PHÁT HIỆN HÀNH VI CHÈN MÃ DỊCH VỤ WEB

Phạm Hoàng Duy, Nguyễn Ngọc Điệp
Bộ môn An toàn thông tin, Khoa CNTT 1,
Học Viện Công Nghệ Bưu Chính Viễn Thông

Tóm tắt: Nhu cầu giám sát truy nhập tới các dịch vụ Web để phát hiện dạng tấn công từ Web log tăng theo sự phát triển của Internet nhằm đảm bảo chất lượng phục vụ và sự an toàn của các dịch vụ này. Báo cáo này khảo sát hiệu năng phân loại của các mô hình dựa trên học máy để phát hiện hiệu quả hành vi tấn công chèn mã tới các dịch vụ Web. Báo cáo đề xuất việc thu thập và xây dựng bộ mẫu tấn công chèn mã dùng cho việc huấn luyện và đánh giá với hơn 400 nghìn mẫu với 9 dạng. Các thử nghiệm được tiến hành trên tập dữ liệu này với các thuật học: Cây quyết định, rừng ngẫu nhiên, SVM, XGB và mạng học sâu (DNN) cho kết quả khả quan, trong đó DNN đạt giá trị F1 lên tới 97,5%.

Từ khóa: IDS, Phát hiện tấn công, Tấn công chèn mã, An toàn thông tin, Dịch vụ Web, Học máy.

I. GIỚI THIỆU

Với sự phát triển của Internet và các ứng dụng trên Web, việc phát hiện bất thường trong các dịch vụ Web không chỉ có phát hiện thao tác sai của người dùng mà còn cần đảm bảo phát hiện được các hành vi có mục đích xấu làm suy giảm chất lượng phục vụ của web-site, cùng với các hành vi gian lận. Một trong những biện pháp quan trọng hỗ trợ phát hiện bất thường là theo dõi và giám sát các truy nhập từ người dùng tới các máy chủ cung cấp dịch vụ Web, qua đó cung cấp thông tin hiệu quả về các hành vi truy nhập của người dùng và có tác dụng to lớn với việc đảm bảo chất lượng cũng như an toàn của dịch vụ được cung cấp. Dựa trên các thông tin thu được từ dữ liệu log của các dịch vụ Web, nhiều kỹ thuật phát hiện truy nhập bất thường được phát triển và triển khai hiệu quả trong thực tế.

Kỹ thuật phát hiện các truy nhập bất thường dựa trên luật được sử dụng phổ biến nhờ tính dễ nắm bắt và tiếp cận với người quản trị dịch vụ Web. Có hai cách tiếp cận cơ bản để sinh ra các luật. Cách thứ nhất là dựa vào luật tĩnh, được tạo ra một cách thủ công thông qua việc phân tích các hành vi truy nhập của người dùng ghi lại trong file nhật ký (Web-log). Cách tiếp cận khác là tạo ra các luật động bằng cách sử dụng các thuật toán của kỹ thuật khai phá dữ liệu hay học máy.

Đối với kỹ thuật sinh luật tĩnh, trước tiên cần tạo ra kịch bản về tình huống mà người quản trị muốn mô phỏng. Chẳng hạn, nếu có hai tham số khác biệt trong truy nhập tới web-site và việc kết hợp của cả hai tham số này gây ra một sự cố bảo mật, người quản trị cần lập mô hình cho trường hợp này. Bên cạnh các quy tắc này, người quản trị phải thực hiện phân tích tương quan để xem trường hợp đang giải quyết là một cuộc tấn công hay không [1]. Luật có thể chứa nhiều tham số như: khung thời gian, lặp lại mẫu, loại dịch vụ, cổng. Thuật toán sau đó kiểm tra dữ liệu từ các file nhật ký và tìm ra các kịch bản tấn công hay hành vi bất thường. Kỹ thuật sinh luật tĩnh có khả năng phát hiện nhanh và chính xác các tấn công đã xét tới. Tuy nhiên, điểm yếu của kỹ thuật này là khối lượng việc làm thủ công lớn và có khả năng bỏ sót các tấn công tiềm ẩn chưa được tính đến. Trong thực tế, các luật tĩnh thường được ứng dụng để phát hiện ra các tấn công đơn giản và có quy luật, dễ dàng tìm ra các đặc trưng tấn công. Đối với các tấn công phức tạp hơn như kiểu tấn công chèn mã nói chung (chèn lệnh, chèn mã, thay đổi tham số, …) thì kỹ thuật này không có nhiều hiệu quả.

Các phương pháp dựa trên việc sinh luật động [2]–[6] bằng mô hình học máy có thể giải quyết vấn đề của việc sinh luật tĩnh và cho phép phát hiện những tấn công tiềm ẩn chưa được người quản trị biết đến. Các thuật toán tiêu biểu thường sử dụng cho cách tiếp cận này có thể kể đến thuật toán luật kết hợp, cây quyết định, rừng ngẫu nhiên … Nhưng các mô hình học sử dụng các thuật toán này đòi hỏi phải giải quyết được sự phức tạp trong việc phân tích dữ liệu nhiều chiều, thuật toán có độ phức tạp tính toán cao. Mặt khác, bộ dữ liệu sử dụng cho học máy có ảnh hưởng quan trọng tới hiệu năng của việc phát hiện các hành vi tấn công dịch vụ Web. Bộ dữ liệu CSIC 2010 [7] được biết tới như bộ dữ liệu mẫu cho các hành vi truy nhập bình thường cũng như tấn công tới dịch vụ thương mại điện tử với tổng cộng khoảng 60.000 truy vấn nhưng không chỉ rõ các truy vấn tấn công thuộc dạng cụ thể nào. Việc xây dựng bộ dữ liệu đủ lớn và cập nhật thuận tiện sẽ có ích trong việc thử nghiệm các kỹ thuật học máy cũng như xây dựng hệ thống phát hiện tấn công chi tiết cho người quản trị dịch vụ Web.

Rõ ràng, các kỹ thuật học máy cho phép phát hiện các hành vi tấn công một cách hiệu quả và linh hoạt ngay cả với những hành vi mới, mà người quản trị Web có thể chưa thực sự nắm rõ. Vì vậy, báo cáo này nghiên cứu việc áp dụng kỹ thuật học máy phát hiện các dạng tấn công chèn mã một cách chi tiết từ việc phân tích dữ liệu Web-log của các truy nhập người dùng. Cụ thể, các tác giả thực hiện khảo sát và đề xuất mô hình phân loại cho việc phát hiện các hành vi

Tác giả liên hệ: Phạm Hoàng Duy,
Email: duyph@ptit.edu.vn
Đến tòa soạn: 8/2020, chỉnh sửa 09/2020, chấp nhận đăng:10/2020

tấn công chèn mã tới dịch vụ Web cũng như phân biệt với hành vi bình thường sử dụng mô hình học máy, với các thuật toán tiêu biểu bao gồm các thuật toán dựa trên cây quyết định, máy véc-tơ tựa (SVM), và mạng học sâu (DNN). Bên cạnh đó, báo cáo đề xuất bộ dữ liệu mẫu cập nhật chứa các dạng hành vi tấn công chèn mã tới dịch vụ Web một cách chi tiết.

Cấu trúc báo cáo như sau. Phần II trình bày các vấn đề lý thuyết và thực tiễn liên quan tới việc phát hiện các hành vi tấn công dựa trên bài toán phân loại của học máy. Cụ thể, phần này trước tiên giới thiệu 8 hành vi tấn công chèn mã tiêu biểu theo OWSAP và các nghiên cứu về việc sử dụng kỹ thuật học máy cho việc phát hiện và phân loại hành vi tấn công dịch vụ Web. Ngoài ra, phần này trình bày mô hình phân loại đa lớp và các thuật toán tiêu biểu được khảo sát cho việc phân loại và phát hiện các hành vi tấn công dịch vụ Web. Vấn đề biểu diễn hành vi truy nhập của người dùng dịch vụ Web cũng được khảo sát trong phần này.

Phần III đề xuất mô hình phân loại đa lớp cho việc phát hiện các hành vi tấn công cũng như cách thức xây dựng bộ dữ liệu mẫu sử dụng cho việc huấn luyện. Phần này cũng trình bày và phân tích mô hình thử nghiệm cùng khảo sát hiệu năng của các thuật toán học máy với bộ dữ liệu mẫu đã xây dựng. Phần cuối trình bày kết luận và hướng phát triển trong tương lai.

II. PHÁT HIỆN VÀ PHÂN LOẠI HÀNH VI CHÈN MÃ DỰA TRÊN HỌC MÁY

A. Các hành vi chèn mã tới dịch vụ Web

Phần dưới đây trình bày các khái niệm liên quan đến các hành vi tấn công liên quan đến việc chèn các dữ liệu bất thường được chuẩn bị một cách có chủ ý tới các dịch vụ Web, theo phân loại của tổ chức OWSAP.

1) *Chèn mã*

Chèn mã (Code injection) là thuật ngữ chung cho các loại tấn công sử dụng mã mà ở đó, mã được chèn vào ứng dụng rồi được ứng dụng dịch và thực thi. Kiểu tấn công này khai thác khả năng dữ liệu không đáng tin cậy được xử lý không an toàn. Các loại tấn công này thường nhằm vào các trường hợp thiếu xác thực dữ liệu đầu vào/đầu ra thích hợp. Kiểu tấn công này cũng giới hạn trong môi trường dịch của ngôn ngữ sử dụng trong chương trình bị tấn công. Ví dụ, nếu kẻ tấn công có thể chèn mã PHP vào một ứng dụng và thực thi nó thì chúng chỉ khai thác trong phạm vi khả năng PHP.

2) *Chèn lệnh*

Chèn lệnh (Command injection) là hình thức tấn công nhằm thực thi các lệnh tùy ý trên hệ điều hành máy chủ thông qua một ứng dụng dễ bị tấn công. Các cuộc tấn công chèn lệnh có thể xảy ra khi một ứng dụng chuyển dữ liệu không an toàn do người dùng cung cấp (biểu mẫu, cookie, mào đầu HTTP, v.v.) sang hệ thống. Trong cuộc tấn công này, các lệnh hệ điều hành do kẻ tấn công cung cấp thường được thực thi với các đặc quyền của ứng dụng dễ bị tấn công. Các cuộc tấn công chèn lệnh có thể phần lớn là do xác nhận đầu vào không thích đáng.

3) *Tràn bộ đệm*

Tràn bộ đệm (Buffer overflow) có lẽ là dạng lỗ hổng bảo mật phần mềm được biết đến nhiều nhất. Hầu hết các nhà phát triển phần mềm đều biết đến lỗ hổng này, nhưng các cuộc tấn công tràn bộ đệm nhằm vào cả các ứng dụng cũ và mới vẫn còn khá phổ biến. Một phần của vấn đề là có nhiều nguyên nhân dẫn đến việc tràn bộ đệm và một phần khác là do các kỹ thuật ngăn chặn dễ bị vượt qua. Kẻ tấn công sử dụng lỗi tràn bộ đệm để làm hỏng ngăn xếp thực thi của ứng dụng Web. Bằng cách gửi dữ liệu đầu vào được xây dựng một cách cẩn thận tới ứng dụng Web, kẻ tấn công có thể khiến ứng dụng Web thực thi mã tùy ý và thực tế kiểm soát máy tính chạy ứng dụng.

Lỗi tràn bộ đệm có thể có trong cả ứng dụng Web hoặc máy chủ Web hỗ trợ Web tĩnh cũng như động. Tràn bộ đệm có thể gây rủi ro đáng kể cho người dùng các ứng dụng này. Khi các ứng dụng Web sử dụng các thư viện ngoài, chẳng hạn như thư viện đồ họa để tạo hình ảnh, thì chúng có rủi ro tự phơi mình ra trước các cuộc tấn công tràn bộ đệm. Tràn bộ đệm cũng có thể được tìm thấy trong mã ứng dụng Web tùy chỉnh. Các lỗi tràn bộ đệm trong các ứng dụng Web tùy chỉnh ít có khả năng bị phát hiện hơn vì thông thường sẽ có ít tin tặc cố gắng tìm và khai thác các lỗ hổng đó trong một ứng dụng cụ thể. Nếu được phát hiện trong một ứng dụng tùy chỉnh, khả năng khai thác lỗ hổng (ngoài việc làm sập ứng dụng) sẽ giảm đáng kể bởi thực tế là mã nguồn và thông báo lỗi chi tiết cho ứng dụng thường không có sẵn cho tin tặc.

4) *Chèn CRLF*

CRLF thực tế là chuỗi ký tự xuống dòng CR và về đầu dòng LF. Tấn công chèn CRLF xảy ra khi người dùng thực hiện được việc gửi chuỗi CRLF tới một ứng dụng Web. Điều này thường được thực hiện bằng cách sửa đổi tham số HTTP hoặc URL.

5) *Chuỗi định dạng*

Tấn công chuỗi định dạng (Format string) xảy ra khi dữ liệu vào được xử lý như một lệnh được thực hiện bởi ứng dụng Web. Bằng cách này, kẻ tấn công có thể thực thi mã, đọc ngăn xếp hoặc gây ra lỗi phân đoạn trong ứng dụng đang chạy, gây ra các hành vi mới có thể làm tổn hại đến bảo mật hoặc tính ổn định của hệ thống.

Cuộc tấn công có thể được thực hiện khi ứng dụng không xác thực đầu vào một cách đúng đắn. Trong trường hợp này, nếu tham số chuỗi định dạng, ví dụ như *%x*, được chèn vào dữ liệu được gửi đến, chuỗi được phân tích cú pháp bởi hàm định dạng và việc chuyển đổi được mô tả trong tham số được thực thi. Tuy nhiên, hàm định dạng cần nhiều đối số hơn làm đầu vào và nếu các đối số này không được cung cấp, dẫn đến hàm này có thể đọc hoặc ghi ngăn xếp.

6) *Sửa đổi tham số*

Tấn công giả mạo tham số Web dựa trên việc sửa đổi các tham số được trao đổi giữa máy khách và máy chủ để thay đổi dữ liệu ứng dụng, chẳng hạn như thông tin và quyền của người dùng, hay thông tin sản phẩm, v.v. Thông thường, thông tin này được lưu trữ trong *cookie*, các trường dữ liệu ẩn hoặc URL. Kiểu tấn công này có thể được thực hiện bởi người dùng có mục đích xấu muốn khai thác ứng dụng vì lợi ích riêng của họ hoặc kẻ tấn công muốn tấn công người thứ ba bằng cách sử dụng cuộc tấn công trung gian (Man-in-the-middle). Thành công của cuộc tấn công phụ thuộc vào các lỗi trong cơ chế xác thực logic và toàn vẹn. Việc khai thác này có thể dẫn đến các hậu quả khác bao gồm XSS, SQLi.

7) *Mã chéo*

Các tấn công mã chéo XSS (Cross-Site Scripting) là một loại tấn công chèn dữ liệu, trong đó các đoạn mã (script) độc hại được chèn vào các trang web. Các tấn công XSS

xảy ra khi kẻ tấn công sử dụng ứng dụng Web để gửi mã độc, thường ở dạng đoạn mã phía trình duyệt, đến một người dùng cuối khác. Các lỗ hổng cho phép các tấn công này thành công khá phổ biến và xảy ra ở bất cứ ứng dụng Web nào sử dụng đầu vào từ người dùng trong kết quả đầu ra nó tạo ra mà không kiểm tra.

Kẻ tấn công có thể sử dụng XSS để gửi đoạn mã độc hại cho người dùng không ngờ tới. Trình duyệt người dùng cuối không có cách nào để biết rằng đoạn mã không đáng tin cậy và sẽ thực thi đoạn mã này. Vì trình duyệt cho rằng đoạn mã đến từ một nguồn đáng tin cậy, đoạn mã này có thể truy cập bất kỳ *cookie*, mã thông báo phiên hoặc thông tin nhạy cảm nào khác được trình duyệt giữ lại và sử dụng với trang web đó. Các đoạn mã này thậm chí có thể viết lại nội dung của trang HTML.

8) *Chèn mã SQL*

Một tấn công chèn mã SQLi (SQL injection) bao gồm chèn một truy vấn SQL thông qua dữ liệu đầu vào từ máy khách đến ứng dụng Web. Việc khai thác SQLi thành công dẫn tới khả năng lộ dữ liệu nhạy cảm từ cơ sở dữ liệu, sửa đổi dữ liệu hay thực thi các thao tác quản trị trên cơ sở dữ liệu và trong một số trường hợp có thể điều khiển hệ điều hành. Các tấn công SQLi là một kiểu tấn công chèn dữ liệu, trong đó các lệnh SQL được đưa vào dữ liệu đầu vào để tác động đến việc thực thi các lệnh SQL được xác định trước.

B. Kỹ thuật học máy trong phát hiện tấn công

Các kỹ thuật phát hiện hành vi truy nhập bất thường dựa trên các luật mà chúng biểu diễn các hành vi của hệ thống. Kết quả phân tích dựa vào tập luật này cho phép xác định hành vi cụ thể thuộc về dạng tấn công hay bình thường. Các kỹ thuật phát hiện hành vi tấn công kiểu này có thể dựa trên bộ phân loại sử dụng kỹ thuật học máy hoạt động theo giả thuyết tổng quát sau đây:

> *"Bộ phân loại có khả năng phân biệt các lớp bình thường và tấn công có thể học được trong không gian đặc trưng nhất định".*

Phân loại đa lớp giả định rằng dữ liệu huấn luyện chứa các mục gán nhãn thuộc nhiều lớp thông thường như trong các báo cáo [8] và [9]. Các kỹ thuật phát hiện hành vi bất thường như vậy dựa vào bộ phân loại để phân biệt giữa từng lớp hành vi bình thường với các lớp còn lại.

Một mẫu dữ liệu cần kiểm tra được coi là bất thường nếu nó không được phân loại là bình thường bởi bất kỳ bộ phân loại nào hay tập các luật tương ứng với lớp đó. Một số kỹ thuật phân loại phụ kết hợp điểm số tin cậy với dự đoán của bộ phân loại. Nếu không có bộ phân loại nào đủ tin cậy để phân loại mẫu dữ liệu cần kiểm tra là bình thường thì mẫu dữ liệu này được coi là bất thường.

Kỹ thuật dựa trên luật cho bài toán đa lớp cơ bản bao gồm hai bước. Bước đầu tiên là học các luật từ dữ liệu huấn luyện bằng thuật toán học như cây quyết định, rừng ngẫu nhiên (Random Forest), ... Mỗi luật có độ tin cậy tương ứng mà giá trị này tỷ lệ với các trường hợp huấn luyện được phân loại chính xác theo luật và tổng số các trường hợp huấn luyện đúng với luận đó. Bước thứ hai là tìm luật biểu diễn tốt nhất cho trường hợp cần kiểm tra. Nghịch đảo của độ tin cậy ứng với các luật tốt nhất là giá trị bất thường của trường hợp cần kiểm tra. Một số biến thể kỹ thuật dựa trên luật cơ bản đã được mô tả trong các nghiên cứu [2]–[6].

Việc khai thác luật kết hợp [10] đã được sử dụng để phát hiện bất thường theo kiểu một lớp bằng cách tạo ra các luật từ dữ liệu theo kiểu không giám sát. Luật kết hợp được tạo từ bộ dữ liệu có phân loại. Để đảm bảo rằng các luật liên kết chặt chẽ với các mẫu, người ta sử dụng ngưỡng hỗ trợ để loại bỏ các luật có mức hỗ trợ thấp. Các kỹ thuật dựa trên khai thác luật kết hợp đã được sử dụng để phát hiện hành vi xâm nhập mạng như trong các nghiên cứu [1], [11], [12].

Mô hình FARM (Fuzzy Association Rule Model) được phát triển bởi Chan và cộng sự [13] nhắm đến các cuộc tấn công giao thức SOAP hoặc tấn công XML tới các dịch vụ Web. Hầu hết các nghiên cứu về hệ thống phát hiện bất thường trên máy chủ và hệ thống mạng chỉ có thể phát hiện các cuộc tấn công ở mức thấp của mạng trong khi các ứng dụng Web hoạt động ở mức ứng dụng cao hơn. Mô hình luật kết hợp mờ FARM là hệ thống phát hiện bất thường cho các vấn đề an ninh mạng đặc biệt đối với các ứng dụng thương mại điện tử dựa trên Web.

Phát hiện hành vi tấn công sử dụng kỹ thuật sinh luật với phân loại đa lớp có thể sử dụng các thuật toán mạnh để phân biệt các trường hợp thuộc các lớp khác nhau. Điều này cho phép xác định một cách chi tiết các nhóm hành vi bình thường cũng như bất thường. Mặt khác, giai đoạn kiểm chứng của kỹ thuật này thường rất nhanh vì mỗi trường hợp cần kiểm tra được so sánh với mô hình tính toán trước.

Ngoài các mô hình dựa trên luật, mô hình học sâu cũng đã được sử dụng cho việc phân biệt và phát hiện cách hành vi truy nhập trái phép từ dữ liệu mạng. Các tác giả của báo cáo [14] sử dụng mạng nơ-ron hồi quy để tự động phân lớp dữ liệu truy nhập, chẳng hạn như các truy vấn *http*, bằng thuật học hồi quy thời gian thực. Sau đó, việc phân loại truy nhập tiếp theo sử dụng kỹ thuật véc-tơ học máy. Việc sử dụng thuật học thời gian thực giúp cho phương pháp đề xuất có khả năng áp dụng cho các hệ thống theo dõi thời gian thực và có thể mở rộng từng bước.

Các báo cáo [15], [16] sử dụng kiến trúc bộ nhớ dài-ngắn hạn LSTM (Long-Short Term Memory) cho mạng nơ-ron hồi quy để xây dựng mô hình phát hiện xâm nhập với tập dữ liệu thử nghiệm KDD 99 [17]. Các tác giả của [15] mở rộng kiến trúc LSTM bằng cách cho phép gán trọng số thích ứng giữa các phần tử trong mạng cho phép các phần tử mạng chống lại trạng thái không mong muốn từ các đầu vào. Kết quả thu được khá khả quan với mức độ phát hiện đạt trên 90%. Tuy nhiên, báo cáo [16] chỉ sử dụng một phần của tập dữ liệu KDD 99 để làm dữ liệu huấn luyện.

Các tác giả [18] đánh giá khả năng của mạng học sâu trong việc phân loại các hành vi truy nhập bất thường với cùng bộ dữ liệu. Kết quả chứng tỏ khả năng học phân loại các hành vi ưu việt của mạng học sâu.

Với việc xây dựng mô hình phân loại dựa trên bộ dữ liệu gán nhãn thì tính chính xác của các nhãn gán cho các lớp bình thường khác nhau có ảnh hưởng quyết định đến hiệu năng của mô hình phân loại, mà trên thực tế thường khó để có được bộ dữ liệu hoàn hảo. Mặt khác, các bộ dữ liệu sử dụng trong các mô hình học máy ít được phổ biến phổ biến rộng rãi, ngoại trừ bộ dữ liệu hành vi tấn công Web CSIC 2010 [7].

C. Mô hình học phân loại cho phát hiện tấn công

1) *Mô hình khái quát*

Bài toán phân loại, một trong những bài toán cơ bản của học máy, nhằm xây dựng mô hình phân loại từ một tập các dữ liệu đã được dán nhãn (giai đoạn huấn luyện), tiếp theo, phân loại các trường hợp cần kiểm tra vào một trong số các

lớp bằng cách áp dụng mô hình đã học (giai đoạn kiểm chứng). Như đã giới thiệu ở phần trước, với việc phát hiện các hành vi truy nhập tấn công các kỹ thuật học máy giúp tạo điều kiện xây dựng các bộ phân loại, tự động tìm hiểu các đặc trưng của mỗi lớp cần phân loại, chẳng hạn như các hành vi tấn công và bình thường bằng cách học từ dữ liệu mẫu. Cách tiếp cận này cho phép nâng cao tính tự động khi đối mặt với các mối đe dọa mới như là sửa đổi kỹ thuật tấn công cũ nhưng vẫn giữ lại một số đặc điểm của việc xâm nhập trước đây.

Để sử dụng kỹ thuật học máy để phát hiện và phân loại các hành vi truy nhập của người dung, trước hết phải xây dựng các tập dữ liệu được gán nhãn để huấn luyện. Mỗi bản ghi của tập dữ liệu mô tả các đặc trưng và một nhãn (cũng được gọi là lớp). Các đặc trưng này bắt nguồn từ một số đặc điểm cụ thể hành vi người dùng, chẳng hạn như kích thước của truy vấn hoặc tần suất của một đoạn tham số nhất định trong truy vấn; nhãn là giá trị nhị phân cho biết truy vấn là bình thường hay không. Việc phân tích để tìm ra các đặc trưng trong hành vi của người dùng có thể áp dụng các kỹ thuật cơ bản như xác định các cấu trúc hay thành phần trong các dữ liệu thu thập được. Các phân tích thống kê bổ sung thêm các đặc trưng về hành vi của người dùng như biểu diễn mức độ tương quan giữa các thành phần dữ liệu hay biểu diễn trừu tượng về cấu trúc dữ liệu thu thập được.

Việc áp dụng mô hình phân loại cho việc phát hiện tấn công sử dụng hai giai đoạn chính. Giai đoạn đầu thực hiện việc huấn luyện nhằm xây dựng mô hình học máy thích ứng với các đặc trưng của dữ liệu đầu vào nhờ thuật toán học máy. Giai đoạn sau thực hiện việc dự đoán, đánh giá chất lượng của mô hình học được bằng việc sử dụng dữ liệu đánh giá hay kiểm tra. Kết quả thu được cho biết chất lượng hay hiệu năng của mô hình thu được.

Thuật toán huấn luyện phân tích các bản ghi được chỉ định để huấn luyện để tạo ra một mô hình toán học ánh xạ mối quan hệ của các đặc trưng và các nhãn của bản ghi truy nhập của người dùng. Mô hình đó, gọi là bộ phân loại, được sử dụng để dự đoán lớp của mỗi bản ghi trong dữ liệu kiểm chứng hoặc các bản ghi được chỉ định để thử nghiệm. Bộ phân loại không thể đọc các nhãn khi đưa ra các dự đoán; nhãn dữ liệu thử nghiệm chỉ được sử dụng khi dự đoán được so sánh với nhãn thực trong phân tích hiệu suất tiếp theo.

2) *Các thuật toán học máy thử nghiệm*

Phần dưới đây giới thiệu một số thuật toán học máy tiêu biểu, sử dụng để xây dựng mô hình phân loại cho nhiều dạng tấn công vào dịch vụ Web. Trước hết, báo cáo giới thiệu các thuật toán tiêu biểu cho việc phân loại như SVM và một số thuật học tăng cường. Ngoài ra, báo cáo cũng đề cập tới mạng học sâu. Đây là mô hình đã có tác động sâu rộng đến ứng dụng mô hình học máy gần đây, đặc biệt trong lĩnh vực như nhận dạng tiếng nói, xử lý ảnh và xử lý ngôn ngữ tự nhiên. Đặc trưng nổi bật của mô hình học sâu là việc sử dụng khối lượng lớn dữ liệu so với cách tiếp cận truyền thống. Các mô hình sử dụng nhiều tham số cho phép khai thác các thông tin trong tập dữ liệu khổng lồ một cách hiệu quả hơn.

a) SVM

SVM [19], Support Vector Machine, được coi như một trong những bộ phân loại chính xác nhất cho văn bản [20]. SVM dựa trên việc sinh ra các hàm từ tập các dữ liệu huấn luyện được dán nhãn. Các hàm có thể là hàm phân loại mà kết quả ở dạng nhị phân. Các hàm cũng có thể là hàm hồi quy khái quát.

Với mục tiêu phân loại, các hàm SVM tìm kiếm một siêu phẳng trong không gian nhiều chiều để phân tách các lớp dữ liệu thành hai phần riêng biệt. Dữ liệu huấn luyện ban đầu được ánh xạ phi tuyến vào không gian các đặc trưng có chiều lớn hơn, và sau đó xây dựng siêu phẳng sao cho các mẫu âm và dương của dữ liệu huấn luyện được phân tách với biên tối đa. Điều này tạo ra ranh giới quyết định phi tuyến trong không gian đầu vào. Các yêu cầu tính toán của SVM khá cơ bản và không có gì đặc biệt.

b) Rừng ngẫu nhiên

Thuật toán rừng ngẫu nhiên [21] RF (Random Forest) thường được sử dụng trong quá trình huấn luyện của mô hình học máy phân loại. Đây thuật toán căn bản cho phép sinh tập luật phân loại từ các dữ liệu đầu vào dựa trên việc kết hợp các cây quyết định riêng lẻ. Trong thực tế, RF đã trở thành một công cụ tin cậy cho phân tích phân loại cũng như hồi quy dữ liệu.

c) XGB

XGB [22], eXtreme Gradient Boost, là một trong những kỹ thuật học máy tăng cường hiệu quả cao để xây dựng cây phân loại. Nhờ vào việc tối ưu hóa về đệm, nén dữ liệu và khả năng mở rộng, XGB có thể hoạt động với khối lượng lớn dữ liệu song sử dụng ít tài nguyên hơn nhiều so với các hệ thống hiện có. XGB được cung cấp dưới dạng bộ phần mềm mã nguồn mở và là một trong những công cụ đã chứng tỏ được năng lực như trong cuộc thi KDD Cup 2015.

d) Mạng học sâu

Trong lĩnh vực an ninh mạng, các mạng học sâu thu hút được sự quan tâm do hiệu suất đáng kinh ngạc và tiềm năng của các mạng học sâu đã được thể hiện trong các vấn đề khác nhau mà từng được coi là không thể giải quyết được trong quá khứ. Học sâu về cơ bản là một lĩnh vực hẹp của học máy qua việc mô phỏng các chức năng của bộ não con người và do đó còn có tên là mạng lưới thần kinh nhân tạo.

Mạng học máy *perceptron* thông thường sử dụng ba lớp (lớp đầu vào, lớp ẩn, và lớp đầu ra) phục vụ cho việc khai thác thông tin nhờ vào việc huấn luyện lớp ẩn và lớp đầu ra theo dữ liệu huấn luyện được cung cấp. Như vậy, mạng học máy này có thể "hình dung" được cách thức biểu diễn của tập dữ liệu. Mạng *perceptron* sâu nhiều lớp, mạng nơ-ron sâu tích chập và mạng nơ-ron hồi quy là cách tiếp cập phổ biến hiện thời trong các mô hình học sâu. Nguyên nhân chủ yếu cho việc dùng mô hình học sâu chính là tính hiệu quả thực tế so với các cách tiếp cận khác. Hơn thế, mô hình học sâu còn cung cấp các kỹ thuật mới và tiên tiến về mặt lý thuyết như các biến thể của các thuật học.

Sự thành công của mô hình học sâu cần phải kể đến sự phổ biến của tính toán hiệu năng cao sử dụng bộ xử lý đồ họa. Khi biểu diễn dưới dạng các ma trận véc-tơ, việc tính toán được tăng tốc nhờ phần cứng và thư viện đồ họa được tối ưu hóa. Kết quả huấn luyện và kiểm chứng mô hình được tiến hành một cách nhanh chóng và hiệu quả.

D. Biểu diễn dữ liệu hành vi người dùng

Các tương tác của người dùng với các dịch vụ Web được lưu giữ lại trong các file Web-log của máy chủ. Thông tin quan trọng trong các file này chính là các truy vấn dịch vụ Web đóng gói theo giao thức HTTP. Các thông tin trong phần mào đầu của truy vấn HTTP này được sử dụng để huấn luyện.

Các thông tin thu thập được từ các truy vấn HTTP cần được phân tích để trích xuất các thông tin quan trọng. Chẳng

hạn như, trước tiên URL được trích xuất và ghép nối với phương thức HTTP (ví dụ: GET, POST, PUT, v.v.). Mặt khác, các truy vấn HTTP cũng chứa các tham số dành cho các chương trình Web, ví dụ: *parameter1 = value1* & *parameter2 = value2*. Các tham số này cũng cần được trích xuất và biến đổi phù hợp theo mô hình học máy được sử dụng.

Phần dưới đây khảo sát một số cách biểu diễn hành vi của người dùng bằng các đặc trưng cơ bản và TF-IDF từ dữ liệu truy vấn thu thập được.

1) *Biểu diễn đặc trưng cơ bản*

Mỗi truy vấn tới các ứng dụng Web qua giao thức HTTP có thể được biểu diễn bằng các đặc trưng cơ bản [23] dựa trên các thống kê đơn giản về các tham số được gửi, thống kê cấu trúc trong các tham số cũng như là các đường dẫn (URI) được người dùng sử dụng.

Đặc trưng về truy vấn bao gồm:

- Độ dài truy vấn
- Độ dài các tham số
- Độ dài mô tả thông tin host
- Độ dài của mào đầu "Accept-Encoding"
- Độ dài của mào đầu "Accept-Language"
- Độ dài của mào đầu "Content-Length"
- Độ dài của mào đầu "User-Agent"
- Giá trị byte nhỏ nhất trong truy vấn
- Giá trị byte lớn nhất trong truy vấn

Đặc trưng về tham số được gửi tới máy chủ dịch vụ:

- Số lượng các tham số
- Số các chữ trong tham số
- Số các ký tự khác trong tham số

Đặc trưng về đường dẫn tới các trang ứng dụng:

- Số chữ số trong đường dẫn tới trang
- Số các ký tự khác trong đường dẫn tới trang
- Số lượng các chữ cái trong đường dẫn tới trang
- Số lượng các ký tự đặc biệt trong đường dẫn tới trang
- Số lượng từ khóa trong đường dẫn tới trang
- Độ dài đường dẫn

Việc phân tích thống kê giúp bổ sung thêm các thông tin về các đặc trưng như: phát hiện các liên kết giữa các đặc trưng với nhau cũng như định lượng mối tương quan với dạng hành vi người dùng mà mô hình học máy muốn phân tách. Việc sử dụng toàn bộ hay một phần các đặc trưng kể trên có tác động trước hết đến hiệu năng của thuật toán phân loại. Nói cách khác, chất lượng của việc phân tích phụ thuộc trực tiếp vào việc lựa chọn đặc trưng sử dụng trong mô hình biểu diễn hành vi truy vấn của người dùng.

2) *Biểu diễn đặc trưng sử dụng TF-IDF*

Chuỗi các tham số trong truy vấn HTTP có thể coi như dạng văn bản trao đổi giữa người dùng và ứng dụng Web, vậy nên chuỗi này có thể được mã hóa bằng các sử dụng độ đo TF-IDF trên các từ khóa hay cụm từ khóa trong các tham số truy vấn HTTP. Độ đo TF-IDF là độ đo phổ biến trong phân tích văn bản [24], nó cho biết tần suất xuất hiện của từ khóa TF (Term Frequency) và nghịch đảo của nó IDF (Inverse Document Frequency). Các độ đo TF và IDF được xác định như công thức sau:

$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}} \quad (1)$$

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

Trong đó f(*t, d*) là số lần xuất hiện của từ khóa *t* trong tham số truy vấn của người dùng; max{f(*w, d*): $w \in d$}: số lần xuất hiện nhiều nhất của từ khóa *w* bất kỳ trong truy vấn; |*D*|: tổng các tham số truy vấn của người dùng; $|d \in D : t \in d|$: số văn bản chứa *t*.

Độ đo TF-IDF này cho phép đánh giá sự tương đồng giữa các tham số truy vấn HTTP. Với dữ liệu từ Web-log, các tham số sử dụng trong các truy vấn được tách ra khỏi nội dung của truy vấn nguyên thủy, tiếp theo các ký tự đánh dấu (như '=') trong tham số truy vấn được loại bỏ. Để xác định độ đo TF-IDF, thông thường chuỗi các tham số được chuyển thành các cụm 3 từ hoặc 3 ký tự và tiến hành xác định TF-IDF cho các cụm 3 này (*n-gram*=3).

3) *Nhận xét*

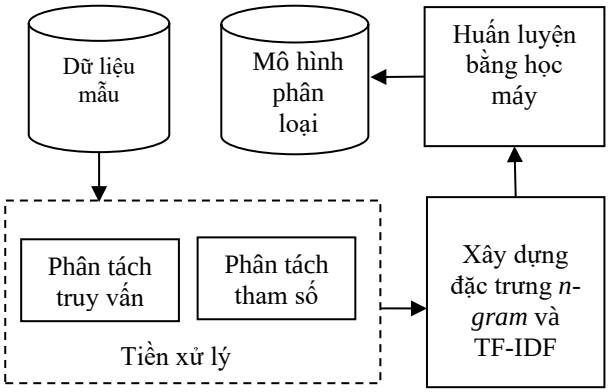
Cách thức biểu diễn hành vi truy nhập dịch vụ Web của người dùng có ảnh hưởng trực tiếp lên hiệu năng của mô hình học máy và việc phát hiện hiệu quả hành vi bất thường. Cách thức biểu diễn kết hợp *n-gram* và TF-IDF cung cấp lượng thông tin lớn hơn rất nhiều so với các thức biểu diễn đặc trưng cơ bản. Vì vậy, thời gian cũng như độ phức tạp để xử lý dữ liệu theo kiểu TF-IDF cũng lớn hơn rất nhiều so với cách thức biểu diễn phổ thông. Điều này khiến cho khi khối lượng dữ liệu tăng lên đáng kể thì đây sẽ là vấn đề quan trọng cần giải quyết. Trong phạm vi báo cáo này, kết hợp *n-gram* và TF-IDF được lựa chọn cho mô hình học máy vì khả năng biểu diễn phong phú các đặc trưng của tượng tác người dùng với ứng dụng Web.

III. PHÂN LOẠI HÀNH VI CHÈN MÃ TỚI DỊCH VỤ WEB

Mô hình học máy phân loại là mô hình đáng quan tâm cho việc phát hiện hành vi chèn mã từ người dùng tới các ứng dụng Web nhờ vào khả năng nhận biết các dạng hành vi của người dùng từ tập dữ liệu mẫu cho trước. Dữ liệu về các tương tác của người dùng tới ứng dụng Web cần được phân chia thành hai nhóm: bình thường và bất thường. Dạng bình thường chứa đựng các truy vấn Web của người dùng thông thường khi sử dụng các dịch vụ Web. Dạng bất thường chính là các dạng truy vấn trong đó chứa được các cấu trúc dữ liệu đặc biệt nhằm thực hiện việc tấn công chèn mã tới các ứng dụng hay máy chủ ứng dụng Web như được giới thiệu trong phần trước đó. Các dạng hành vi bình thường và bất thường trong tập dữ liệu mẫu cần được gán nhãn phân biệt với nhau.

Phần dưới đây trình bày chi tiết về hiệu năng mô hình học phân loại áp dụng cho việc phát hiện các hành vi chèn mã tới ứng dụng Web dựa trên các mô hình cây quyết định, học tăng cường, SVM và mạng học sâu.

A. Mô hình thử nghiệm



Hình 1. Mô hình huấn luyện phân loại hành vi tấn công

Hình 1 biểu diễn mô hình thử nghiệm phân loại cách hành vi tấn công chèn mã tới máy chủ Web. Dữ liệu đầu vào của mô hình là các thông tin thu thập từ file nhật ký của máy chủ dịch vụ Web được người quản trị xác nhận bình thường và các mẫu tấn công. Việc tiền xử lý nhằm loại bỏ dữ liệu dư thừa như các truy vấn trùng nhau, có lỗi hay các từ khóa đặc trưng cho công cụ sinh các mẫu tấn công. Việc này làm giảm số lượng các mẫu truy nhập không bình thường song cũng làm giảm việc huấn luyện bị thiên lệch về một số các mẫu truy nhập tấn công cụ thể. Chi tiết về bộ dữ liệu dùng để huấn luyện sẽ được trình bày trong phần sau.

Sau tiền xử lý, dữ liệu được tiến hành phân tích TF-IDF dựa trên các cụm 3 ký tự thay vì các cụm 3 từ (*n-gram*=3). Thực nghiệm cho thấy các cụm 3 ký tự giúp giảm thiểu không gian biểu diễn các truy vấn cũng như mang lại hiệu năng tốt hơn là các cụm 3 từ.

Việc huấn luyện bộ phân loại triển khai các thuật toán học máy trình bày ở phần trước bao gồm *SVM* tuyến tính, rừng ngẫu nhiên, *xgb* và mạng học sâu để phân biệt các hành vi tấn công kiểu chèn mã tới các dịch vụ Web. Việc này cho phép đánh giá hiệu năng của các thuật toán để lựa chọn thuật toán tối ưu cho mô hình phân loại. Mô hình phân loại đã huấn luyện được được lưu lại phục vụ cho việc phân tích các hành vi truy nhập ghi nhận được.

Mô hình thử nghiệm trên được triển khai bằng môi trường lập trình Python và bộ thư viện *scikit-learn, xgboost, keras*, cho phép triển khai nhanh chóng và hiệu quả các mô hình học máy. Mặt khác, môi trường phát triển này cũng cho phép kết nối thuận tiện với các hệ quản trị cơ sở dữ liệu phù hợp với việc quản lý dữ liệu Web log lớn như MongoDB, Spark.

B. Xây dựng bộ dữ liệu

Bộ dữ liệu HTTP CSIC 2010 [7] là bộ dữ liệu mẫu phổ biến được sử dụng trong việc đánh giá và thử nghiệm hiệu năng của các mô hình phát hiện truy nhập bất thường trong lĩnh vực nghiên cứu. Mặc dù, bộ dữ liệu chứa các mẫu truy nhập bất thường ở nhiều dạng như tấn công XSS, chèn mã SQL song việc áp dụng bộ dữ liệu này cho việc phân tích hành vi truy nhập người dùng tới dịch vụ Web cụ thể không thực sự phù hợp. Vì vậy, việc xây dựng bộ dữ liệu phù hợp cho các dịch vụ Web cần được theo dõi và giám sát đóng vai trò quan trọng trong việc theo dõi và phân tích truy nhập người dùng. Báo cáo đề xuất cách thức bán tự động để xây dựng bộ dữ liệu dùng cho việc xây dựng mô hình phân loại sử dụng kỹ thuật học máy sử dụng công cụ đánh giá an ninh .

Các công cụ kiểm tra đánh giá an ninh cho các dịch vụ Web cung cấp nguồn quan trọng các mẫu truy nhập bất thường dưới nhiều dạng khác nhau như tấn công chiếm quyền, XSS, hay chèn mã SQL. Các mẫu này rất hữu ích cho việc xây dựng bộ dữ liệu mẫu. Tuy nhiên, các định dạng mẫu có thể không hoàn toàn phù hợp với hệ thống phân tích truy nhập dịch vụ Web của người dùng cuối. Bên cạnh các mẫu truy nhập tấn công chèn mã, các mẫu truy nhập bình thường được sinh tự động bằng công cụ dò quét cấu trúc dịch vụ Web. Với các trang Web động, quá trình dò quét được hỗ trợ thủ công bởi người quản trị. Như vậy, hệ thống sẽ được cung cấp 2 nhóm dữ liệu truy nhập mẫu bao gồm mẫu tấn công chi tiết và mẫu bình thường.

OWASP Zed Attack Proxy (ZAP) là một trong những công cụ bảo mật mã nguồn mở phổ biến nhất và được duy trì một cách tích cực nhờ cộng đồng người dùng đông đảo. ZAP có thể giúp người quản trị dịch vụ Web tự động tìm vấn đề an ninh trong các ứng dụng Web nhất là trong giai đoạn phát triển và thử nghiệm các ứng dụng. Không những thế, ZAP cũng là một công cụ hữu ích cho người kiểm thử xâm nhập có kinh nghiệm sử dụng để kiểm tra bảo mật thủ công.

Trong báo cáo này, ZAP được sử dụng để sinh ra các mẫu truy nhập tấn công tới các dịch vụ Web cần được theo dõi và giám sát. ZAP được cài đặt ở chế độ hoạt động tối đa để thu được các dạng truy nhập chèn mã (tấn công XSS, chèn mã SQL, hay thay đổi tham số...) cũng như số mẫu sinh ra nhiều nhất. Các mẫu này được phân loại và lưu lại dưới dạng file bán cấu trúc tương ứng để xử lý sau này cho các loại tấn công khác nhau.

Bên cạnh các công cụ phân tích an ninh, ZAP cung cấp cơ chế dò quét cấu trúc dịch vụ Web thông qua dịch vụ *spider* và *AJAX spider*. Các công cụ này cho phép lưu lại các thông tin về cấu trúc các trang dịch vụ và lưu vào trong file riêng làm các mẫu hành vi bình thường. Ngoài ra, các dữ liệu về hành vi bình thường cũng được thu thập từ file nhật ký của ứng dụng Web.

Các tác giả tiến hành sử dụng công cụ ZAP để thu thập dữ liệu về truy nhập dịch vụ Web tới 2 web-site thử nghiệm trong đó có 1 web-site dựng nên từ dịch vụ web nổi tiếng DVWA cho việc thử nghiệm kiểm tra các lỗ hổng Web phổ biến. Các dữ liệu thu thập từ file nhật ký kết hợp với dữ liệu được sinh từ bộ công cụ ZAP, sau khi loại bỏ trùng lặp và từ khóa liên quan đến công cụ ZAP tạo thành bộ dữ liệu mẫu. Chi tiết bộ dữ liệu này như trong bảng dưới đây.

Bảng 1. Bộ dữ liệu mẫu

Phân loại	**Số lượng**	**Tỷ lệ**	**Kiểu truy nhập**
normal	8479	3%	Bình thường
codeinj	43746	11%	Chèn mã
cmdinj	62942	16%	Chèn lệnh
buf	2435	1%	Tràn bộ đệm
crlf	17815	5%	Chèn CRLF
fstr	1333	1%	Chuỗi định dạng
param	5485	2%	Thay đổi tham số
sqli	264100	65%	Chèn mã SQL
xss	2144	1%	Tấn công XSS

Trong tổng số hơn 400.000 dữ liệu khác biệt về các truy nhập của người dùng, có trên 65% dữ liệu là về hành vi chèn mã SQL theo sau là chèn lệnh(*cmdinj*) và chèn mã (*codeinj*) với tỷ lệ khoảng 10% trên tổng số dữ liệu. Tỷ lệ các truy nhập bình thường của người dùng chỉ chiếm số nhỏ là 3%. Điều này cũng phản ánh thực tế cấu trúc dữ liệu trong truy nhập thông thường của người dùng không đa dạng và phong phú như trong các truy vấn bất thường có mục đích xấu. Rõ ràng, dữ liệu không cân bằng đặt ra nhiều thách thức cho các thuật toán học máy.

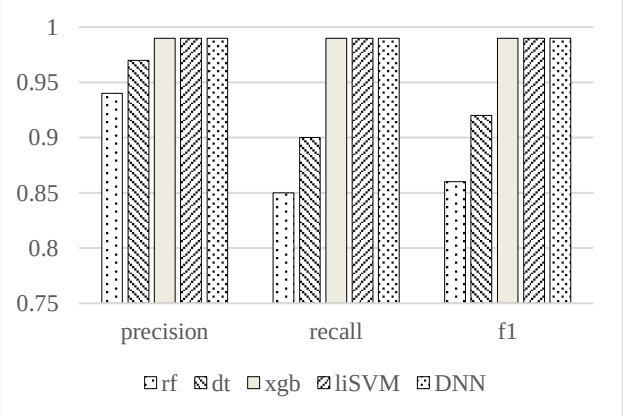
C. Thử nghiệm và đánh giá

1) *Thử nghiệm*

Như đã trình bày ở phần trên, hệ thống phát hiện truy nhập tấn công được phát triển dựa trên môi trường Python 3.7 và bộ thư viện scikit-learn. Bộ dữ liệu mẫu thu được từ phần trên được phân chia theo tỷ lệ 8:2 trên tổng số hơn 400.000 mẫu để huấn luyện và kiểm tra. Việc phân chia được thực hiện một cách ngẫu nhiên song vẫn duy trì tỷ lệ phân bố các mẫu tương đương nhau với cả hai tập dữ liệu huấn luyện và kiểm tra.

Các dữ liệu huấn luyện và kiểm tra được biến đổi sử dụng cấu trúc bộ ba với 3 ký tự (*n-gram = 3* và đơn vị là ký tự) và tính toán TF-IDF từ các cụm 3 này từ chuỗi truy nhập của người dùng. Thực tế, rất nhiều chuỗi truy nhập của người dùng và các tham số trong truy vấn được mã hóa dưới dạng các chuỗi số nên việc lựa chọn đơn vị phân tích là ký tự phù hợp hơn là đơn vị từ. Hơn thế, đơn vị phân tích là ký tự đảm bảo kích cỡ của bộ từ vựng hợp lý và hiệu năng tốt so với việc sử dụng đơn vị phân tích là từ.

Thuật toán cây quyết định (decision tree) được sử dụng ngoài các thuật học đã nêu trong phần trước để xây dựng năng lực học cơ bản cho các thuật học khác, bao gồm *dt*: cây quyết định; *rf*: rừng ngẫu nhiên, *liSVM*: SVM tuyến tính dùng chiến lược *ovr* (one-versus-rest), *xgb*: eXtreme Gradient Boost, *DNN*: mạng học sâu. Hiệu năng của các thuật toán xem xét sử dụng các độ đo tiêu chuẩn là *precision*, *recall*, và *f1*. Các tham số cấu hình tương ứng của các bộ phân loại sau khi được tối ưu như sau. Số cây trong *rf* là 150. Hàm đánh giá *softmax* được sử dụng để phân loại trong *xgb*. Mạng học sâu được xây dựng với kiến trúc 4 lớp ẩn kết nối đầy đủ (fully-connected), với kích cỡ các lớp là 512, 384, 128 và 80 nút, hàm kích hoạt *ReLU* và cuối cùng là hàm *softmax*. Mạng được huấn luyện sử dụng thư viện Keras, sử dụng batch_size là 256 và tốc độ học (*learning rate*) là 0,001.



Hình 2. Hiệu năng của các thuật học thử nghiệm.

Hình 2 thể hiện hiệu năng phân loại chung của của các thuật toán học máy. Kết quả trong hình 2 cho thấy các thuật toán học máy cho kết quả khá tốt khi phân biệt chính xác các hành vi tấn công cũng như bình thường với giá trị lớn hơn 90%. Trong số này thuật học *rf* và *dt* có hiệu năng tổng thể kém hơn đáng kể so với các thuật học còn lại khi đánh giá tổng thể qua chỉ số *f1* và *recall*.

Bảng 2 dưới đây thể hiện độ đo hiệu năng chi tiết của 3 thuật toán *DNN, rf,* và *xgb*. Xét về khả năng phát hiện chi tiết các hành vi chèn mã, toàn bộ các thuật toán học máy khảo sát đều có khả năng phân biệt tốt các hành vi chèn mã *codeinj*, chèn lệnh *cmdinj*, chèn sql *sqli*, chèn *xss*. Số lượng các mẫu này hiếm phần lớn (hơn 90%) trong bộ dữ liệu mẫu. Các kết quả thực nghiệm đều có giá trị tuyệt đối với chỉ số *f1* đạt 100%.

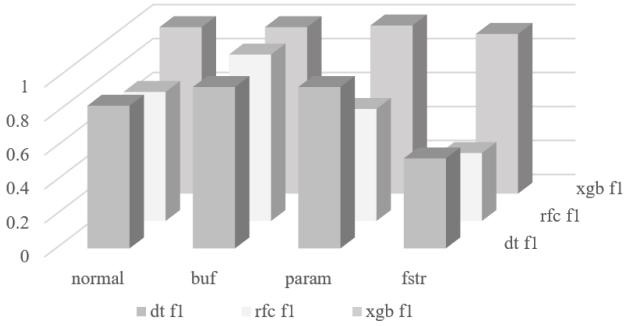
Bảng 2. Hiệu năng của một số thuật học

Thuật toán	**Hành vi**	***precision***	***recall***	***f1***
DNN	*normal*	0.98	0.98	0.98
	codeinj	1	1	1
	cmdinj	1	1	1
	buf	1	0.96	0.98
	crlf	1	1	1
	fstr	0.91	0.99	0.95
	param	0.99	0.99	0.99
	sqli	1	1	1
	xss	1	1	1
rf	*normal*	0.65	0.91	0.76
	codeinj	0.99	0.99	0.99
	cmdinj	1	0.99	0.99
	buf	1	0.96	0.98
	crlf	1	1	1
	fstr	0.91	0.25	0.4
	param	0.95	0.51	0.66
	sqli	0.97	1	0.99
	xss	1	1	1
xgb	*normal*	0.99	0.97	0.98
	codeinj	1	1	1
	cmdinj	1	1	1
	buf	1	0.97	0.98
	crlf	1	1	1
	fstr	0.9	0.98	0.94
	param	0.98	1	0.99
	sqli	1	1	1
	xss	1	1	1

Ngoại trừ hành vi tấn công *xss* khá khác biệt, 3 hành vi chèn mã nêu trên đều có các thể hiện khá giống nhau. Chẳng hạn như, người tấn công có thể sử dụng *sqli* để kích hoạt

các câu lệnh tấn công vào hệ điều hành máy chủ hay hệ quản trị cơ sở dữ liệu. Về mặt hình thức, kiểu tấn công này trùng hợp với kiểu tấn công chèn lệnh. Dù vậy, các thuật toán đều thể hiện khả năng tốt của mình khi phân biệt chính xác các kiểu tấn công này.

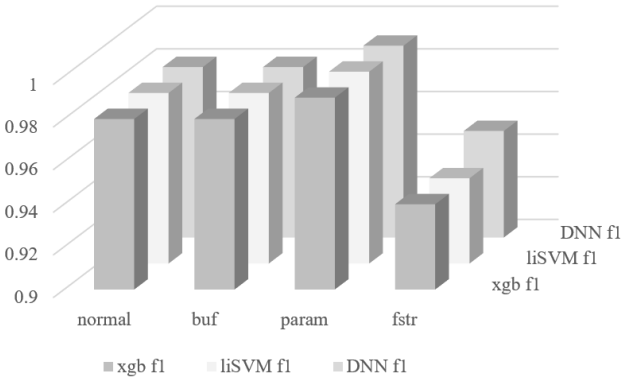
Sự khác biệt về hiệu năng của các thuật toán khảo sát thể hiện chủ yếu trong nhóm nhỏ các hành vi tấn công còn lại (*buf*, *crlf*, *fstr*, và *param*) và nhất là hành vi truy nhập bình thường.



Hình 3. Giá trị f1 của các thuật toán cây quyết định.

Hình 3 cho thấy hiệu năng vượt trội của thuật toán *xgb* so với rừng ngẫu nhiên *rf* và cây quyết định *dt*, đặc biệt là với hành vi tấn công dạng chuỗi định dạng *fstr* và thay đổi tham số *param*. Với cả hai dạng tấn công này thuật toán *xgb* đều đạt giá trị *f1* trên 80% trong khi hai thuật toán còn lại chỉ đạt dưới 40% với *fstr*. Với hành vi bình thường của người dùng, giá trị *f1* của *rf* và *dt* chỉ đạt được khoảng 75% trong khi đó *xgb* vượt hơn 90%.

Các thuật toán còn lại bao gồm mạng học sâu *DNN* và *liSVM* đều có hiệu năng tốt ngang bằng *xgb* với giá trị *f1* đều đạt trên 92% xét trên cả 4 loại hành vi cần phân biệt, mà chúng chiếm tỷ lệ nhỏ trong bộ mẫu đánh giá. Cụ thể thuật học *DNN* cho kết quả tốt giúp phân biệt rõ ràng hành vi tấn công *fstr* so với 2 thuật học còn lại.



Hình 4. Giá trị f1 của các thuật toán xgb, liSVM và DNN.

2) *Đánh giá*

Với sự phát triển của Internet và các ứng dụng trên Web, các hành vi bất thường của người dùng trong các dịch vụ Web có thể biến đổi từ việc dùng sai cho tới các hành vi có chủ đích nhằm làm suy giảm chất lượng phục vụ của web-site cho tới các hành vi gian lận tài chính, gây tổn thất uy tín và danh tiếng của nhà cung cấp dịch vụ.

Kỹ thuật học máy với các thuật học được khảo sát cho thấy khả năng học các đặc trưng kết hợp *n-gram* và TF-IDF của các hành vi tấn công và hành vi bình thường của người dùng. Kết quả này hứa hẹn khả năng ứng dụng tốt vào thực tiễn giám sát và đảm bảo chất lượng phục vụ của các dịch vụ Web. Mặt khác, hiệu quả của mô hình học máy phân loại các hành vi tấn công cũng như bình thường phụ thuộc vào chất lượng của bộ dữ liệu mẫu sử dụng. Báo cáo đã đề xuất việc xây dựng cách thức xây dựng bộ dữ liệu đáp ứng được nhu cầu giám sát và phân tích hành vi truy nhập người dùng dựa trên công cụ ZAP cho phép kiểm thử an toàn các dịch vụ Web. Các mẫu dữ liệu tấn công và bình thường này được lưu vào trong các file bán cấu trúc tương ứng với tên của hành vi tấn công, chẳng hạn như các mẫu truy nhập bình thường *normal.csv*. Điều này cho phép hỗ trợ hiệu quả cho công việc quản trị, phân tích và giám sát các dịch vụ Web với nguồn lực hạn chế. Cấu trúc đơn giản cho phép người quản trị bổ sung thêm các mẫu truy nhập rõ ràng là bình thường hay mẫu tấn công. Nói cách khác, người quản trị hay vận hành các dịch vụ Web có thể tự duy trì thư viện về các hành vi truy nhập của người dùng tùy theo nhu cầu riêng của mình.

Các thuật toán sinh luật dựa trên cây quyết định giúp cho người quản trị dễ dàng hình dung được cách thức hoạt động của hệ thống. Mặt khác các mô hình sử dụng thuật toán cây quyết định cho phép xây dựng mô hình phân loại một cách nhanh chóng và tương đối hiệu quả. Báo cáo đã cho thấy mô hình học máy kết hợp sử dụng các đặc trưng *n-gram* và TD-IDF để biểu diễn truy nhập người dùng mang lại kết quả khá tốt nhất là với các hành vi tấn công có tỷ lệ lớn trong bộ dữ liệu mẫu như *codeinj*, *sqli*, hay *xss*.

Mô hình phân loại dựa trên thuật toán SVM tuyến tính với chiến lược *ovr* cho kết quả rất khả quan, nhất là khi xem xét tốc độ huấn luyện và hiệu năng phân loại. Mô hình SVM này cho kết quả phân loại chi tiết tốt hơn nhiều so với cây quyết định cũng như rừng ngẫu nhiên. Dù vậy, SVM có nguy cơ quá thiên lệch (*over-fitting*) về dữ liệu huấn luyện. Trong khi đó mạng học sâu *DNN*, như thể hiện trong thử nghiệm, có kết quả tốt hơn và được trang bị nhiều biện pháp hiệu quả để hạn chế vấn đề *over-fitting*.

Với việc sử dụng các phần cứng đặc biệt, việc huấn luyện và phân loại hành vi truy nhập được cải thiện đáng kể. Vấn đề chỉ trở nên phức tạp khi khối lượng dữ liệu cần cho việc xây dựng và triển khai mô hình phân loại tăng mạnh. Với các dịch vụ Web có quy mô vừa và nhỏ, việc sử dụng phần cứng hỗ trợ tính toán cho mạng học sâu có thể là trở ngại đáng kể cũng như hạn chế về khả năng mở rộng để đáp ứng việc gia tăng của khối lượng dữ liệu.

Mô hình sử dụng thuật toán *xgb* cung cấp khả năng cân bằng giữa tốc độ huấn luyện, hiệu quả phân loại, mức độ chính xác và khả năng mở rộng sau này cho người quản trị dịch vụ Web. Kết quả thực nghiệm cho thấy mô hình dựa trên *xgb* có hiệu năng phân loại kém hơn một chút so với mạng học sâu DNN. Bộ phần mềm *xgb* trang bị sẵn các cơ chế đối phó với vấn đề *over-fitting*. Ngoài khả năng tận dụng các phần cứng hỗ trợ tính toán như mạng học sâu *DNN*, cách thức xây dựng thuật toán *xgb* thuận tiện cho việc chia-trộn gần với các cơ chế xử lý và tính toán dữ liệu lớn như Apache Spark.

IV. KẾT LUẬN

Việc phát triển mạnh mẽ của các dịch vụ Web làm cho vấn đề quản trị và giám sát các hành vi truy nhập của người dùng càng trở nên cấp bách nhằm đảm bảo chất lượng phục vụ cũng như sự an toàn của dịch vụ Web. Việc phân biệt và phát hiện các hành vi tấn công chèn mã tới các dịch vụ Web

có tác dụng cảnh báo và cung cấp thông tin hiệu quả cho người quản trị các dịch vụ Web trong quá trình vận hành.

Báo cáo nghiên cứu cách thức phát hiện 8 hành vi tấn công kiểu chèn mã tới dịch vụ Web như *sqli*, *xss*, hay tràn bộ đệm cũng như phân loại hành vi bình thường từ các dữ liệu nhật ký truy nhập trên máy chủ Web. Các truy nhập của người dùng được biểu diễn thông qua đặc trưng *n-gram* và TF-IDF do khả năng phân loại hành vi được cải thiện đáng kể theo cách biểu diễn này. Báo cáo đã trình bày cách thức xây dựng và duy trì bộ dữ liệu dùng cho việc xây dựng mô hình phân loại dựa trên công cụ kiểm thử an toàn ZAP. Người quản trị dịch vụ Web có thể dễ dàng duy trì và cập nhập bộ dữ liệu theo nhu cầu quản lý và giám sát của riêng mình dưới dạng các file bán cấu trúc.

Kết quả thực nghiệm chứng tỏ các mô hình học máy có khả năng xây dựng mô hình phân loại tốt từ bộ dữ liệu thử nghiệm với độ đo hiệu năng *f1* đạt được trên 90%. Trong đó mô hình phân loại dựa trên mạng học sâu cho hiệu năng tốt nhất.

Với việc được hỗ trợ từ môi trường phát triển Python, các mô hình thử nghiệm này có khả năng tích hợp dễ dàng và thuận tiện với các nền tảng khác nhau. Đặc biệt, mô hình dựa trên *xgb* có nhiều đặc tính thuận lợi cho việc tích hợp với các nền tảng xử lý dữ liệu lớn đáp ứng tốt hơn việc mở rộng quản trị và giám sát với các dịch vụ Web.

TÀI LIỆU THAM KHẢO

[1] M. V Mahoney and P. K. Chan, "Learning rules for anomaly detection of hostile network traffic," in Data Mining, 2003. ICDM 2003. Third IEEE International Conference on, 2003, pp. 601–604.

[2] W. Fan, M. Miller, S. Stolfo, W. Lee, and P. Chan, "Using artificial anomalies to detect unknown and known network intrusions," Knowl. Inf. Syst., vol. 6, no. 5, pp. 507–527, 2004.

[3] G. G. Helmer, J. S. K. Wong, V. Honavar, and L. Miller, "Intelligent agents for intrusion detection," in Information Technology Conference, 1998. IEEE, 1998, pp. 121–124.

[4] W. Lee, S. J. Stolfo, and P. K. Chan, "Learning patterns from unix process execution traces for intrusion detection," in AAAI Workshop on AI Approaches to Fraud Detection and Risk Management, 1997, pp. 50–56.

[5] S. Salvador, P. Chan, and J. Brodie, "Learning States and Rules for Time Series Anomaly Detection.," in FLAIRS conference, 2004, pp. 306–311.

[6] H. S. Teng, K. Chen, and S. C. Lu, "Security audit trail analysis using inductively generated predictive rules," Sixth Conf. Artif. Intell. Appl., pp. 24–29, 1990.

[7] C. T. Gimnez, A. P. Villegas, and G. Á. Marañón, "HTTP data set CSIC 2010." 2010.

[8] C. De Stefano, C. Sansone, and M. Vento, "To reject or not to reject: that is the question-an answer in case of neural classifiers," IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev., vol. 30, no. 1, pp. 84–94, 2000.

[9] D. Barbara, N. Wu, and S. Jajodia, "Detecting novel network intrusions using bayes estimators," in Proceedings of the 2001 SIAM International Conference on Data Mining, 2001, pp. 1–17.

[10] R. Agrawal and R. Srikant, "Mining sequential patterns," in Data Engineering, 1995. Proceedings of the Eleventh International Conference on, 1995, pp. 3–14.

[11] M. V Mahoney, P. K. Chan, and M. H. Arshad, "A machine learning approach to anomaly detection," 2003.

[12] G. Tandon and P. K. Chan, "Weighting versus pruning in rule validation for detecting network and host anomalies," in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007, pp. 697–706.

[13] G.-Y. Chan, C.-S. Lee, and S.-H. Heng, "Discovering fuzzy association rule patterns and increasing sensitivity analysis of XML-related attacks," J. Netw. Comput. Appl., vol. 36, no. 2, pp. 829–842, 2013.

[14] L. O. Anyanwu, J. Keengwe, and G. A. Arome, "Scalable Intrusion Detection with Recurrent Neural Networks," in 2010 Seventh International Conference on Information Technology: New Generations, 2010, pp. 919–923.

[15] S. Althubiti, W. Nick, J. Mason, X. Yuan, and A. Esterline, "Applying Long Short-Term Memory Recurrent Neural Network for Intrusion Detection," in SoutheastCon 2018, 2018, pp. 1–5.

[16] J. Kim, J. Kim, H. L. Thi Thu, and H. Kim, "Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection," in 2016 International Conference on Platform Technology and Service (PlatCon), 2016, pp. 1–5.

[17] S. Hettich and S. D. Bay, "The UCI KDD Archive [http://kdd.ics.uci.edu]," Univ. California, Dep. Inf. Comput. Sci., 1999.

[18] H. D. Pham and N. D. Nguyen, "Intrusion detection using deep neural network," Southeast Asian J. Sci., vol. 5, no. 2, pp. 111–125, 2017.

[19] V. Vapnik, The nature of statistical learning theory. Springer science & business media, 2000.

[20] S. Chakrabarti, Mining the Web: Discovering knowledge from hypertext data. Elsevier, 2002.

[21] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[22] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[23] H. T. Nguyen, C. Torrano-Gimenez, G. Alvarez, S. Petrović, and K. Franke, "Application of the generic feature selection measure in detection of web attacks," in Computational Intelligence in Security for Information Systems, Springer, 2011, pp. 25–32.

[24] R. R. Larson, "Introduction to information retrieval," J. Am. Soc. Inf. Sci. Technol., vol. 61, no. 4, pp. 852–853, 2010.

DETECT CODE INJECTION BEHAVIORS IN WEB SERVICE

Abstract: The need to monitor access to Web services to detect attacks from Web log increases with Internet development in order to maintain service quality and safety of these services. This paper examines the performance of machine learning-based models to effectively detect code injection attacks to Web services. Also, the paper proposes to build a set of a labelled data-set of these attacks with about 400 thousand samples in 9 categories including normal accesses. Experiments conducted on this data-set using the following algorithms: decision tree, random forest, SVM, XGB and deep learning network (DNN) showed positive results, of which DNN reached F1 value up to 97, 5%.

Keywords: IDS, Web attack detection, Code injection, Web log, Information security, Web service, Machine learning

Phạm Hoàng Duy tham gia giảng dạy tại Khoa CNTT 1 từ năm 2000; hoàn thành nghiên cứu Tiến sỹ 2005-2009 tại Đại học Queensland, Australia về Trí tuệ nhân tạo; lĩnh vực giảng dạy và nghiên cứu quan tâm: các hệ thống thông minh và ứng dụng.
Email: duyph@ptit.edu.vn

Nguyễn Ngọc Điệp tham gia giảng dạy về An toàn thông tin tại Khoa CNTT 1, Học viện Công nghệ Bưu chính Viễn thông từ năm 2013; hoàn thành nghiên cứu Tiến sỹ tại Học viện Công nghệ Bưu chính Viễn thông năm 2017 về các phương pháp học máy cho nhận dạng hoạt động người; lĩnh vực nghiên cứu quan tâm: nhận dạng hoạt động, xử lý ngôn ngữ tự nhiên, an toàn thông tin.

Email:diepnn@ptit.edu.vn