PHƯƠNG PHÁP XÂY DỰNG VECTOR ĐẶC TRƯNG DỰA TRÊN CHUYỂN ĐỔI CẤU TRÚC VÀ THỐNG KÊ CHUỖI TRUY VẤN TRONG MÔ HÌNH NHẬN DẠNG BẤT THƯỜNG TƯỜNG LỬA ỨNG DỤNG WEB

Huỳnh Hoàng Tân[1*], Trần Văn Hoài[2]

[1]*Trường Đại học Công nghệ Đồng Nai*
[2]*Trường Đại học Bách khoa TP. HCM*
*Tác giả liên hệ: *Huỳnh Hoàng Tân, huynhhoangtan@dntu.edu.vn*

THÔNG TIN CHUNG

Ngày nhận bài: 28/02/2024

Ngày nhận bài sửa: 02/05/2024

Ngày duyệt đăng: 30/05/2024

TỪ KHOÁ

Bảo mật ứng dụng web;
Nhận dạng bất thường truy vấn web;
Nhận dạng tấn công web.

TÓM TẮT

Ngày nay, internet đã trở nên phổ biến, cùng với sự phát triển mạnh mẽ công nghệ điện toán đám mây, IoT và điện thoại thông minh đã thúc đẩy sự gia tăng nhanh chóng của ứng dụng phát triển trên nền tảng web. Để bảo vệ các ứng dụng web, hệ thống phát hiện/ngăn chặn xâm nhập trái phép được phát triển được gọi là tường lửa ứng dụng web (WAF). Chức năng nhận dạng tấn công trên WAF thường được phân loại thành hai phương pháp là dựa trên quy tắc và bất thường. Mô hình dựa trên bất thường về lý thuyết có thể nhận dạng các truy vấn độc hại chưa được biết đến bằng cách quan sát các dữ liệu truy vấn. Trong nghiên cứu này, chúng tôi đề xuất phương pháp xây dựng vector đặc trưng bằng cách chuyển đổi cấu trúc và thống kê các thành phần của chuỗi truy vấn. Sau đó, vector đặc trưng sẽ là đầu vào cho các thuật toán phân loại không giám sát để nhận dạng truy vấn bất thường. Kết quả thử nghiệm với thuật toán K-means, DBSCAN, Isolation Forest cho thấy DBSCAN có độ chính xác cao nhất (Accuracy>96%, F1-Score >97%), ngay cả đối với ứng dụng web dễ nhận dạng nhầm như xác thực và đăng ký. Tính hiệu quả của phương pháp là sử dụng dữ liệu không cần dán nhãn trước nên giúp việc triển khai trên WAF dễ dàng hơn.

1. GIỚI THIỆU

Ngày nay, ứng dụng web đã trở nên phổ biến với các ưu điểm là truy cập ở mọi nơi chỉ cần có kết nối internet, triển khai và cập nhật dễ dàng, yêu cầu hệ thống đơn giản hơn (thường chỉ yêu cầu cao ở máy chủ web) so với ứng dụng truyền thống (phát triển dưới dạng cài đặt tại máy tính để bàn). Do đó, ứng dụng web trở thành đối tượng tấn công của tội phạm mạng máy tính. Theo báo cáo Verizon Data Breach Investigations Report (DBIR) 2023 (Langlois et al., 2023) có đến 80% hành động tấn công gây sự cố hệ thống là nhằm vào ứng dụng web.

WAF được xem là công cụ hữu hiệu để bảo vệ ứng dụng web trước các tấn công. WAF là một lớp bảo mật trung gian giữa ứng dụng web

và người dùng nhằm phát hiện, ngăn chặn truy vấn trái phép. Chức năng kiểm tra truy vấn (WAF inspection) của WAF có thể phân tích các luồng dữ liệu truy cập vào ứng dụng web bao gồm giao thức HTTP và HTTPS. Đối với các kết nối mã hóa HTTPS, chức năng này sử dụng một chứng chỉ SSL/TLS được cấp để chuyển dữ liệu chuyển từ dạng mã hóa sang dạng văn bản thô để kiểm tra nội dung. Chức năng nhận dạng tấn công trên WAF thường bao gồm hai phương pháp là dựa trên quy tắc và bất thường. Phương pháp dựa trên quy tắc dễ dàng xây dựng và có hiệu quả cao khi bảo vệ chống lại các truy vấn tấn công đã biết hoặc tạo ra chính sách phù hợp với ứng dụng web. Hạn chế của phương pháp này là yêu cầu hiểu rõ chi tiết cụ thể của các mối đe dọa và phụ thuộc vào cơ sở dữ liệu các quy tắc. Phương pháp dựa trên sự bất thường sẽ quan sát các truy cập đến ứng dụng web để xây dựng một mô hình có thể phát hiện các truy vấn bất hợp pháp. Do đó, mô hình có thể nhận ra một truy vấn tấn công chưa được biết đến hay không có trong cơ sở dữ liệu. Tuy nhiên, phương pháp dựa trên sự bất thường gặp phải một số thách thức sau:

+ Khi triển khai WAF thông thường sẽ không biết trước được đặc điểm của ứng dụng web cần bảo vệ như: nền tảng web phát triển, ngôn ngữ phát triển, hệ thống quản lý nội dung sử dụng, … dù rằng điều này ảnh hưởng rất lớn đến loại tấn công, khai thác lỗ hổng đối với hệ thống. Do đó, phương pháp nhận dạng bất thường phải có khả năng xử lý độc lập với các đặc điểm của ứng dụng web.

+ Thông thường tỷ lệ nhân dạng nhầm các truy vấn bình thường thành tấn công trong các hệ thống dựa trên bất thường cao hơn so với các hệ thống dựa trên quy tắc (Dong et al., 2018; Sureda Riera et al., 2020; Dau et al., 2022).

+ Phương pháp dựa trên bất thường đòi hỏi rất nhiều tài nguyên tính toán để xây dựng mô hình (Dau et al., 2022).

+ Khi triển khai WAF, các truy vấn có thể bao gồm các truy vấn độc hại và bình thường. Vì thế, dữ liệu được thu thập tự động để huấn luyện các thuật toán phân loại có giám sát khó thực hiện.

+ WAF xử lý được truy vấn ở chế độ thời gian thực, tốc độ xử lý nhanh điều này có thể không phù hợp một số mô hình có thời gian huấn luyện lớn và thuật toán có mức độ tính toán phức tạp.

Trong báo cáo này, chúng tôi đề xuất phương pháp trích xuất thông tin của chuỗi truy vấn bằng kỹ thuật chuyển đổi cấu trúc và thống kê chuỗi truy vấn nhằm tạo ra vector đặc trưng. Sau đó, vector là đầu vào cho phương pháp học không giám sát để phân loại thành hai lớp bất thường và bình thường. Phương pháp không bị ảnh hưởng bởi kỹ thuật chuyển đổi ký tự (URL Encoding) và giao thức mã hóa https do được thực hiện sau khi chức năng kiểm tra của WAF thực thi nên dữ liệu kết nối lúc này đã được giải mã thành dạng dữ liệu văn bản thô. Những đóng góp chính của nghiên cứu này như sau:

(1) Đề xuất cách tiếp cận chuyển đổi cấu trúc và thống kê chuỗi truy vấn để xây dựng vector đặc trưng mà không phụ thuộc vào đặc điểm ứng dụng web cụ thể.

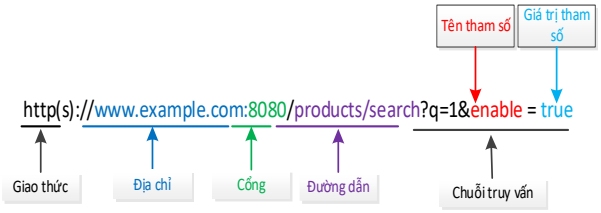
(2) Áp dụng phương pháp học không giám sát phù hợp để phát hiện truy vấn bất thường dựa vào bộ dữ liệu vector đặc trưng.

Phần 2 cung cấp cơ sở và nghiên cứu liên quan về các phương pháp và mô hình phát hiện tấn công web dựa trên sự bất thường. Phần 3 mô tả chi tiết về kỹ thuật chuyển đổi cấu trúc và thống kê chuỗi truy vấn để xây dựng vector đặc trưng. Phần 4 trình bày việc thu thập và xử lý trước dữ liệu để thực hiện kiểm nghiệm. Tiếp theo, phần 5 đánh giá hiệu quả của phương pháp bằng cách sử dụng các phương pháp học không giám sát. Cuối cùng, phần 6 kết luận và các hướng nghiên cứu trong tương lai.

2. NHỮNG CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN

Một trong những vấn đề cơ bản trong bảo mật ứng dụng web là kết nối từ phía người dùng rất khó để kiểm soát như mong muốn. Người

dùng có thể tạo hoặc tùy chỉnh bất kỳ dữ liệu đầu vào nào để được chuyển trở lại máy chủ web xử lý. Ứng dụng web trao đổi thông tin với người dùng thông qua giao thức HTTP/S trên môi trường mạng. Một URL (Uniform Resource Locator) được sử dụng để xác định duy nhất một tài nguyên trên Web. Một URL có cấu trúc minh họa như hình 1:



Hình 1. mô tả cấu trúc của truy vấn ứng dụng web

Cấu trúc URL bao gồm giao thức, địa chỉ máy chủ, cổng kết nối, đường dẫn và chuỗi truy vấn. Ví dụ: "/Products/search" là đường dẫn nhằm xác định tài nguyên cụ thể trong máy chủ. Tuy nhiên, đối với ứng dụng web đường dẫn thường có nghĩa là một chức năng (hành động) cụ thể. Hình 1, mô tả chức năng tìm kiếm sản phẩm, ký tự "?" xác định bắt đầu chuỗi truy vấn, ký tự "&" phân cách các tham số trong chuỗi truy vấn, "q=1&enable=true" là nội dung chuỗi truy vấn. "q=1" là một tham số với "q" là tên, "1" là giá trị.

Để đảm bảo tính ổn định và an toàn của hệ thống, máy chủ web thường cấu hình mặc định giới hạn kích thước tối đa của chuỗi truy vấn (bảng 1) mặc dù trong tiêu chuẩn chính thức RFC 2616 (HTTP/1.1) không qui định một cách rõ ràng. Ngoài ra, qua khảo sát các thiết bị WAF của các nhà cung cấp thiết bị bảo mật nổi tiếng Barracuda, Fortinet và Imperva cho thấy cấu hình mặc định, khuyến nghị kích thước nhỏ hơn 4096 bytes (bảng 2).

Bảng 1. Độ dài tối đa chuỗi truy vấn trong cấu hình mặc định của các máy chủ web

Tên máy chủ	Tên thuộc tính	Giá trị (byte)
Microsoft IIS (2022)	maxQueryString	2048
Apache (Core - Apache HTTP Server Version 2.4, n.d.)	LimitRequestLine	8190
Nginx (Module Ngx_Http_Core_ Module, n.d.)	large_client_head er_buffers	4096

Bảng 2. Thông tin cấu hình khuyến nghị, mặc định của các nhà cung cấp WAF

Nhà cung cấp	Tên thuộc tính	Giá trị (byte)
Barracuda (Configuring Request Limits, 2020)	- Chiều dài tối đa truy vấn - Chiều dài tối đa URL	4096 4096
Fortinet (Configuring an HTTP Protocol Constraint Policy, n.d.)	- Chiều dài tên tham số URL tối đa - Chiều dài giá trị tham số URL tối đa	1024 4096
Imperva (Imperva Documentation Portal, n.d.)	Chiều dài giá trị tham số URL tối đa	4096

Các truy vấn độc hại tấn công vào ứng dụng web thường thay đổi sự phân phối và trình tự của ký tự trong chuỗi. Trong danh sách Top 10 OWASP (dự án mở về bảo mật ứng dụng Web) có nhiều lỗ hổng bảo mật sử dụng các kỹ thuật thay đổi nội dung của http, url như: lỗi nhúng mã (A03); phá vỡ kiểm soát truy cập (A01); sử dụng thành phần đã tồn tại lỗ hổng (A06), các kỹ thuật khai thác thông tin dựa trên cấu trúc http như: http parameter pollution, http verb tampering, http flood, http host header, http header injection, …

Nhiều mô hình nhận dạng tấn công dựa trên sự bất thường đã được nghiên cứu và chủ yếu tập trung vào phân tích yêu cầu http. Các mô hình nhận dạng bất thường trên một yêu cầu đầu vào có thể phân thành các nhóm là: dựa trên kỹ thuật thống kê các tham số http (Kruegel & Vigna, 2003; Kruegel et al., 2005), dựa trên chuyển đổi chuỗi url sang dạng các vector đặc trưng hoặc ma trận sau đó dùng các thuật toán

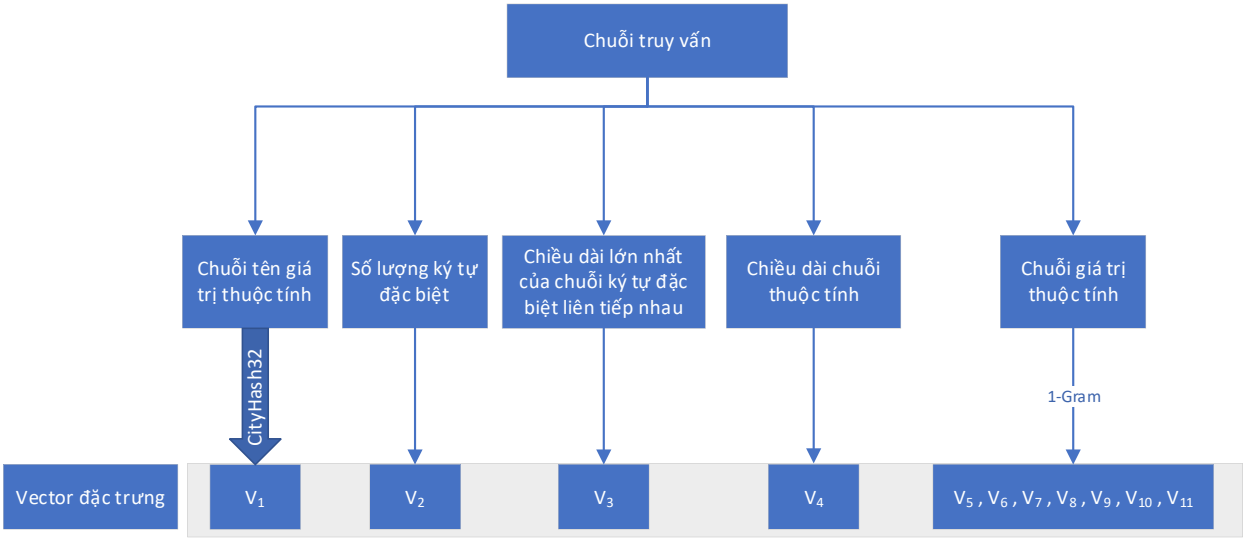
phân lớp để phát hiện bất thường (Kruegel et al., 2005; Vartouni et al., 2018; Le Dinh & Xuan, 2017; Dau et al., 2022), kết hợp nhiều thông tin của một yêu cầu http.

Theo nghiên cứu của Kruegel, các mô hình nhận dạng dựa vào cấu trúc yêu cầu http sẽ kiểm tra cấu trúc và tính logic của một yêu cầu gửi đến máy chủ, nhằm phát hiện các cấu trúc lỗi và độc hại bao gồm: đường dẫn, tham số của của một truy vấn (thông tin loại tham số, giá trị, phân bố ký tự, chiều dài, …) (Kruegel & Vigna, 2003; Kruegel et al., 2005). Ngoài ra, mô hình Markov ẩn (Corona et al., 2009) được sử dụng phân tích các thuộc tính và các giá trị tương ứng trong các truy vấn.

Truy vấn http bao gồm các tham số là chuỗi ký tự, vì vậy nhiều mô hình đã sử dụng kỹ thuật trích xuất dạng ký tự, chuỗi ký tự sang vector đặc trưng. Một cách tiếp cận phổ biến là chia

Đối với mô hình kết hợp nhiều thông tin của một yêu cầu http, Yao Pan , Fangzhou Sun, và nhóm đề xuất áp dụng mô hình end-to-end deep learning để phát hiện các cuộc tấn công, sử dụng công cụ RSMT (robust software modeling tool) tự động theo dõi và mô tả hành vi thời gian thực của các ứng dụng web (Pan et al., 2019). Tianlong Liu, Yu Qi, Liang Shi và Jianan Yan đề xuất mô hình Locate-Then-Detect dùng nhận dạng tấn công thời gian thực thông qua Attention-based Deep Neural Networks (Liu et al., 2019), mô hình chia thành 2 chức năng chính là Payload Location Network (PLN) để tạo ra các khu vực có khả năng là bất thường từ số lượng lớn các truy cập và Payload Classification Network (PCN) để phát hiện chính xác các cuộc tấn công trong các khu vực do PLN tạo ra.

3. PHƯƠNG PHÁP ĐỀ XUẤT



Hình 2. mô tả thành phần của vector đặc trưng

url thành nhiều ký tự theo các quy tắc nhất định và sau đó sử dụng các vectơ khác nhau để đại diện cho từng ký tự (Li et al., 2020). Theo Sureda Riera, kỹ thuật N-Gram được các nghiên cứu sử dụng nhiều nhất với số bài báo là 12, tiếp theo là kỹ thuật Bag-Of-Word với 2 bài báo (Sureda Riera et al., 2020). Ngoài ra, mạng neural nhân tạo cũng được áp dụng chuyển đổi các url dạng chuỗi ký tự thành các vectơ hoặc ma trận trong đó chủ yếu là Stacked Auto-encoder và Word2vec.

Mô hình dựa trên kỹ thuật thống kê mô tả một đặc trưng của một yêu cầu hợp pháp, chẳng hạn như độ dài tham số truy vấn, phân phối ký tự trong tham số và chế độ chuyển đổi ký tự của tham số (Li et al., 2020). Tuy nhiên, mô hình dựa vào các tính năng được trích xuất thủ công nên chỉ có thể phát hiện các kiểu tấn công cụ thể.

Đối với mô hình xây dựng vector đặc trưng dựa trên dựa trên kỹ thuật *N-Gram*, nhưng *N-*

gram ngắn dễ bị tấn công bắt chước, trong đó kẻ tấn công cẩn thận thêm các ký tự để tiến gần hơn đến phân phối *N-gram* dự kiến (Betarte et al., 2018). Các cuộc tấn công bắt chước trở nên khó khăn hơn nhiều khi sử dụng *N-gram* bậc cao hơn. Tuy nhiên, nếu *N* càng lớn thì kích thước của đặc trưng tăng lên theo cấp số nhân dẫn đến chi phí tính toán lớn và tốn bộ nhớ (Vartouni et al., 2018).

$$S = \{N - gram_i | \ i = 1, 2, 3 \ldots\ldots, C^N\} \quad (1)$$

Trong đó, *S* là chuỗi phát sinh, *C* là số lượng ký tự, *N* là số *N-gram* lựa chọn. Ví dụ với n = 2, C = 63 thì S = 632 = 3,969. Do đó, một số mô hình sử dụng các kỹ thuật giảm số lượng ký tự và giảm số chiều đặc trưng như PCA, StackAutoEncoder (Vartouni et al., 2018; Betarte et al., 2018; Dau et al., 2022).

Betarte và Li đưa ra kỹ thuật Word Embedding sẽ chuyển yêu cầu http về dạng các token đặc trưng để xây dựng ma trận các đặc trưng sau đó dùng các thuật toán phân loại để xác định yêu cầu bất thường (Betarte et al., 2018; Li et al., 2020). Hạn chế là độ chính xác của mô hình là phụ thuộc vào kinh nghiệm phân tích chuỗi thành các token và điều chỉnh các thông số phù hợp cho mô hình huấn luyện.

Với các phân tích trên, chúng tôi đề xuất xây dựng vector đặc trưng dựa trên kỹ thuật chuyển đổi cấu trúc chuỗi truy vấn sang dạng mã hóa và 1-gram kết hợp các thông số thống kê về chiều dài, số lượng ký tự đặc biệt, chiều dài lớn nhất của chuỗi ký tự đặc biệt liên tiếp nhau, chiều dài chuỗi thuộc tính (mô tả hình 2).

3.1 Chuyển đổi cấu trúc chuỗi truy vấn

Cho một chức năng web có chuỗi truy vấn

$q = ((a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n))$ với n là số lượng các thuộc tính, trong đó a_i, b_i lần lượt là tên và giá trị của thuộc tính tạo ra hai chuỗi con lần lượt là:

Một là: chuỗi tên thuộc tính

$$\text{strA} = (\text{"1"} + a_1 + \text{"2"} + a_2 + \ldots. + \text{“n”} + a_n) \quad (2)$$

Việc sử dụng công thức (2) nhằm đảm bảo tránh trường hợp có cùng kết quả đối với hai chuỗi khác nhau khi gom các tên thuộc tính thành một chuỗi. Ví dụ: xem xét hai chuỗi truy vấn (chuỗi 1) id=123&passwd=123, (chuỗi 2) idpasswd=123

Áp dụng công thức (2), ta có:

(chuỗi 1) => 1id2passwd

(chuỗi 2) => 1idpasswd

Như vậy, kết quả là hai chuỗi khác nhau so với việc ghép các ký tự với nhau sẽ có cùng giá trị là “idpasswd”

Hai là: chuỗi giá trị thuộc tính

$$\text{strB} = (b_1 + b_2 + \ldots. + b_n) \quad (3)$$

Trong đó giá trị b_i được mã hóa theo bảng sau:

Bảng 3. Nội dung mã hóa giá trị thuộc tính

Giá trị thuộc tính	Giá trị mã hóa
Chỉ toàn ký tự alphabet	<C>
Chỉ toàn là số	<N>
Chỉ toàn ký tự đặc biệt	<S>
Chỉ toàn ký tự alphabet và số	<M>
Chỉ toàn ký tự alphabet và ký tự đặc biệt	<Q>
Chỉ toàn số và ký tự đặc biệt	<P>
Bao gồm ký tự alphabet, số và ký tự đặc biệt	<Z>

Bảng 3 mã hóa giá trị thuộc tính nhằm biểu diễn sự phân bố và phân loại ký tự của thuộc tính. Theo William mô hình phân phối ký tự của thuộc tính dựa trên khái niệm về sự "bình thường" hoặc "thường xuyên" trong các tham số truy vấn bằng cách quan sát việc phân phối các ký tự trong giá trị của tham số (William et al., 2006). Cách tiếp cận này dựa trên quan sát cho thấy các thuộc tính có cấu trúc thông thường ít

thay đổi, chủ yếu là con người có thể đọc được, và gần như chỉ bao gồm các ký tự in được. Ngoài ra, các truy vấn độc hại trong các cuộc tấn công chèn mã và các truy vấn thông thường khác nhau về phân bố ký tự và chuỗi ký tự (Dong et al., 2018).

Đối với phân loại ký tự của thuộc tính thì thông thường các thuộc tính có giá trị thuộc một dạng dữ liệu cụ thể nào đó ví dụ: tên đăng nhập thường là sự kết hợp các ký tự đọc được trong bảng chữ cái, trong khi mật khẩu là tập các chữ cái và số, ký tự đặc biệt vì vậy việc phân loại các nhóm ký tự cho giá trị của một thuộc tính là cần thiết giúp phát hiện các giá trị bất thường trong truy vấn. Một số nghiên cứu gần đây các tấn công khai thác lỗi XSS, khai thác lỗi nhúng mã độc, khai thác lỗi SQL Injection có sử dụng nhiều ký tự đặc biệt (Abikoye et al., 2020; Tang et al., 2020; Kuppa et al., 2022). Ngoài ra, khảo sát website chuyên cung cấp thông tin lỗ hổng bảo mật lớn là exploit-db.com, cve.mitre.org cho thấy các ký tự non-alphanumeric thường được sử dụng trong kỹ thuật tấn công ứng dụng web. Do đó, cần thiết phải phân biệt các ký tự đặc biệt và alphanumeric trong chuỗi các ký tự của giá trị thuộc tính.

Ngược lại, việc phân biệt giữa các số hay giữa các ký tự alphabetic thì không hiệu quả trong việc phát hiện tấn công. Corona và Ming Zhang khi xây dựng mô hình để nhận dạng bất thường đối với giá thuộc tính đã đề xuất chuyển chuỗi ký tự giá trị của thuộc tính thành chuỗi mới theo quy tắc như sau: với giá trị là số chuyển thành N, giá trị ký tự chuyển thành A, các giá trị còn lại sẽ được giữ nguyên (Corona et al., 2009; Zhang et al., 2017).

Ví dụ: truy vấn đến chức năng registed.aspx

http://example.com/registed.aspx?ID=123456 &Content=200&Role=admin

Áp dụng kỹ thuật chuyển đổi cấu trúc chuỗi truy vấn ta có:

strA = "1ID2Content3Role" và strB = "NNC"

3.2 Xây dựng vector dựa trên kỹ thuật chuyển đổi cấu trúc và các thông số thống kê chuỗi truy vấn

Theo nghiên cứu của Kruegel, các mô hình dựa vào việc thống kê cấu trúc cấu trúc và tính logic của truy vấn trong quá trình huấn luyện từ đó phát hiện truy vấn có cấu trúc khác với cấu trúc đã được học, từ đó có thể xem đó là truy vấn bất thường trong đó đường dẫn và tham số của truy vấn (Kruegel & Vigna, 2003; Kruegel et al., 2005). Một số mô hình đề xuất:

- Mô hình đường dẫn thống kê danh sách các đường dẫn được truy cập hợp pháp. Nếu các yêu cầu đường dẫn không nằm trong danh sách được xem là bất thường.

- Mô hình tham số xem xét sự tồn tại, thứ tự của các tham số trong yêu cầu, từ đó nhận dạng các yêu cầu có sự thay đổi số lượng, sự xuất hiện và trình tự tham số để đánh giá có thể là truy vấn tấn công.

- Mô hình thống kê các quan hệ hợp lệ giữa đường dẫn và các tham số được học cho mỗi đường dẫn duy nhất.

- Mô hình kiểu giá trị xem xét các kiểu đặc tính giá trị của tham số. Ví dụ: kiểu string, integer, boolean hay double… nhằm nhận ra việc chèn dữ liệu không hợp lệ.

- Mô hình chiều dài giá trị xem xét chiều dài của mỗi giá trị tham số. sau đó tính trung bình mẫu và phương sai mẫu. Một truy vấn bị cho là bất thường khi so sánh chiều dài của tham số với một giá trị ngưỡng.

Trên cơ sở nghiên cứu các mô hình trên, chúng tôi xây dựng vector đặc trưng từ dựa vào kỹ thuật 1-Gram với số lượng từ vựng chính là số lượng ký tự mã hóa theo bảng 3 kết hợp thông số thống kê bao gồm: chiều dài, số lượng ký tự đặc biệt, chiều dài lớn nhất của chuỗi ký tự đặc biệt liên tiếp nhau, chiều dài chuỗi thuộc tính được mô tả trong hình 2. Cụ thể như sau:

Cho một chức năng web với chuỗi truy vấn là

$q = ((a_1, b_1), (a_2, b_2), (a_3, b_3) \ldots, (a_n, b_n))$ với n là số lượng các thuộc tính, trong đó a_i, b_i lần lượt là tên và giá trị của thuộc tính.

Hàm C(q) chuyển đổi chuỗi truy vấn q thành $V = (v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11})$

Trong đó:

v_1 : đại diện sự chuyển đổi tên thuộc tính

$v_1 = H(\text{strA})$ với H là hàm cityhash32. Chuỗi tên thuộc tinh sẽ chuyển về ký tự thường.

Ví dụ: strA = 1ID2Content3Role

=> strA = "1id2content3role". Suy ra:

$H(\text{strA})$=H("1id2content3role") = 61,258,6730

v_2 : số lượng ký tự đặc biệt trong chuỗi giá trị thuộc tính.

v_3 : chiều dài lớn nhất của chuỗi ký tự đặc biệt liên tiếp nhau. v_3 là tham số hỗ trợ cho v_2 nhằm tăng độ chính xác khi nhận dạng bất thường đối với các ứng dụng cho phép sử dụng ký tự đặc biệt trong chuỗi truy vấn như xác thực người dùng, quản lý đăng nhập, đăng ký tài khoản, ... ví dụ: trong các tấn công khai thác lỗi XSS, SQL Injection, Directory traversal, …có sử dụng nhiều ký tự đặc biệt.

v_4 : đại diện cho chiều dài chuỗi giá trị thuộc tính.

$v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}$: là tần suất xuất hiện của các giá trị mã hóa <C,N,S,M,Q,P,Z>.

4. THU THẬP VÀ XỬ LÝ DỮ LIỆU

Tập dữ liệu CSIC 2010 chứa lưu lượng truy cập đến ứng dụng web thương mại điện tử được phát triển bởi hội đồng nghiên cứu quốc gia Tây Ban Nha (CSIC). Tập dữ liệu CSIC 2010 là một trong những tập dữ liệu tiêu biểu được sử dụng trong quá trình thử nghiệm các phương pháp, mô hình bảo mật ứng dụng web (Pastrana et al., 2015; Nico Epp et al., 2018; Liu et al., 2020; Jemal et al., 2021). Trong ứng dụng web này, người dùng có thể mua hàng thông qua giỏ hàng và đăng ký tài khoản bằng cách cung cấp một số thông tin cá nhân.

Tập dữ liệu được bao gồm 36.000 truy vấn bình thường và hơn 25.000 truy vấn bất thường. Các truy vấn được gắn nhãn là bình thường (hợp pháp) hoặc bất thường (độc hại). Các truy vấn bất thường được thu thập từ các kỹ thuật tấn công chèn SQL, tràn bộ đệm, thu thập thông tin, tiết lộ tập tin, chèn CRLF, XSS, giả mạo tham số, v.v (HTTP DATASET CSIC 2010, n.d.).

5. THỬ NGHIỆM VÀ ĐÁNH GIÁ

5.1. Thử nghiệm

Mục tiêu của nghiên cứu là tạo ra vector đặc trưng từ chuỗi truy vấn từ đó phân loại là bất thường hay bình thường bằng cách sử dụng các thuật toán học không giám sát nên không yêu cầu dữ liệu phải được gắn nhãn trước. Do đó, việc gắn nhãn dữ liệu chỉ phục vụ việc kiểm tra độ chính xác của các thuật toán và hiệu quả của phương pháp chuyển đổi vector.

Giả định là số lượng truy vấn bình thường sẽ cao hơn nhiều so với truy vấn bất thường, tập dữ liệu huấn luyện Q và tập kiểm tra T bao gồm 7200 truy vấn bình thường (90%) và 800 truy vấn bất thường (10%) được lấy ra từ CSIC 2010. Ngoài ra, các truy vấn dùng trong tập huấn luyện và kiểm tra là truy vấn có thuộc tính. Các truy vấn chỉ đơn thuần là cần lấy tài nguyên tĩnh không được sử dụng.

Áp dụng kỹ thuật xây dựng vector (mục 3.2) đối với chuỗi truy vấn trong tập Q và T, ta có hai tập dữ liệu chứa các vector tương ứng là tập huấn luyện X và tập kiểm tra Y. Ví dụ cho một URL:*"http://localhost:8080/tienda1/publico/pagar.jsp?modo=insertar&precio=514&B1=Pasar+por+caja"*

Bảng 4. Mô tả các bước tạo vector đặc trưng

Nội dung	Kết quả thực hiện
chuỗi truy vấn	"modo=insertar&precio=514&B1=Pasar por caja"
Các thuộc tính	modo=insertar, precio=514, B1=Pasar por caja'

Chuỗi tên thuộc tính	1modo2precio3B1
Chuỗi tên thuộc tính chuyển sang ký tự thường	1modo2precio3b1
Sử dụng hàm cityhash32 với chuỗi tên thuộc tính	3746138844
Chuỗi giá trị thuộc tính	insertar514Pasar por caja
Số lượng ký tự đặc biệt	0
Chiều dài lớn nhất của chuỗi ký tự đặc biệt liên tiếp nhau	0
Chiều dài chuỗi giá trị thuộc tính	25
Tần suất xuất hiện của các giá trị mã hóa (bỏ qua ký tự khoảng trắng)	<C>: 2 <N>: 1 <S>: 0 <M>: 0 <Q>: 0 <P>: 0 <Z>: 0
Vector được chuyển đổi	(3746138844,0,0,25,2, 1,0,0,0,0,0)

Để đo lường độ chính xác nhận dạng bình thường và bất thường, nghiên cứu sử dụng ma trận nhầm lẫn (confusion matrix) mô tả trong bảng 5 bao gồm các thành phần sau:

+ True Positives (TP): số lượng truy vấn thuộc lớp positive (bất thường) được dự đoán đúng.

+ False Positives (FP): số lượng truy vấn thuộc lớp negative (bình thường) được mô hình dự đoán sai là positive (bất thường).

+ True Negatives (TN): số lượng truy vấn thuộc lớp negative (bình thường) được mô hình dự đoán đúng.

+ False Negatives (FN): Số lượng truy vấn thuộc lớp positive (bất thường) được mô hình dự đoán sai là negative (bình thường).

Bảng 5. Ma trận nhầm lẫn

Nhãn thực tế	**Nhãn dự đoán**	
	Lớp Positive	Lớp Negative
Lớp Positive	TP	FN
Lớp Negative	FP	TN

Dựa trên các thành phần này, nghiên cứu sử dụng các độ đo để đánh giá như sau:

+ Độ chính xác (Accuracy): tỉ lệ giữa số lượng dự đoán đúng và tổng số lượng dự đoán.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

+ Độ chính xác dự báo (Precision): tỉ lệ giữa số lượng dự đoán đúng positive và tổng số lượng dự đoán positive.

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

+ Độ chính xác phát hiện (Recall): Tỉ lệ giữa số lượng dự đoán đúng positive và tổng số lượng thực tế positive.

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

+ F1-score: kết hợp giữa precision và recall.

$$\text{F1 score} = 2.\frac{\text{Precision}.\,\text{Recall}}{\text{Precision} + \text{Recall}}$$

Các thuật toán phân lớp học không giám sát thông dụng được thử nghiệm là K-means, DBSCAN, Isolation Forest trong bộ thư viện scikit-learn (một trong những thư viện Python phổ biến nhất được sử dụng cho học máy). Do dữ liệu thực hiện thử nghiệm có sự phân bố chênh lệch lớn nên khi sử dụng các thuật toán trên nếu cụm nào có số điểm phân bố lớn được

gắn nhãn là bình thường và ngược lại là bất thường.

K-Means là một thuật toán phân cụm dữ liệu đơn giản và phổ biến trong lĩnh vực học máy và khai phá dữ liệu. K-Means được cấu hình để phân loại thành hai lớp n_clusters=2.

DBSCAN là một thuật toán phân cụm dựa trên mật độ. Thuật toán này được sử dụng để phân chia dữ liệu thành các cụm dựa trên mật độ của các điểm dữ liệu trong không gian đặc trưng, mà không cần phải xác định trước số lượng cụm. Cấu hình tham số như sau DBSCAN(eps=0.5, min_samples=20). Trong đó eps là khoảng cách tối đa giữa hai mẫu để một mẫu được coi là ở lân cận của mẫu kia, min_sample là số lượng mẫu trong một vùng lân cận để một điểm được coi là điểm cốt lõi.

Isolation Forest là một thuật toán học máy không giám sát dựa trên cây quyết định và hoạt động bằng cách cố gắng tách các điểm bất thường ra khỏi dữ liệu bằng cách sử dụng các cây quyết định đơn giản. Cấu hình được để mặc định theo scikit-learn.

5.2. Đánh giá kết quả

Kết quả so sánh hiệu suất của các thuật toán với bộ dữ liệu thử nghiệm được trình bày ở bảng 7. Các thuật toán có các chỉ số accuracy (>89%) và F1 Score (>90%), precision (>85%), recall (>95%) tương đối cao. Điều này cho thấy các thuật toán hoạt động tốt trên tập dữ liệu kiểm tra trong đó thuật toán DBSCAN đạt kết quả tốt nhất.

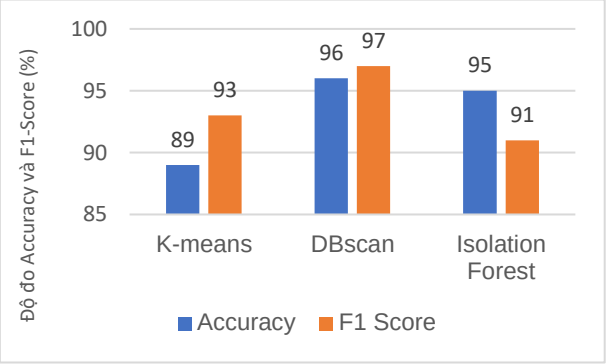
Các truy vấn bất thường có thay đổi về thứ tự xuất hiện, tên hay tăng giảm số lượng thuộc tính kiểm tra cho thấy các thuật toán phân loại chính xác. Nguyên nhân chủ yếu là sự thay đổi ở trên dẫn đến sự khác nhau về chuỗi tên thuộc tính, dẫn đến hàm cityhash32 cho ra kết quả là hai số nguyên thường có khoảng cách lớn và điều này giúp các thuật toán sử dụng độ đo khoảng cách dễ phân lớp.

Bảng 6. Thay đổi tên thuộc tính và sự khác nhau về giá trị tính bằng thuật toán Cityhash32

Chuỗi tên thuộc tính	Cityhash32
1modo2precio3b1	3746138844
Tăng thuộc tính 1modo2precio3b14km	2125970402
Giảm thuộc tính 1modo2precio	498026761
Thay đổi thứ tự 1precio2modo3b1	2250015360

Đối với các truy vấn bất thường chỉ thay đổi nội dung của giá trị thuộc tính như các kỹ thuật tấn công chèn SQL, tràn bộ đệm, chèn CRLF, XSS, ... thuật toán dựa vào khoảng cách như K-Means (khoảng cách giữa mỗi điểm dữ liệu và trọng tâm của cụm) hoặc Isolation Forest (tính toán độ dài trung bình của đường dẫn từ gốc cây đến điểm dữ liệu) không có độ chính xác cao bằng DBSCAN dựa trên mật độ của các điểm dữ liệu (hình 3).

Hình 3. So sánh kết quả theo độ đo accuracy và F1 Score của các thuật toán



Bảng 7. Kết quả đánh hiệu suất các thuật toán

Thuật toán	Accuracy	Precision	Recall	F1 Score
K-means	0.89	0.90	0.97	0.93

DBscan	0.96	0.97	0.98	0.97
Isolation Forest	0. 95	0.86	0.96	0.91

6. KẾT LUẬN

6.1. Kết luận

Để tăng hiệu quả phương pháp nhận dạng tấn công dựa trên sự bất thường của WAF, chúng tôi đã đề xuất xây dựng một vector đặc trưng dựa trên kỹ thuật chuyển đổi cấu trúc và các thông số thống kê chuỗi truy vấn. Thử nghiệm sử dụng các thuật toán học không giám sát đối với bộ dữ liệu vector được tạo ra từ các truy vấn cho kết quả khả năng phân lớp tốt. Phương pháp này có ưu điểm so với các mô hình khác là không phụ thuộc vào loại ứng dụng web và dữ liệu không cần gắn nhãn trước để tiến hành phân lớp . Kết quả của nghiên cứu có thể là đầu vào để thực hiện các mô hình nhận dạng bất thường có hiệu suất tốt hơn và theo thời gian thực.

6.2. Hướng phát triển

Thu thập dữ liệu truy vấn của các ứng dụng web mới để hoàn thiện hơn mô hình nhận dạng. Đặc biệt, nghiên cứu sâu hơn về tấn công với chức năng xác thực nhằm đưa ra các bổ sung, điều chỉnh vector giúp hạn chế nhận dạng sai.

TÀI LIỆU THAM KHẢO

Abikoye, O. C., Abubakar, A., Dokoro, A. H., Akande, O. N., & Kayode, A. A. (2020, August 18). A novel technique to prevent SQL injection and cross-site scripting attacks using Knuth-Morris-Pratt string match algorithm. EURASIP Journal on Information Security, 2020(1). https://doi.org/10.1186/s13635-020-00113-y

Applebaum, S., Gaber, T., & Ahmed, A. (2021). Signature-based and Machine-Learning-based Web Application Firewalls: A Short Survey. Procedia Computer Science, 189, 359–367. https://doi.org/10.1016/j.procs.2021.05.105

Aras Pranckevičius. (2016, August 9). More Hash Function Tests Aras' website. Aras' Website. Retrieved February 12, 2024, from https://aras-p.info/blog/2016/08/09/More-Hash-Function-Tests/

Babiker, M., Karaarslan, E., & Hoscan, Y. (2018, March). Web application attack detection and forensics: A survey. 2018 6th International Symposium on Digital Forensic and Security (ISDFS). https://doi.org/10.1109/isdfs.2018.8355378

Betarte, G., Gimenez, E., Martinez, R., & Pardo, A. (2018, December). Improving Web Application Firewalls through Anomaly Detection. 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). https://doi.org/10.1109/icmla.2018.00124

Blázquez-García, A., Conde, A., Mori, U., & Lozano, J. A. (2021, April 17). A Review on Outlier/Anomaly Detection in Time Series Data. ACM Computing Surveys, 54(3), 1–33. https://doi.org/10.1145/3444690

Configuring an HTTP Protocol Constraint policy. (n.d.). https://help.fortinet.com/fadc/4-8-0/olh/Content/FortiADC/handbook/waf_protocol.htm

Configuring Request Limits. (2020, June 3). Barracuda Campus. https://campus.barracuda.com/product/webapplicationfirewall/doc/4259870/configuring-request-limits

Core - Apache HTTP Server Version 2.4. (n.d.). https://httpd.apache.org/docs/2.4/mod/core.html

Corona, I., Ariu, D., & Giacinto, G. (2009, June). HMM-Web: A Framework for the Detection of Attacks Against Web Applications. 2009 IEEE International Conference on Communications. https://doi.org/10.1109/icc.2009.5199054

Dau, H. X., Trang, N. T. T., & Hung, N. T. (2022, June 8). A Survey of Tools and Techniques for Web Attack Detection. Journal of Science and Technology on Information Security, 1(15), 109–118. https://doi.org/10.54654/isj.v1i15.852

Dik, D., Polyakova, E., Chelovechkova, A., & Moskvin, V. (2019, October). Web Attacks Detection Based on Patterns of Sessions. 2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon). https://doi.org/10.1109/fareastcon.2019.8934015

Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., & Keogh, E. (2008, August). Querying and mining of time series data: Experimental comparison of representations and distance measures. Proceedings of the VLDB Endowment, 1(2), 1542–1552. https://doi.org/10.14778/1454159.1454226

Dong, Y., Zhang, Y., Ma, H., Wu, Q., Liu, Q., Wang, K., & Wang, W. (2018, February 2). An adaptive system for detecting malicious queries in web attacks. Science China Information Sciences, 61(3). https://doi.org/10.1007/s11432-017-9288-4

HTTP DATASET CSIC 2010. (n.d.). HTTP DATASET CSIC 2010. https://www.tic.itefi.csic.es/dataset

Imperva Documentation Portal. (n.d.). https://docs.imperva.com/bundle/v14.3-web-application-firewall-user-guide/page/1178.htm

Jemal, I., Haddar, M. A., Cheikhrouhou, O., & Mahfoudhi, A. (2021). Malicious Http Request Detection Using Code-Level Convolutional Neural Network. Lecture Notes in Computer Science, 317–324. https://doi.org/10.1007/978-3-030-68887-5_19

Jenkins , & Appleby. (n.d.). CityHash, a family of hash functions for strings. Google Github. Retrieved February 12, 2024, from https://github.com/google/cityhash

Juvonen, A., Sipola, T., & Hämäläinen, T. (2015, November). Online anomaly detection using dimensionality reduction techniques for HTTP log analysis. Computer Networks, 91, 46–56. https://doi.org/10.1016/j.comnet.2015.07.019

Kruegel, C., & Vigna, G. (2003, October 27). Anomaly detection of web-based attacks. Proceedings of the 10th ACM Conference on Computer and Communications Security. https://doi.org/10.1145/948109.948144

Kruegel, C., Vigna, G., & Robertson, W. (2005, August). A multi-model approach to the detection of web-based attacks. Computer Networks, 48(5), 717–738. https://doi.org/10.1016/j.comnet.2005.01.009

Kuppa, K., Dayal, A., Gupta, S., Dua, A., Chaudhary, P., & Rathore, S. (2022, May). ConvXSS: A deep learning-based smart ICT framework against code injection attacks for HTML5 web applications in sustainable smart city infrastructure. Sustainable Cities and Society, 80, 103765. https://doi.org/10.1016/j.scs.2022.103765

Langlois, Pinto, Hylender, & Widup. (2023, June). DBIR 2023 Data Breach Investigations Report 10K 20K 30K About the cover. Verizon. https://doi.org/10.13140/RG.2.2.32362.70085

Le Dinh, T., & Xuan, T. P. (2017, October). On the usage of character distribution for the detection of web attacks. 2017 9th International Conference on Knowledge and Systems Engineering (KSE). https://doi.org/10.1109/kse.2017.8119435

Li, J., Fu, Y., Xu, J., Ren, C., Xiang, X., & Guo, J. (2020). Web Application Attack

Detection Based on Attention and Gated Convolution Networks. IEEE Access, 8, 20717–20724. https://doi.org/10.1109/access.2019.2955674

Li, J., Zhang, H., & Wei, Z. (2020). The Weighted Word2vec Paragraph Vectors for Anomaly Detection Over HTTP Traffic. IEEE Access, 8, 141787–141798. https://doi.org/10.1109/access.2020.3013849

Liang, J., Zhao, W., & Ye, W. (2017, December 8). Anomaly-Based Web Attack Detection. Proceedings of the 2017 VI International Conference on Network, Communication and Computing. https://doi.org/10.1145/3171592.3171594

Liu, C., Yang, J., & Wu, J. (2020, February 3). Web intrusion detection system combined with feature analysis and SVM optimization. EURASIP Journal on Wireless Communications and Networking, 2020(1). https://doi.org/10.1186/s13638-019-1591-1

Liu, T., Qi, Y., Shi, L., & Yan, J. (2019, August). Locate-Then-Detect: Real-time Web Attack Detection via Attention-based Deep Neural Networks. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. https://doi.org/10.24963/ijcai.2019/656

Module ngx_http_core_module. (n.d.). http://nginx.org/en/docs/http/ngx_http_core_module.html

Nico Epp, Ralf Funk, & Cristian Cappo. (2018). Anomaly-based Web Application Firewall using HTTP-specific features and One-Class SVM. Zenodo, 2(1). https://doi.org/10.5281/zenodo.1336812

Pan, Y., Sun, F., Teng, Z., White, J., Schmidt, D. C., Staples, J., & Krause, L. (2019, August 27). Detecting web attacks with end-to-end deep learning. Journal of Internet Services and Applications, 10(1). https://doi.org/10.1186/s13174-019-0115-x

Panopoulos, V. (2016, October 7). Near Real-time Detection of Masquerade attacks in Web applications: catching imposters using their browsing behavor. Near Real-time Detection of Masquerade Attacks in Web Applications : Catching Imposters Using Their Browsing Behavor. https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-183777

Park, S., Kim, M., & Lee, S. (2018). Anomaly Detection for HTTP Using Convolutional Autoencoders. IEEE Access, 6, 70884–70901. https://doi.org/10.1109/access.2018.2881003

Pastrana, S., Torrano-Gimenez, C., Nguyen, H. T., & Orfila, A. (2015). Anomalous Web Payload Detection: Evaluating the Resilience of 1-Grams Based Classifiers. Intelligent Distributed Computing VIII, 195–200. https://doi.org/10.1007/978-3-319-10422-5_21

R. A. (2022, April 6). Request Limits <requestLimits>. Microsoft Learn. Retrieved May 2, 2024, from https://docs.microsoft.com/en-us/iis/configuration/system.webserver/security/requestfiltering/requestlimits

Sriraghavan, R. G., & Lucchese, L. (2008, October). Data processing and anomaly detection in web-based applications. 2008 IEEE Workshop on Machine Learning for Signal Processing. https://doi.org/10.1109/mlsp.2008.4685477

Sureda Riera, T., Bermejo Higuera, J. R., Bermejo Higuera, J., Martínez Herraiz, J. J., & Sicilia Montalvo, J. A. (2020, June 17). Prevention and Fighting against Web Attacks through Anomaly Detection Technology. A Systematic Review. Sustainability, 12(12), 4945. https://doi.org/10.3390/su12124945

Tang, P., Qiu, W., Huang, Z., Lian, H., & Liu, G. (2020, February). Detection of SQL injection based on artificial neural network. Knowledge-Based Systems, 190, 105528. https://doi.org/10.1016/j.knosys.2020.105528

Tang, R., Yang, Z., Li, Z., Meng, W., Wang, H., Li, Q., Sun, Y., Pei, D., Wei, T., Xu, Y., & Liu, Y. (2020, July). ZeroWall: Detecting Zero-Day Web Attacks through Encoder-Decoder Recurrent Neural Networks. IEEE INFOCOM 2020 - IEEE Conference on Computer Communications. https://doi.org/10.1109/infocom41043.2020.9155278

Tran, T. M., & Nguyen, K. V. (2019, March). Fast Detection and Mitigation to DDoS Web Attack Based on Access Frequency. 2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF). https://doi.org/10.1109/rivf.2019.8713762

Vartouni, A. M., Kashi, S. S., & Teshnehlab, M. (2018, February). An anomaly detection method to detect web attacks using Stacked Auto-Encoder. 2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS). https://doi.org/10.1109/cfis.2018.8336654

WEN, K., GUO, F., & YU, M. (2013, August 26). Adaptive anomaly detection method of Web-based attacks. Journal of Computer Applications, 32(7), 2003–2006. https://doi.org/10.3724/sp.j.1087.2012.02003

William, Giovanni, Christopher, & Richard. (2006). Using Generalization and Characterization Techniques in the Anomaly-based Detection of Web Attacks. InProc. Network and Distributed System Security Symposium (NDSS). Internet Society.

Wu, Y., Sun, Y., Huang, C., Jia, P., & Liu, L. (2019, November 22). Session-Based Webshell Detection Using Machine Learning in Web Logs. Security and Communication Networks, 2019, 1–11. https://doi.org/10.1155/2019/3093809

Zhang, M., Lu, S., & Xu, B. (2017, December). An Anomaly Detection Method Based on Multi-models to Detect Web Attacks. 2017 10th International Symposium on Computational Intelligence and Design (ISCID). https://doi.org/10.1109/iscid.2017.223

METHOD FOR BUILDING A FEATURE VECTOR IN THE WEB APPLICATION FIREWALL ANOMALY DETECTION MODEL BY UTILIZING QUERY STATISTICS AND STRUCTURAL CONVERSION

Huynh Hoang Tan[1*], Tran Van Hoai[2]

[1] *Dong Nai Technology University*

[2] *Ho Chi Minh City University of Technology*

* Corresponding author: *Huynh Hoang Tan, huynhhoangtan@dntu.edu.vn*

GENERAL INFORMATION

Received date: 28/02/2024

Revised date: 02/05/2024

ABSTRACT

The widespread use of the Internet today, along with the rapid growth of cloud computing, the Internet of Things and smartphones have fueled the need for web-based apps. A web

Accepted date: 30/05/2024

KEYWORD

Detecting abnormal web queries;
Detecting the web attacks.
Web application security;

application firewall (WAF) is a type of unauthorized intrusion detection and prevention system designed to protect web applications. On WAF, attack recognition is often divided into two categories: anomalous and rule-based. Through observation of query data, models based on theoretical anomalies are able to detect undiscovered harmful queries. In this paper, we suggest an approach to characterizing vector construction by modification of the query string's component parts' structure and statistics. The unsupervised classification algorithm will then use the feature vector as input to determine which requests are anomalous. The results of testing the DBSCAN, K-means, and Isolation forest algorithms reveal that DBSCAN has the highest accuracy (F1-score >97%, accuracy >96%), especially for online applications like registration and authentication that are prone to misidentification. The effectiveness of this method stems from its ability to use data without pre-labeling, which facilitates deployment on the WAF.