DOI: 10.15625/vap.2022.0213

ỨNG DỤNG HỌC MÁY PHÁT HIỆN MÃ ĐỘC TRONG LUỒNG DỮ LIỆU MÃ HÓA

Nguyễn Ngọc Hưng[1], Nguyễn Thị Thuý Quỳnh[1], Nguyễn Việt Hùng[1]

[1] Học viện Kỹ thuật Quân sự

tamquy999@gmail.com, nguyenthuyquynhcvp@gmail.com, hungnv@lqdtu.edu.vn

TÓM TẮT: *Trong giai đoạn bùng nổ các kết nối mạng của quá trình chuyển đổi số hiện nay, xuất hiện ngày càng nhiều các phần mềm độc hại với nhiều hành vi tinh vi nhằm vượt qua các hệ thống phát hiện mã độc. Nhiều mẫu mã độc đã lợi dụng việc mã hoá đường truyền thông qua giao thức TLS để che giấu hoạt động của chúng. Điều này đã làm cho việc phát hiện các hành vi của mã độc trở nên khó khăn hơn. Một trong những hướng nghiên cứu được quan tâm gần đây là ứng dụng trí tuệ nhân tạo để phát hiện các kết nối của mã độc trong các luồng dữ liệu mã hóa. Trong bài báo này, chúng tôi đề xuất mô hình phát hiện luồng dữ liệu độc hại sử dụng các đặc trưng của giao thức TLS được phân loại bằng thuật toán XGBoost. Các đánh giá thử nghiệm với các bộ dữ liệu tiêu chuẩn và bộ dữ liệu thực tế chứng minh các mô hình học máy với đặc trưng TLS lựa chọn có khả năng phân loại tốt các luồng dữ liệu độc hại và luồng dữ liệu bình thường.*

Từ khóa: *Phát hiện mã độc, kết nối độc hại, XGBoost, luồng dữ liệu mã hóa.*

I. TỔNG QUAN VỀ PHÁT HIỆN MÃ ĐỘC TRÊN ĐƯỜNG TRUYỀN

Mã độc (Malware) là một thuật ngữ chung để chỉ các phần mềm độc hại được xây dựng với mục đích thực hiện các hành vi bất hợp pháp nhằm vào người dùng cá nhân, cơ quan, tổ chức [1]. Mã độc thường được cài đặt vào hệ thống một cách bí mật để phá hoại hoặc lấy cắp thông tin, làm gián đoạn, tổn hại tới tính bí mật, tính toàn vẹn, tính sẵn sàng của hệ thống máy tính nạn nhân. Trong xu hướng cuộc cách mạng công nghiệp lần thứ tư, con người kết nối giao tiếp với nhau, với các hệ thống ứng dụng, gửi nhận dữ liệu,… đều được thực hiện qua Internet. Điều này trở thành điều kiện thuận lợi để mã độc lây lan, phá hoại. Mã độc trên đường truyền khi tới các thiết bị trên hệ thống, tiến hành lây nhiễm, dẫn tới các nguy cơ khó lường ảnh hưởng đến không chỉ các cá nhân, tổ chức mà còn cả kinh tế, an ninh, quốc phòng của các quốc gia [1]. Chính vì vậy, vấn đề phát hiện và phòng, chống mã độc nói chung và vấn đề phát hiện mã độc trên đường truyền nói riêng đã và đang được quan tâm, nghiên cứu rộng rãi trong nhiều năm qua.

Các phương pháp phát hiện mã độc truyền thống đối với luồng dữ liệu không mã hoá có thể được chia thành hai nhóm chính: căn cứ vào dấu hiệu (signature-based) và căn cứ vào hành vi (behavior-based) [2].

Phương pháp phát hiện mã độc dựa vào dấu hiệu là quá trình so khớp các đặc trưng của các mã độc đã biết sau quá trình phân tích, trích rút thông tin. Các kỹ thuật điển hình của phương pháp này là:

- Kỹ thuật dò quét mẫu: Sau khi thu thập được một mẫu mã độc, các chuyên gia sẽ tiến hành phân tích và tìm ra các đặc trưng của chúng. Mỗi loại mã độc sẽ được biểu diễn bởi một hoặc nhiều mẫu hoặc các dấu hiệu (signature) - là chuỗi các byte tuần tự được coi là đặc trưng chính xác để nhận dạng mã độc. Các mẫu đặc trưng này sẽ được lưu trữ vào cơ sở dữ liệu (CSDL) mã độc. Kỹ thuật dò quét mẫu sẽ sử dụng các thuật toán so khớp mẫu nhằm phát hiện có hay không sự tồn tại của các mẫu này trong các file cần kiểm tra. Kỹ thuật này phù hợp với dò quét các loại mã độc đính kèm với vật chủ như virus, hay một số loại trojan.

- Kỹ thuật kiểm tra mã băm: Kỹ thuật này tập trung phát hiện những mã độc tồn tại độc lập, không cần kí sinh vào các file vật chủ như sâu máy tính (worm), phần mềm gián điệp (spyware), rootkit, backdoor. Các loại mã độc này sau khi được thu thập sẽ được tính toán giá trị băm với hàm băm xác định (ví dụ như MD5 hoặc SHA1, SHA2) để tính toán đặc trưng và lưu vào CSDL. File cần được kiểm tra xem có phải mã độc hay không sẽ được băm và so sánh với các mã băm trong CSDL này. Nhờ tính chất của hàm băm, nếu mã băm của một file nằm trong CSDL mã độc sẽ được xác định là mã độc.

Phương pháp phát hiện mã độc dựa vào hành vi là cách tiếp cận nghiên cứu mã độc máy tính dưới góc độ thi hành của tập mã lệnh. Cũng là chương trình máy tính, nhưng khác với các phần mềm thông thường, mã độc chứa các hành động gây nguy hại cho người dùng. Các kỹ thuật điển hình của phương pháp này là:

- Kỹ thuật phân tích mã nguồn: Là kỹ thuật sử dụng các công cụ khác nhau như dịch ngược nhằm phân tích mã nguồn của mã độc. Các chuyên gia, phân tích sẽ tập trung nghiên cứu trật tự, quy luật hình thành các lệnh của mã độc, phân tích các hành vi để xác định chúng.

- Kỹ thuật phân tích động (emulation): Là kỹ thuật phát hiện mã độc dựa vào giám sát các hành vi của chúng khi cho phép thực thi mã chương trình trên các môi trường giả lập. Các hành vi của file thực thi trong môi trường ảo sẽ được phân tích và kết luận có độc hại hay không.

Mặt khác, Internet phát triển cùng với đó là tất cả các dữ liệu thông tin cá nhân, giao dịch tài chính ngân hàng, mua sắm, giao tiếp đều được thực hiện trực tuyến. Chính vì vậy việc bảo đảm cho các thông tin nhạy cảm trao đổi hàng ngày này vẫn riêng tư, không bị rò rỉ trở thành vấn đề quan trọng hàng đầu. Mã hoá trở thành phương pháp cần thiết để bảo vệ quyền riêng tư của người dùng cuối và tính bảo mật, tính toàn vẹn của thông tin liên lạc. Hầu hết lưu lượng truy cập Internet được bảo vệ bởi giao thức mã hoá được gọi là Transport Layer Security (TLS). Mặc dù mã hoá lưu lượng có thể đảm bảo an toàn cho giao tiếp, nhưng nó lại trở thành phương tiện để các phần mềm độc hại lợi dụng che giấu thông tin và tránh bị phát hiện [3]. Hai phương pháp phát hiện truyền thống kể trên phát huy hiệu quả cao với đường truyền không mã hoá nhưng lại trở nên khó khăn khi có sự can thiệp của mã hoá. Việc phát hiện lưu lượng phần mềm độc hại khó và phức tạp hơn rất nhiều.

Trong phần tiếp theo, bài báo sẽ giới thiệu các phương pháp, nghiên cứu đã có để phát hiện kết nối của mã độc đối với luồng dữ liệu mã hoá. Phần III sẽ phân tích đặc trưng của luồng dữ liệu mã hoá và khả năng áp dụng trong phân loại luồng dữ liệu độc hại và luồng dữ liệu thường. Phần IV sẽ trình bày thuật toán, thực nghiệm và đánh giá. Phần cuối cùng là kết luận cùng một số hướng phát triển tiếp theo.

II. CÁC NGHIÊN CỨU LIÊN QUAN

Giải pháp phổ biến nhất để đối phó với lưu lượng truy cập mã hoá trong các tổ chức là cài đặt proxy. Điều này liên quan đến việc cài đặt các chứng chỉ đặc biệt trong máy tính của tổ chức để mở và kiểm tra lưu lượng truy cập của các thành viên. Bộ chặn này được cài đặt giữa máy khách và máy chủ. Cơ chế hoạt động của nó là giải mã lưu lượng đến, quét tìm phần mềm độc hại, mã hoá lại lưu lượng sau đó gửi nó đến đích nếu nó được cho là đáng tin cậy. Nhược điểm của việc sử dụng này là tốn kém và đòi hỏi tính toán. Quá trình mã hoá và giải mã yêu cầu một lượng lớn tài nguyên của thiết bị trung gian, đồng thời với đó là việc trao đổi thông tin bị chậm đi nhiều. Triển khai hệ thống phức tạp và tốn kém. Và quan trọng hơn hết là nó vi phạm các tiêu chuẩn về sự riêng tư người dùng, đi ngược lại mục đích thiết kế ban đầu của giao thức mã hoá TLS [4].

Khắc phục những điểm còn tồn tại của giải pháp phát hiện trên, đã có những nghiên cứu áp dụng trí tuệ nhân tạo vào xây dựng phương pháp mới để phát hiện mã độc. Các nghiên cứu này xoay quanh việc phân tích lưu lượng TLS mã hoá. Anderson và cộng sự [5] đã nghiên cứu việc sử dụng TLS của phần mềm độc hại, tập trung vào sự khác biệt giữa lưu lượng mã độc và lưu lượng lành tính bằng cách sử dụng kết hợp giữa siêu dữ liệu NetFlow và quá trình bắt tay TLS. Nghiên cứu đánh giá khả năng phát hiện mã độc và họ mã độc mà không cần giải mã dữ liệu, sử dụng 4 bộ đặc trưng khác nhau: siêu dữ liệu luồng, chuỗi độ dài và thời gian gói tin, sự phân phối byte, và thông tin tiêu đề của TLS đồng thời sử dụng mô hình phân loại L1 Logistic Regression để thu các kết hợp khác nhau của dữ liệu này để tạo ra 98,5 % - 99,6 % tỉ lệ độc hại.

David McGre [6] đề xuất sử dụng TLS flow, DNS flow, HTTP headers và TLS unencrypted header để phát hiện luồng mã độc HTTPS mà không giải mã. Kết quả nghiên cứu có khả năng phát hiện luồng mã độc với độ chính xác cao và cảnh báo nhầm thấp. Một nghiên cứu khác của nhóm tác giả Lastline và cộng sự [7] tập trung vào TLS metadata để phân loại các luồng TLS. Nghiên cứu này sử dụng 5 mô hình khác nhau gồm: Logistic Regression, Random Forest, K-Nearest neightbors, Linear Discriminant Analysis, và Linear Support Vector Classifier. Thực nghiệm của nghiên cứu này mang lại độ chính xác khoảng 97,6 % trong việc phát hiện luồng mã độc.

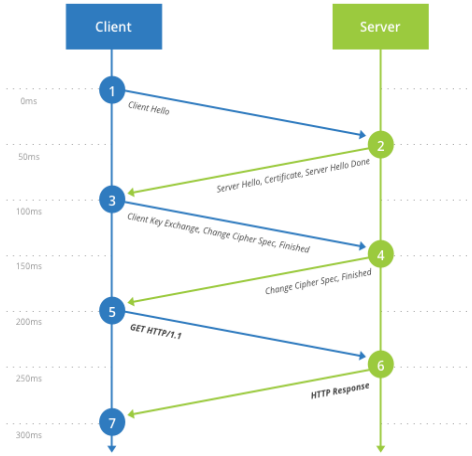
Nhiều mô hình đã chứng minh được sự thành công khi sử dụng TLS handshake metadata. Tuy nhiên một số nhóm nghiên cứu sử dụng bộ dữ liệu thiếu cập nhật, dẫn đến khả năng cho ra kết quả thiếu chính xác. Nhóm tác giả Bryan Scarbrough [8] đã sử dụng đặc trưng TLS handshake metadata và 3 mô hình phân loại khác nhau để phân loại một kết nối TLS là độc hại hay lành tính: Support Vector Machine, One-Class Support Vector Machine và Autoencoder Neural Network. Nghiên cứu của chúng tôi cũng sử dụng bộ dữ liệu cùng với những đặc trưng trích xuất từ TLS handshake metadata của bài báo này và đề xuất phương pháp học máy cho kết quả tốt hơn.

III. PHÂN TÍCH ĐẶC TRƯNG CỦA LUỒNG DỮ LIỆU MÃ HÓA

A. Giao thức TLS

TLS (Transport Layer Security – Bảo mật tầng truyền tải) là một giao thức mật mã, cung cấp quyền riêng tư và sự toàn vẹn dữ liệu trong giao tiếp giữa các ứng dụng. TLS là kế thừa của giao thức SSL (Secure Sockets Layer), cung cấp sự bảo mật và quyền riêng tư mà không yêu cầu khả năng tương thích ngược. TLS thường được triển khai trên các giao thức ứng dụng phổ biến chẳng hạn như HTTP cho web hoặc SMTP cho email. Có hai giai đoạn trong TLS: Bắt tay và truyền dữ liệu. Trong giai đoạn bắt tay, các thuật toán và thông số khác nhau được yêu cầu cho truyền dữ liệu an toàn. Các khoá đối xứng được thoả thuận sau khi giai đoạn bắt tay hoàn thành. Sau giai đoạn bắt tay, dữ liệu được chuyển giữa máy khách và máy chủ dưới dạng các bản ghi mã hoá [9].

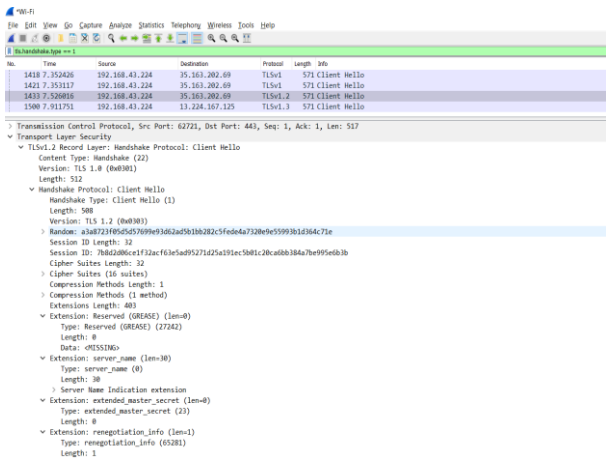
Một quá trình bắt tay TLS được thể hiện trong Hình 1.



Hình 1. Quá trình bắt tay TLS [10]

Một phiên làm việc TLS phiên bản 1.2 (TLS 1.2) yêu cầu 2 lượt thông báo vòng giữa máy khách và máy chủ để thiết lập một đường truyền an toàn và tuân theo một quy trình được xác định từ trước. Đầu tiên, một ứng dụng khách, ví dụ như trình duyệt web, sẽ gửi một thông điệp ClientHello để chỉ định phiên bản TLS được hỗ trợ, các thuật toán mã hóa và các tính năng được hỗ trợ khác. Máy chủ sau đó sẽ phản hồi bằng một loạt thông báo về việc thực hiện các chức năng như: chọn bộ mã hóa, phiên bản TLS, gửi chứng chỉ máy chủ và cung cấp một số thông tin cần thiết cho việc tạo khóa mã hóa. Máy khách sẽ xác thực chứng chỉ đã nhận được từ máy chủ, gửi dữ liệu tạo khóa cho máy chủ và cung cấp khóa mã hóa dùng chung. Cuối cùng, máy chủ sẽ phản hồi với một thông điệp ChangeCipherSpec, sau đó quá trình giao tiếp giữa máy khách và máy chủ sẽ được mã hóa và quá trình bắt tay được hoàn thành [8].

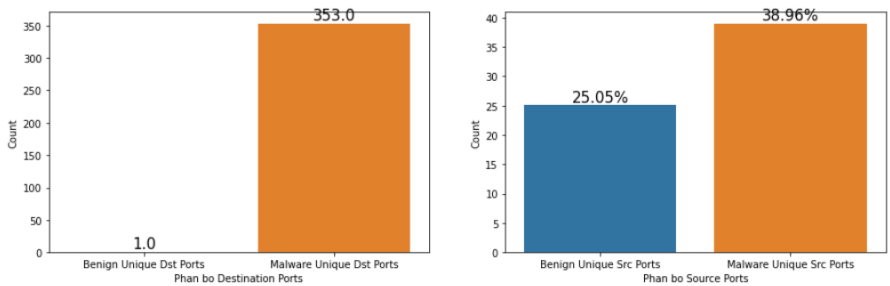
Mỗi một cặp client – server sẽ lựa chọn bộ mã hóa (cipher suites), phiên bản TLS (TLS version), chứng chỉ máy chủ, thông tin cần thiết cho việc tạo khóa mã hoá, … là khác nhau cho mỗi lần bắt tay (Hình 2). Dựa vào đặc trưng của các quá trình bắt tay TLS thu thập được để chọn ra các thuộc tính đặc trưng phân biệt benign và malware.



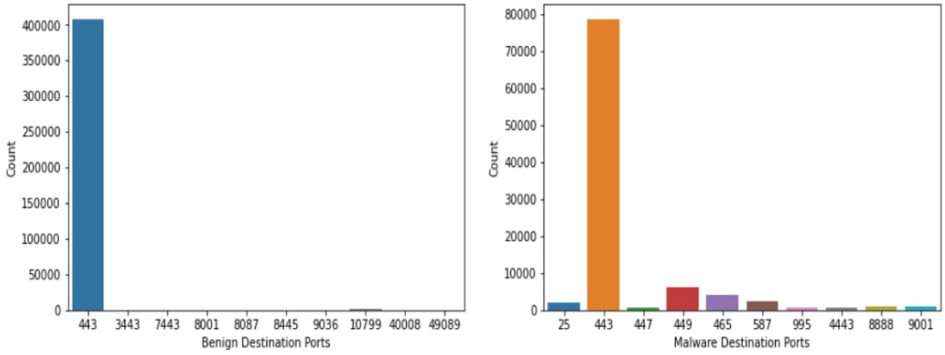
Hình 2. TLS features của một gói tin

B. Đặc trưng trích xuất của luồng dữ liệu mã hóa

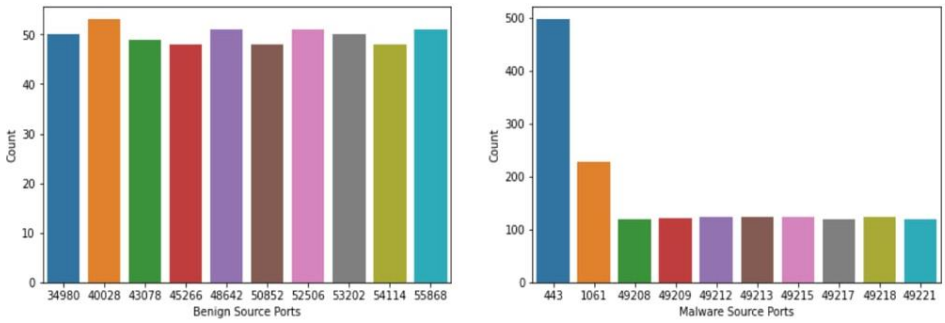
Trong phần này, chúng tôi phân tích, so sánh dựa trên kết quả thống kê về các đặc trưng khác biệt nhau nổi bật nhất giữa hai luồng độc hại và không độc hại (malware TLS handshake và benign TLS handshake). Sau đó lựa chọn ra bộ các đặc trưng với khả năng phân loại cao nhất.



Hình 3. Phân bố Destination Ports và Source Ports



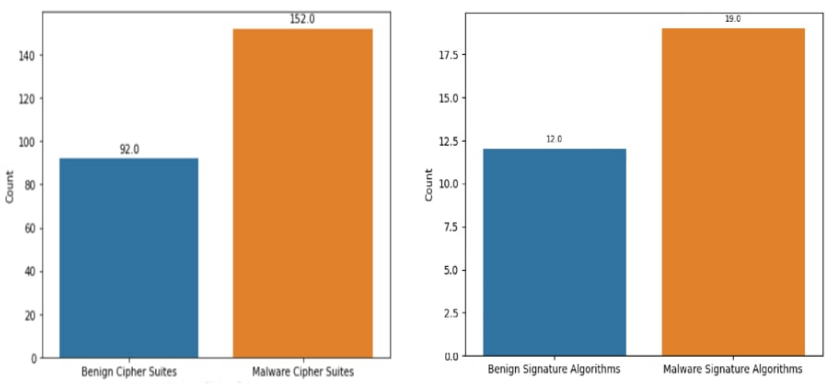
Hình 4. Destination Ports



Hình 5. Source Ports

Ta xem xét đặc trưng source port và destination port. Theo thống kê trên bộ dữ liệu hơn 100.000 gói tin thu thập được, sự phân bố source port và destination port giữa hai lớp dữ liệu độc hại và lành tính rất khác nhau. Trong khi tất cả các mẫu dữ liệu lành tính có số destination port chỉ phân bổ chủ yếu trên 10 cổng thì dữ liệu độc hại lại có sự phân bổ rộng hơn rất nhiều, trên 903 cổng (Hình 3). Nhìn vào biểu đồ thống kê cụ thể số lượng lưu lưu lượng trên các cổng, nhận thấy rõ cổng TCP 443 là phổ biến nhất đối với cả luồng độc hại và lành tính. Bên cạnh đó, luồng lành tính có đặc trưng destination port trên các cổng 3443, 7443, 8001, 8087, 8445, 9036, 10799, 40008, 49089; luồng độc hại lại tập trung nhiều hơn vào các cổng 25, 447, 449, 465, 587, 995, 4443, 8888, 9001 (Hình 4).

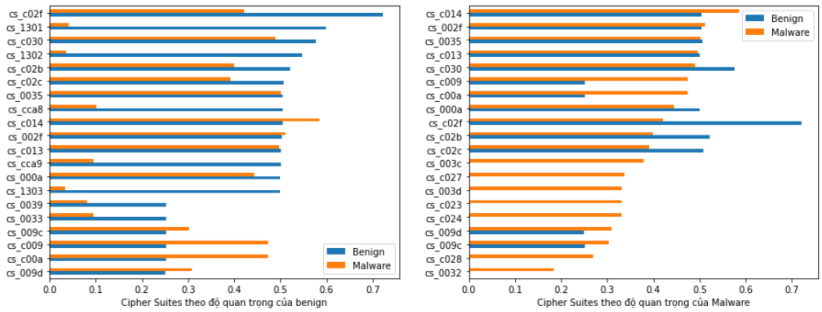
Sự phân bổ này thay đổi đáng kể hơn đối với số source port. Đối với luồng lành tính phân bổ trên 14151 cổng, luồng độc hại trên 16102 cổng (Hình 3). Cụ thể, các cổng tập trung nhiều luồng lành tính có thể kể đến: 34980, 40028, 43078, 45266, 50582, 52506, 53202, 54114, 55868; còn luồng độc hại lại phân bổ chủ yếu trên các cổng 443, 1061, 49208, 49212, 49213, 49215, 49217, 49218, 49221 (Hình 5).



Hình 6. Phân bố Cipher Suites và Signature Algorithms

Tiếp theo, chúng tôi phân tích thống kê các đặc trưng Cipher Suites và Signature Algorithms trong metadata. Điều nổi bật là số lượng các giá trị khác nhau được mã độc sử dụng trong cả hai đặc trưng. Mặc dù lưu lượng độc hại chỉ có 22,624 % phân phối dữ liệu trong tập dữ liệu, nhưng vẫn sử dụng thêm 7 signature algorithms và hơn 60 cipher suites so với lưu lượng truy cập lành tính (Hình 6).

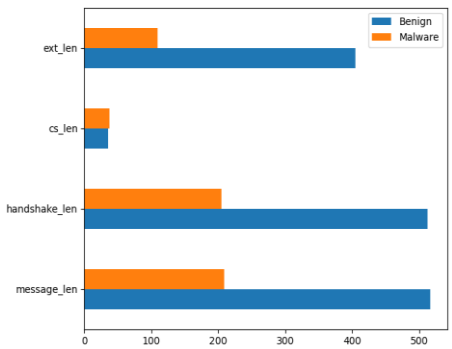
Ta cũng thấy rằng mã độc không chỉ sử dụng nhiều cipher suites và signature algorithms hơn, mà nó còn ưa thích các phiên bản rất khác so với các phiên bản của luồng lành tính.



Hình 7. Cipher Suites

Một số cipher suites trong các kết nối lành tính được sử dụng nhiều trong khi mã độc hầu như không sử dụng: cs_1302, cs_1301, cs_1303, cs_0039, cs_009f. Ngược lại, có cipher suites mã độc sử dụng nhiều nhưng luồng lành tính lại gần như không có: cs_000a, cs_0032 (Hình 7).

Một nhóm đặc trưng dữ liệu khác đại diện cho các trường mà mã độc có ảnh hưởng ít nhất và khó sửa đổi hơn các trường trước đó. Chúng có thể dễ dàng thay đổi source port và destination port, sửa đổi thông tin máy khách được sử dụng để phát triển mở rộng hay thay đổi hoàn toàn việc phân phối cipher suites và signature algorithms được cung cấp. Tuy nhiên các trường dữ liệu như handshake metadata và message length phức tạp hơn. Trong Hình 8 dưới đây, chúng tôi đánh giá giá trị trung bình của các giá trị độ dài lành tính lớn hơn đáng kể so với giá trị được mã độc sử dụng.



Hình 8. Độ dài các trường dữ liệu

Từ những so sánh, phân tích giữa luồng độc hại và luồng lành tính trên, cùng quá trình phân tích thống kê dữ liệu thu thập được, chúng tôi đề xuất bộ 18 đặc trưng sử dụng trong quá trình huấn luyện. Tuy nhiên, dữ liệu của 18 đặc trưng này ở dạng dữ liệu thô thuật toán học máy chưa hiểu được. Vì vậy chúng tôi đã tiền xử lý dữ liệu, biến đổi vector dữ liệu từ 18 chiều ở dạng thô thành vector 510 chiều dạng số để đưa vào các mô hình học máy.

Bảng 1. Các đặc trưng TLS Handshake Metadata được lựa chọn

Đặc trưng	Kích cỡ	Kiểu dữ liệu
Source Port	1	Int
Destination Port	1	Int
TLS Record Type	1	Int
Client TLS Version	1	Int
Message Length	1	Int
Cipher Suite Length	1	Int
Cipher Suite	351	Float
Extension Length	1	Int
Handshake Type	1	Int
Handshake Length	1	Int
Handshake Version	1	Int
Signature Algorithms	36	Float
Supported Groups	49	Float
Supported Points	3	Int
Server OCSP Stapling	1	Int
Server TLS Version	1	Int
Server Supported TLS Version	1	Int
Server Extensions	59	Float
Total	510	

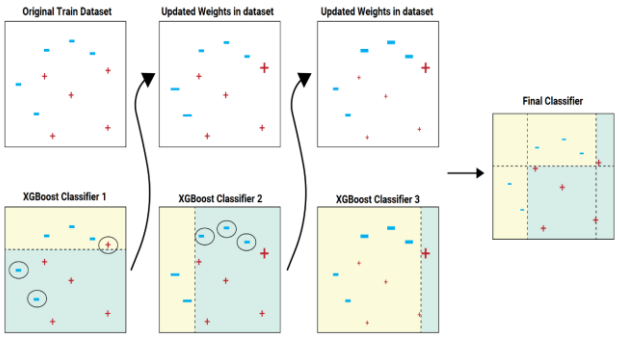
IV. ỨNG DỤNG HỌC MÁY XGBOOST TRONG PHÁT HIỆN MÃ ĐỘC

A. Thuật toán XGBoost

Do đặc thù phân tích luồng dữ liệu trên mạng cần xử lý với tốc độ nhanh, chúng tôi đề xuất thuật toán học máy XGBoost để cải thiện kết quả phát hiện các kết nối của mã độc. XGBoost là viết tắt của Extreme Gradient Boosting. Đây là thuật toán được sử dụng nhiều trong các mô hình huấn luyện có giám sát với tốc độ xử lý nhanh và độ chính xác cao bên cạnh mô hình điển hình như SVM, Neural Network, Random Forest. XGboost có tốc độ huấn luyện nhanh, có khả năng tính toán song song cao trên nhiều máy, có thể tăng tốc bằng cách sử dụng GPU, nhờ vậy mà việc xử lý dữ liệu lớn như xử lý dữ liệu mạng sẽ là ưu điểm của mô hình này khi triển khai thực tế [10].

Ý tưởng của thuật toán Boosting là xây dựng một lượng lớn các mô hình phân lớp (thường là cùng loại). Mỗi mô hình sau sẽ học cách sửa những lỗi của mô hình trước (dữ liệu mà mô hình trước dự đoán sai), tạo thành một chuỗi các mô hình mà mô hình sau sẽ tốt hơn mô hình trước bởi trọng số được cập nhật qua mỗi mô hình (cụ thể ở đây là trọng số của những dữ liệu dự đoán đúng sẽ không đổi, còn trọng số của những dữ liệu dự đoán sai sẽ được tăng thêm) (Hình 9). Kết quả của mô hình cuối cùng trong chuỗi mô hình này sẽ được lấy làm kết quả trả về [10].

XGBoost là một giải thuật được dựa trên Gradient Boosting, tuy nhiên kèm theo đó là những cải tiến về mặt tối ưu thuật toán, cùng khả năng hỗ trợ cao của phần cứng tính toán song song, giúp đạt được những kết quả vượt trội cả về mặt thời gian huấn luyện cũng như bộ nhớ sử dụng.

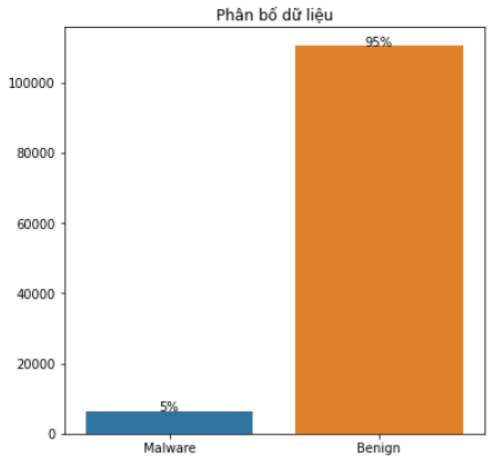


Hình 9. Thuật toán XGBoost [11]

B. Dữ liệu thử nghiệm

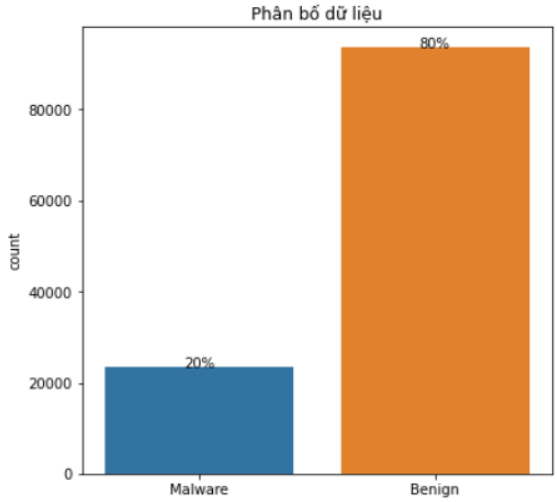
Trong bài báo này chúng tôi sử dụng hai bộ dữ liệu. Một bộ dữ liệu mẫu chúng tôi đã đề cập ở trên và một bộ dữ liệu chúng tôi tự thu thập.

Bộ dữ liệu thứ nhất là bộ dữ liệu mẫu. Bộ dữ liệu này do nhóm nghiên cứu từ Viện Nghiên cứu An toàn thông tin Canada (CIC) cung cấp [8]. Tập dữ liệu này được tạo trong vài tuần đầu năm 2020 và cung cấp các tập dữ liệu được gắn nhãn chứa cả mạng độc hại và lành tính. Tập dữ liệu lành tính được lấy từ bộ dữ liệu của Google Chrome và Mozilla Firefox bằng cách sử dụng Cloudflare và Google DNS để phân tích. Tập dữ liệu độc hại đều lấy từ trang web malware-trafficanalysis.net. Trang web này chứa hàng trăm file capture mạng khác nhau. Tổng cộng có 255 files capture được lấy từ nhiều họ mã độc khác nhau: Dridex, TrickBot, Emotet, Zbot/Zloader, IcedID, Quakbot,… Công cụ Netcap (netwwork metadata capture tool) được sử dụng để xử lý file capture và trích xuất TLS client và server handshake. Bộ dữ liệu xử lý gồm tổng cộng 110490 luồng lành tính và 6422 luồng độc hại (Hình 10) [12].



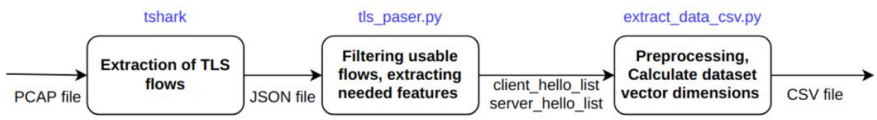
Hình 10. Phân bố dữ liệu (bộ 1)

Bộ dữ liệu thứ hai chúng tôi tự thu thập. Bộ này có tổng cộng 94444 luồng chứa mã độc và 417080 luồng lành tính. Đối với tập dữ liệu lành tính, các kết nối TLS được thu thập từ lưu lượng truy cập tới server đặt tại Phòng thí nghiệm An toàn thông tin của HVKTQS. Có rất nhiều nguồn dữ liệu lưu lượng mạng có chứa malware trên Ineternet. Tuy nhiên hầu hết chúng đều tập trung vào các giao thức không được mã hoá như HTTP hoặc DNS. Ngoài ra, sự phân biệt giữa luồng lành tính và phần mềm độc hại trong các file capture này thường không rõ ràng hoặc các tác giả chỉ cung cấp tệp .csv chứ không phải các file pcap gốc, trong khi chúng tôi cần tận dụng toàn bộ luồng TLS. Vì vậy đối với tập dữ liệu độc hại, chúng tôi thu thập dữ liệu từ nguồn malware-traffic-analysis.net. Tất cả các tệp thu thập từ tháng 6 năm 2013 đến tháng 3 năm 2022 có chứa luồng TLS đều đã được thu thập, tương ứng với 94444 luồng TLS trên 1082 tệp thu thập được (Hình 11).



Hình 11. Phân bố dữ liệu (bộ 2)

Tập dữ liệu được chia theo tỉ lệ 70 % dành cho huấn luyện, 30 % dành cho thử nghiệm. Hình 12 dưới đây là quá trình xử lý các thuộc tính thành các thông tin dạng số để học máy có thể hiểu được.



Hình 12. Quá trình trích xuất đặc trưng

C. Đánh giá kết quả

Do trong các dữ liệu thử nghiệm mất can bằng với số lượng luồng mã độc ít hơn so với luồng lành tính nên chúng tôi đánh giá các độ đo "Precision", "Recall" và "F1 scores" để đánh giá so sánh kết quả phân loại của các phương pháp với phương pháp đề xuất. Bên cạnh đó vẫn sử dụng kết quả "Accuracy" để kiểm tra độ chính xác của thuật toán.

Accuracy là chỉ số thể hiện độ chính xác của phép toán, được tính bằng cách chia số lần dự đoán đúng cho tổng số lần dự đoán:

$$\text{accuracy} = \frac{Số\ lần\ dự\ đoán\ đúng}{Tổng\ số\ lần\ dự\ đoán}$$

Precision là chỉ số thể hiện cho tỉ lệ các nhãn dương tính được xác định chính xác, được tính bằng cách chia số lượng dương tính thật cho số nhãn dương tính đã được xác định:

$$\text{precision} = \frac{TP}{TP+FP}$$

Recall là tỷ lệ giữa các lần xác định dương tính đúng của bộ phân loại trên tổng số các nhãn dương tính lẽ ra phải gán:

$$\text{recall} = \frac{TP}{TP+FN}$$

F1 scores tính toán giá trị trung bình của precision và recall, cung cấp trọng lượng cân bằng cho hai chỉ số này.

$$\text{F1 measure} = \frac{2 * precision * recall}{precision + recall}$$

(TP: true positive, FP: false positive, FN: false negative).

Bảng 2 thể hiện kết quả thử nghiệm trên bộ dữ liệu của CIC. Từ kết quả thử nghiệm ở Bảng 2 với bộ dữ liệu 1, ta có thể thấy bộ dữ liệu có tính phân loại cao. Các chỉ số kết quả đánh giá cao nhất tương ứng với thuật toán như sau:

Accuracy = 0.9976 – XGBoost, Precision = 0.9979 – Decision Tree, Recall = 1.0 – XGBoost, KNN, F1 scores = 0.9984 – XGBoost. Thuật toán XGBoost cho kết quả phân loại tốt nhất với ¾ độ đo đánh giá.

Bảng 2. Kết quả các thuật toán với 25000 mẫu chứa 20% malware (bộ 1)

Algorithm	Accuracy	Precision	Recall	F1 scores
SVM	0,9868	0,98529	0,99975	0,99247
XGBoost	**0,9976**	0,99700	**1,0**	**0,99849**
MLP	0,997	0,9965	0,99974	0,99812
KNN	0,9962	0,99526	**1,0**	0,99762
Decision Tree	0,9964	**0,9979**	0,99749	0,99774

Bảng 3. Kết quả các thuật toán với 100000 mẫu chứa 20 % malware (bộ 2)

Algorithm	Accuracy	Precision	Recall	F1 scores
SVM	0,99285	0,99152	0,99962	0,99555
XGBoost	**0,9979**	**0,99805**	0,99868	**0,99863**
MLP	0,9978	0,99730	**0,99993**	0,99862
KNN	0,9973	0,99680	0,99981	0,99830
Decision Tree	0,9969	0,99786	0,99824	0,99805

Kết quả thống kê kết quả đối với bộ dữ liệu tự thu thập thể hiện ở Bảng 3. Các chỉ số kết quả cao nhất ứng với thuật toán như sau: Accuracy = 0.9979 – XGBoost, Precision = 0,99805 – XGBoost, Recall = 0,99993 – MLP, F1 scores = 0.99863 – XGBoost. Với 100000 mẫu chứa 20% malware, thuật toán XGBoost cho kết quả phân loại tốt nhất so với các thuật toán còn lại đem so sánh.

Như vậy với hai bộ dữ liệu đem thử nghiệm và kết quả thống kê ở Bảng 2 và 3, có thể thấy bộ các đặc trưng trích xuất từ TLS handshake metadata có khả năng phân loại tốt, đạt hiệu quả cho giải pháp phát hiện kết nối của mã độc trên đường truyền mã hoá. Đồng thời thuật toán XGBoost là thuật toán tốt nhất, cho kết quả đánh giá cao nhất đối với hai bộ dữ liệu đem thử nghiệm này.

V. KẾT LUẬN

Bài báo đã trình bày phương pháp phát hiện mã độc trong lưu lượng mã hoá TLS sử dụng các đặc trưng của quá trình bắt tay TLS. Với bộ 510 đặc trưng này có thể đạt được hiệu suất phân loại luồng lành tính và luồng độc hại tốt với các thuật toán học máy phổ biến đồng thời giải quyết được những điểm hạn chế của phương pháp sử dụng proxy. Kết quả thực nghiệm cho thấy thuật toán XGBoost cung cấp kết quả tổng thể tốt nhất. Trong thời gian tới, chúng tôi tiếp tục thử nghiệm với số lượng dữ liệu lớn và sự đa dạng của lưu lượng truy cập, cải thiện độ chính xác phát hiện mã độc trong luồng mã hóa. Thêm vào đó, chúng tôi sẽ tiếp tục nghiên cứu thêm các đặc trưng trích xuất khác của luồng dữ liệu để nâng cao hiệu suất phân loại cũng như kết quả của bài toán phát hiện mã độc trên đường truyền.

TÀI LIỆU THAM KHẢO

[1] Sikorski, M., Honig, A.: Practical malware analysis: the hands-on guide to dis-secting malicious software. no starch press, San Francisco, 2012.

[2] Ekta Gandotra, Divya Bansal, and Sanjeev Sofat, Malware analysis and classification: A survey. Journal of Information Security, 2014.

[3] Odin Jenseg.: A machine learning approach to detecting malware in TLS traffic using resilient network features, 2019.

[4] František Střasák. "Detection of HTTPS Malware Traffic". Czech Technical University in Prague. May 2017.

[5] Anderson, B., Paul, S., & McGrew.: Deciphering malware's use of TLS (without decryption), 2016.

[6] David McGre, Blake Anderson. Identifying Encrypted Malware Traffic with Contextual Flow Data. 2016.

[7] Roques, O. (2019, September). Detecting Malware in TLS Traffic. The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05). doi:10.1109/lcn.2005.35.

[8] Bryan Scarbrough. Malware Detection in Encrypted TLS Traffic Through Machine Learning. Sans Institute Information Security Reading Room, 2021.

[9] Jay Shah.: Detection of malicious Encrypted Web Traffic Using Machine Learning, 2018.

[10] Corey Wade, Kevin Glynn - Hands-On Gradient Boosting with XGBoost and scikit-learn, Packt Publishing, 2020.

[11] XGBoost By Heart. Website was accessed on 2022. https://medium.com/almabetter/xgboost-by-heart-b494a471845e.

[12] Yaser M. Banadaki. Detecting Malicious DNS over HTTPS Traffic in Domain Name System using Machine Learning Classifiers. Journal of Computer Sciences and Applications. 2020.

MACHINE LEARNING APPROACH FOR MALICIOUS ENCRYPTED TRAFFIC DETECTION

Nguyen Ngoc Hung, Nguyen Thi Thuy Quynh, Nguyen Viet Hung

ABSTRACT: *In the explosive phase of network connections of the current digital transformation, more and more malicious software has appeared with many sophisticated behaviors to bypass malicious code detection systems. Many malicious code samples have taken advantage of the encryption of communication through the TLS protocol to conceal their activities. This approach has made it more difficult to detect malicious behavior. One of the research directions of recent interest is the application of artificial intelligence to detect malicious connections in encrypted data streams. In this paper, we propose a malicious data stream detection model using the features of the TLS protocol classified by the XGBoost algorithm. Experimental evaluations demonstrate that proposed method achieved a good result of malicious and normal encrypted traffic classification.*