|  |
| --- |
| BAN CƠ YẾU CHÍNH PHỦ  **HỌC VIỆN KỸ THUẬT MẬT MÃ**  ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯  Logo HvKTMM |
| CHUYÊN ĐỀ KỸ NGHỆ AN TOÀN MẠNG  ***Nghiên cứu giải pháp bypass tường lửa ứng dụng web*** |
| Ngành: An toàn thông tin  Mã số: 7.48.02.02  *Sinh viên thực hiện*:  **Lưu Văn Tùng – AT160456**    *Người hướng dẫn*:  **TS. Nguyễn Mạnh Thắng   Th.S Nguyễn Thị Thu Thủy**  Khoa An toàn thông tin – Học viện Kỹ thuật mật mã |
| Hà Nội, 2023 |

Mục lục

[Danh mục kí hiệu và viết tắt 4](#_Toc122478073)

[MỞ ĐẦU 5](#_Toc122478074)

[I. Tính cấp thiết của đề tài 5](#_Toc122478075)

[II. Mục tiêu thực hiện đề tài 5](#_Toc122478076)

[Chương 1. Giới thiệu về tường lửa ứng dụng web 6](#_Toc122478077)

[1.1. Tồng quan về tường lửa ứng dụng web (WAF) 6](#_Toc122478078)

[1.1.1. Tường lửa ứng dụng web (WAF) 6](#_Toc122478079)

[1.1.2. Lới ích của tường lửa ứng dụng web (WAF) 8](#_Toc122478080)

[1.1.3. Tầm quan trọng của tường lửa ứng dụng web (WAF) 9](#_Toc122478081)

[1.2. Kiến trúc của tường lửa ứng dụng web (WAF) 10](#_Toc122478082)

[1.2.1. Máy chủ và vị trí đặt WAF 10](#_Toc122478083)

[1.2.2. Mô hình bảo mật WAF 10](#_Toc122478084)

[1.2.3. Mô hình hoạt động. 10](#_Toc122478085)

[1.2.4. Một số các cấu hình tường lửa ứng dụng web (WAF ) 11](#_Toc122478086)

[1.2.5. Ví dụ 1 số sản phẩn phổ biến cho tường lửa ứng dụng web (WAF ) 13](#_Toc122478087)

[1.3. Một số loại tường lửa ứng dụng web (WAF) 13](#_Toc122478088)

[1.3.1. Tường lửa ứng dụng web host-based 13](#_Toc122478089)

[1.3.2. Tường lửa ứng dụng web netword-base. 13](#_Toc122478090)

[1.3.3. Tường lửa ứng dụng web cloud& hybrid 14](#_Toc122478091)

[1.4. Ưu nhược điểm của tường lửa ứng dụng web 14](#_Toc122478092)

[Kết luận chương 1 15](#_Toc122478093)

[Chương 2. kĩ thuật bypass ứng dụng web (WAF ) 16](#_Toc122478094)

[2.1. Khai thác bộ xử lý mới 16](#_Toc122478095)

[2.1.1. Bỏ qua xác minh tham số 16](#_Toc122478096)

[2.1.2. Những phương thức HTTP không đúng định dạng 16](#_Toc122478097)

[2.1.3. Làm quá tải tường lửa ứng dụng web (WAF) 16](#_Toc122478098)

[2.2. Impedance Mismatch 16](#_Toc122478099)

[2.2.1. HTTP parameter Pollution 16](#_Toc122478100)

[2.2.2. Double URL Encoding 18](#_Toc122478101)

[2.3. Bỏ qua bộ quy tắc ứng dụng tường lửa web mới 18](#_Toc122478102)

[2.3.1. Brute force bang cách liệt kê các payload khác nhau 18](#_Toc122478103)

[2.3.2. Dịch ngược bộ quy tắc ứng dụng tường lửa web (WAF) 19](#_Toc122478104)

[2.4. Một số phương pháp tiếp cận để kiểm thử xâm nhập 19](#_Toc122478105)

[2.4.1. Tắt tường lửa ứng dụng web 19](#_Toc122478106)

[2.4.2. Thu thập thông tin 19](#_Toc122478107)

[2.4.3. Một số lỗ hổng thường gặp 21](#_Toc122478108)

[2.5. Giới thiệu về công cụ WAFNinja 21](#_Toc122478109)

[2.5.1. Định nghĩa công cụ WAFNinja 21](#_Toc122478110)

[2.5.2. Công dụng của công cụ WAFNinja 22](#_Toc122478111)

[2.5.3. Triển khai 22](#_Toc122478112)

[Kết luận chương 2 22](#_Toc122478113)

[Chương 3. Triển khai bypass tường lửa ứng dụng web 23](#_Toc122478114)

[3.1. Triển khai cầu hình tường lửa ModSecurity . 23](#_Toc122478115)

[3.1.1. Chức năng của ModSecurity. 23](#_Toc122478116)

[3.1.2. Quy trình xử lý trong ModSecurity 23](#_Toc122478117)

[3.1.3. Cấu trúc tập luật trong ModSecurity 25](#_Toc122478118)

[3.1.4. Chị tiết cấu hình và cài đặt hệ thống ModSecurity 26](#_Toc122478119)

[3.2. Triển khai thực nghiệm bypass tường lửa ứng dụng web (WAF) 29](#_Toc122478120)

[3.2.1. Kịch bản triển khai 29](#_Toc122478121)

[3.2.2. Triển khai thức nghiệm 29](#_Toc122478122)

[3.3. Đánh giá 33](#_Toc122478123)

[Kết luận chương 3 33](#_Toc122478124)

[Kết luận 34](#_Toc122478125)

[Tài liệu tham khảo 35](#_Toc122478126)

Danh mục kí hiệu và viết tắt

|  |  |
| --- | --- |
| WAF | (Web Application Firewall): tường lửa ứng dụng we |
| SQL | (Structured Query Language): ngôn ngữ truy vấn cơ sở dữ liệu |
| XSS | (Cross Site Scripting): kịch bản chéo trang |
| CSRF | (Cross-site request Fogery): tấn công giả mạo chính chủ thể của nó |
| CC | Challenge Collapsar |
| NGFW | Next Generation Firewall |
| IIS | Internet Information Services |
| Ddos | Distributed Denial of Service |
| HPP | HTTP parameter pollution |
| GUI | Graphical User Interface |
| IDS | ids Intrusion Detection System |
| IPS | ips Intrusion prevention system |

# MỞ ĐẦU

## Tính cấp thiết của đề tài

Hiện nay ngày càng nhiều công ty sử dụng tường lửa ứng dụng web điều đó làm cho việc kiểm thử thâm nhập phần mềm khó khăn hơn nhiều. Các chuyên gia bảo mật thực hiện đánh giá bảo mật của các ứng dụng web để xác định các lỗ hổng có thể bị khai thác bởi người dùng độc hại. Tường lửa ứng dụng Web tăng cường thêm lớp bảo vệ cho các ứng dụng web để giảm thiểu các lỗ hổng. Nỗ lực vượt qua tường lửa ứng dụng web là một khía cạnh quan trọng của đánh giá bảo mật và cần thiết để đảm bảo kết quả kiểm thử chính xác.

Những gì người kiểm thử thâm nhập cần làm là tìm lỗ hổng trên ứng dụng web và sau đó báo cáo lỗ hổng đã tìm được cho khách hàng hoặc cho công ty. Các công ty có xu hướng sử dụng tường lửa ứng dụng web, khiến công việc của người kiểm thử thâm nhập trở lên khó khăn hơn. Những người kiểm thử thâm nhập cần biết cách bypass tường lửa để đạt được kết quả kiểm thử tốt nhất.

## Mục tiêu thực hiện đề tài

Mục tiêu của đề tài này là hiểu được tường lửa ứng dụng web hoạt động như thế nào, những giới hạn của nó và tìm hiểu các phương pháp bypass tường lửa ứng dụng web, đưa ra cách tiếp cận để vượt qua WAF cho việc kiểm thử thâm nhập đảm bảo kết quả chính xác.

**Tên đề tài: “Nghiên cứu giải pháp bypass tường lửa ứng dụng web”.**

Mục tiêu đặt ra khi thực hiện đề tài:

1. Hiểu được tường lửa web hoạt động như thế nào và những giới hạn của nó

2. Đưa ra một số giải pháp bypass tường lửa ứng dụng web

Đề tài gồm 3 chương:

Chương 1: Giới thiệu về tường lửa ứng dụng web

Chương 2: Những phương pháp và kỹ thuật bypass WAF

Chương 3: Triển khai bypass tường lửa ứng dụng web

|  |  |
| --- | --- |
|  |  |

# Giới thiệu về tường lửa ứng dụng web

## Tồng quan về tường lửa ứng dụng web (WAF)

### **Tường lửa ứng dụng web (WAF)**

WAF là một thiết bị proxy có thể xử lý giao thức HTTP nhằm bảo vệ ứng dụng web. WAF kiểm tra lượng truy cập và sẽ lọc ra các yêu cầu có mối đe dọa xâm hại đến website trước khi đến ứng dụng web.

Tường lửa ứng dụng Web (Web Application Firewall – WAF) được triển khai ở đường biên mạng (Network Edge). Nó thực hiện việc kiểm tra lưu lượng truy cập đến và đi khỏi các ứng dụng Web. WAF có thể lọc và giám sát lưu lượng truy cập để bảo vệ chống lại các cuộc tấn công như SQL Injection, Cross Site Scripting (XSS) hay Cross-site Request Forgery – CSRF (tấn công giả mạo yêu cầu Cross-site).

WAF hoạt động tại lớp 7 (tầng ứng dụng). Mặc dù WAF có thể bảo vệ chống lại hàng loạt các cuộc tấn công ở lớp ứng dụng nhưng nó không thể tự hoạt động một mình mà phải kết hợp với các công cụ bảo mật khác. Có vậy mới đảm bảo chống lại các cuộc tấn công nhắm vào các lớp mạng khác hoặc các phần khác của môi trường bảo mật.

Tường lửa ứng dụng Web là thiết bị bảo vệ chuyên biệt cho ứng dụng, có thể phân tích, hiểu sâu được cấu trúc, thuộc tính và hoạt động của website cụ thể của mỗi tổ chức. Nhờ vậy nó có thể phát hiện và ngăn chặn cả tấn công chưa biết và những hành vi bất thường trên ứng dụng, tấn công tinh vi, sử dụng kỹ thuật lẩn tránh mà có thể dễ dàng vượt qua tường lửa mức mạng và IPS. Tường lửa có khả năng kết hợp với các giải pháp đánh giá điểm yếu để đưa ra bảo vệ điểm yếu được phát hiện ngay lập tức với bản vá ảo. Giải pháp tường lửa ứng dụng Web hỗ trợ nhiều mô hình triển khai cho phép triển khai hoàn toàn trong suốt mà không làm thay đổi tới tới mạng và ứng dụng đang hoạt động .Giải pháp cho phép:

* Bảo vệ địa chỉ IP và tên miền: WAF cung cấp đám mây và các phiên bản chuyên dụng để bảo vệ các trang web của bạn. Bạn có thể thêm tên miền hoặc địa chỉ IP vào WAF.
* Bảo vệ dịch vụ HTTP / HTTPS: WAF giữ cho các ứng dụng ổn định và an toàn. Nó kiểm tra các yêu cầu HTTP và HTTPS để phát hiện và chặn các cuộc tấn công, chẳng hạn như đưa vào ngôn ngữ truy vấn cấu trúc (SQL), viết mã trang web chéo (XSS), tải lên web shell, tiêm lệnh hoặc mã, bao gồm tệp, truy cập tệp nhạy cảm, lỗ hổng của bên thứ ba khai thác, tấn công CC, trình thu thập dữ liệu độc hại và giả mạo yêu cầu trên nhiều trang web (CSRF).
* WebSocket / WebSockets: WAF hỗ trợ giao thức WebSocket / WebSockets, được bật theo mặc định.
* Bảo vệ web cơ bản: Với cơ sở dữ liệu danh tiếng mở rộng được cài đặt trước, WAF bảo vệ chống lại 10 mối đe dọa hàng đầu của Dự án Bảo mật Ứng dụng Web Mở (OWASP), khai thác lỗ hổng, trình bao web và các mối đe dọa khác.
* Bảo vệ toàn diện: WAF phát hiện và chặn các cuộc tấn công khác nhau, chẳng hạn như SQL injection, XSS, lỗ hổng tràn từ xa, bao gồm tệp, lỗ hổng Bash, thực thi lệnh từ xa, tấn công truyền qua thư mục (đường dẫn), truy cập tệp nhạycảm trái phép, tiêm lệnh / mã và đưa vào XML hoặc Xpath các cuộc tấn công.
* Phát hiện vỏ web: WAF bảo vệ khỏi trình bao web từ giao diện tải lên.
* Nhận dạng chính xác.
* Kiểm tra sâu: WAF xác định và chặn các cuộc tấn công trốn tránh, chẳng hạn như các cuộc tấn công sử dụng sự xáo trộn ký tự đồng hình, chèn lệnh với các ký tự đại diện biến dạng, UTF7, lược đồ URI dữ liệu và các kỹ thuật khác.
* Phát hiện tiêu đề: WAF phát hiện tất cả các trường tiêu đề trong các yêu cầu.
* Phòng chống tấn công CC: Có thể tùy chỉnh quy tắc bảo vệ chống tấn công CC để hạn chế quyền truy cập vào một URL cụ thể trên trang web của mình dựa trên địa chỉ IP, cookie hoặc Người giới thiệu, giảm thiểu các cuộc tấn công CC. Các hành động bảo vệ của các quy tắc bảo vệ chống tấn công CC bao gồm Mã xác minh , Chặn , Chặn động và Chỉ nhật ký .
* Dữ liệu bảo mật dựa trên GUI: WAF cung cấp giao diện dựa trên GUI để ngườidùng theo dõi thông tin tấn công và nhật ký sự kiện trong thời gian thực.
* Bảo vệ kết nối: Nếu một số lượng lớn lỗi 502 Bad Gateway và 504 Gateway Timeout được phát hiện, bạn có thể bật tính năng bảo vệ sự cố WAF và bảo vệ kết nối để cho phép WAF tạm ngưng trang web của bạn và bảo vệ máy chủ gốccủa bạn không bị hỏng. Khi yêu cầu lỗi 502/504 và yêu cầu URL đang chờ xử lý đạt đến ngưỡng bạn định cấu hình, WAF sẽ bật tính năng bảo vệ tương ứng cho trang web.
* Mặt nạ dữ liệu: WAF che thông tin nhạy cảm, chẳng hạn như tên người dùng vàmật khẩu, trong nhật ký sự kiện.
* Ngăn ngừa rò rỉ thông tin: WAF ngăn thông tin nhạy cảm của người dùng bị tiết lộ trên các trang web, chẳng hạn như số ID, số điện thoại và địa chỉ email.
* Thông báo báo động: Người dùng có thể bật thông báo cho nhật ký tấn công. Khi chức năng này được kích hoạt, WAF sẽ gửi nhật ký tấn công cho người dùng bằng phương pháp định cấu hình.

### **Lới ích của tường lửa ứng dụng web (WAF)**

WAF có lợi thế hơn tường lửa truyền thống vì nó cung cấp khả năng hiển thị tốt hơn đối với dữ liệu ứng dụng nhạy cảm được giao tiếp bằng cách sử dụng lớp ứng dụng HTTP. Nó có thể ngăn chặn các cuộc tấn công lớp ứng dụng thường vượt qua tường lửa mạng truyền thống, bao gồm những điều sau:

* Các cuộc tấn công cross-site scripting (XSS) cho phép kẻ tấn công đưa và thực thi các đoạn mã độc hại trong trình duyệt của người dùng khác.
* Các cuộc tấn công đưa vào ngôn ngữ truy vấn có cấu trúc (SQL) có thể ảnh hưởng đến bất kỳ ứng dụng nào sử dụng cơ sở dữ liệu SQL và cho phép kẻ tấn công truy cập và có khả năng thay đổi dữ liệu nhạy cảm.
* Hack phiên web cho phép kẻ tấn công chiếm đoạt ID phiên và giả dạng người dùng được ủy quyền. ID phiên thường được lưu trữ trong cookie hoặc Uniform Resource Locator (URL).
* Các cuộc tấn công từ chối dịch vụ (DDoS) phân tán áp đảo một mạng bằng cách làm ngập nó với lưu lượng truy cập cho đến khi nó không thể phục vụ người dùng. Cả tường lửa mạng và WAF đều có thể xử lý kiểu tấn công này nhưng tiếp cận nó từ các lớp khác nhau.

Một ưu điểm khác của WAF là nó có thể bảo vệ các ứng dụng dựa trên web mà không nhất thiết phải có quyền truy cập vào mã nguồn của ứng dụng. Trong khi WAF dựa trên máy chủ có thể được tích hợp vào mã ứng dụng, WAF được lưu trữ trên đám mây có khả năng bảo vệ ứng dụng mà không cần có quyền truy cập. Ngoài ra, WAF đám mây rất dễ triển khai và quản lý, đồng thời cung cấp các giải pháp vá lỗi ảo nhanh chóng cho phép người dùng tùy chỉnh nhanh các cài đặt của họ để thích ứng với các mối đe dọa mới được phát hiện.

### **Tầm quan trọng của tường lửa ứng dụng web (WAF)**

WAF có thể đặc biệt có lợi cho công ty cung cấp trang thương mại điện tử e-Commerce, những dịch vụ tài chính trực tuyến hoặc bất kỳ loại sản phẩm hay dịch vụ dựa trên web nào khác liên quan đến tương tác với khách hàng hoặc đối tác kinh doanh. Trong những trường hợp này, WAF có thể đặc biệt hữu ích trong việc ngăn chặn gian lận và đánh cắp dữ liệu.

Ví dụ: các ngân hàng có thể sử dụng WAF để giúp họ đáp ứng Tiêu chuẩn bảo mật dữ liệu ngành thẻ thanh toán (PCI DSS), là một bộ chính sách để đảm bảo rằng dữ liệu của chủ thẻ (CHD) được bảo vệ. Cài đặt tường lửa là một trong 12 yêu cầu tuân thủ PCI DSS. Việc tuân thủ này áp dụng cho bất kỳ doanh nghiệp nào xử lý CHD. Vì nhiều công ty mới hơn sử dụng các ứng dụng di động và Internet vạn vật (IoT) ngày càng phát triển, ngày càng có nhiều giao dịch diễn ra ở lớp ứng dụng bằng cách sử dụng web. Vì lý do này, WAF là một phần quan trọng trong mô hình bảo mật của một doanh nghiệp hiện đại.

WAF rất quan trọng, và nó hiệu quả nhất khi kết hợp với các thành phần bảo mật khác, bao gồm IPS, IDS và tường lửa cổ điển hoặc thế hệ tiếp theo (NGFW). Một mô hình bảo mật doanh nghiệp toàn diện lý tưởng sẽ đặt WAF bên cạnh các loại tường lửa khác, chẳng hạn như NGFW và các thành phần bảo mật, chẳng hạn như IPS và IDS, thường được bao gồm trong NGFW.

Diagram

Description automatically generated

Hình 1.1 :Mô hình của một hệ thống Tường lửa ứng dụng Web (WAF)

## Kiến trúc của tường lửa ứng dụng web (WAF)

### Máy chủ và vị trí đặt WAF

Tường lửa ứng dụng web có 2 hình thức triển khai chinh : dạng phần cứng và dạng cloudbased

Các thiết bị tường lửa ứng dụng web dạng phần cứng thường đặt sau tường lửa mạng và đặt trước máy chủ ứng dụng web, việc đặt WAF được thực hiện sao cho tất cả lưu lượng đến ứng dụng web cần qua tường lửa ứng dụng web trước .

Tuy nhiên đôi khi cũng có 1 số ngoại lệ khi WAF chỉ được dung để giám sát cổng đang mở trên máy chủ web .

Ngoài ra ,các chương trình WAF còn được cài đặt trực tiếp lên máy chủ web và thực hiện các chức năng tương tự như các thiết bị WAF là giám sát các lưu lượng đến và ra khỏi ứng dụng web.

Diagram

Description automatically generated

Hình 1.2: Ví dụ triển khai WAF trong máy chủ web Windows

### Mô hình bảo mật WAF

Một tường lửa ứng dụng web (WAF) hoạt động dựa trên 2 mô hình bảo mật là : positive và negative .

* Mô hình Positive chỉ cho phép các lưu lượng hợp lệ được định nghĩa sẵn đi qua va chặn tất cả các lưu lượng còn lại.
* Mô hình Negative sẽ cho phép tất cả các lưu lượng vượt qua và chỉ chặn được các lưu lượng mà WAF cho là nguy hại.

Đôi khi cũng có các WAF cung cấp cả 2 mô hình trên, tuy nhiên thông thường WAF chỉ cung cấp 1 trong 2 mô hình.

Với mô hình Postitive thì đòi hỏi nhiều cấu hình và tùy chỉnh. Còn mô hình Negative chu yếu dựa vào khả năng học hỏi và phân tích hành vi của lưu lượng mạng.

### Mô hình hoạt động.

Tường lửa ứng dụng web (WAF) có thể hoạt động riêng biệt .

Một số mô hình điển hình :

• Reverse Proxy: đây là chức năng được sử dụng phổ biến khi triển khai WAF. Trong mô hình này, WAF giám sát tất cả các lưu lượng đi đến ứng dụng web, sau đó thay vì cho các địa chỉ IP bên ngoài gửi yêu cầu trực tiếp đến máy chủ web thì WAF đứng ra làm trung gian để gửi các yêu cầu này đến máy chủ web thay cho trình duyệt gốc rồi gửi trả lại kết quả cho các địa chỉ IP kia. Mô hình này có nhược điểm là tạo ra độ trễ khi kết nối từ trình duyệt đến ứng dụng web.

• Transparent Proxy: Ở mô hình này, WAF đứng giữa tường lửa mạng và máy chủ web và hoạt động tương tự ở mô hình Reverse Proxy nhưng không đứng ra làm trung gian kết nối như bên Reverse Proxy. Mô hình này không đòi hỏi phải thay đổi điều gì trong hạ tầng mạng nhưng có thể không cung cấp được một số dịch vụ như mô hình Reverse Proxy có thể.

• Layer 2 Brigde: Ở mô hình này, WAF đứng giữa tường lủa mạng và máy chủ web, nhưng hoạt động giống như một thiết bị Switch ở lớp .Mô hình này giúp mạng hoạt động với hiệu năng cao và mạng thay đổi không đáng kể, tuy nhiên nó lại không thể cung cấp các dịch vụ cao cấp khác mà các mô hình WAF khác có thể.

• Host/Server Based: Đây là các phần mềm được cài trực tiếp lên máy chủ web. Các loại Host based không cung cấp các tính năng tương tự như các loại WAF network base. Tuy nhiên mô hình này có thể khắc phục được vài điểm yếu mà các mô hình network base (các thiết bị WAF cứng) có. Tuy nhiên nó cũng làm tăng mức độ tải của máy chủ web

### Một số các cấu hình tường lửa ứng dụng web (WAF )

a. Backlisting

Danh sách đen là một trong những thuật toán lâu đời nhất trong bảo mật máy tính và nó được hầu hết các phần mềm chống vi-rútsử dụng để chặn các thực thể không mong muốn. Quá trình lập danh sách đen các ứng dụng liên quan đến việc tạo một danh sách chứa tất cả các ứng dụng hoặc tệp thực thi có thể gây ra mối đe dọa cho mạng, dưới hình thức tấn công phần mềm độc hại hoặc đơn giản bằng cách cản trở trạng thái năng suất của nó.Danh sách đen có thể được coi là một phương pháp lấy mối đe dọa làm trung tâm. Lợi ích rõ ràng của danh sách đen là sự đơn giản của nó. Quản trị viên có thể dễ dàng chỉ chặn phần mềm độc hại đã biết và chạy mọi thứ khác. Bằng cách này, người dùng sẽ có quyền truy cập vào tất cả các ứng dụng mà họ yêu cầu, giảm khối lượng quản trị viên tăng hoặc các ứng dụng thiết yếu bị chặn. Danh sách đen là một cách tiếp cận tốt cho các doanh nghiệp muốn thực hiện một cách tiếp cận thoải mái hơn để kiểm soát ứng dụng. Tuy nhiên đây chỉ đơn giản là chặn mọi thứ không đáng tin cậy. Khoảng 230.000 mẫu phần mềm độc hại được tạo ra hàng ngày, khiến quản trị viên không thể giữ một danh sách toàn diện và cập nhật cũng như danh sách các ứng dụng độc hại. Và xét rằng 30% phần mềm độc hại có xu hướng nhắm mục tiêu vào các lỗ hổng zero-day, có khả năng vi phạm bảo mật có thể xảy ra trước khi các ứng dụng bị ảnh hưởng được đưa vào danh sách đen.Thật không may, trong trường hợp tấn côngzero-day, các doanh nghiệp sẽ dễ bị tấn công bất kể hệ thống bảo mật mà họ có.Sự gia tăng gần đây trong các cuộc tấn công có chủ đích nhằm đánh cắp dữ liệubí mật từ các doanh nghiệp cũng là điều mà các quản trị viên cần phải lo lắng.

Việc dự đoán và ngăn chặn những kiểu tấn công này bằng cách sử dụng danh sách đen sẽ không hiệu quả.

b. Whitelisting

Cách tiếp cận danh sách trắng có nghĩa là WAF sẽ từ chối tất cả các yêu cầu theo mặc định và chỉ cho phép các yêu cầu được biết là đáng tin cậy. Danh sách trắng đối lập với danh sách đen, trong đó danh sách các thực thể đáng tin cậy như ứng dụng và trang web được tạo và được phép hoạt động độc quyền trong mạng. Danh sách trắng có nhiều cách tiếp cận lấy lòng tin hơn và được coi là an toàn hơn. Phương pháp kiểm soát ứng dụng này có thể dựa trên các chính sách như tên tệp, sản phẩm và nhà cung cấp hoặc nó có thể được áp dụng ở cấp độ thực thi, nơi chứng chỉ kỹ thuậtsố hoặc hàm băm mật mã của tệp thực thi được xác minh. Danh sách trắng chỉ cho phép một số ứng dụng chạy hạn chế, giúp giảm thiểu bề mặt tấn công một cách hiệu quả. Ngoài ra, việc xây dựng danh sách trắng dễ dàng hơn nhiều, vì số lượng ứng dụng đáng tin cậy chắc chắn sẽ thấp hơn khi so sánh với số lượngứng dụng không đáng tin cậy. Các doanh nghiệp tuân thủ các thông lệ tuân thủ quy định nghiêm ngặt có thể được hưởng lợi từ việc đưa vào danh sách trắng. Việc xây dựng một danh sách trắng có vẻ dễ dàng, nhưng một động thái vô tìnhcó thể dẫn đến các truy vấn của bộ phận trợ giúp chồng chất lên quản trị viên. Không thể truy cập các ứng dụng thiết yếu sẽ khiến các tác vụ quan trọng khác nhau bị tạm dừng.

c. Hybrid security: Mô hình bảo mật kết hợp sử dụng các yếu tố của cả blacklisting và whitelisting dựa trên nhu cầu cụ thể của ứng dụng. Đồng thời cho phép lưu lượng truy cập hợp pháp và chặn lưu lượng truy cập độc hại với thông tin xác thực sai tối thiểu. Nó có thể được sử dụng trên cả mạng nội bộ và mạng bên

Tóm lại cả 3 cấu hình hoạt động đều để phân tích các tương tác HTTP và giảm hoặc lý tưởng nhất là loại bỏ lưu lượng độc hại trước khi đến máy chủ để xử lý

### Ví dụ 1 số sản phẩn phổ biến cho tường lửa ứng dụng web (WAF )

ModSecurity: Đây là phần mềm nguồn mở có thể hoạt động như một module trong máy chủ Apache hoặc là một thành phần độc lập. ModSecurity sử dụng biểu thức chính quy trong việc bảo vệ máy chủ web từ các cuộc tấn công được xác định trước dựa theo các dấu hiệu hoặc các cuộc tấn công bất thường khác. Bên cạnh đó, ModSecurity cũng có khả năng lọc các siêu ký tự do người dùng chèn vào ứng dụng web.

URLScan: Đây là một sản phẩm của Microsoft dành riêng cho các máy chủ web IIS. URL scan không chỉ bảo vệ máy chủ IIS 6 khỏi các điểm yếu từ các phiên bản cũ hơn mà còn cung cấp thêm các biện pháp bảo vệ khác như lọc dữ liệu mã hóa trên URL hoặc lọc các siêu ký tự do người dùng chèn vào để chống lại các loại tấn công như XSS, SQL Injection…

WebKnight: Đây cũng là 1 sản phẩm dành cho các máy chủ web IIS.

## Một số loại tường lửa ứng dụng web (WAF)

### Tường lửa ứng dụng web host-based

Host-based WAFs (WAF dựa trên máy chủ) có thể được tích hợp hoàn toàn vào mã ứng dụng. Các lợi ích của việc triển khai WAF dựa trên máy chủ bao gồm chi phí thấp hơn và các tùy chọn tùy chỉnh tăng lên. WAF dựa trên máy chủ có thể là một thách thức để quản lý vì chúng yêu cầu thư viện ứng dụng và phụ thuộc vào tài nguyên máy chủ cục bộ để chạy hiệu quả. Do đó, có thể cần thêm nguồn nhân viên, bao gồm cả nguồn lực của lập trình viên, nhà phân tích hệ thống và DevOps/DevSecOps, có thể được yêu cầu.

### Tường lửa ứng dụng web netword-base.

Network-based WAFs (WAF dựa trên mạng) thường dựa trên phần cứng và có thể giảm độ trễ vì chúng được cài đặt cục bộ tại cơ sở thông qua một thiết bị chuyên dụng, càng gần ứng dụng càng tốt. Hầu hết các nhà cung cấp WAF dựa trên mạng lớn đều cho phép sao chép các quy tắc và cài đặt trên nhiều thiết bị, do đó có thể triển khai, cấu hình và quản lý quy mô lớn. Hạn chế lớn nhất đối với loại sản phẩm WAF này là chi phí – có chi phí vốn trả trước, cũng như chi phí vận hành liên tục để bảo trì.

### Tường lửa ứng dụng web cloud& hybrid

Cloud-hosted WAFs (WAF được lưu trữ trên đám mây) cung cấp giải pháp chi phí thấp cho các tổ chức muốn có sản phẩm chìa khóa trao tay yêu cầu tài nguyên tối thiểu để triển khai và quản lý. Cloud WAF dễ triển khai, có sẵn trên cơ sở đăng ký và thường chỉ yêu cầu hệ thống tên miền (DNS) đơn giản hoặc thay đổi proxy để chuyển hướng lưu lượng ứng dụng. Mặc dù có thể khó khăn khi đặt trách nhiệm lọc lưu lượng ứng dụng web của tổ chức với nhà cung cấp bên thứ ba, nhưng chiến lược này cho phép các ứng dụng được bảo vệ trên nhiều vị trí lưu trữ và sử dụng các chính sách tương tự để bảo vệ khỏi các cuộc tấn công lớp ứng dụng. Ngoài ra, các bên thứ ba này có thông tin tình báo về mối đe dọa mới nhất và có thể giúp xác định và chặn các mối đe dọa bảo mật ứng dụng mới nhất.

## Ưu nhược điểm của tường lửa ứng dụng web

1. Ưu điểm

* Tổng chi phí sở hữu (TCO) thấp: Là một giải pháp cloud tổng thể, WAF không yêu cầu phần cứng, phần mềm, chi phí vận hành và bảo trì, cũng như không cần rack space hoặc chi phí điện. Không cần kỹ sư WAF in-house ở phía khách hàng.
* Bảo mật toàn diện cho trang web: Ngăn chặn việc hack và các cuộc tấn công từ DDoS trước khi tiếp cận server web của bạn cũng như bảo vệ khỏi việc khai thác các lỗ hổng ứng dụng web và code bị lỗi.
* Chi phí băng thông (bandwidth) hiệu quả: Giải pháp tập trung vào việc giảm thiểu tấn công hơn là hấp thụ nhằm đảm bảo rằng băng thông chỉ được sử dụng bởi lưu lượng sạch chứ không phải bởi lưu lượng tấn công độc hại

1. Nhược điểm

* Bảo vệ không đầy đủ: Theo Ponemon, 65% người được hỏi nói rằng các cuộc tấn công đang vượt qua WAF. Tệ hơn nữa, chỉ có 9% người trả lời khảo sát chobiết WAF của họ chưa bao giờ bị vi phạm . Chúng tôi tin rằng điều này là do thiết kế kiến trúc của WAFs (phòng thủ theo chu vi) và vì chúng tập trung vào việc bảo vệ chống lại các cuộc tấn công dựa trên khai thác. Các trình điều khiểnnày dẫn đến phạm vi hiệu quả thấp đối với các rủi ro phổ biến như OWASP Top 10. Ngoài ra, WAFs không thể bảo vệ khỏi các lỗi thiết kế như Kiểm soát truy cập bị hỏng và Giả mạo tham số.
* Dương tính giả: WAF dựa trên các quy tắc và mô hình danh sách đen, và phải "học" ứng dụng trong quá trình cấu hình. Cách tiếp cận này không hiệu quả và kết quả là các WAF thường phân loại nhầm lưu lượng truy cập hợp pháp là độc hại (dương tính giả). Sai lầm này gây thiệt hại đáng kể cho trải nghiệm của người dùng. Trên thực tế, 43% người trả lời Ponemon chỉ ra rằng WAF của họ đang ở "Chế độ chỉ phát hiện / cảnh báo" , có nghĩa là các cuộc tấn công sẽ không bị chặn.
* Phạm vi hẹp: WAF không giúp các nhà phát triển xây dựng các ứng dụng an toàn . Dựa trên các câu trả lời cho cuộc khảo sát của Ponemon, có tới 34% người dùng WAF cho rằng tuân thủ là nhiệm vụ chính của WAF, trái ngược với việc giảm thiểu lỗ hổng bảo mật và / hoặc bảo vệ tấn công.
* Thiếu hỗ trợ đám mây: Những người trả lời Ponemon nói rằng hầu hết các WAF thường được triển khai dưới dạng phần cứng tại chỗ hoặc như một thiết bị được quản lý. Phong cách triển khai này không thích ứng tốt với các chiến lược triển khai hiện đại dựa trên nền tảng đám mây linh hoạt. WAF khóa các ứng dụng vào các mô hình cấu hình tĩnh.
* Đói tài nguyên: WAF tiêu tốn nhiều tài nguyên từ tổ chức vì chúng phức tạp để duy trì và giấy phép rất tốn kém. Trên thực tế, theo Ponemon, việc quản lý WAF yêu cầu số nhân viên trung bình là 2,5 FTE và ngân sách hàng năm vượt quá 400 nghìn đô la

## Kết luận chương 1

Trong chương 1 này, em đã giới thiệu tổng quan về tường lửa ứng dụng web, những lợi ích của WAF, tầm quan trọng của WAF, kiến trúc của tường lửa ứng dụng web, vị trí đặt WAF, mô hình bảo mật của WAF và mô hình hoạt động của tường lửa ứng dụng web cùng với một số cách cấu hình bảo mật của WAF, đưa ra một số loại của tường lửa ứng dụng web.

# kĩ thuật bypass ứng dụng web (WAF )

## Khai thác bộ xử lý mới

### Bỏ qua xác minh tham số

• PHP xóa khoảng trắng khỏi tên tham số hoặc biến đổi chúng thành dấu gạch dưới:



• ASP xóa ký tự % mà không được theo sau bởi hai chữ số thập phân:



• Một WAF không từ chối không xác định có thể được bypass.

### Những phương thức HTTP không đúng định dạng

• Máy chủ web được định dạnh cấu hình sai có thể chấp nhận HTTP không đúng định dạng:

Graphical user interface, application

Description automatically generated

• Một WAF chỉ có kiểm tra những yêu cầu GET và POST có thể bị bypass

### Làm quá tải tường lửa ứng dụng web (WAF)

* + Có thể định cấu hình WAF để bỏ qua xác thực đầu vào nếu tải hiệu suất quá cao
  + Thường áp dựng cho những WAF nhúng
  + Hacker gửi nhiều yêu cầu độc hại đến với khả năng làm quá tải WAF và để một số yêu cầu được thông qua .

## Impedance Mismatch

### HTTP parameter Pollution

HTTP parameter pollution (HPP) là một kỹ thuật tấn công web mà kẻ tấn công sẽ tạo ra các tham số trùng nhau trong HTTP request. Lợi dụng các phản ứng khác nhau của các công nghệ web khi xử lý các tham số trùng nhau này để inject những đoạn mã độc hại nhằm tấn công server và người sử dụng

Kĩ thuật đơn giản nên dễ kể tấn công có thể lợi dụng để:

* Sửa đổi các tham số HTTP
* Thay đổi các hành vi của ứng dụng web
* Truy cập và khai thác các biến không được kiểm soát chặt chẽ
* Bypass WAF và các cơ chế kiểm tra dữ liệu đầu vào

Do đó, nếu một ứng dụng web tồn tại lỗ hổng để thực hiện cuộc tấn công HPP, kẻ tấn công có thể dễ dàng chèn các đoạn mã độc hại để tấn công web server cũng như người sử dụng.

HPP có 2 loại :

Các máy chủ web khác nhau hoạt động theo những cách khác nhau khi chúng nhận được nhiều tham số có cùng tên do các cách phân tích tham số khác nhau. Các cuộc tấn công HPP phía máy khách hoặc phía máy chủ có thể xảy ra tùy thuộc vào cách xử lý các yêu cầu.

• Server -side (HPP từ phía mãy chủ )

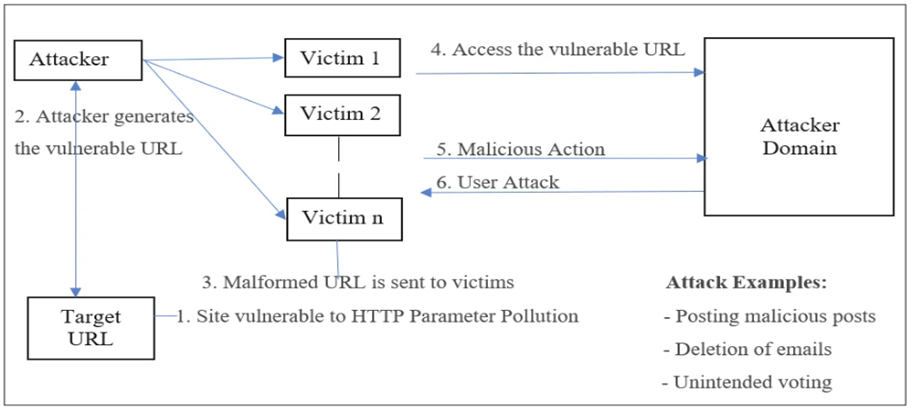
Mục tiêu của kẻ tấn công trong một cuộc tấn công phía máy khách là tấn công những người dùng khác. Trong phía máy chủ, kẻ tấn công có thể sử dụng ứng dụng web có nguy cơ theo nhiều cách. Họ có thể truy cập dữ liệu được bảo vệ hoặc thực hiện các hành động không được phép hoặc không được phép chạy ở phía máy chủ.

Server-side attacks có thể được thực hiện bằng cách đưa một tham số tương tự vào các giá trị hiện có hoặc bằng cách sử dụng các tham số không hiển thị cho người dùng cuối.

• Client-side (HPP phía máy khách )

Một cuộc tấn công HTTP Parameter Pollution phía máy khách có liên quan đến môi trường máy khách hoặc người dùng, nghĩa là hành động của người dùng bị ảnh hưởng và sẽ kích hoạt hành động độc hại hoặc ngoài ý muốn mà họ không biết. Nguy cơ này phát sinh khi một ứng dụng nhúng đầu vào của người dùng vào URL theo cách không an toàn. Kẻ tấn công có thể sử dụng phần mở này để tạo một URL, nếu được người dùng khác truy cập, sẽ sửa đổi các URL trong phản hồi bằng cách chèn thêm các tham số chuỗi truy vấn ghi đè lên các tham số hiện có. Điều này có thể dẫn đến các liên kết và biểu mẫu hoạt động theo cách chúng không được sử dụng. Ví dụ: có thể sửa đổi biểu mẫu lời mời bằng cách sử dụng HPP để lời mời được gửi tới người nhận khác.

Điều này có thể có tác động lớn hơn hoặc ít hơn tùy thuộc phần lớn vào những gì ứng dụng bị ảnh hưởng có thể thực hiện. Ngay cả với một ứng dụng không có tác động trực tiếp đến chính nó, kẻ tấn công có thể sử dụng nó cùng với các lỗ hổng khác để làm cho cuộc tấn công tổng thể trở nên tồi tệ hơn.



### Double URL Encoding

Kỹ thuật tấn công này bao gồm chuyển hóa các tham số yêu cầu của người dùng hai lần ở định dạng thập lục phân để vượt qua kiểm soát bảo mật hoặc gây ra hành vi không mong muốn từ ứng dụng. Điều đó có thể xảy ra vì máy chủ web chấp nhận và xử lý các yêu cầu của khách hàng ở nhiều dạng được chuyển hóa.

Bằng cách sử dụng mã hóa kép, có thể bỏ qua các bộ lọc bảo mật chỉ giải mã đầu vào của người dùng một lần. Quá trình giải mã thứ hai được thực thi bởi nền tảng phụ trợ hoặc các mô-đun xử lý dữ liệu được mã hóa đúng cách, nhưng không có sẵn các biện pháp kiểm tra bảo mật tương ứng.

• Thông thường WAF chuyển hóa các ký tự URL thành các chữ trong ASCII

• WAF có thể được cấu hình để giải mã những ký tự chỉ một lần

• Chuyển hóa URL hai lần một payload có thể dẫn đến bypass WAF



Ví dụ payload chứa chuyển hóa URL hai lần các ký tự



## Bỏ qua bộ quy tắc ứng dụng tường lửa web mới

### Brute force bằng cách liệt kê các payload khác nhau

Brute Force Attack là phương thức tấn công dò ra mật khẩu và tài khoản của người quản trị cao nhất. Hình thức tấn công này dễ phòng chống nhưng lại rất dễ bị dính nếu người quản trị chủ quan trong việc đặt mật khẩu và username. Thường thì sẽ dễ bị tấn công kiểu này khi:

• Đặt username là admin, administrator hoặc tương tự.

• Mật khẩu không an toàn, dễ đoán ra, sử dụng phổ biến.

• Không bảo mật đường dẫn đăng nhập.

• Không thay đổi mật khẩu thường xuyên.

• Như vậy, các vấn đề liên quan đến bảo mật tài khoản đăng nhập sẽ đều giúp cho hacker sử dụng brute force attack để tấn công.

### Dịch ngược bộ quy tắc ứng dụng tường lửa web (WAF)

Dịch ngược bộ quy tắc để xem bộ quy tắc đó hoạt động như thế nào, từ đó hiểu được quy tắc hoạt động của bộ quy tắc và sẽ thuận lợi hơn trong việc tìm cách bypass tường lửa ứng dụng web.

- Dịch ngược bộ quy tắc theo cách tiếp cận thử nghiệm và quan sát lỗi:

* Gửi các ký hiệu và từ khóa có thể hữu ích để tạo nên payload
* Quan sát xem những ký hiệu hay từ khóa nào bị chặn
* Cố gắng phát triển khai thác dựa trên kết quả của các bước trên

## Một số phương pháp tiếp cận để kiểm thử xâm nhập

### Tắt tường lửa ứng dụng web

a. Điều kiện

* + - Thông tin đăng nhập đã được lấy.
    - Chế độ WAF để bảo vệ tên miền là Enabled hoặc Bypassed

b. Thủ tục

* Đăng nhập vào bảng điều khiển quản lý.
* Nhấp vào ở góc trên bên trái của bảng điều khiển quản lý và chọn một khu vực hoặc dự án.
* Nhấp vào Danh sách dịch vụ ở đầu trang. Chọn Bảo mật > Tường lửa ứng dụng web . Trong ngăn điều hướng ở bên trái, chọn Miền
* Trong hàng chứa tên miền mong muốn, nhấp vào Chế độ chuyển đổi trong cột Hoạt động .
* Trong hộp thoại Switch Mode , chọn Enabled rồi nhấn OK .

### Thu thập thông tin

Thu thập thông tin là bước đầu tiên của quá trình kiểm thử, và đây là bước quan trọng có ảnh hướng to lớn đến kết quả kiểm thử. Nếu thông tin thu thập đầy đủ và chính xác sẽ giúp rút ngắn thời gian kiểm thử và mang lại kết nhiều kết quả chính xác. Sau khi thực hiện thu thập thông tin chúng ta sẽ biết chi tiết về mô hình mạng của mục tiêu, các phiên bản phần mềm, open port và hiểu được các hoạt động ở bước tiếp theo để đánh giá hệ thống.Ngoài ra chúng ta có thể xác định loại hệ thống nào đang đang có trong mạng như: Thông tin hệ điều hành, dịch vụ đang chạy trên chúng, từ đó sẽ giúp người kiểm thử lựa chọn các công cụ hợp lý để thực hiện kiểm thử.

Các loại thu thập thông tin:

1. Thu thập thông tin bị động:

Chúng ta sẽ cố gắng thu thập tất cả thông tin của mục tiêu mà không tác động trực tiếp đến hệ thống mục tiêu,các thông tin thu thập không nằm trên hệ thống mục tiêu mà chúng được cung cấp từ bên thứ 3. Các thông tin của đối tượng là những thông tin công khai, vì vậy rất nhiều phương thức search được thực thi. Các thông tin cần thu thập không liên quan trực tiếp đến hệ thống mạng của mục tiêu như: thông tin nhân viên, hoạt động kinh doanh.

• Web presence:

Bước này cung cấp một số lượng thông tin rất lớn về công ty của khách hàng bao gồm: thông tin nhân viên, vị trí địa lý, loại hệ thống (hệ điều hành, phiên bản) và kiến trúc mạng

1. Thu thập thông tin chủ động

Để có được các thông tin về đối tượng chúng ta phải kết nối trực tiếp đến mục tiêu. Ví dụ: thông tin về OS, port, range IP, … Với hình thức thu thập này chúng ta sẽ hiểu rõ hơn về phạm vi, loại hệ thống, số lượng hệ thống trong dự án. Từ đó sẽ có cái nhìn chi tiết, hiểu biết hơn về đối tượng kiểm thử.

* Công cụ chính : Port scanning
* Thông tin thu thập được khi foootprinting :

• Thông tin tổ chức: chi tiết nhận viên , địa chỉ , nền tảng tổ chức , công nghệ web …

• Thông tin mạng lưới : tên miền ( domain) , tiền miền phụ (subdomain) , khối mạng , địa chỉ IP của các hệ thồng có thể tiếp cận , whois , DNS…

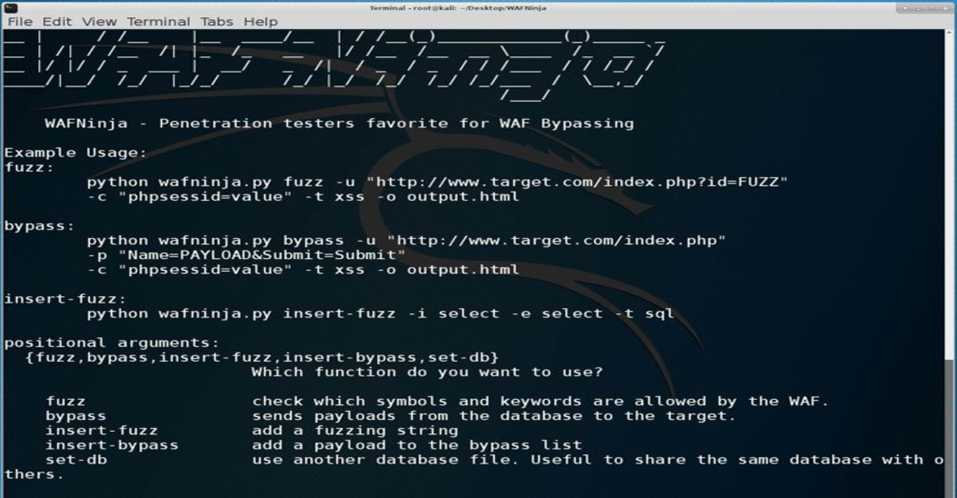
• Thông tin hệ thống : hệ điều hành và vị trí của web server ,user và password …

* Các nguồn có thể tìm kiếm thông tin:
* Sử dụng công cụ tìm kiếm
* Sử dụng dịch vụ web
* Sử dụng mạng xã hội
* Sử dụng thông tin trên website
* Sử dụng thông tin email
* Sử dụng thông tin cạnh tranh
* Sử dụng Whois
* Sử dụng DNS
* Sử dụng kỹ nghệ xã hội

### Một số lỗ hổng thường gặp

* Tấn công thông qua cookie của trình duyệt
* Tấn công qua lỗ hổng cross Site Scripting (XSS)
* Tấn công qua lỗ hổng Cross Site Request Forgery (CSRF)
* Tấn công qua lỗi nhúng mã (SQL) Injection
* Tấn công vào các dữ liệu đã được ẩn Lỗi tràn bộ đệm
* Sử dụng các ứng dụng giả mạo

## Giới thiệu về công cụ WAFNinja



### Định nghĩa công cụ WAFNinja

WAFNinja là một công cụ CLI được viết bằng Python. Nó sẽ giúp những người kiểm tra thâm nhập bỏ qua WAF bằng cách tự động hóa các bước cần thiết để bỏ qua xác thực đầu vào. Công cụ này được tạo ra với mục tiêu dễ dàng mở rộng, sử dụng đơn giản và có thể sử dụng được trong môi trường nhóm. Nhiều tải trọng và chuỗi làm mờ, được lưu trữ trong tệp cơ sở dữ liệu cục bộ được vận chuyển cùng với công cụ. WAFNinja hỗ trợ các kết nối HTTP, các yêu cầu GET và POST cũng như việc sử dụng Cookies để truy cập các trang bị hạn chế đối với người dùng đã xác thực. Ngoài ra, một proxy chặn có thể được thiết lập.

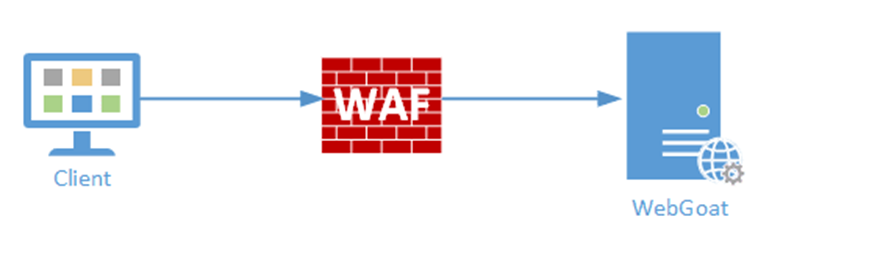
WAFNinja đã được trình bày bởi Khalil Bijjou tại OWASP Stammtisch Frankfurt 2015 và PHDays 2016.

### Công dụng của công cụ WAFNinja

Nhiều tải trọng và chuỗi làm mờ, được lưu trữ trong tệp cơ sở dữ liệu cục bộ được vận chuyển cùng với công cụ.

* WAFNinja hỗ trợ:
* Kết nối HTTP
* NHẬN yêu cầu
* Yêu cầu POST
* Sử dụng Cookie (đối với các trang có xác thực)
* Chặn proxy

### Triển khai



Sơ đồ thử nghiệm

* Công cụ này có các chế độ làm việc khác nhau:

**+ fuzz** cố gắng tìm các từ khóa và ký hiệu không bị WAF chặn

**+ bypass** gửi tải trọng từ cơ sở dữ liệu đến mục tiêu

Ngoài ra insert-fuzz, insert-bypassbạn có thể chỉnh sửa chuỗi mờ hoặc thêm tải trọng vào danh sách bỏ qua.

## Kết luận chương 2

Trong chương 2 này em đã đưa ra một số giải pháp bypass tường lửa ứng dụng web với ba phương pháp chính như là: khai thác bộ xử lý mới: làm WAF bỏ qua xác thực đầu vào, Impedance Mismatch: WAF diễn giải đầu vào khác với backend, Bypass bộ quy tắc WAF: sử dụng những payload không bị phát hiện bởi WAF. Và có một số phương pháp để tiếp cận kiểm thử thâm nhập, giới thiệu công cụ WAFNinja cùng với những công dụng của WAFNinja.

# Triển khai bypass tường lửa ứng dụng web

## Triển khai cầu hình tường lửa ModSecurity .

ModSecurity là phần mềm mã nguồn mở có thể hoạt động như một module trong máy chủ Apache hoặc là một thành phần độc lập. ModSecurity sử dụng biểu thức chính quy trong việc bảo vệ máy chủ web từ các cuộc tấn công được xác định trước dựa theo các dấu hiệu hoặc các cuộc tấn công bất thường khác. Bên cạnh đó, ModSecurity cũng có khả năng lọc các siêu ký tự do người dùng chèn vào ứng dụng web.

### Chức năng của ModSecurity.

Mod Security đứng trước Web Server, làm nhiệm vụ như một firewall để kiểm soát truy cập vào ra Web Server. Các thông tin đi từ bên ngoài vào và bên trong ra sẽ được kiểm soát chặt chẽ để tránh những thông tin có thể gây hại cho Web Server hay là việc rò rỉ các thông tin đặc biệt từ Web Server đến Client.

Các chức năng cụ thể :

- Parsing: Phân tách các dữ liệu luân chuyển qua hệ thống thành cấu trúc dữ liệu mà ModSecurity định nghĩa sẵn. Cấu trúc này sẽ được chuyển qua cơ chế so trùng mẫu trong tập rule để phân tích nguy cơ

- Buffering (đệm ): Khi các request gửi đến ứng dụng web thì phải thông qua Mod Security trước khi đến ứng dụng xử lý và những response cũng sẽ được phân tích trước khi trả về phía client. Cơ chế này là cách duy nhất để có thể ngăn chặn các cuộc tấn công thời gian thực, các dữ liệu mà Mod Security nhận được và phân tích sẽ được lưu trữ trong RAM (bao gồm request body và response data)

- Logging: Mod Security hỗ trợ ghi nhật ký các gói tin HTTP: request headers, request body, response header, response body nhằm hỗ trợ người quản trị phân tích nguy cơ mà hệ thống đang gặp phải để có thể ra quyết định kiểm soát

- Rule Engine: Các tập mẫu trong Mod Security đóng vai trò quan trọng trong việc phát hiện các dạng tấn công và thực hiện phòng chống. Mod Security cùng phát triển với dự án OWASP phát triển các mẫu để phân tích và phòng chống các tấn công hệ thống web

### Quy trình xử lý trong ModSecurity

Trong ModSecurity mỗi phiên phân tích sẽ thực hiện lần lượt qua 5 bước (pha), tại mỗi bước ModSecurity sẽ thực thi các rule tương ứng nhằm phát hiện và phòng chống các khai thác.

1. Phase Request Header

Rule được đặt tại đây sẽ được thực hiện ngay sau khi Web Server đọc request header, lúc này phần request body vẫn chưa được đọc. Đây là bước đầu tiên trong quá trình thực hiện phân tích gói tin. Mục đích của bước này nhằm cho phép người viết rule tương tác với các request trước khi thực hiện các yêu cầu trong phần HTTP body. Phần này khá quan trọng để phân tích các khai thác dựa vào HTTP method cũng như dựa vào URL như SQL Injection, Reflect XSS, Local file include

1. Phase Request body

Đây là thời điểm các thông tin chức năng chung đưa vào được phân tích và xem xét, các rule mang tính application-oriented thường được đặt ở đây. Bước này là quá trình kiểm tra chính trong quá trình client gởi request đến server, phần này sẽ có hiệu quả khi người dùng cố sử dụng phương thức POST hoặc PUT để upload tập tin lên phía server. Việc kiểm tra này bảo đảm dữ liệu đưa lên server là an toàn, tránh tình trạng upload mã độc hoặc các dạng tấn công như Stored XSS, Ajax Injection... ModSecurity hỗ trợ ba loại mã hóa request

body :

+ Application/x-www-form-urlencoded dùng để truyền form dữ liệu.

+Multipart/form-data dùng để truyền file.

+ Text/xml : dùng để phân tích dữ liệu XML.

1. Phase Response headers

Những request đã được xử lý tại server sẽ được trả về cho ModSecurity kiểm tra trạng thái trong phần respone header. Trước khi phần respone body được đọc thì ModSecurity sẽ dựa vào tập rule để xác định có cần kiểm tra nội dung dữ liệu trong phần body hay không.

Ví dụ: mã trạng thái trả về là 404 (Not found) thì lúc này sẽ không cần kiểm tra nội dung gói tin trả về.

1. Phase Response body

Sau khi ModSecurity đã hoàn thành việc kiểm tra tại respone header thì nội dung trong phần body sẽ được kiểm tra so trùng với mẫu trong tập lệnh.Việc này là khá hiệu quả để phát hiện và phòng chống xâm nhập trong trường hợp bước 1 và 2 không phát hiện đượctấn công.

Ví dụ: trong khai thác SQL injection, nếu hacker cố gắng sử dụng một số công nghệ evasion thì việc phát hiện khi request là khó khăn. Khi khai thác thành công, ModSecurity sẽ phân tích kết quả trong gói tin trả về để phát hiện nếu như câu truy vấn thành công.

1. Phase Logging

Đây là thời điểm các hoạt động log được thực hiện. Các rules đặt ở đây sẽ định rõ việc log sẽ như thế nào, nó sẽ kiểm tra các error message log của Web Server. Đây cũng là thời điểm cuối cùng để chặn các connection không mong muốn, kiểm tra các response header không thể kiểm tra ở phase response header và phase response body

### Cấu trúc tập luật trong ModSecurity

SecRule được sử dụng để tạo các rule cho ModSecurity, cú pháp như sau:

SecRule VARIABLES OPERATOR [ACTIONS]

• Variables : xác định vị trí dữ liệu mà modSecurity sẽ tìm kiếm mẫu

REMOTE\_ADDR: Địa chỉ IP của client.

REMOTE\_HOST: Hostname của client (nếu tồn tại) .

REMOTE\_USER: Authenticated username (nếu tồn tại) .

REMOTE\_IDENT: Remote Username (lấy từ inetd, ít dùng) .

REQUEST\_METHOD: Request Method (GET, HEAD, POST..) .

SCRIPT\_FILENAME: Đường dẫn đầy đủ của script được thực thi.

PATH\_INFO: Phần mở rộng của URI phía sau tên của một script.

Ví dụ: /archive.php/5 thì PATH\_INFO là /5

QUERY\_STRING: URI phía sau dấu ?.

Ví dụ /index.php?i=1 thì QUERY\_STRING là i=1.

AUTH\_TYPE: Basic hoặc Digest Authentication.

DOCUMENT\_ROOT: Đường dẫn đến documentroot.

SERVER\_ADMIN: Email của Server Administrator.

SERVER\_NAME: Hostname của Server.

SERVER\_ADDR: Địa chỉ IP của Server.

SERVER\_PORT: Server port.

SERVER\_PROTOCOL: Protocol, (ví dụ HTTP/1.1).

SERVER\_SOFTWARE: Apache version.

TIME\_YEAR: Năm hiện tại.

TIME\_MON: Tháng hiện tại.

TIME\_DAY: Ngày.

TIME\_HOUR: Giờ.

TIME\_MIN: Phút.

TIME\_SEC: Giây.

TIME\_WDAY: Thứ tự ngày trong tuần (ví dụ 4 - Thursday).

TIME: Thời điểm hiện tại được viết theo cấu trúc : YmdHMS.

ví dụ: 20200328144530 : 28/03/2020 14h 45' 30''.

API\_VERSION

THE\_REQUEST: Dòng đầu tiên của request. vd: GET / HTTP/1.1.

REQUEST\_URI: Request URI.

FILENAME: Tên file được yêu cầu đến.

IS\_SUBREQ

• Operator : - Các toán tử kiểm tra trong ModSecurity có nhiệm vụ phân tích các biến đầu vào Variables để ra quyết định. Hầu hết các rule sẽ sử dụng các regular expression cho việc phân tích, nhưng trong một số trường hợp cụ thể thì các phân nhóm toán tử khác sẽ hữu ích hơn. Ta xét trường hợp cần so sánh các giá trị là số (numberic) thì việc sử dụng Regular expression là khá bất lợi cho việc tạo rule và tài nguyên khi thực thi so sánh rule. ModSecurity hỗ trợ một nhóm phương thức so sánh khác nhau nhằm tăng hiệu năng cho phần kiểm tra. Trong trường hợp này thì việc sử dụng các toán tử về số học sẽ hiệu quả hơn nhiều so với regular expression. ModSecurity hỗ trợ 4 nhóm:

* String-matching operators
* Numberical operators
* Validation operators
* Miscellancous operators

• Action : Các hành vi (action) là điểm mạnh của ModSecurity cho phép hệ thống web có khả năng miễn dịch với một số loại khai thác đã biết đến. Các action là thành phần cuối cùng trong một rule, Apache sẽ quyết định kết quả trả về phía người dùng (thông báo lỗi, hủy kết nối hoặc cho phép truy cập...)

ModSecurity chia các action thành 7 phân mục:

* Disruptive actions
* Flow actions
* Metadata actions
* Metadata actions
* Variable actions
* Logging actions
* Special actions
* Miscellaneous actions

### Chị tiết cấu hình và cài đặt hệ thống ModSecurity

1. Cài đặt hệ thống ModSecurity

Bước 1 :ModSecurity có thể được cài đặt bằng cách chạy lệnh sau trong thiết bị đầu cuối của bạn:



Bước 2: Sau khi cài đặt ModSecurity, kích hoạt headers mô-đun Apache 2 bằng cách chạy lệnh sau:

|  |
| --- |
| sudo a2enmod headers |

Bước 3: Sau khi cài đặt ModSecurity và bật mô-đun tiêu đề, bạn cần khởi động lại dịch vụ apache2, điều này có thể được thực hiện bằng cách chạy lệnh sau:

|  |
| --- |
| sudo systemctl restart apache2 |

1. Cấu hình hệ thống Modsecurity:

• Chuẩn bị tệp cấu hình Modsecurity

• Xóa .recommendedphần mở rộng khỏi tên tệp cấu hình ModSecurity bằng lệnh sau:

|  |
| --- |
| sudo cp /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf |

• Với trình soạn thảo văn bản như vim, hãy mở /etc/modsecurity/modsecurity.confvà thay đổi giá trị cho SecRuleEngine thành On:



|  |
| --- |
| # -- Rule engine initialization ----------------------------------------------  # Enable ModSecurity, attaching it to every transaction. Use detection  # only to start with, because that minimises the chances of post-installation  # disruption.  #  SecRuleEngine On  .... |

• Khởi động lại Apache để áp dụng các thay đổi:

|  |
| --- |
| sudo systemctl restart apache2 |

1. Thiết lập bộ quy tắc OWASP MdSecurity

Bộ quy tắc cốt lõi ModSecurity OWASP (CRS) là một bộ quy tắc phát hiện tấn công chung để sử dụng với ModSecurity hoặc tường lửa ứng dụng web tương thích.. CRS cung cấp khả năng bảo vệ chống lại nhiều loại tấn công phổ biến, bao gồm SQL Injection, Cross Site Scripting và Local File Inclusion.

Quy trình thiết lập :

Bước 1 : xóa toàn bộ quy tắc có sẵn bằng cách chạy lệnh sau :

|  |
| --- |
| sudo rm -rf /usr/share/modsecurity-crs |

Bước 2:cài đặt GIT

|  |
| --- |
| Sudo apt install git |

Bước 3: Sao chép kho lưu trữ OWASP-CRS GitHub vào thư mục /usr/share/modsecurity-crs:

|  |
| --- |
| sudo git clone https://github.com/coreruleset/coreruleset /usr/share/modsecurity-crs |

Bước 4: Đổi tên crs-setup.conf.examplethành crs-setup.conf:

|  |
| --- |
| sudo mv /usr/share/modsecurity-crs/crs-setup.conf.example /usr/share/modsecurity-crs/crs-setup.conf |

Bước 5 : Đổi tên tệp quy tắc loại trừ yêu cầu mặc định:

|  |
| --- |
| sudo mv /usr/share/modsecurity-crs/rules/REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf.example /usr/share/modsecurity-crs/rules/REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf |

1. Kiểm tra ModSecurity

Kiểm tra ModSecurity bằng cách thực hiện một cuộc tấn công bao gồm tệp cục bộ đơn giản bằng cách chạy lệnh sau:

|  |
| --- |
| curl http://<SERVER-IP/DOMAIN>/index.php?exec=/bin/bash |

Nếu ModSecurity đã được cấu hình đúng và đang tích cực ngăn chặn các cuộc tấn công, lỗi sau sẽ được trả về:

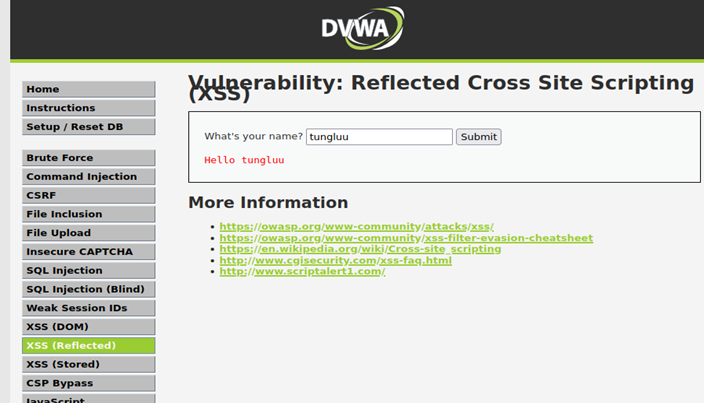


## Triển khai thực nghiệm bypass tường lửa ứng dụng web (WAF)

### Kịch bản triển khai

Triển khai ứng dụng web DVWA có chứa nhiều lỗ hổng bảo mật và sẽ triển khai khai thác lỗ hổng reflected XSS, cài đặt tường lửa modsecurity với core rule set ver.2.2.9. Sử dụng công cụ WAFNinja đã giới thiệu ở chương 2 để phân tích cách response của tất cả các request và brute forcing tường lửa bằng cách liệt kê những payload, gửi những payload này tới website để tìm những function có thể bypass được.

### Triển khai thức nghiệm

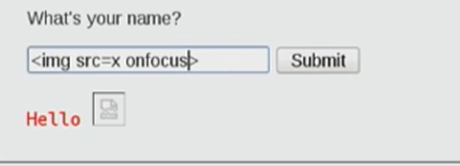


Trên màn hình ta thấy có một trường input, khi nhập vào tên của mình và submit thì website sẽ reflect Hello + input.



Do đã cài tường lửa lên khi submit một payload XSS để alert một box thì đã bị chặn và thông báo Forbidden. Thử một vài payload khác ví dụ như:





Khi tham chiếu <img src=x, điều này gây ra lỗi vì ứng dụng không thể tìm thấy hình ảnh x. Điều này được thực hiện có chủ ý vì để sử dụng các function xử lý sự kiện của javascript. Nhưng tất cả các payload này đều bị tường lửa chặn, vậy ta cần tìm một function có thể bypass qua tường lửa nhưng trong javascript có hơn 150 functions, để quá trình liệt kê các payload nhanh chóng thì em sẽ sử dụng công cụ WAFNinja.

Text

Description automatically generated Sử dụng fuzz function có các tham số khác nhau như hình trên.

Text

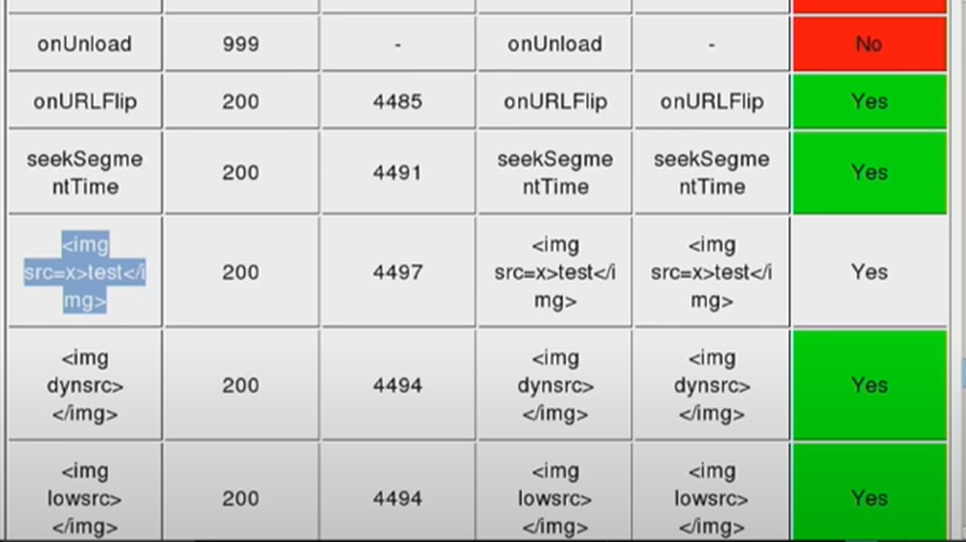
Description automatically generated

Trong câu lệnh trên -u là url mục tiêu, FUZZ là nơi thử các input khác nhau sau đó ta cần có cookie -c, -t là kiểu tấn công, ở đây chọn kiểu tấn công XSS, và kết quả sẽ được xuất ra file output.html. Sau khi gửi xong có thể thấy có 157 Fuzz đã gửi và output đã lưu tới output.html.

A screenshot of a computer

Description automatically generated with medium confidence

Mở file output.html ra có thể thấy được các cột như Fuzz là thử các input, HTTP Status và cuối cùng là cột xem chuỗi có được chấp nhận hay không sẽ được thể hiện là yes hoặc no.



Có thể thấy thẻ img có hoạt động, thẻ img này không bị chặn bởi trang web, WAF chấp nhận nó vậy thì bây giờ chúng ta chỉ cần tìm một function javascirpt có thể được thực thi trong payload



Function onwheel không bị chặn

Graphical user interface, text, application

Description automatically generated

Vậy em sẽ thử với payload: <img src=x onwheel=alert(1)> và submit

Graphical user interface

Description automatically generated

Sau khi submit và ấn vào biểu tượng cạnh hello thì sẽ hiện lên popup 1. Vậy là đã bypass được WAF

## Đánh giá

Công cụ hỗ trợ WAFNinja rất hữu ích trong việc có thể tìm một số phương pháp bypass waf, có thể liệt kê các payload và gửi chúng tới mục tiêu và phân tích response của tất cả các request, kết quả hiển thị rất rõ ràng.

## Kết luận chương 3

Trong chương 3 này chúng em đã xây dựng và cấu hình tường lửa modsecurity với bộ quy tắc sử dụng là core rule set ver.2.2.9. Triển khai tường lửa bảo vệ web DVWA và sử dụng công cụ WAFNinja hỗ trợ triển khai tìm các cách để có thể bypass được tường lửa.

Kết luận

Ba chương của đồ án đã thể hiện được rằng những mục tiêu đặt ra khi thực hiện đồ án đều đã đạt được. Cụ thể:

Chương 1 đã hệ thống lại những kiến thức tổng quan về tường lửa ứng dụng web (WAF). Chương này cũng đã chỉ ra lợi ích và tầm quan trọng của tường lửa ứng dụng web trong bảo mật .

Chương 2 lần lượt xem này em đã đưa ra một số giải pháp bypass tường lửa ứng dụng web với ba phương pháp chính như là: khai thác bộ xử lý mới: làm WAF bỏ qua xác thực đầu vào, Impedance Mismatch: WAF diễn giải đầu vào khác với backend, Bypass bộ quy tắc WAF: sử dụng những payload không bị phát hiện bởi WAF. Và có một số phương pháp để tiếp cận kiểm thử thâm nhập, giới thiệu công cụ WAFNinja cùng với những công dụng của WAFNinja.

Trong chương 3 đã xây dựng và cấu hình tường lửa ModSecurity với bộ quy tắc sử dụng là core rule set ver.2.2.9. . Triển khai tường lửa bảo vệ web DVWA và sử dụng công cụ WAFNinja hỗ trợ triển khai tìm các cách để có thể bypass được tường lửa.

Tài liệu tham khảo

[1] https://www.linuxbabe.com/security/modsecurity-apache-debian-ubuntu

[2] https://iratoon.medium.com/bypassing-web-application-firewall-part-5-108eea42b200

[3] https://devdays.lt/2019/wp-content/uploads/2019/06/2019-05 14\_WAF\_Bypassing\_-\_Khalil\_Bijjou.pdf

[4] https://www.vietsunshine.com.vn/2020/09/14/waf/

[5] https://owasp.org/www-pdf-archive/OWASP\_Stammtisch\_Frankfurt\_-\_Web\_Application\_Firewall\_Bypassing\_-\_how\_to\_defeat\_the\_blue\_team\_-\_2015.10.29.pdf

[6] https://www.linkedin.com/pulse/how-we-bypassed-libmodsecurity-aka-modsecurity-soufiane-tahiri

[7] https://www.youtube.com/watch?v=iQqwQXHwQk0&t=340s