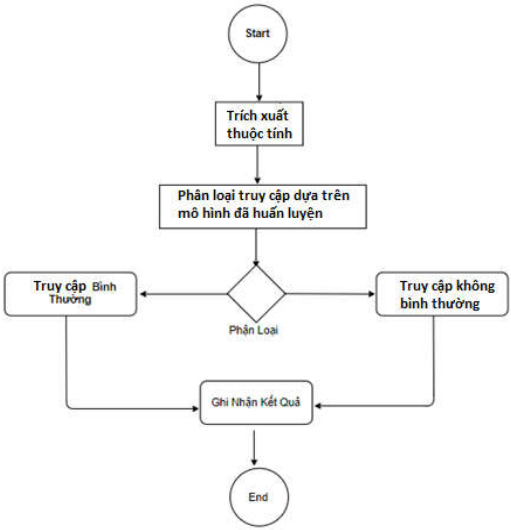
Để giải quyết bài toán phát hiện tấn công dựa trên log file, luồng mạng thì cần thực hiện các giai đoạn sau đây :

- Giai đoạn 1: Thu thập dữ liệu và tiền xử lý dữ liệu phục vụ cho quá trình học của mô hình.
- Giai đoạn 2: Lựa chọn thuật toán xây dựng mô hình.
- Giai đoạn 3: Huấn luyện mô hình với dữ liệu đã xử lý



Hình 2. 2 - Sơ đồ tạo mô hình phân lớp

- Giai đoạn 4: Kiểm tra huấn luyện trên mô hình mới



Hình 2. 3 - Mô hình phân lớp dữ liệu đầu vào

2.2. Tổng quan về thuật toán Random Forest

Thuật toán Random Forest lần đầu tiên được đề xuất vào năm 1995, là phương pháp học máy kết hợp, tức là sử dụng cách kết hợp các phương pháp học máy đơn giản để xây dựng một mô hình có độ chính xác cao hơn. Random Forest có thể được sử dụng để giải cả bài toán phân loại và hồi quy. Nó làm việc bằng cách xây dựng một tập hợp các cây quyết định trong quá trình training, sau đó kết hợp kết quả trả về của mỗi cây đưa ra quyết định dự đoán cuối cùng. Rừng ngẫu nhiên là một thuật toán học có giám sát.

Random Forest (RF) dựa trên cơ sở :

- Random = Tính ngẫu nhiên ;
- Forest = nhiều cây quyết định (decision tree).

Đơn vị của RF là thuật toán cây quyết định, với số lượng hàng trăm. Mỗi cây quyết định được tạo ra một cách ngẫu nhiên từ việc : Tái chọn mẫu (bootstrap, random sampling) và chỉ dùng một phần nhỏ tập biến ngẫu nhiên (random features) từ toàn bộ các biến trong dữ liệu. Ở trạng thái sau cùng, mô hình RF thường hoạt động rất chính xác, nhưng đổi lại, ta không thể nào hiểu được cơ chế hoạt động bên trong mô hình vì cấu trúc quá phức tạp.

Nhiều cây quyết định được tạo theo các ngẫu nhiên sẽ tạo ra một rừng ngẫu nhiên (random forest). Một rừng ngẫu nhiên phân lớp bao gồm một tổ hợp cây quyết định phân lớp:

$$\{\delta(\varepsilon,\omega_k,k=1,\ldots)\}$$

Với điều kiện $\{\omega_k\}$ là các cây quyết định được tạo độc lập ngẫu nhiên và mỗi cây quyết định sẽ bình chọn cho kết quả lớp phổ biến nhất với giá trị đầu vào x.

2.2.1. Cách làm việc của thuật toán

Thuật toán Random Forest bao gồm 2 giai đoạn chính:

- Quá trình tạo ra rừng cây ngẫu nhiên
- Quá trình thực hiện dự đoán dựa trên rừng đã tạo

a. Quá trình tạo ra rừng ngẫu nhiên

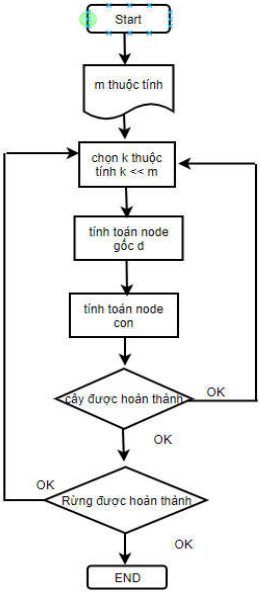
Một rừng ngẫu nhiên là một tập hợp của rất nhiều cây quyết định (decision tree) Để tạo mới cây quyết định, thuật toán Random Forest luôn luôn bắt đầu với 1 cây quyết định rỗng. Đó là cây quyết định chỉ có điểm bắt đầu và liên kết thẳng tới câu

trả lời. Thuật toán sẽ tìm ra câu hỏi đầu tiên tốt nhất để bắt đầu, và sau đó xây dựng cây quyết định. Mỗi khi thuật toán tìm được 1 câu hỏi tốt để hỏi, nó sẽ tạo ra 2 nhánh (trái và phải) của cây. Khi không còn câu hỏi nào thú vị nữa, thuật toán sẽ dừng lại và kết thúc quá trình xây dựng cây quyết định.

Để chắc chắn rằng tất cả các cây quyết định là không giống nhau, Random Forest sẽ tự động thay đổi ngẫu nhiên đối tượng cần theo dõi. Nói một cách chính xác hơn, thuật toán sẽ xóa ngẫu nhiên 1 vài đối tượng, và nhân bản 1 vài đối tượng khác. Tiến trình này được gọi là "bootstrapping". Ngoài ra để đảm bảo rằng cây quyết định có sự khác biệt, Random Forest sẽ ngẫu nhiên loại bỏ có mục đích một vài câu hỏi khi xây dựng cây quyết định. Trong trường hợp này, nếu câu hỏi tốt nhất không được kiểm tra, thì các câu hỏi khác sẽ được chọn để tạo ra cây. Quá trình được gọi là "attribute sampling".

Quá trình tao ra rừng cây ngẫu nhiên được thể hiện qua các bước sau :

- Bước 1 : Chọn ngẫu nhiên k thuộc tính từ tổng m thuộc tính sao cho k<<m
- Bước 2 : Trong số *k* thuộc tính, tính toán node gốc (root) *d* sử dụng phương pháp chọn thuộc tính tốt nhất (best split point).
- Bước 3: Chọn các node trong (internal node) bằng cách sử dụng phương pháp chọn thuộc tính tốt nhất (best split).
- Bước 4: Lặp lại bước 1 đến bước 3 cho đến khi cây được hoàn thành.
- Bước 5: Lặp lại bước 1 dến bước 4 cho đến khi rừng cây được hoàn thành.



Hình 2. 4 - Sơ đồ tạo rừng ngẫu nhiên

b. Quá trình thực hiện dự đoán dựa trên rừng đã tạo

Sau quá trình tạo rừng ngẫu nhiên, thuật toán sẽ dự đoán trên rừng đã được tạo các bước cho quá trình dự đoán như sau :

- Bước 1: Lấy tập thuộc tính kiểm thử và sử dụng tập luật được tạo ra bởi cây quyết định ngẫu nhiên trong quá trình tạo rừng cây ngẫu nhiên, để dự đoán đầu ra.
- Bước 2: Tính toán số phiếu bầu – bình chọn của mỗi cây ngẫu nhiên đưa ra.
- Bước 3: Coi số phiếu bầu – bình chọn cao nhất trong cả rừng cây ngẫu nhiên là kết quả cuối cùng

2.2.2. Thuật toán lựa chọn thuộc tính cho Random Forest

Trong thuật toán Random Forest, để lựa chọn ra thuộc tính nào phù hợp nhất để làm node gốc (root node) và các thuộc tính nào phù hợp để làm các node trong (internal node) tiếp theo, thì thuật toán Random Forest sử dụng chủ yếu thuật toán Information Gain.

Thuật toán Information Gain là một thuật toán được thực hiện dựa trên việc dùng Entropy làm độ đo.

Công thức tính Entropy như sau:

$$H(X) = E_X[I(x)] = -\sum_{x \in X} p(x) log p(x)$$

Hàm số Entropy

Cho một phân phối xác suất của một biến rời rạc x có thể nhận được n giá trị khác nhau x1,x2,.., xn. Giả sử rằng xác xuất để x nhận các giá trị này là p_i = p(x= xi)

Ký hiệu phân phối này là $\mathbf{p} = (p_1, p_2, \ldots, p_n)$.

Entropy của phân phối này là :

$$H(\mathbf{p}) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

Từ đồ thị ta thấy, hàm Entropy sẽ đạt giá trị nhỏ nhất nếu có một giá trị *pi* =1, đạt giá trị lớn nhất nếu tất cả các *pi* bằng nhau. Hàm Entropy càng lớn thì độ ngẫu nhiên của các biến rời rạc càng cao.

Với cây quyết định, ta cần tạo cây như thế nào để cho ta nhiều thông tin nhất, tức là Entropy là cao nhất.

Bài toán của ta trở thành, tại mỗi tầng của cây, cần chọn thuộc tính nào để độ giảm Entropy là thấp nhất.

Người ta có khái niệm Information Gain được tính bằng :

Gain(S,f) = H(S) - H(f,S)

trong đó:

H(*S*) là Entropy tổng của toàn bộ tập dataset *S*.

H(*f*,*S*) là Entropy được tính trên thuộc tính *f*.

Do *H*(*S*) là không đổi với mỗi tầng, ta chọn thuộc tính *f* có Entropy nhỏ nhất để thu được *Gain*(*S*,*f*) lớn nhất.

2.3. Tập dữ liệu huấn luyện (CSIC 2010)

Bộ dữ liệu trong đồ án sử dụng được lấy từ tập dữ liệu Bộ dữ liệu HTTP CSIC 2010 [1] được phát triển tại Viện An toàn Thông tin thuộc Hội đồng Nghiên cứu Quốc gia Tây Ban Nha, chuyên để thử nghiệm các giải pháp tường lửa ứng dụng web. Trong tập dữ liệu này có tổng cộng 36.000 câu truy vấn an toàn và 25.000 câu truy vấn có tấn công. Tập dữ liệu chứa hầu hết các loại tấn công phổ biến của ứng dụng web, bao gồm các cuộc tấn công như SQL, tràn bộ đệm, thu thập thông tin, tiết lộ

tệp, tiêm CRLF, XSS, bao gồm phía máy chủ, giả mạo tham số, v.v.

2.4. Phương pháp đánh giá

Hiệu năng của một mô hình thường được đánh giá dựa trên tập dữ liệu kiểm thử (test data), được phân tách từ tập dữ liệu. Ở bước này thực hiện chia ngẫu nhiên dữ liệu thành 2 phần theo tỉ lệ: 70% dùng cho training và 30% cho testing. Dữ liệu này được chia với 2 nhãn, với các luồng mạng bình thường gán nhãn 0 còn các luồng độc hại được gán nhãn 1.

Có rất nhiều cách đánh giá một mô hình phân lớp. Các phương pháp thường được sử dụng là: accuracy score, confusion matrix, ROC curve… Precision and Recall, F1 score… Trong luận văn sẽ sử dụng phương pháp Precision and Recall, F1 score do tập dữ liệu của các lớp là chênh lệch nhau nhiều.

- Precision (độ chính xác): Trong tất cả các dự đoán thuộc lớp được đưa ra, bao nhiêu dự đoán là chính xác.
- Recall (độ hồi tưởng): Trong tất cả các trường hợp thuộc lớp dương, bao nhiêu trường hợp đã được dự đoán chính xác.
- F1-Score: Tiêu chí đánh giá F1 là sự kết hợp của 2 tiêu chí đánh giá Precision và Recall. F1 là một trung bình điều hòa (harmonic mean) của các tiêu chí Precision và Recall.

2.5. Kết quả thử nghiệm

Tổng hợp dữ liệu mã độc tấn công có chủ đích từ tập dữ liệu HTTP CSIC 2010. Từ tập dữ liệu này, sau khi tiền xử lý dữ liệu phù hợp với mô hình học máy, trộn và chia ngẫu nhiên dữ liệu thành 2 phần cho training và testing, tách nhãn phân loại cho quá trình training và lưu ra file, cho vào thuật toán học máy tạo model.

Kết quả thực hiện

	Precision	Recall	F1-score
Bình thường	0,82	0.91	0.86
Tấn công	0.84	0.72	0.77
Avg/total	0.83	0.83	0.83

Hình 2. 5 -Kết quả học máy và in ma trận nhầm lẫn

Từ các thông số trên, ta rút ra một số kết luận cho mô hình này:

- Precision: Mô hình dự đoán đúng 83% requests bình thường trong tổng số các requests mà nó phân loại là bình thường.
- Recall : Mô hình dự đoán đúng 83% request tấn công trong tổng số các request mà nó phân loại là tấn công.

2.6. Kết luận chương

Trong chương này đã xây dựng được mô hình học máy phát hiện tấn công và thực hiện đánh giá hiệu quả của mô hình dựa trên các tập dữ liệu HTTP CSIC 2010. Qua kịch bản thử nghiệm cho thấy mô hình với dữ liệu từ tập dữ liệu HTTP CSIC 2010 đạt kết quả tốt. Vì vậy, hệ thống sẽ được xây dựng dựa trên mô hình và tập dữ liệu này. Chương kế tiếp sẽ trình bày mô hình thực nghiệm ứng dụng.

CHƯƠNG III – XÂY DỰNG HỆ THỐNG THỰC NGHIỆM

3.1. Xây dựng hệ thống

Đây là demo hệ thống phát hiện tấn công dựa trên việc đọc gói tin trực tiếp hoặc đọc file log sử dụng bộ dữ liệu học HTTP CSIC 2010.

Chương trình có hai chức năng chính:

- Đọc trực tiếp từ pcap: ứng dụng sẽ lọc các request được gửi tới server, đem phân loại từng request, in kết quả lên màn hình.
- Đọc file pcap ngoại tuyến: Khi các gói tin gửi đến được tổng hợp và lưu dưới dạng file pcap, có thể dùng ứng dụng mở lên đọc các kết nối HTTP và phân loại.
- Đọc trực tiếp file log từ webserver: Mỗi khi có kết nối đến web server được ghi vào file log, ứng dụng sẽ đọc các dòng mới này và đem đi phân loại request..
- Ngoài ra, ta có thể đọc lại các file log từ Apache, Nginx đã ghi sẵn cho quá trình phân loại và in kết quả.

Chương trình sẽ lấy các dữ liệu tại URL và Payload làm tiền đề để phân loại, vì vậy trước mắt chương trình chỉ có thể phân biệt các tấn công có tác động đến URL và Payload ví dụ như Command Injection, SQL Injection, XSS, Weak Session ID,…

3.1.1. Thu thập dữ liệu log và tiền xử lý dữ liệu

Quá trình tiền xử lý dữ liệu vào và lọc dữ liệu, ta cần có các trường Host, Method, Content Type (lưu vị trí của các giá trị đó trong mảng), URL, Payload và Content Length (lưu độ dài chuỗi).

Với bộ CSIC Dataset ban đầu, ta gộp hai file normal và anomalous, đổi sang định dạng .csv với tất cả các trường. Tiến hành xóa các trường không dùng đến, ta sẽ giữ lại các trường: Method, Host, Index, URL, Payload, ContentLength, Label. Đổi các trường ít giá trị: Method, Host, Label sang mảng và gán số thứ tự.

Tiếp đến xử lí URL và Payload, tại mỗi trường ta cần lưu thêm những thông tin sau: Tổng số ký tự; Số ký tự số; Số

ký tự đặc biệt; Số ký tự không phải chữ. Với Payload ta cần thêm trường số lượng đối số (agrument) nhập vào.

Lọc các HTTP requests, sau đó tìm phương thức và gán số thứ tự của phương thức trong mảng vào trường method. Tiếp đó đưa chuỗi tin vào hàm phân loại.

Tại hàm phân loại, ta trích xuất các thông tin Features yêu cầu trong chuỗi. User agent ở giữa chuỗi "User-Agent" và chuỗi ngắt dòng "\r\n". URL ở giữa dấu cách đầu tiên và chuỗi "HTTP". Đối với phương thức GET, payload nằm trong URL sau dấu "?". Các phương thức còn lại payload nằm ở dòng cuối cùng của gói tin.

Với chức năng nghe trực tiếp qua web server log (ở đây là apache và nginx), ta mở file server log sinh từ config, chạy một vòng lặp vô hạn để đọc các dòng log mới. Vòng lặp dừng khi ta bấm Stop trên GUI. Mỗi khi đọc được dòng log mới sẽ đưa vài hàm phân loại.

Dòng log vào được phân tách các trường bằng chuỗi regex. Chuỗi regex này dùng cho định dạng log chuẩn (Common Log Format). Phân tách các trường như pcap và đưa vào mảng numpy. Gọi hàm phân loại lấy kết quả, kết hợp ghi ra file log riêng và in ra GUI.

Với chức năng đọc log hoặc đọc file pcap, ta sử dụng FileChooser của Gtk để lấy đường dẫn tới file và gọi hàm đọc từng dòng giống như nghe trực tiếp. Nếu dòng thuộc log Apache sẽ được đưa vào module phân loại apache phía trên, thuộc pcap thì đưa vào hàm phân loại pcap ở phía trên.

3.1.2. Cấu trúc thư mục:

- Dataset: Chứa bộ dữ liệu gốc và bộ dữ liệu mẫu sau khi trích xuất các trường.
- Features Extract: Code để tách lọc các trường cần thiết cho mỗi mô hình
- RF: Chứa các modules phục vụ cho phân tích dữ liệu:

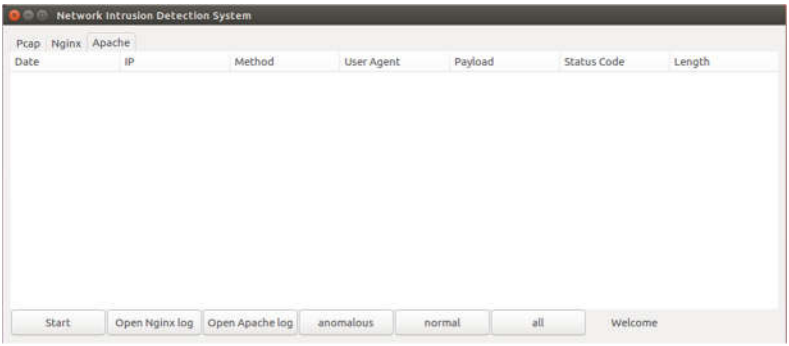
+ *RF.py:* Phần xử lý tạo model và học dữ liệu, tính giá trị độ chính xác, độ phân loại, độ nhạy với mẫu thử. Phân loại log từ bên ngoài.
+ *live_test.py:* Xử lý dữ liệu vào từ gói tin trực tiếp, phân loại và in kết quả.

- Application: Chứa dữ liệu chạy chương trình dạng đồ hoạ

3.1.3. Cài đặt hệ thống:

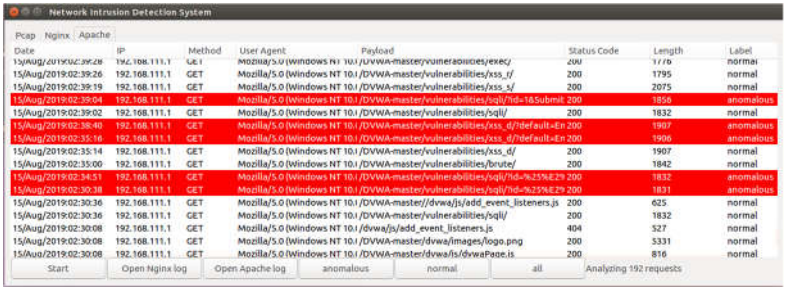
Chương trình xây dựng chạy trên môi trường Python2. Ứng dụng sử dụng một số thư viện cần thiết như sau:

- Numpy
- Scapy
- Scikit-learn
- Nltk
- Gtk

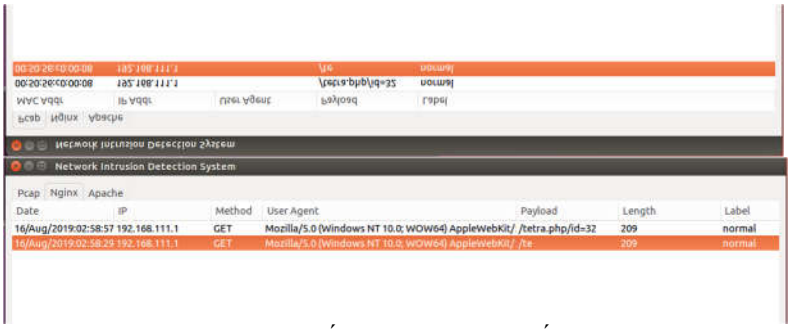


Hình 3. 1 - Giao diện chính chương trình

- Sử dụng giao diện Gtk, phần chính là một notebook với ba list dùng để hiển thị request từ pcap, nginx, apache,… Hiển thị một số trường cơ bản.
- Button Start dùng cho chế độ nghe trực tiếp request từ pcap, nginx log và apache log, đồng thời phân loại normal hoặc anomalous.
- Hai buttons “Open log” để mở các log sinh từ server và phân loại requests.
- Ba buttons tiếp để phân loại requests theo nhãn của chúng.



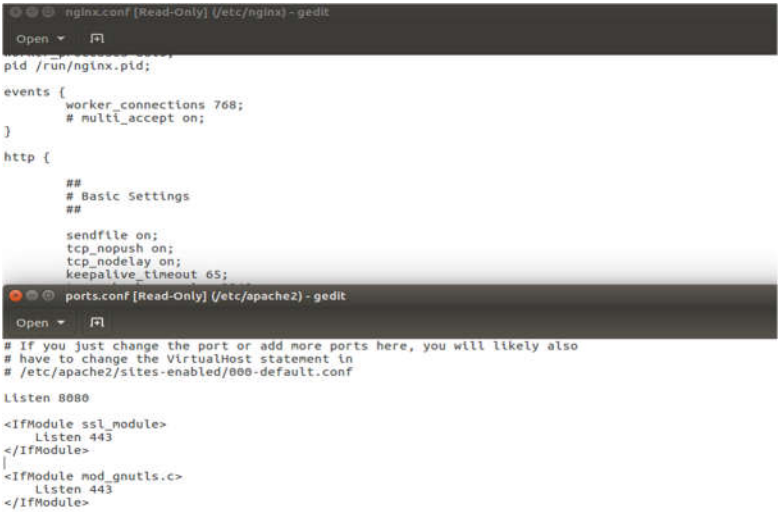
Hình 3. 2 -Đọc một file log từ Apache và phân loại



Hình 3. 3 - Đọc log trực tiếp từ pcap và trực tiếp từ nginx log

3.2. Một số kết quả thử nghiệm hệ thống

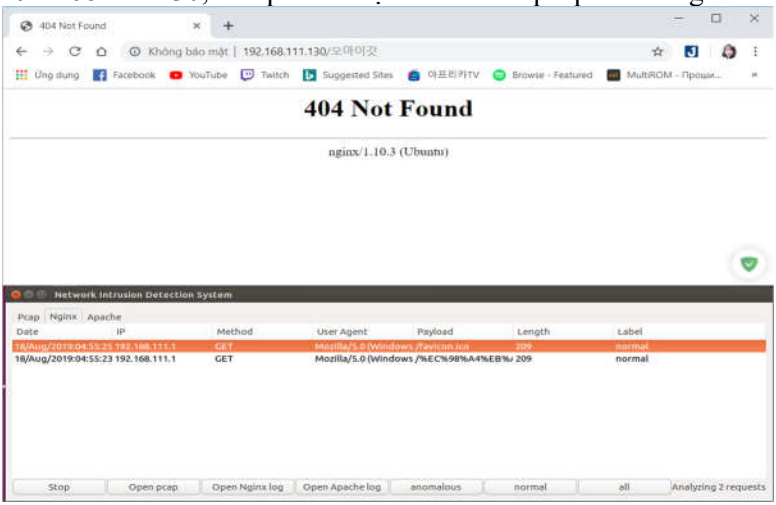
Máy server chạy ubuntu có IP 192.168.111.130. Được cài đặt hai web server apache và nginx. Apache được set vitual host nghe kết nối ở cổng 8080, nginx chạy mặc định ở cổng 80



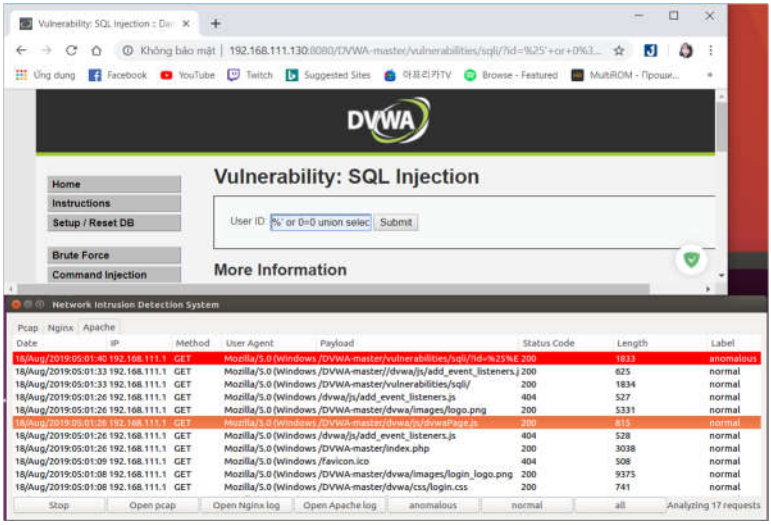
Hình 3. 4 -Cấu hình Apache và Nginx trên server

Mở chương trình bằng file nids.py.

Bắt đầu chế độ nghe bằng nút Start, chương trình sẽ lắng nghe các kết nối từ scapy sniff và file log apache và nginx. Sử dụng một máy bên ngoài truy cập vào nginx thông qua: 192.168.111.130, kết quả sẽ được in ra ở tab pcap và tab nginx



Hình 3. 5 - Nghe gói tin thông qua nginx log

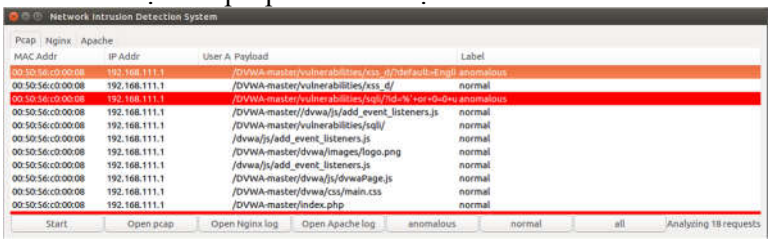


Hình 3. 6 -Nghe gói tin thông qua apache log

Sử dụng máy ngoài truy cập web của apache server thông qua máy chủ có IP 192.168.111.130:8080, khi có kết nối đến, kết quả gói tin được in tại tab pcap và tab apache.

Thử nghiệm lỗi SQL Injection thông qua DVWA được cài trên Apache server sẽ có kết quả như hình trên.

Mở đọc file pcap mới thu được:



Hình 3. 7 - Mở file pcap

Chương trình về cơ bản có thể đọc trực tiếp gói tin, đọc từ file log và phân loại các request tấn công đơn giản theo hướng URL và đối số Payload, hiển thị filter theo nhãn bình thường hay nguy hiểm.

KẾT LUẬN VÀ KIẾN NGHỊ

4.1. Những đóng góp của luận văn

Trong khuôn khổ của luận văn tác giả đã tìm hiểu cơ sở lý thuyết và một số phương thức phát hiện tấn công ứng dụng web dựa trên log truy cập. Tác giả cũng đã tập trung nghiên cứu về thuật toán Random Forest và phương pháp ứng dụng thuật toán áp dụng vào việc phát hiện tấn. Từ những kết quả thực nghiệm trên bộ dữ liệu HTTP CSIC 2010, chúng ta thấy kết quả tương đối tốt. Tuy nhiên phương pháp này có nhược điểm là thời gian chạy chương trình hơi lâu khi phân tích khối lượng dữ liệu lớn.

4.2. Hướng phát triển luận văn

Để giải quyết hạn chế của mô hình học máy được đề xuất ở trên, trong thời gian tới tác giả sẽ chú trọng tìm hiểu, cải tiến nhằm tăng tốc độ phân lớp của giải thuật. Đồng thời, tiến hành thử nghiệm phương pháp trên nhiều bộ dữ liệu khác nhau nhằm đánh giá độ chính xác và ổn định của phương pháp đối với từng loại dữ liệu cụ thể.

Tìm hiểu thêm một số phương pháp phân lớp khác như phương pháp hỗ trợ véc tơ (SVM) để so sánh với thuật toán random forest khi đánh giá kết quả dự đoán. So sánh hiệu quả giữa các phương pháp để có thêm lựa chọn khi phát triển các ứng dụng phát hiện tấn công bằng phương pháp phân lớp dữ liệu.

CÁC TÀI LIỆU THAM KHẢO

[1] M. Zolotukhin, T. Hämäläinen, T. Kokkonen and J. Siltanen, "Analysis of HTTP Requests for Anomaly Detection of Web Attacks," *2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing*, Dalian, 2014, pp. 406-411.

[2] Michie, Donald, David J. Spiegelhalter, and C. C. Taylor. "Machine learning." Neural and Statistical Classification 13.1994 (1994): 1-298.

[3] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.

[4] Meyer, Roger, and Carlos Cid. "Detecting attacks on web applications from log files." *Sans Institute* (2008).

Tài liệu tham khảo từ Internet:

[5] CLASSIFICATION – PART 3, https://tech.3si.vn/2016/03/31/ml-classification-part-3/. Truy cập 18/12/2019