**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**

****

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN: AN TOÀN ỨNG DỤNG WEB VÀ CƠ SỞ DỮ LIỆU**

**MÃ HỌC PHẦN: INT14105**

**ĐỀ TÀI: TÌM HIỂU VỀ TƯỜNG LỬA ỨNG DỤNG WEB**

**Các sinh viên thực hiện:**

**Phạm Hải Dương :B21DCAT072**

**Trần Văn Chính :B21DCAT048**

**Nguyễn Đức Anh :B21DCAT032**

**Tên nhóm: 13**

**Giảng viên hướng dẫn:** **ThS Vũ Minh Mạnh**

**HÀ NỘI 2024**

**PHÂN CÔNG NHIỆM VỤ NHÓM THỰC HIỆN**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Công việc / Nhiệm vụ** | **SV thực hiện** | **Thời hạn  hoàn thành** |
| 1 | Tìm hiểu tổng quan về WAF, triển khai WAF, so sánh WAF với các hệ thống bảo mật khác, demo cài đặt hệ thống IDS hỗ trợ Mod Security phát hiện các cuộc tấn công vào ứng dụng web, viết báo cáo | Phạm Hải Dương | 15/11/2024 |
| 2 | Tìm hiểu về Mod Security, cài đặt Mod Security ngăn chặn tấn công SQLi, XSS và thực hiện tấn công vượt qua Mod Security | Trần Văn Chính | 15/11/2024 |
| 3 | Tìm hiểu về WAF, các dịch vụ WAF hiện nay (Akamai, Azure, …), cài đặt môi trường demo tấn công vào ứng dụng web, làm slide | Nguyễn Đức Anh | 15/11/2024 |

**NHÓM THỰC HIỆN TỰ ĐÁNH GIÁ**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TT** | **SV thực hiện** | **Thái độ tham gia** | **Mức hoàn thành CV** | **Kỹ năng giao tiếp** | **Kỹ năng hợp tác** | **Kỹ năng lãnh đạo** |
| 1 | Phạm Hải Dương | 5 | 5 | 5 | 5 | 5 |
| 2 | Trần Văn Chính | 5 | 5 | 5 | 5 | 5 |
| 3 | Nguyễn Đức Anh | 5 | 5 | 5 | 5 | 5 |
| 4 |  |  |  |  |  |  |

MỤC LỤC

[**MỤC LỤC 3**](#_Toc183165704)

[**DANH MỤC CÁC HÌNH VẼ 5**](#_Toc183165705)

[**DANH MỤC CÁC BẢNG BIỂU 6**](#_Toc183165706)

[**DANH MỤC CÁC TỪ VIẾT TẮT 7**](#_Toc183165707)

[**MỞ ĐẦU 8**](#_Toc183165708)

[**CHƯƠNG 1. Tổng quan về tường lửa ứng dụng web 9**](#_Toc183165709)

[**1.1 Khái niệm 9**](#_Toc183165710)

[**1.2 Kiến trúc và vai trò của tường lửa ứng dụng web 9**](#_Toc183165711)

[**1.2.1 Kiến trúc của tường lửa ứng dụng web 9**](#_Toc183165712)

[**1.2.2 Vai trò của tường lửa ứng dụng web 10**](#_Toc183165713)

[**1.3 Cách thức hoạt động của tường lửa ứng dụng web 12**](#_Toc183165714)

[**1.4 Phân loại tường lửa ứng dụng web 12**](#_Toc183165715)

[**1.4.1 Phân loại theo mô hình bảo mật 12**](#_Toc183165716)

[**1.4.2 Phân loại theo mô hình triển khai 13**](#_Toc183165717)

[**1.5 Kết chương 14**](#_Toc183165718)

[**CHƯƠNG 2. Cách triển khai và các vấn đề của tường lửa ứng dụng web 15**](#_Toc183165719)

[**2.1 Triển khai tường lửa ứng dụng web 15**](#_Toc183165720)

[**2.2 Các trường hợp sử dụng tường lửa ứng dụng web 15**](#_Toc183165721)

[**2.2.1 Bảo vệ Website và ứng dụng 15**](#_Toc183165722)

[**2.2.2 Tuân thủ các tiêu chuẩn bảo mật và quy định 15**](#_Toc183165723)

[**2.2.3 Kiểm soát Bot và Ngăn chặn Tấn công DDoS 15**](#_Toc183165724)

[**2.2.4 Khắc phục các Lỗ hổng Bảo mật 16**](#_Toc183165725)

[**2.2.5 Phát hiện Xâm nhập trong Thời gian Thực 16**](#_Toc183165726)

[**2.2.6 Thiết lập Chính sách 16**](#_Toc183165727)

[**2.3 Các vấn đề của tường lửa ứng dụng web 16**](#_Toc183165728)

[**2.3.1 Cấu hình 16**](#_Toc183165729)

[**2.3.2 Dễ bị bypass 16**](#_Toc183165730)

[**2.3.3 Chi phí và hiệu suất 16**](#_Toc183165731)

[**2.4 Một số nền tảng dịch vụ cung cấp tường lửa ứng dụng web 16**](#_Toc183165732)

[**2.4.1 Azure WAF 16**](#_Toc183165733)

[**2.4.2 AWS WAF 17**](#_Toc183165734)

[**2.4.3 Akamai WAF 17**](#_Toc183165735)

[**2.4.4 Cloudflare WAF 17**](#_Toc183165736)

[**2.5 Kết chương 17**](#_Toc183165737)

[**CHƯƠNG 3. SO SÁNH TƯỜNG LỬA ỨNG DỤNG WEB VỚI CÁC CÔNG CỤ BẢO MẬT ỨNG DỤNG WEB KHÁC 18**](#_Toc183165738)

[**3.1 Tường lửa ứng dụng web với tường lửa truyền thống 18**](#_Toc183165739)

[**3.2 Tường lửa ứng dụng web với tường lửa thế hệ mới 18**](#_Toc183165740)

[**3.3 Tường lửa ứng dụng web với hệ thống ngăn chặn xâm nhập 18**](#_Toc183165741)

[**3.4 Kết chương 19**](#_Toc183165742)

[**CHƯƠNG 4. Thực Nghiệm 20**](#_Toc183165743)

[**4.1 Tìm hiểu về Mod Security 20**](#_Toc183165744)

[**4.2 Cài đặt môi trường 22**](#_Toc183165745)

[**4.3 Thử nghiệm tấn công SQL Injection và XSS khi chưa tích hợp Mod Security 23**](#_Toc183165746)

[**4.3.1 Thử nghiệm SQLi 23**](#_Toc183165747)

[**4.3.2 Thử nghiệm XSS 23**](#_Toc183165748)

[**4.4 Triển khai Mod Security 27**](#_Toc183165749)

[**4.4.1 Cấu hình Mod Security với CRS của OWASP 27**](#_Toc183165750)

[**4.4.2 Thực hiện tấn công lại vào trang web 29**](#_Toc183165751)

[**4.5 Thử nghiệm bypass qua Mod Security 31**](#_Toc183165752)

[**4.6 Thử nghiệm cài đặt hệ thống IDS hỗ trợ Mod Security phát hiện tấn công SQLi 34**](#_Toc183165753)

[**4.6.1 Cài đặt và cấu hình IDS (Suricata) 34**](#_Toc183165754)

[**4.6.2 Thử nghiệm 35**](#_Toc183165755)

[**4.7 Kết luận 37**](#_Toc183165756)

[**KẾT LUẬN 38**](#_Toc183165757)

[**TÀI LIỆU THAM KHẢO 39**](#_Toc183165758)

DANH MỤC CÁC HÌNH VẼ

[Hình 1- Tổng quan về tường lửa ứng dụng web 9](#_Toc183165759)

[Hình 2- Vai trò của tường lửa ứng dụng web 11](#_Toc183165760)

[Hình 3- Vai trò của tường lửa ứng dụng web 12](#_Toc183165761)

[Hình 4- quá trình xử lý của Mod Security 21](#_Toc183165762)

[Hình 5- Giao diện DVWA 22](#_Toc183165763)

[Hình 6: thử nghiệm SQLi khi chưa triển khai Mod Security 23](#_Toc183165764)

[Hình 7- thử nghiệm XSS Reflected khi chưa triển khai Mod Security 24](#_Toc183165765)

[Hình 8- Giao diện XSS DOM 24](#_Toc183165766)

[Hình 9- thử nghiệm XSS DOM khi chưa triển khai Mod Security 25](#_Toc183165767)

[Hình 10- thử nghiệm XSS DOM khi chưa triển khai Mod Security 25](#_Toc183165768)

[Hình 11- thử nghiệm XSS Store khi chưa triển khai Mod Security 26](file:///D:\hocdeee\4.1\atw&csdl\BTL-nhóm%2013.docx#_Toc183165769)

[Hình 12- thử nghiệm XSS Store khi chưa triển khai Mod Security 26](#_Toc183165770)

[Hình 13- tải về Mod Security 27](#_Toc183165771)

[Hình 14- Cài đặt coreruleset 27](#_Toc183165772)

[Hình 15- Cài đặt Mod Security Config 28](#_Toc183165773)

[Hình 16- Thử nghiệm tấn công SQLi khi đã cài đặt Mod Security 29](#_Toc183165774)

[Hình 17- Thử nghiệm tấn công XSS Reflected khi đã cài đặt Mod Security 29](#_Toc183165775)

[Hình 18- Thử nghiệm tấn công XSS DOM khi đã cài đặt Mod Security 30](#_Toc183165776)

[Hình 19- Thử nghiệm tấn công XSS Store khi đã cài đặt Mod Security 30](#_Toc183165777)

[Hình 20- Kiểm tra log của Mod Security 31](#_Toc183165778)

[Hình 21- Tấn công SQLi bypass qua Mod Security 32](#_Toc183165779)

[Hình 22- Tấn công SQLi bypass qua Mod Security 32](#_Toc183165780)

[Hình 23- Tấn công XSS bypass qua Mod Security 33](#_Toc183165781)

[Hình 24- Tấn công XSS bypass qua Mod Security 34](#_Toc183165782)

[Hình 25- Cấu hình dải mảng trong Suricata 34](#_Toc183165783)

[Hình 26- Cấu hình luật Suricata 35](#_Toc183165784)

[Hình 27- Thêm/sửa các luật Suricata 35](#_Toc183165785)

[Hình 28- Kiểm tra cấu hình Suricata 35](#_Toc183165786)

[Hình 29- Khởi động Suricata 36](#_Toc183165787)

[Hình 30- Sử dụng Suricata đẻ giám sát 36](#_Toc183165788)

[Hình 31- Kiểm tra log Suricata để phát hiện tấn công SQLi 36](#_Toc183165789)

DANH MỤC CÁC BẢNG BIỂU

DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Từ  viết tắt** | **Thuật ngữ tiếng Anh/Giải thích** | **Thuật ngữ tiếng Việt/Giải thích** |
| WAF | Web Application Fire Wall | Tường lửa ứng dụng web |
| HTTP | HyperText Transfer Protocal | là một giao thức mạng được sử dụng để truyền tải dữ liệu giữa các máy khách (client) và máy chủ (server) trên mạng internet. |
| GDPR | General Data Protection Regulation | quy định bảo vệ dữ liệu chung của Liên minh Châu Âu (EU), có hiệu lực từ ngày 25 tháng 5 năm 2018, nhằm bảo vệ quyền riêng tư và dữ liệu cá nhân của người dùng trong EU. |
| HPIAA | Health Insurance Portability and Accountability Act | là một luật của Hoa Kỳ, được ban hành năm 1996, nhằm bảo vệ dữ liệu sức khỏe cá nhân (Protected Health Information - PHI). |
| PCI DSS | Payment Card Industry Data Security Standard | là một bộ tiêu chuẩn bảo mật dữ liệu do các tổ chức thẻ tín dụng lớn (Visa, Mastercard, American Express, Discover, JCB) đưa ra, nhằm bảo vệ dữ liệu thanh toán. |
| DDOS | Distributed Denial of Service | một hình thức tấn công mạng nhằm làm gián đoạn hoặc ngăn chặn việc truy cập hợp pháp vào một hệ thống, dịch vụ, hoặc mạng. |
| … |  |  |

MỞ ĐẦU

Trong bối cảnh các cuộc tấn công mạng ngày càng tinh vi và đa dạng, việc bảo vệ ứng dụng web trở thành một yêu cầu thiết yếu đối với mọi tổ chức. Tường lửa ứng dụng web (Web Application Firewall - WAF) ra đời như một giải pháp bảo mật hiệu quả, giúp ngăn chặn các lỗ hổng phổ biến như SQL Injection, Cross-Site Scripting (XSS), và các mối đe dọa trong danh sách OWASP Top 10. Không chỉ dừng lại ở việc phân tích và lọc lưu lượng HTTP/S, WAF còn cung cấp khả năng tùy chỉnh linh hoạt để đáp ứng nhu cầu bảo mật riêng biệt của từng ứng dụng. Nghiên cứu về WAF không chỉ giúp hiểu rõ hơn về cách hoạt động và triển khai công nghệ này, mà còn cung cấp cơ sở để xây dựng các giải pháp bảo mật toàn diện, đảm bảo an toàn cho hệ thống ứng dụng web trước các mối đe dọa hiện đại.

Báo cáo bài tập lớn gồm 3 chương với nội dung chính như sau:

- Chương 1 tìm hiểu tổng quan về tường lửa ứng dụng web, bao gồm khái niệm, kiến trúc, vai trò, cách thức hoạt động và phân loại.

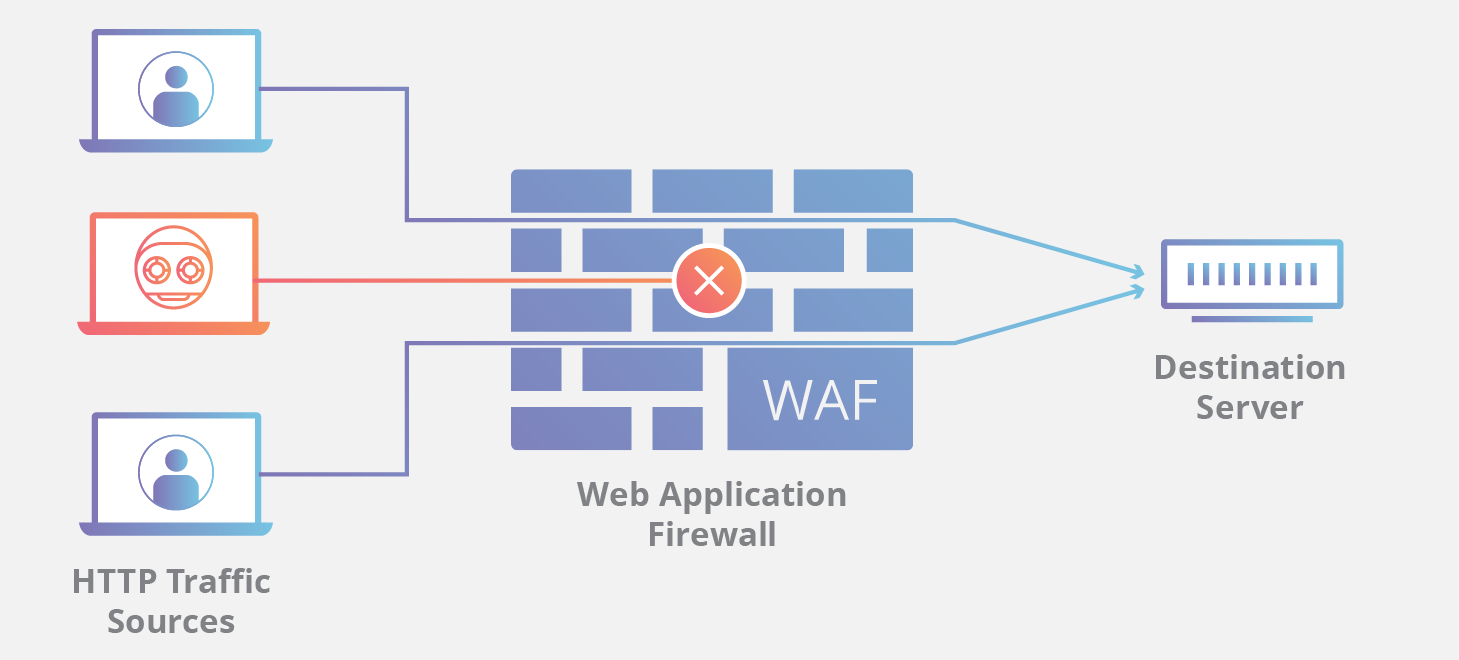
- Chương 2 thực hiện việc tìm hiểu về cách triển khai tường lửa ứng dụng web cũng như nêu ra các vấn đề thường gặp của nó.

- Chương 3 phân tích tường lửa ứng dụng web và so sánh nó với các công cụ bảo mật ứng dụng web khác.

- Chương 4 thực nghiệm cài đặt cấu hình Mod Security, cũng như thử nghiêm các phương pháp tấn công và Mod Security.

1. Tổng quan về tường lửa ứng dụng web
   1. Khái niệm

Tường lửa ứng dụng web (WAF - WAF) là một hệ thống bảo mật được thiết kế để bảo vệ các ứng dụng web khỏi các cuộc tấn công mạng và các lỗ hổng bảo mật. Bằng cách kiểm tra và lọc lưu lượng HTTP/HTTPS của các ứng dụng web, WAF giúp phát hiện, ngăn chặn và bảo vệ các ứng dụng web khỏi các lỗ hổng bảo mật và các loại tấn công phổ biến.



Hình 1- Tổng quan về tường lửa ứng dụng web

Tường lửa ứng dụng web sử dụng các chính sách hoặc quy tắc để loại bỏ lưu lượng độc hại, cho phép điều chỉnh nhanh chóng để đối phó với các loại tấn công khác nhau

WAF có thể ở dạng thiết bị, plugin máy chủ hoặc bộ lọc và có thể được tùy chỉnh cho một ứng dụng web.

Bằng cách triển khai tường lửa ứng dụng web trước ứng dụng web, một “tấm chắn” sẽ được đặt giữa ứng dụng web và Internet. Trong khi máy chủ proxy bảo vệ danh tính của máy khách bằng cách sử dụng một trung gian, là một loại proxy ngược, bảo vệ máy chủ khỏi bị lộ bằng cách yêu cầu máy khách đi qua Tường lửa ứng dụng web trước khi đến máy chủ.

* 1. Kiến trúc và vai trò của tường lửa ứng dụng web
     1. Kiến trúc của tường lửa ứng dụng web

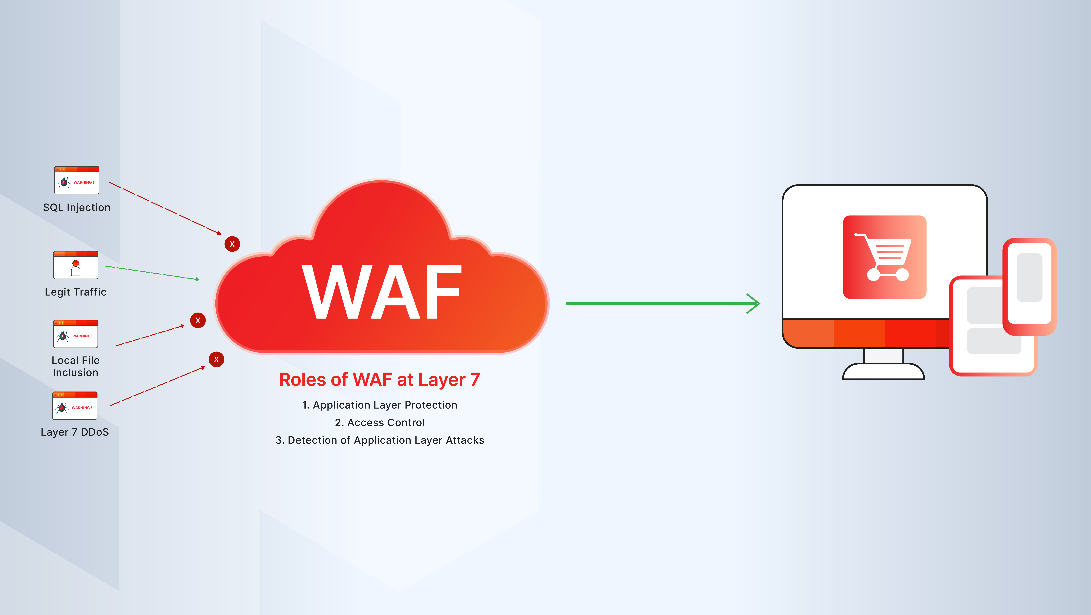
Kiến trúc ứng dụng web tiên tiến nhất vẫn chứa một lượng đáng kể các lỗ hổng bảo mật, khiến chúng dễ bị tổn thương trước các cuộc tấn công phổ biến như brute force và cả những cuộc tấn công phức tạp như khai thác thực thể XML (XML External Entities - XXE) hay tấn công giả mạo yêu cầu giữa các trang (Cross-Site Request Forgery - CSRF).Vì vậy, có các phương pháp cấu hình kiến trúc bảo mật WAF để giảm thiểu mức độ hiệu quả và tần suất của nhiều cuộc tấn công khác nhau.

Khi thiết kế hệ thống nên xem xét các cấu trúc khác nhau cho kiến trúc ứng dụng web để tối ưu hóa hiệu quả của WAF cho cấu hình cụ thể đó. Có các kiến trúc ứng dụng web an toàn cơ bản (một tầng hoặc hai tầng) trong đó máy chủ web và máy chủ cơ sở dữ liệu của ứng dụng sử dụng chung một máy chủ lưu trữ. Kiến trúc này có thể hữu ích trong giai đoạn đầu của quá trình phát triển dự án. Tuy nhiên, nó không phù hợp cho các ứng dụng chạy sản xuất do nguy cơ trở thành điểm lỗi duy nhất (single point of failure). Kiến trúc đa tầng chia các thành phần của ứng dụng thành nhiều tầng khác nhau tùy theo chức năng, và mỗi tầng chạy trên một hệ thống riêng biệt. Cách này tạo ra sự phân tách và tránh được rủi ro về điểm lỗi duy nhất. Trong hầu hết các kiến trúc ứng dụng, WAF thường được đặt phía sau tầng cân bằng tải (load balancing) để tối đa hóa hiệu quả sử dụng, hiệu suất, độ tin cậy và khả năng hiển thị.

WAF là một dịch vụ bảo mật dựa trên proxy ở tầng 7 và có thể triển khai ở bất cứ đâu trong đường dẫn dữ liệu. Tuy nhiên nên đặt WAF gần nhất với ứng dụng mà nó bảo vệ và nằm sau tầng cân bằng tải để tối ưu hóa kiến trúc cho việc sử dụng, hiệu suất, độ tin cậy và khả năng hiển thị.

* Sử dụng tài nguyên của WAF: Các tài nguyên CPU là yếu tố rất quan trọng đối với WAF vì chúng xử lý toàn bộ lưu lượng để đánh giá tính hợp lệ và an toàn của các yêu cầu, do đó tối đa hóa việc sử dụng từ vị trí đặt cụ thể của WAF trong đường truyền dữ liệu là rất quan trọng.
* Hiệu suất của WAF: Triển khai WAF phía sau tầng cân bằng tải giúp loại bỏ sự cần thiết của một tầng cân bằng tải dành riêng cho WAF ở phía trên, từ đó đơn giản hóa toàn bộ đường truyền dữ liệu và tăng hiệu suất cho các ứng dụng.
* Độ tin cậy của WAF: Để tăng độ tin cậy của WAF, các bộ cân bằng tải có thể được sử dụng để mở rộng WAF theo chiều ngang. WAF đặt phía trước tầng cân bằng tải dễ bị ảnh hưởng bởi lưu lượng tăng đột biến hoặc các cuộc tấn công, điều này yêu cầu thêm một tầng cân bằng tải, dẫn đến chi phí và độ phức tạp gia tăng.
* Khả năng hiển thị của WAF: Kiến trúc WAF đa đám mây hoặc đám mây lai hiện đại tạo ra một hệ thống bảo mật ứng dụng web phân tán. Hệ thống này thực thi bảo mật bằng cách sử dụng phân tích tích hợp để đưa ra các quyết định bảo mật thông minh. Cách tiếp cận này cũng cho phép triển khai theo từng ứng dụng và mở rộng linh hoạt qua các trung tâm dữ liệu và môi trường đa đám mây, đồng thời đáp ứng các yêu cầu tuân thủ GDPR, HIPAA, và PCI.
  + 1. Vai trò của tường lửa ứng dụng web

Tường lửa ứng dụng web nằm ở tầng thứ 7 trong mô hình mô hình OSI, có vai trò bảo vệ tầng ứng dụng, kiểm soát lượng truy cập vào phát hiện những đợt tấn công với các hình thức khác nhau.



Hình 2- Vai trò của tường lửa ứng dụng web

Bảo vệ tầng ứng dụng: Tường lửa ứng dụng web giúp bảo vệ ứng dụng web khỏi các cuộc tấn công mục tiêu tại mức ứng dụng. Hoạt động này bao gồm việc ngăn chặn các cuộc tấn công như SQL injection, cross-site scripting (XSS) và nhiều hình thức khác.

Kiểm soát truy cập: Tường lửa ứng dụng web cho phép quản trị viên kiểm soát truy cập vào nội dung và chức năng của ứng dụng web dựa trên các đặc điểm của yêu cầu như thông tin người dùng, địa chỉ IP và nhiều yếu tố khác.

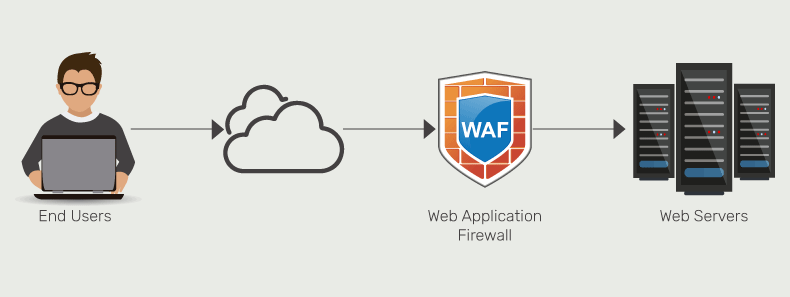
Tường lửa ứng dụng web giúp phát hiện tấn công tầng ứng dụng: Tường lửa ứng dụng web có khả năng phát hiện các hoạt động đáng ngờ tại tầng ứng dụng, cho phép xác định sự khác biệt giữa các yêu cầu hợp lệ của người dùng, hệ thống và đặc điểm của các cuộc tấn công. Trong trường hợp phát hiện có dấu hiệu nguy hiểm, hệ thống sẽ tự động ngăn chặn để tránh các ảnh hưởng đến ứng dụng và cơ sở dữ liệu.

Cụ thể WAF cung cấp khả năng bảo vệ chống lại các mối đe dọa đa dạng, bao gồm:

* SQL injection, command spam
* Cross-site scripting (XSS)
* Tấn công từ chối dịch vụ phân tán (DDoS)
* Các cuộc tấn công dành riêng cho ứng dụng

Và các lợi ích khác bao gồm:

* Bộ quy tắc mặc định mạnh mẽ
* Báo cáo và ghi nhật ký theo thời gian thực để có khả năng hiển thị tức thì
* …
  1. Cách thức hoạt động của tường lửa ứng dụng web



Hình 3- Vai trò của tường lửa ứng dụng web

WAF có thể được tích hợp vào các thiết bị phần cứng, plugin phần mềm phía máy chủ hoặc lọc lưu lượng truy cập dưới dạng dịch vụ. Bảo mật WAF bảo vệ các ứng dụng web khỏi các điểm cuối độc hại và về cơ bản là đối lập với máy chủ proxy (tức là proxy ngược), bảo vệ thiết bị khỏi các ứng dụng độc hại.

Để đảm bảo tính bảo mật, WAF chặn và kiểm tra tất cả các yêu cầu HTTP. Lưu lượng truy cập giả chỉ bị chặn hoặc kiểm tra bằng các CAPTCHA được thiết kế để ngăn chặn các chương trình máy tính và bot có hại.

Quản trị WAF dựa trên các quy trình bảo mật được xây dựng dựa trên các chính sách tùy chỉnh, nhằm giải quyết các lỗi bảo mật ứng dụng web hàng đầu được liệt kê bởi OWASP

Theo truyền thống, các chính sách này có thể phức tạp, yêu cầu quản trị viên chuyên môn phải định cấu hình WAF phù hợp với chính sách bảo mật của công ty. Những quản trị viên này chịu trách nhiệm đặt, định cấu hình, quản trị và giám sát WAF một cách chính xác để đảm bảo tính bảo mật tối đa.

* 1. Phân loại tường lửa ứng dụng web
     1. Phân loại theo mô hình bảo mật

Blocklist (Danh sách đen) và Allowlist (Danh sách cho phép) là hai cơ chế quan trọng mà WAF (WAF) sử dụng để kiểm soát lưu lượng truy cập đến máy chủ.

* + - 1. Blocklist WAF

Blocklist là danh sách các địa chỉ IP, tên miền, hoặc kiểu yêu cầu được chặn bởi tường lửa ứng dụng web. Nếu một yêu cầu đến từ các nguồn này hoặc có các đặc điểm nằm trong danh sách blocklist, nó sẽ bị tường lửa ứng dụng web từ chối và không được chuyển tiếp đến máy chủ.

Cách hoạt động: tường lửa ứng dụng web kiểm tra lưu lượng truy cập dựa trên blocklist để tìm các mẫu, địa chỉ hoặc hành vi đã bị liệt kê trước. Nếu có sự trùng khớp, yêu cầu sẽ bị từ chối.

Blocklist thường được sử dụng để chặn:

* Địa chỉ IP của kẻ tấn công đã biết.
* Các miền hoặc quốc gia cụ thể.
* Các kiểu tấn công đã xác định trước (SQL injection, XSS, brute-force, v.v.).

Nhược điểm: Blocklist chỉ chặn được các mối đe dọa đã biết trước. Các mối đe dọa mới, chưa được liệt kê, có thể vượt qua.

* + - 1. Allowlist WAF

Allowlist là danh sách các địa chỉ IP, tên miền, hoặc kiểu yêu cầu mà tường lửa ứng dụng web sẽ cho phép truy cập vào máy chủ. Bất kỳ yêu cầu nào không nằm trong allowlist sẽ bị từ chối bởi tường lửa ứng dụng web.

Cách hoạt động: tường lửa ứng dụng web kiểm tra yêu cầu truy cập và so sánh với allowlist. Chỉ những yêu cầu hoặc nguồn được cho phép mới được chuyển đến máy chủ.

Allowlist thường được sử dụng khi muốn chỉ cho phép:

* Những nguồn truy cập đáng tin cậy đã biết (ví dụ: địa chỉ IP của các đối tác, nhân viên nội bộ).
* Các kiểu lưu lượng truy cập nhất định (ví dụ: chỉ cho phép truy cập qua HTTPs).

Nhược điểm: Allowlist có thể quá nghiêm ngặt, dẫn đến việc chặn nhầm các yêu cầu hợp pháp nếu chúng không nằm trong danh sách. Điều này đòi hỏi quản trị viên phải duy trì danh sách chính xác và liên tục cập nhật.

* + 1. Phân loại theo mô hình triển khai
       1. Hardware base WAF

Tường lửa ứng dụng web dựa trên phần cứng thường được triển khai thông qua một thiết bị phần cứng và được cài đặt trong mạng cục bộ (LAN), gần ứng dụng và máy chủ web hơn. Hệ điều hành hỗ trợ cấu hình và cập nhật phần mềm khi nó chạy trong thiết bị. Ưu điểm lớn nhất của nó là tốc độ nhanh và hiệu suất cao.

* + - 1. Software based WAF

Không giống như WAF dựa trên phần cứng được triển khai thông qua thiết bị phần cứng vật lý, WAF dựa trên phần mềm được cài đặt trong máy ảo . Về cơ bản, điều này giống như các thành phần của WAF phần cứng, với điểm khác biệt duy nhất là người dùng cần bộ ảo hóa của riêng họ để chạy máy ảo. WAF dựa trên phần mềm là một lựa chọn tuyệt vời vì tính linh hoạt của nó

* + - 1. Cloud base WAF

Tường lửa ứng dụng web dựa trên đám mây được cung cấp như một dịch vụ bảo mật từ nhà cung cấp bên thứ ba. WAF này bảo vệ ứng dụng web bằng cách phân tích lưu lượng truy cập thông qua hạ tầng đám mây của nhà cung cấp trước khi đến máy chủ ứng đụng web.Nó cũng có thể cung cấp giải pháp được cập nhật liên tục để bảo vệ khỏi các mối đe dọa mới nhất mà không phải tốn thêm công sức hoặc chi phí nào từ phía người dùng. Hạn chế của tường lửa ứng đụng web dựa trên đám mây là nó bị phụ thuộc vào nhà cung cấp dịch vụ bên thứ ba.

* 1. Kết chương

Chương 1 tìm hiểu tổng quan về tường lửa ứng dụng web, bao gồm khái niệm, kiến trúc, vai trò, cách thức hoạt động và phân loại. Các kiến thức được nêu ra ở chương này cũng là nền tảng quan trọng để tìm hiểu sâu hơn về Tường lửa ứng dụng web.

1. Cách triển khai và các vấn đề của tường lửa ứng dụng web
   1. Triển khai tường lửa ứng dụng web

Triển khai tường lửa ứng dụng web là quá trình tích hợp nó vào ứng dụng web hoặc cơ sở hạ tầng phân phối ứng dụng web để cung cấp thêm một lớp bảo vệ chống lại các cuộc tấn công dựa trên web khác nhau. Việc triển khai thường liên quan đến việc định cấu hình tường lửa ứng dụng web để thực thi một bộ chính sách bảo mật được thiết kế để phát hiện và chặn lưu lượng truy cập độc hại nhắm mục tiêu vào ứng dụng web. Điều này có thể bao gồm việc xác định các quy tắc để kiểm tra lưu lượng truy cập HTTP/HTTPS đến cho các mẫu và sự bất thường cho thấy một cuộc tấn công, chẳng hạn như SQL injection hoặc cross-site scripting (XSS).

Việc triển khai tường lửa ứng dụng web cũng có thể yêu cầu tích hợp WAF với các giải pháp bảo mật khác, chẳng hạn như hệ thống ngăn chặn xâm nhập (IPS) và nền tảng quản lý sự kiện và thông tin bảo mật (SIEM), để cho phép khả năng phát hiện và ứng phó mối đe dọa tốt hơn.

Nhìn chung, triển khai WAF là một khía cạnh quan trọng của bảo mật ứng dụng web giúp bảo vệ chống lại một loạt các cuộc tấn công nhắm vào các ứng dụng web và dịch vụ web.

* 1. Các trường hợp sử dụng tường lửa ứng dụng web
     1. Bảo vệ Website và ứng dụng

Đúng như tên gọi, một trong những trường hợp sử dụng chính của WAF là bảo vệ các ứng dụng giao tiếp qua HTTP, như website, chức năng serverless và các điểm cuối API. Nếu không có WAF, các cuộc tấn công đã biết và chưa biết sẽ không được phát hiện, rò rỉ dữ liệu khó được ngăn chặn, và việc truy cập vào các URL cũng như cổng sẽ không có nhiều kiểm soát.

* + 1. Tuân thủ các tiêu chuẩn bảo mật và quy định

Cùng với việc ngăn chặn các mối đe dọa, thiết lập WAF là cách nhanh nhất và hiệu quả nhất để tuân thủ các tiêu chuẩn và quy định bảo mật. Chẳng hạn, các trang web xử lý dữ liệu thẻ tín dụng phải tuân thủ tiêu chuẩn PCI-DSS (Payment Card Industry Data Security Standard). Các công ty thẻ tín dụng và các công ty yêu cầu mức độ bảo mật cao có thể hưởng lợi rất nhiều từ WAF.

* + 1. Kiểm soát Bot và Ngăn chặn Tấn công DDoS

Khi bot được sử dụng ngày càng nhiều, WAF giúp kiểm soát mức độ truy cập của các bot vào hệ thống. Các bot có ích giúp duy trì hoạt động, trong khi WAF quản lý các bot xấu, ngăn chặn spam, trộm cắp thông tin, sao chép nội dung trang web, khởi xướng các cuộc tấn công DDoS và nhiều hành vi khác.

* + 1. Khắc phục các Lỗ hổng Bảo mật

Trong quá trình bảo mật ứng dụng, vẫn có khả năng các lỗ hổng xuất hiện trong môi trường sản xuất. Việc tìm ra giải pháp và phát hành bản vá sẽ cần thời gian. WAF là cách tốt nhất để bảo mật các lỗ hổng tạm thời này.

* + 1. Phát hiện Xâm nhập trong Thời gian Thực

Cách hiệu quả nhất để phát hiện tấn công là theo dõi lưu lượng truy cập theo thời gian thực để đội ngũ bảo mật có thể phản ứng kịp thời. Trong hệ thống phân tán, việc này có thể khó khăn vì nhật ký bị phân tán. WAF giúp tập trung việc ghi log và thu thập số liệu, đóng vai trò là điểm trung tâm để phát hiện và đánh giá các cuộc tấn công trước đó.

* + 1. Thiết lập Chính sách

Một trường hợp sử dụng khác của WAF là khả năng kiểm tra và lọc các gói HTTP. Với điều này, các tổ chức có thể đặt các quy tắc để cho phép hoặc chặn kết nối dựa trên nội dung, giúp kiểm soát và điều chỉnh nội dung truy cập phù hợp với chính sách bảo mật.

* 1. Các vấn đề của tường lửa ứng dụng web

Mặc dù WAF có rất quan trọng và đã trở thành một phần không thể thiếu trong bảo mật ứng dụng web, song nó vẫn tồn tại một số hạn chế nhất định

* + 1. Cấu hình

Tường lửa ứng dụng web có thể chặn lưu lượng truy cập hoặc yêu cầu hợp pháp nếu nhầm lẫn chúng với hoạt động độc hoặc không thể ngăn chặn các yêu cầu độc hại nếu nó không được cấu hình một cách chính xác.

* + 1. Dễ bị bypass

Các kẻ tấn công có thể tìm ra những cách để né tránh tường lửa ứng dụng web bằng cách làm lệch hướng các quy tắc hoặc tận dụng các lỗ hổng cấu hình bằng cáchw sử dụng các kỹ thuật encoding (mã hóa đặc biệt), obfuscation (làm rối mã), hoặc fragmentation (phân mảnh gói dữ liệu) có thể giúp kẻ tấn công né tránh sự phát hiện của tường lửa ứng dụng web.

* + 1. Chi phí và hiệu suất

Tường lửa ứng dụng web có thể làm tăng thêm chi phí vào ứng dụng web và có thể ảnh hưởng đến hiệu suất của ứng dụng. Đây có thể là mối lo ngại đối với các ứng dụng quan trọng có lưu lượng truy cập cao hoặc yêu cầu hiệu suất nghiêm ngặt

Ngoài ra , nhiều cuộc tấn công có thể được thực hiện đối với một ứng dụng nằm ngoài khả năng của WAF.Ta cần có WAF cải tiến hoặc cần nhiều hệ thống hơn ngoài WAF để tăng cường bảo mật như WAAP (Web Application and API Protection),RASP (Runtime Application Self-Protection),IDPS (Intrusion Detection and Prevention System)

* 1. Một số nền tảng dịch vụ cung cấp tường lửa ứng dụng web
     1. Azure WAF

Microsoft cung cấp Azure Application Gateway WAF dưới dạng giải pháp bảo mật lớp 7 được quản lý tập trung, tích hợp vào trung tâm bảo mật Azure và cung cấp khả năng quản lý bảo mật thuận tiện mà không yêu cầu thay đổi ứng dụng. Giá WAF của Cổng ứng dụng Azure được tích hợp vào mô hình định giá tổng thể, tùy thuộc vào lượng dữ liệu được xử lý bởi các cổng và khoảng thời gian cổng được cung cấp và khả dụng.

* + 1. AWS WAF

Amazon WAF cho phép người dùng thêm tùy chọn tường lửa ứng dụng web cho các giải pháp AWS hiện có. Không giống như các nhà cung cấp khác, người dùng không phải trả phí trọn gói cho bảo mật ứng dụng WAF mà bị tính phí cho số lượng quy tắc AWS WAF được thêm vào và số yêu cầu web nhận được mỗi tháng. Để giảm nhu cầu định cấu hình các chính sách bảo mật tùy chỉnh, tính năng Tự động hóa bảo mật AWS WAF sẽ tự động cung cấp ACL web với các quy tắc AWS WAF để lọc các cuộc tấn công dựa trên web phổ biến. Sau đó, người dùng có thể chọn những tính năng bảo vệ nào sẽ được đưa vào AWS WAF của họ.

* + 1. Akamai WAF

Tường lửa ứng dụng web Akamai được gọi là Kona WAF. Nó được phân phối trên Nền tảng thông minh Akamai. Kona WAF được triển khai ở rìa mạng thay vì trung tâm dữ liệu. Dịch vụ tường lửa ứng dụng web của Kona xử lý lưu lượng truy cập bị đe dọa mà không ảnh hưởng đến hiệu suất của máy chủ gốc.

* + 1. Cloudflare WAF

Tường lửa ứng dụng web Cloudflare bao gồm 148 quy tắc WAF tích hợp có thể được áp dụng chỉ bằng một cú nhấp chuột. Sự bảo vệ này bổ sung cho các biện pháp bảo vệ khỏi 10 lỗ hổng hàng đầu của OWASP, được cung cấp theo mặc định. Khách hàng doanh nghiệp và doanh nghiệp có thể yêu cầu các quy tắc WAF tùy chỉnh cho các mối đe dọa cụ thể. Cloudflare WAF thách thức khách truy cập web bằng bài kiểm tra CAPTCHA. Nó cũng cung cấp một loạt các cài đặt bảo mật từ thấp đến cao.

* 1. Kết chương

Chương 2 đã trình bày về cách thức triển khai tường lửa ứng dụng web cũng như các vấn đề gặp phải khi sử dụng. Đồng thời chương cũng có giới thiệu về 1 số bên cũng cấp dịch vụ tường lửa ứng dụng web và các đặc điểm nổi trội của chúng

1. SO SÁNH TƯỜNG LỬA ỨNG DỤNG WEB VỚI CÁC CÔNG CỤ BẢO MẬT ỨNG DỤNG WEB KHÁC
   1. Tường lửa ứng dụng web với tường lửa truyền thống

Tường lửa truyền thống bảo vệ luồng thông tin giữa các máy chủ trong khi WAF có thể lọc lưu lượng truy cập cho một ứng dụng web cụ thể. Tường lửa mạng và tường lửa ứng dụng web bổ sung cho nhau và có thể hoạt động cùng nhau. Khả năng của WAF và bảo mật tường lửa truyền thống có thể kết hợp kiểm tra cổng/giao thức và kiểm tra cấp ứng dụng để ngăn chặn sự xâm nhập và sử dụng các nguồn thông tin bên ngoài. Một điểm khác biệt giữa tường lửa truyền thống và tường lửa ứng dụng web là các phương pháp bảo mật truyền thống bao gồm tường lửa mạng, hệ thống phát hiện xâm nhập (IDS) và hệ thống ngăn chặn xâm nhập (IPS). Chúng có hiệu quả trong việc chặn lưu lượng tầng 3- tầng 4 bất hợp pháp, dựa trên mô hình Kết nối hệ thống mở (OSI). Tùy thuộc vào giao thức đang chạy, tường lửa truyền thống có thể hoạt động bằng phương pháp không trạng thái hoặc phương pháp trạng thái. Tường lửa truyền thống không thể phát hiện các cuộc tấn công duy nhất đối với các lỗi bảo mật trong ứng dụng web vì chúng không hiểu Giao thức truyền siêu văn bản (HTTP) xảy ra ở lớp 7 của mô hình OSI. Họ cũng chỉ cho phép mở hoặc đóng cổng gửi và nhận các trang web được yêu cầu từ máy chủ HTTP. Đây là lý do tại sao WAF lại quan trọng trong việc ngăn chặn các cuộc tấn công như SQLi, chiếm quyền điều khiển phiên và XSS.

* 1. Tường lửa ứng dụng web với tường lửa thế hệ mới

Tường lửa ứng dụng web (WAF - WAF) và tường lửa thế hệ mới (Next-Generation Firewall - NGFW) đều là các công nghệ bảo mật nhưng có sự khác biệt về mục tiêu và phạm vi bảo vệ. WAF tập trung bảo vệ các ứng dụng web khỏi các cuộc tấn công phổ biến như SQL injection, cross-site scripting (XSS), hay các lỗ hổng logic ứng dụng, bằng cách phân tích và lọc các yêu cầu HTTP/S dựa trên chính sách bảo mật của ứng dụng. Ngược lại, NGFW là sự kết hợp giữa tường lửa truyền thống và các tính năng bảo mật nâng cao như kiểm tra sâu gói tin (DPI), phát hiện và ngăn chặn xâm nhập (IPS), kiểm soát ứng dụng, và tích hợp khả năng bảo vệ chống phần mềm độc hại. Trong khi WAF chủ yếu được triển khai tại lớp ứng dụng (Layer 7), NGFW hoạt động ở nhiều lớp trong mô hình OSI (Layer 3-7) để cung cấp bảo vệ toàn diện hơn cho toàn bộ mạng. Vì vậy, WAF phù hợp để bảo vệ các ứng dụng web cụ thể, còn NGFW là lựa chọn để bảo vệ tổng thể hạ tầng mạng với khả năng giám sát và kiểm soát lưu lượng mạng ở mức độ sâu hơn.

* 1. Tường lửa ứng dụng web với hệ thống ngăn chặn xâm nhập

Tường lửa ứng dụng web (WAF) cung cấp bảo mật web cho các dịch vụ trực tuyến khỏi các cuộc tấn công bảo mật độc hại như SQL SQL, cross-site scripting (XSS). Hệ thống ngăn chặn xâm nhập (IPS) cũng tương tự khi nó cố gắng xác định các mối đe dọa được phát hiện và ngăn chặn chúng. IPS liên tục theo dõi mạng để tìm kiếm các sự kiện nguy hiểm tiềm ẩn và thu thập thông tin về chúng. Sự khác biệt lớn nhất giữa WAF và IPS là mức độ thông minh cần thiết để phân tích lưu lượng truy cập trên Lớp 7. WAF cảnh báo mọi nỗ lực truy cập ứng dụng web, cho dù đó là cuộc họp, người dùng và chương trình trong khi IPS chỉ dựa trên chữ ký và không biết về các phiên và người dùng đang cố gắng truy cập vào ứng dụng web.

* 1. Kết chương

Chương đã so sánh tường lửa ứng dụng web với các công cụ bảo mật ứng dụng web khác như tường lửa truyền thống, tương lửa kiểu mới hay IDS/IPS. Các so sánh chỉ ra kỹ các điểm nổi bật cũng như sự khác nhau giữa các bên, làm nổi bật nên điểm mạnh cũng như điểm yếu của từng loại.

1. Thực Nghiệm
   1. Tìm hiểu về Mod Security

ModSecurity là một web application firewall(WAF) được Ivan Ristic phát triển dành cho Apache Web Server. Giống như những firewall thông thường khác, nó lọc những lưu lượng dữ liệu vào và ra để có thể quyết định chặn lại những lưu lượng mà nó nghi ngờ là độc hại dựa theo tập lệnh nó định nghĩa. Nó còn có nhiều tính năng vượt trội khác như : HTTP transaction logging và content injection…

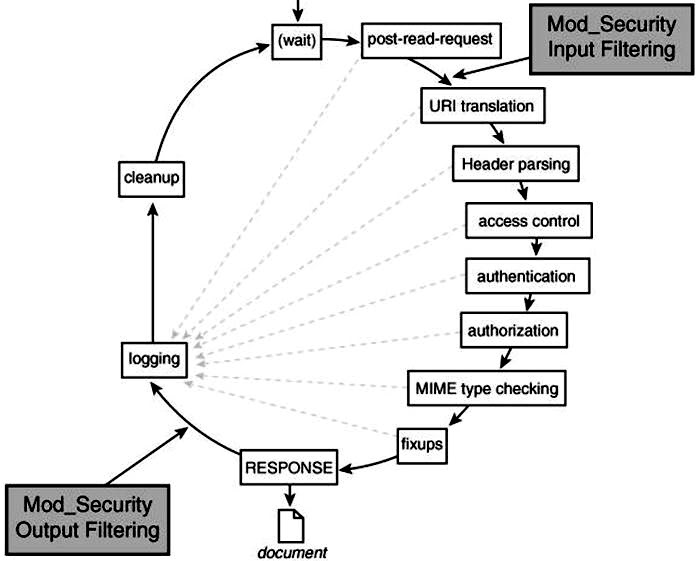
Các luật được tạo và chỉnh sửa sử dụng một định dạng văn bản đơn giản, nó làm cho việc viết rules trở nên đơn giản hơn. Một khi đã quen với cú pháp của ModSecurity, chúng ta có thể nhanh chóng viết được những rules để block một exploit mới hoặc ngăn chặn một lỗ hổng.

Có thể hình dung ModSecurity như là một trạm trung gian giữa HTTP request và httpd(dịch vụ web server). Khi phát hiện ra một cuộc tấn công, những chi tiết về vụ tấn công sẽ được lưu vào log file hoặc một email có thể được gửi tới người quản trị viên để báo hiệu có một cuộc tấn công xảy ra trên hệ thống.

ModSecurity cho phép bạn bảo vệ server của mình thông qua việc viết các luật nhằm bao phủ một dải các viễn cảnh tấn công có thể. Do đó, ModSecurity là một lớp bổ sung có thể giúp bạn bảo vệ theo một cách không cần bản vá.

ModSecurity giải quyết các vấn đề tầm nhìn : nó giúp bạn nhìn thấy lưu lượng web của mình. Đó là chìa khóa trong security : mỗi khi bạn có thế nhìn thấy HTTP traffic, bạn có thể phân tích nó trong thời gian thực, record nó khi cần thiết và phàn ứng lại với các sự kiện. Điểm nổi bật ở đây là bạn có thể làm tất cả mà không tác động gì tới các ứng dụng web. Thậm chí tốt hơn, nó còn có thể áp dụng với bất kì ứng dụng nào, ngay cả khi bạn không thể truy được vào source code.

Quá trình xử lý các request của Apache và ModSecurity :

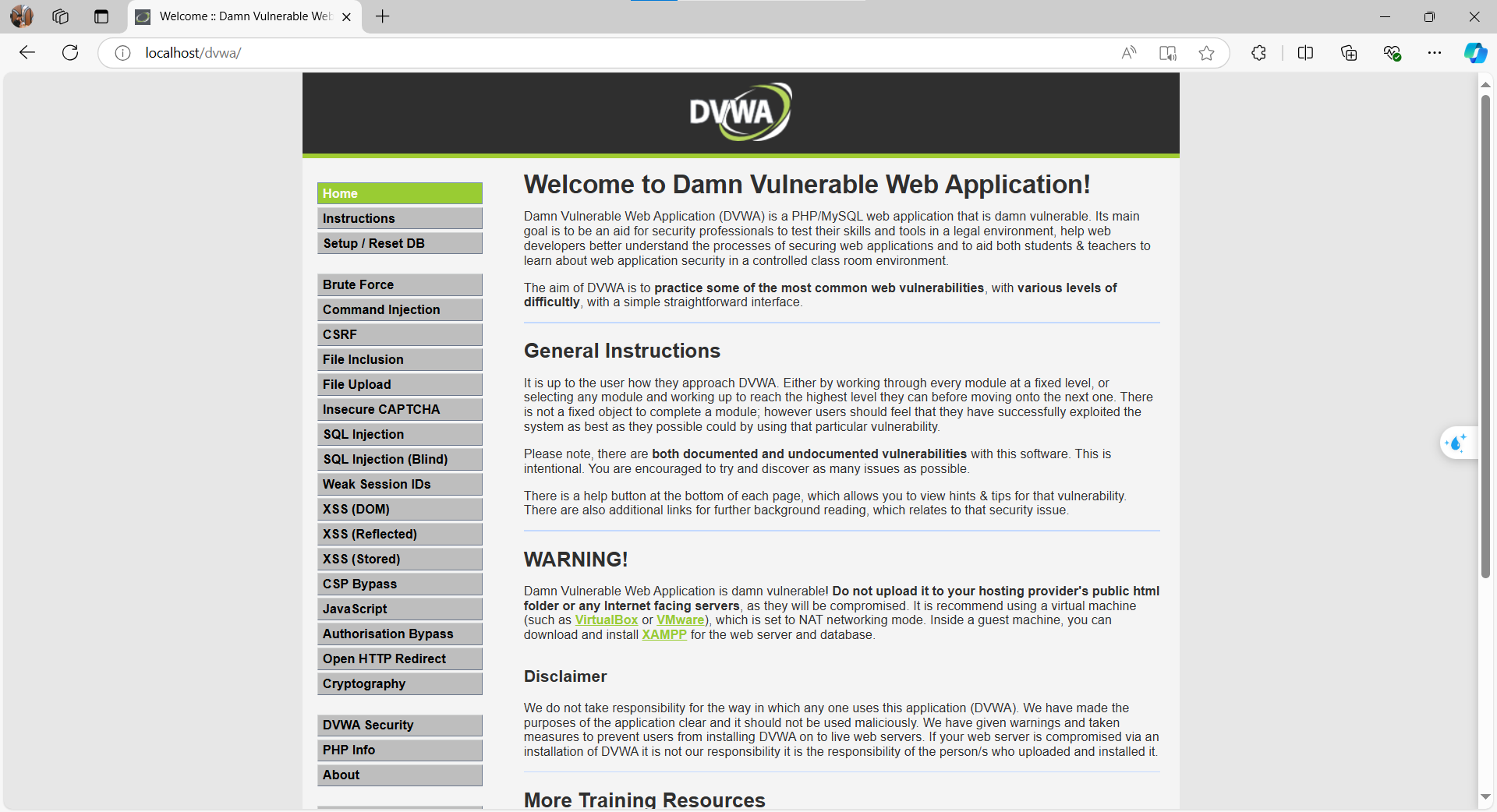


Hình 4- quá trình xử lý của Mod Security

* Phase Request Header : rule được đặt tại đây sẽ được thực hiện ngay sau khi Apache 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 …
* 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.
* Phase Response Header : 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.
* 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 ở phase request header và phase request body không phát hiện được tấn 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.
* 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 Apache. Đây cũng là thời điểm cuối cùng để bạn chặn các connection không mong muốn, kiểm tra các response header mà bạn không thể kiểm tra ở phase response header và phase response body.
  1. Cài đặt môi trường

nhóm sử dụng Damn Vulnerable Web Application (DVWA) là một ứng dụng mã nguồn PHP/MySQL tập hợp sẵn các lỗi logic về bảo mật ứng dụng web trong mã nguồn PHP.

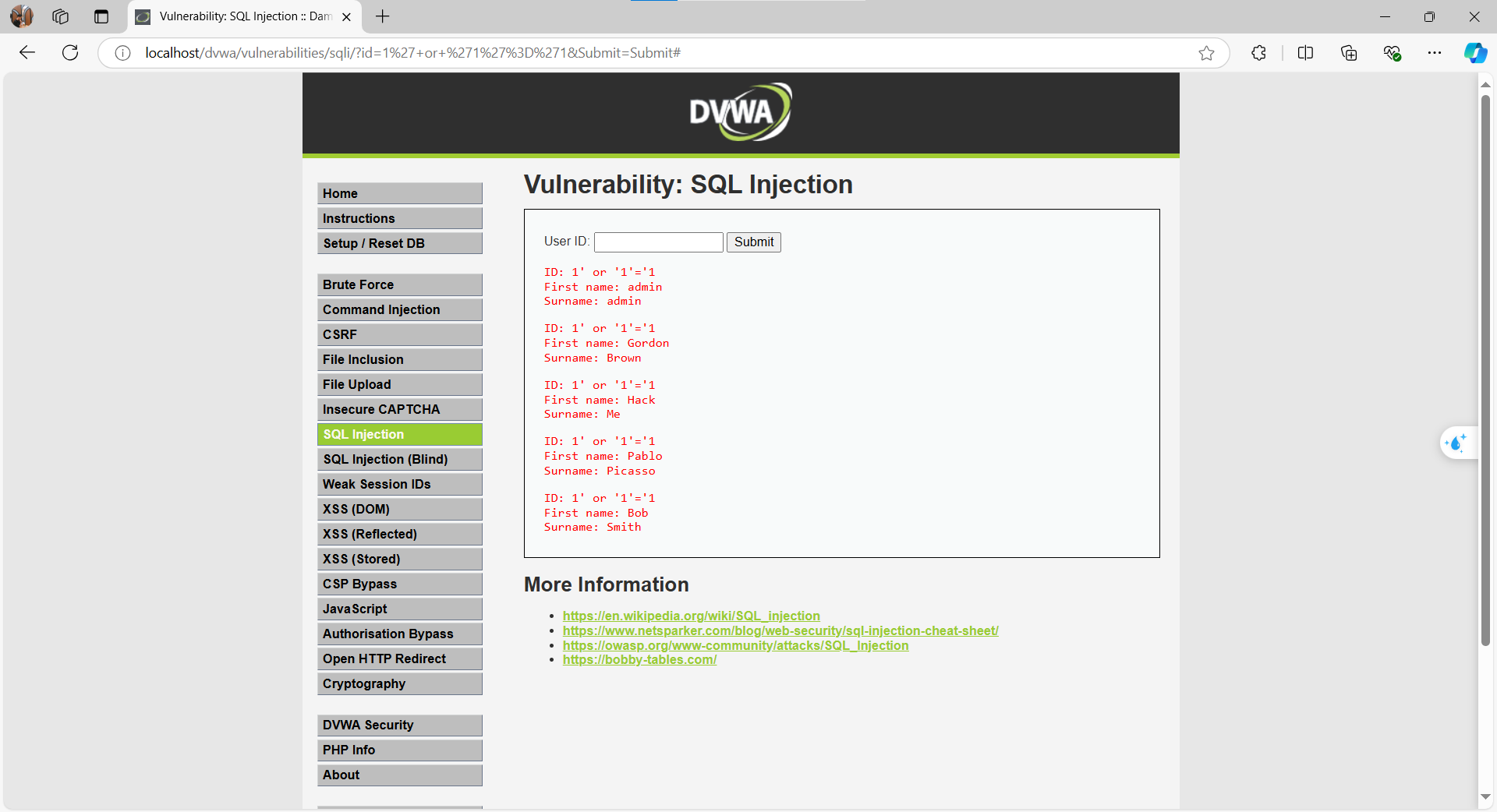
Sau khi tải mã nguồn từ : <https://github.com/digininja/DVWA>, giải nén vào thư mục *C:\xampp\htdocs*, khởi chạy và đăng nhập bằng tài khoản username/password: *admin/password* thì giao diện trang web như sau:



Hình 5- Giao diện DVWA

* 1. Thử nghiệm tấn công SQL Injection và XSS khi chưa tích hợp Mod Security
     1. Thử nghiệm SQLi

Vào SQL Injection nhập vào ***1’ or ‘1’ = ‘1*** và submit thì ta nhận được kết quả sau



Hình 6: thử nghiệm SQLi khi chưa triển khai Mod Security

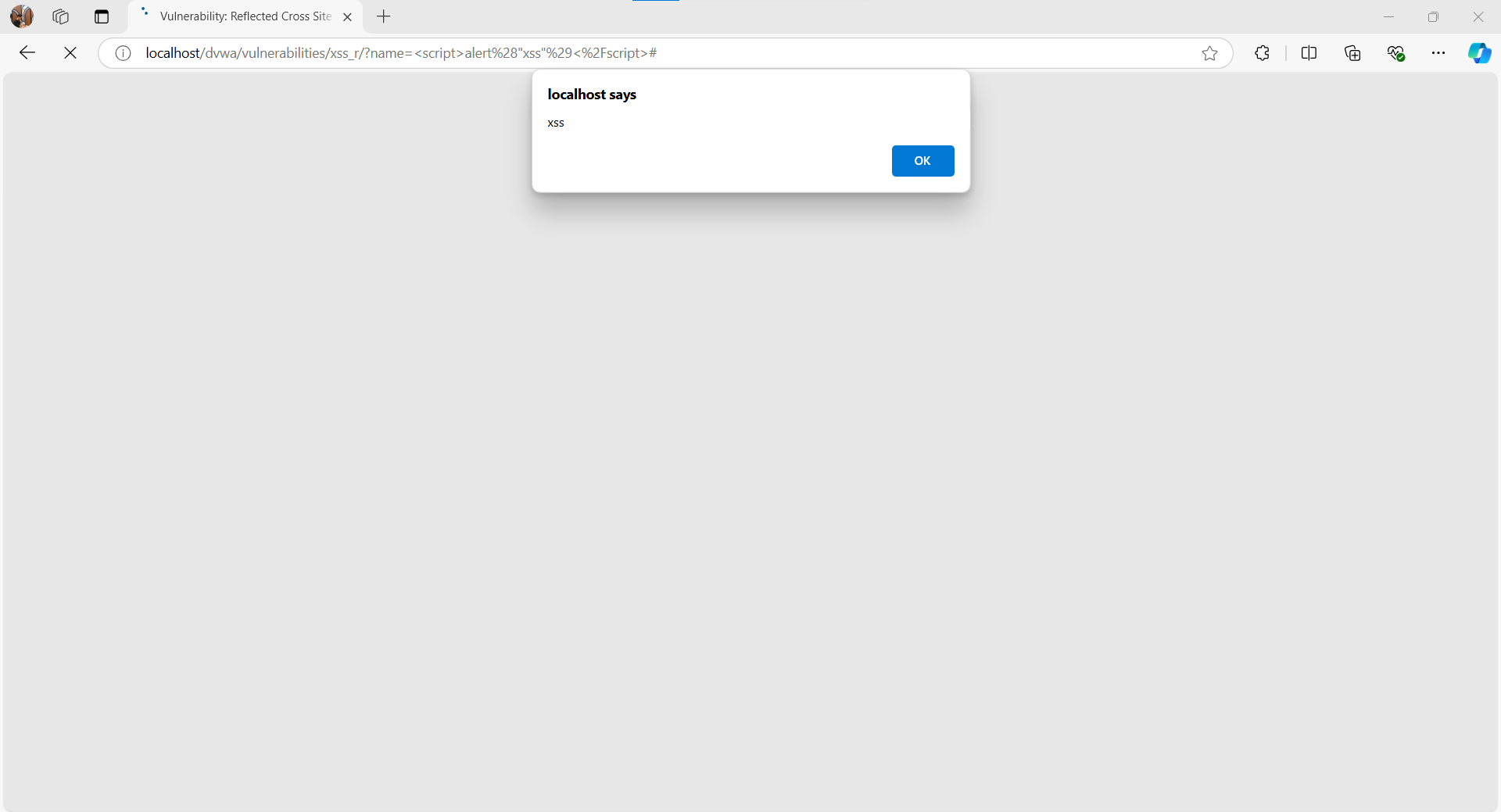
Kết quả sẽ nhận được thông tin của tất cả các users có trong cơ sở dữ liệu. Bởi vì câu truy vấn lúc này được hiểu là : ***SELECT \* FROM users WHERE user\_id = ‘1’ or ‘1’ = ‘1’***

* + 1. Thử nghiệm XSS
       1. XSS Reflected

Vào XSS Reflected và submit đoạn script sau: ***<script> alert('xss') </script>***

hoặc ***<script> console.log('xss') </script>***

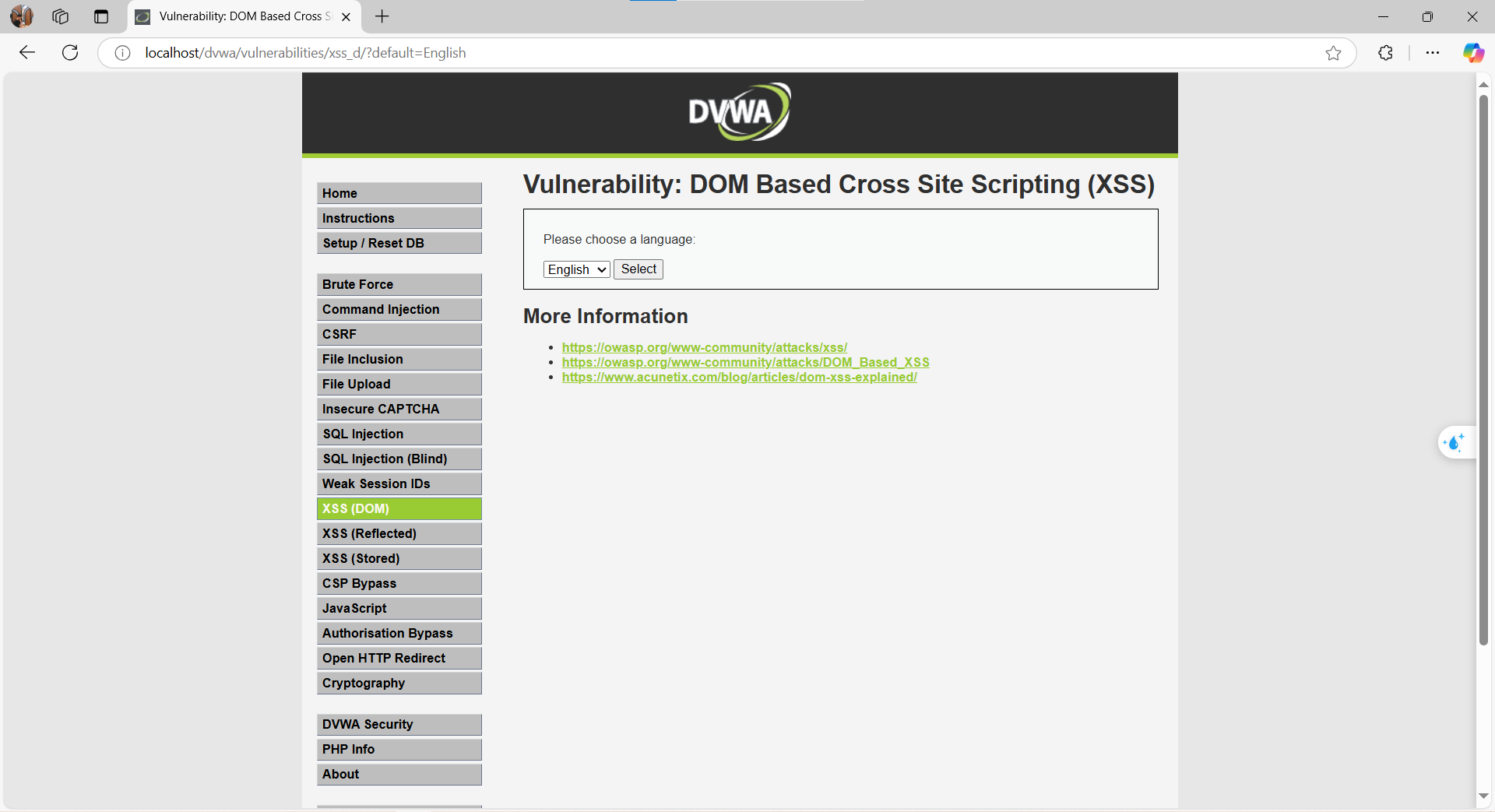
Khi đó trang web sẽ hiển thị thông báo xss hoặc trong console của trang web sẽ hiển thị dòng chữ xss



Hình 7- thử nghiệm XSS Reflected khi chưa triển khai Mod Security

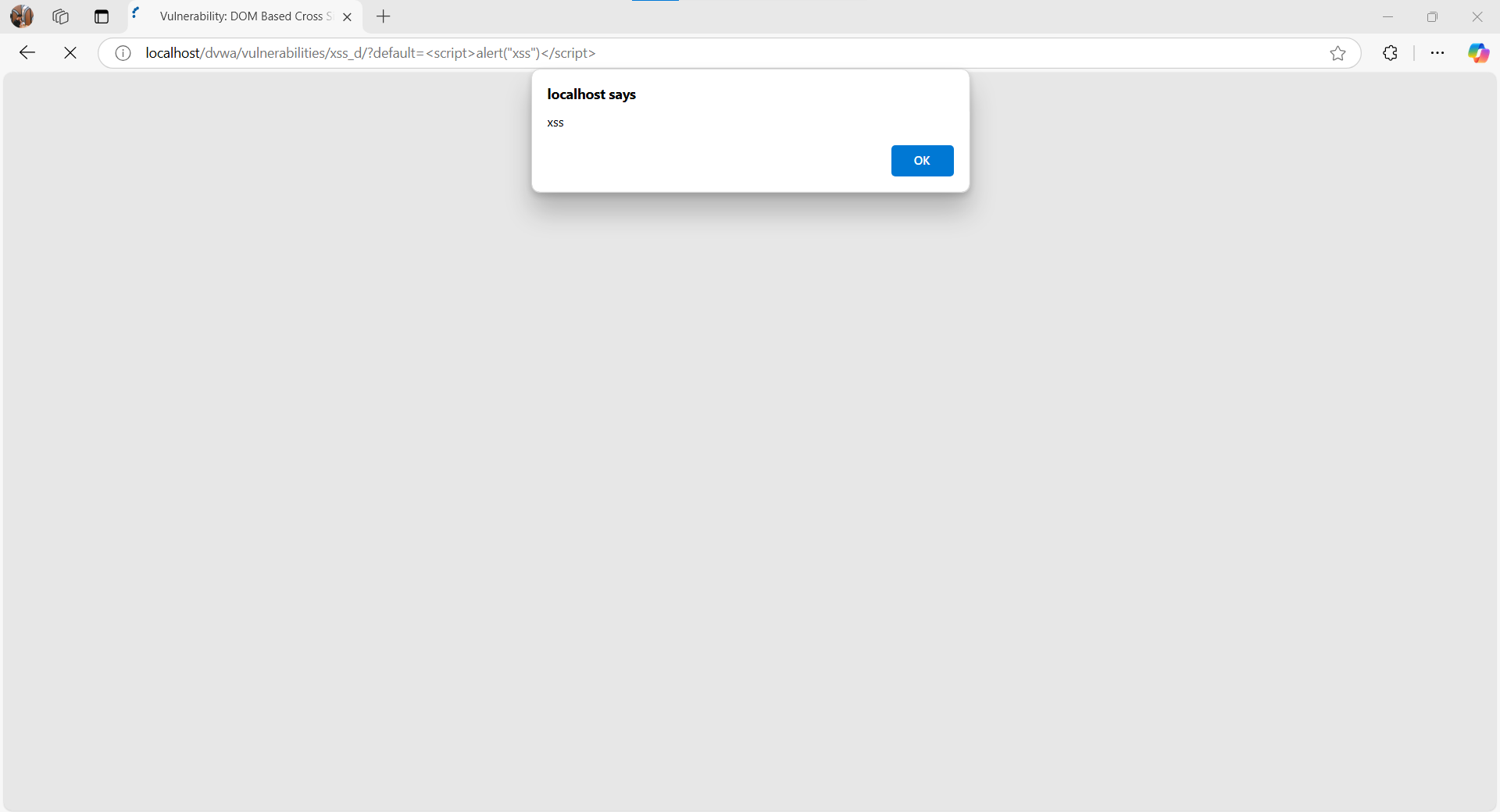
* + - 1. XSS DOM

Vào XSS DOM chọn một ngôn ngữ, khi đó trang web sẽ có đường dẫn là: http://localhost/dvwa/vulnerabilities/xss\_d/?default=English và hiển thị như sau



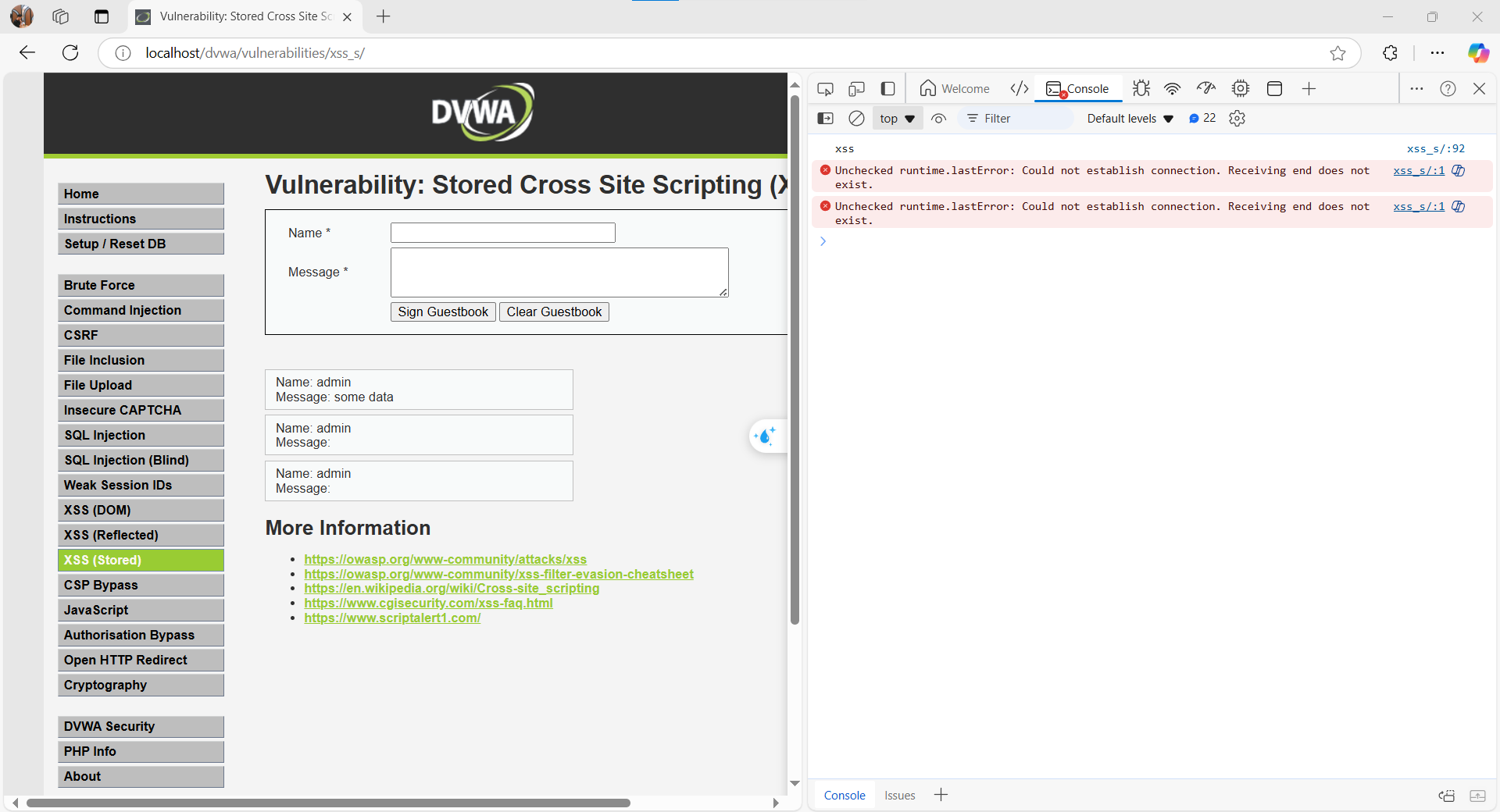
Hình 8- Giao diện XSS DOM

Ta thực hiển sửa giá trị của tham số default thành ***<script>alert("xss")</script>*** khi đó trang web sẽ hiển thị thông báo xss



Hình 9- thử nghiệm XSS DOM khi chưa triển khai Mod Security

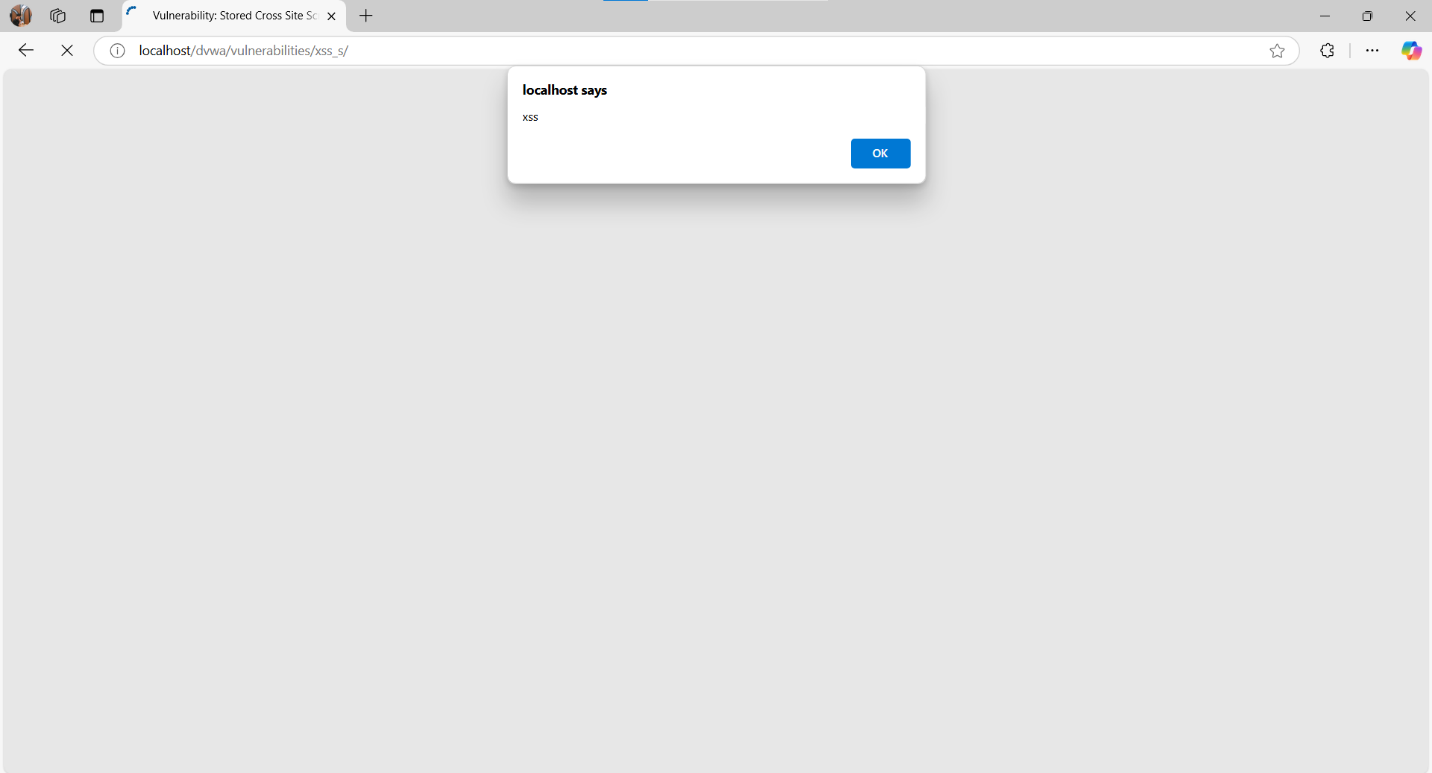
Ta thực hiện tương tự với giá trị của tham số default là ***<script>console.log(“xss”)</script>*** ta nhận được kết quả là hiển thị dòng chữ “xss” trong console của trang web



Hình 10- thử nghiệm XSS DOM khi chưa triển khai Mod Security

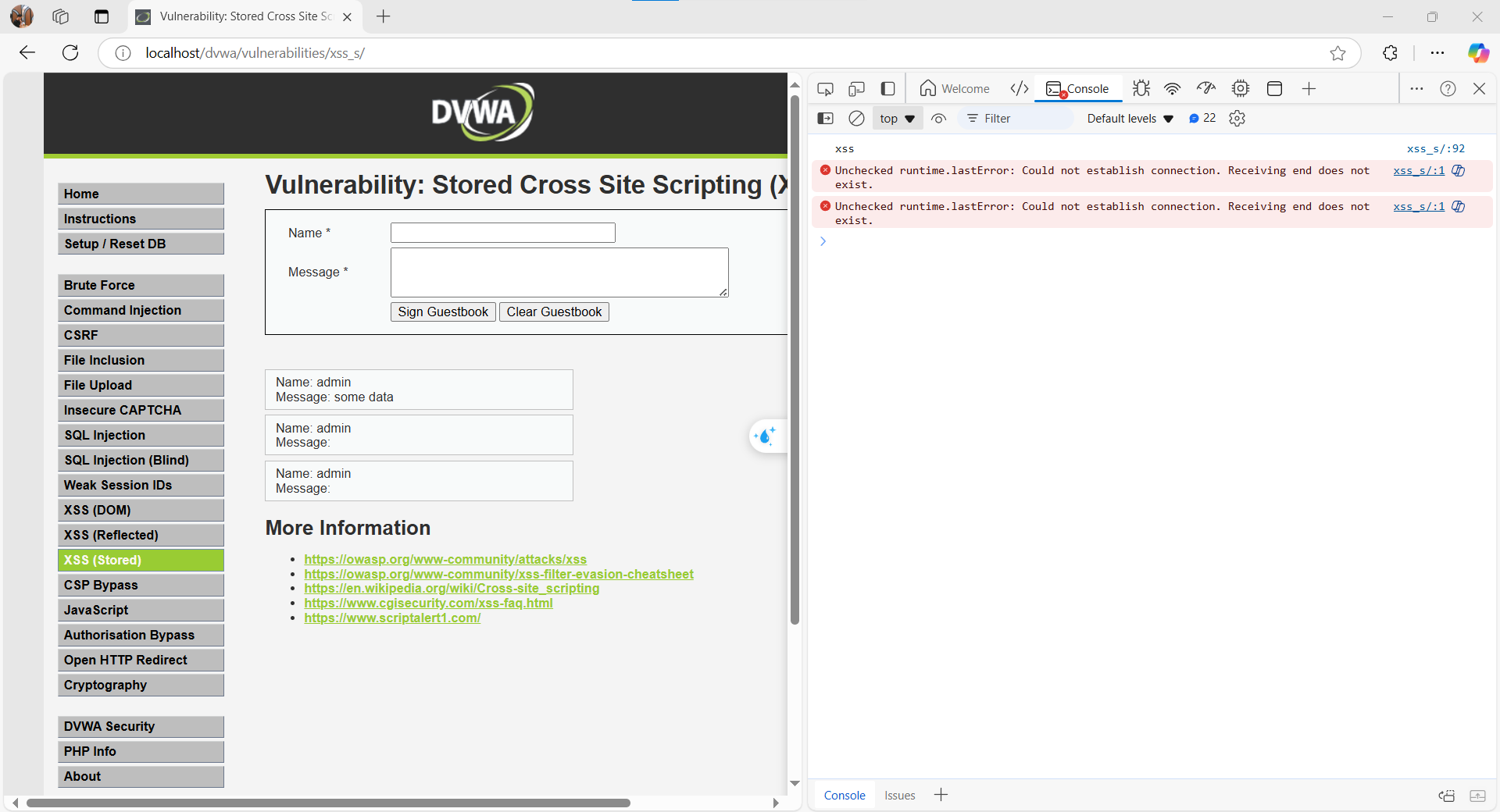
* + - 1. XSS Store

Ta thực hiện ghi vào trong guestbook giá trị của message là ***<script>alert("xss")</script>*** ta nhận được thông báo xss hiện trên màn hình



Hình 11- thử nghiệm XSS Store khi chưa triển khai Mod Security

Ta thực hiện ghi vào trong guestbook giá trị của message là ***<script>console.log("xss")</script>*** ta nhận được dòng chữ xss hiện trên cửa sổ console của trang web



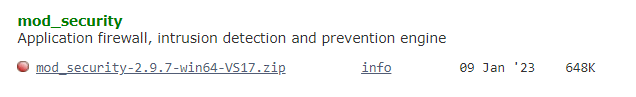
Hình 12- thử nghiệm XSS Store khi chưa triển khai Mod Security

Với một vài thao tác đơn giản, chúng ta đã có thể thấy được lỗi SQL Injection và XSS rất dễ dàng khai thác. Vì vậy cần có phương pháp để ngăn ngừa các dạng tấn công này.

* 1. Triển khai Mod Security
     1. Cấu hình Mod Security với CRS của OWASP

Nhóm em sử dụng Mod Security kết hợp với core ruler set của owasp. OWASP CRS là một bộ quy tắc phát hiện tấn công chung để sử dụng với Mod Security hoặc tường lửa ứng dụng web tương thích. Nó nhằm mục đích bảo vệ các ứng dụng web khỏi nhiều loại tấn công, bao gồm OWASP Top Ten , với tối thiểu các cảnh báo sai. CRS cung cấp khả năng bảo vệ chống lại nhiều loại tấn công phổ biến, bao gồm SQL Injection, Cross Site Scripting, Local File Inclusion, v.v

Truy cập vào <https://www.apachelounge.com/download/> và tải về mod\_security tương ứng



Hình 13- tải về Mod Security

Sau khi dowload ta giải nén và đọc file readme.txt và làm theo các bước sau:

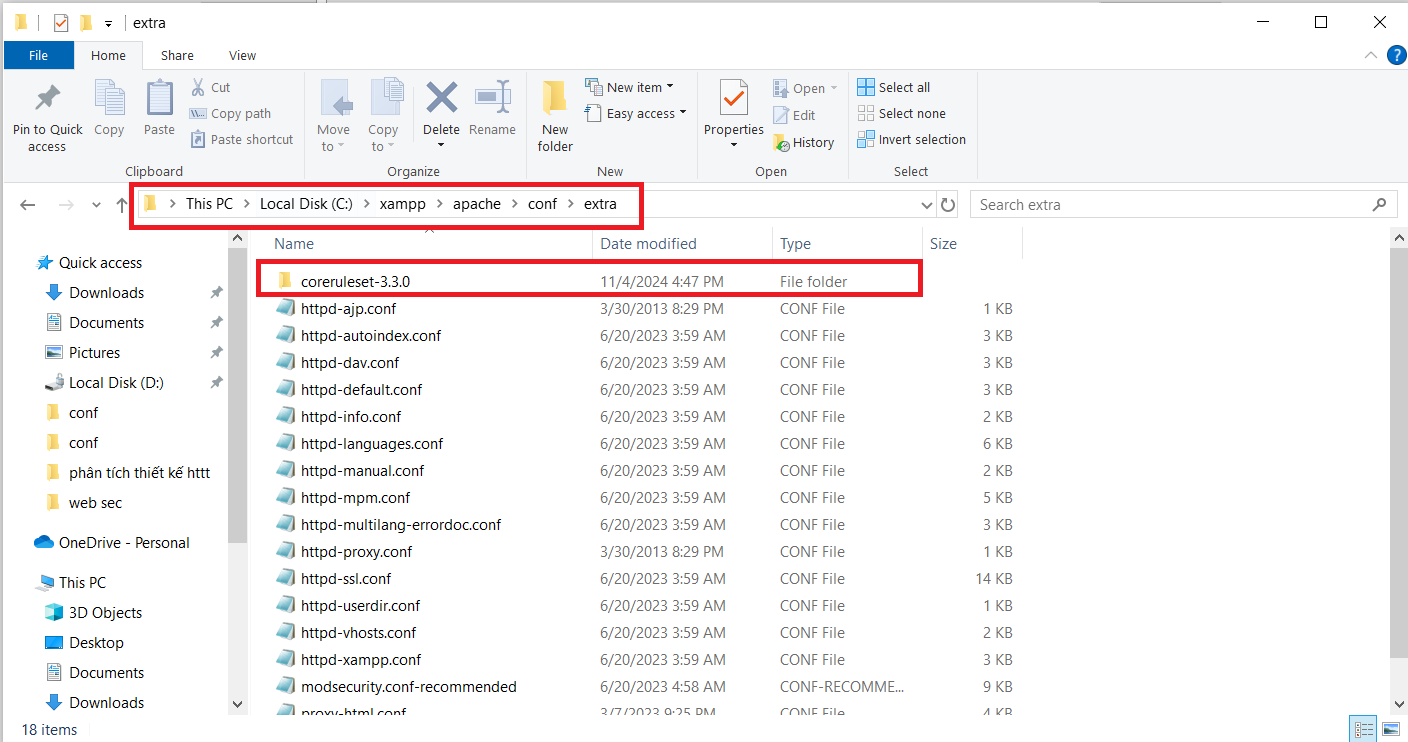
* Copy mod\_security2.so vào *C:\xampp\apache\modules*
* Copy ***yajl.dll*** vào *C:\xampp\apache\bin*
* Mở file ***httpd.conf*** của Xampp và thêm vào dòng sau:

***LoadModule security2\_module modules/mod\_security2.so***

* Bỏ comment trước dòng:

***LoadModule unique\_id\_module modules/mod\_unique\_id.so***

* Truy cập github <https://github.com/coreruleset/coreruleset> và tải về ruleset, đặt vào trong đường dẫn sau của Xampp: *C:\xampp\apache\conf\extra*



Hình 14- Cài đặt coreruleset

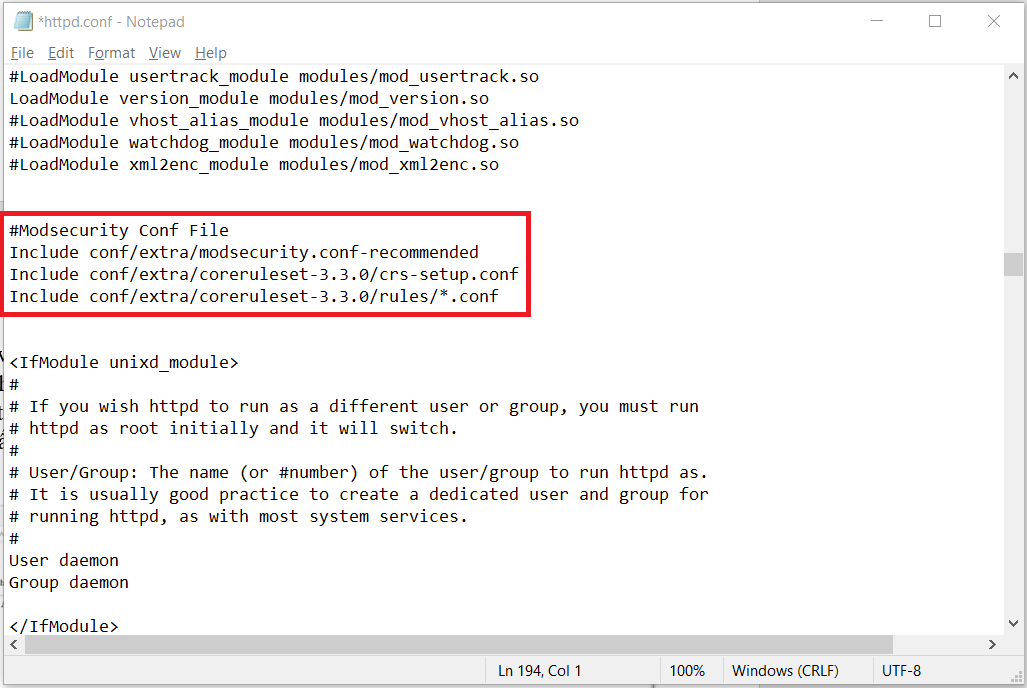
Ta tải file Mod Security.conf-recommended từ [https://raw.githubusercontent.com/SpiderLabs/Mod Security/v2/master/Mod Security.conf-recommended](https://raw.githubusercontent.com/SpiderLabs/ModSecurity/v2/master/modsecurity.conf-recommended) và đặt vào trong đường dẫn *C:\xampp\apache\conf\extra*

Sửa nội dung trong file ***Mod Security.conf-recommended*** như sau:

* Comment lại dòng ***SecRuleEngine DetectionOnly***
* Thêm dòng code sau và tạo file ***modsec\_audit.log*** tại đường dẫn *C:\var\log* để ghi lại log của Mod Security:
* **SecAuditLogType Serial**
* **SecAuditLog /var/log/modsec\_audit.log**

Sửa file ***httpd.conf*** thêm dòng config sau:

* **Include conf/extra/Mod Security.conf-recommended**
* **Include conf/extra/coreruleset-3.3.0/crs-setup.conf**
* **Include conf/extra/coreruleset-3.3.0/rules/\*.conf**

****

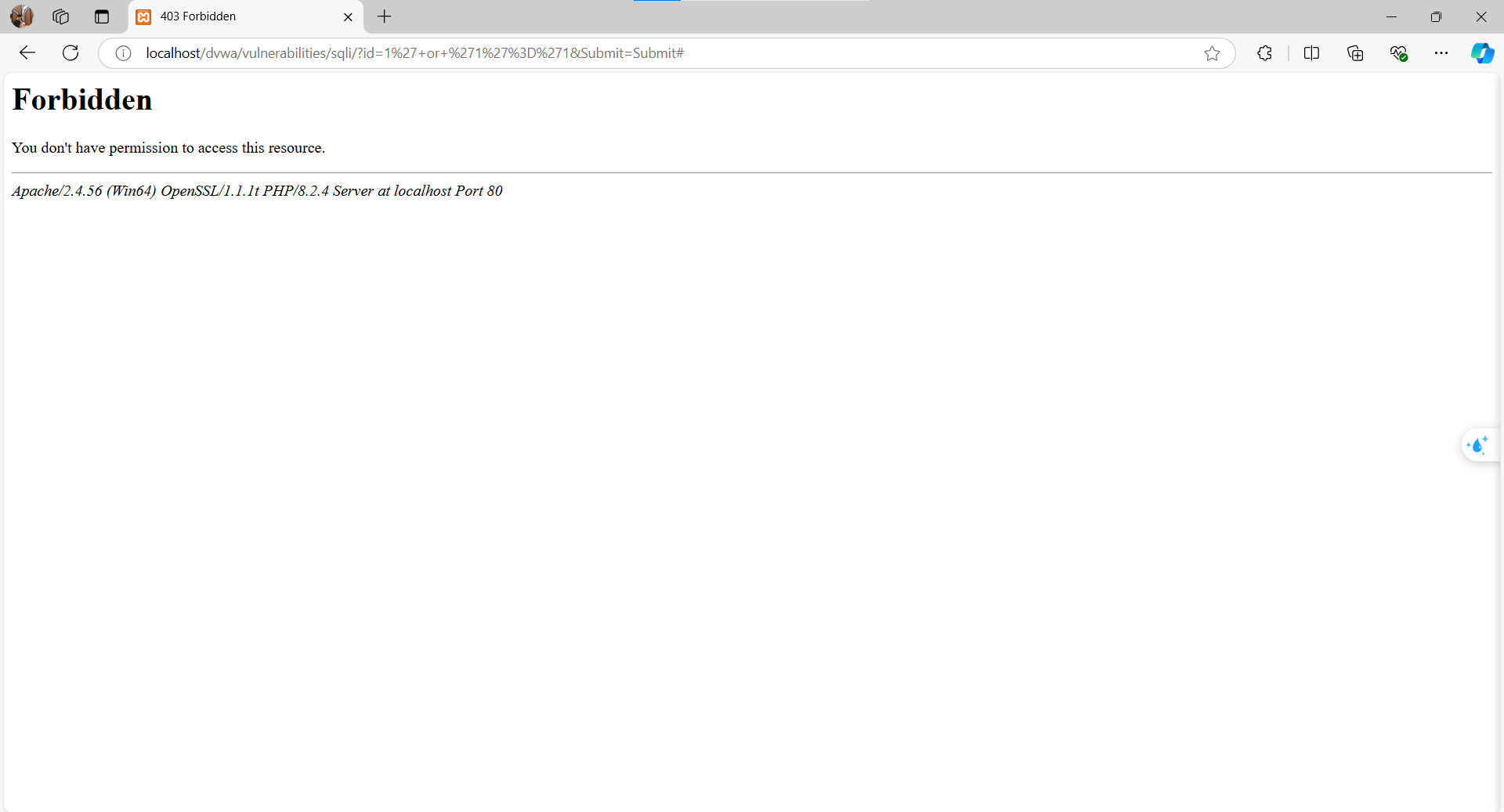
Hình 15- Cài đặt Mod Security Config

Như vậy là đã thành công cấu hình xong Mod Security với Core Ruler Set

Tiến hành restart lại Xampp Apache. Thử nghiệm lại tấn công SQL Injection và XSS như ở phần trước đã tấn công.

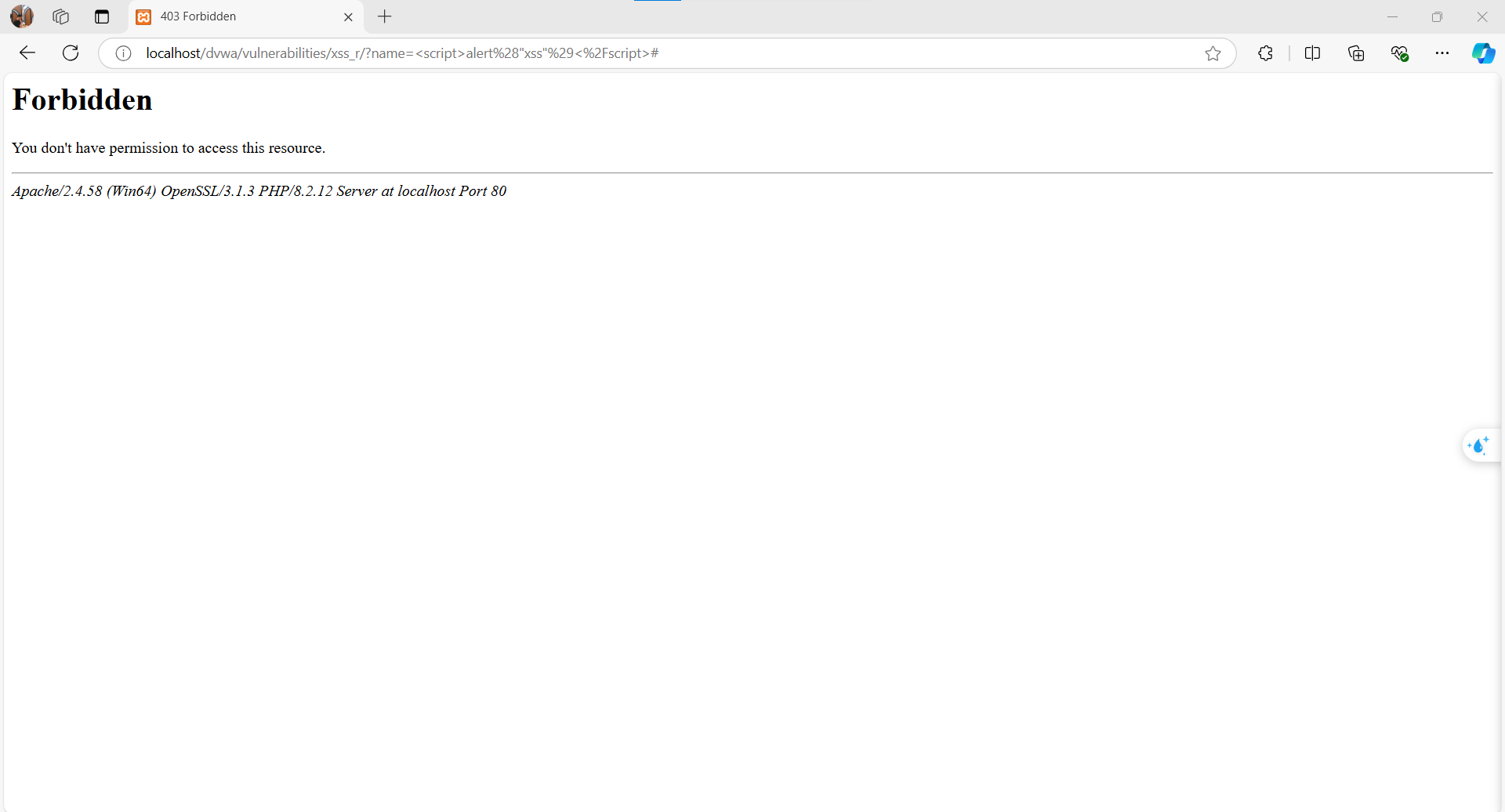
* + 1. Thực hiện tấn công lại vào trang web

Thực hiện tấn công SQLi vào trang web, ta nhận được thông báo không thể truy cập và yêu cầu bị chặn bởi Mod Security

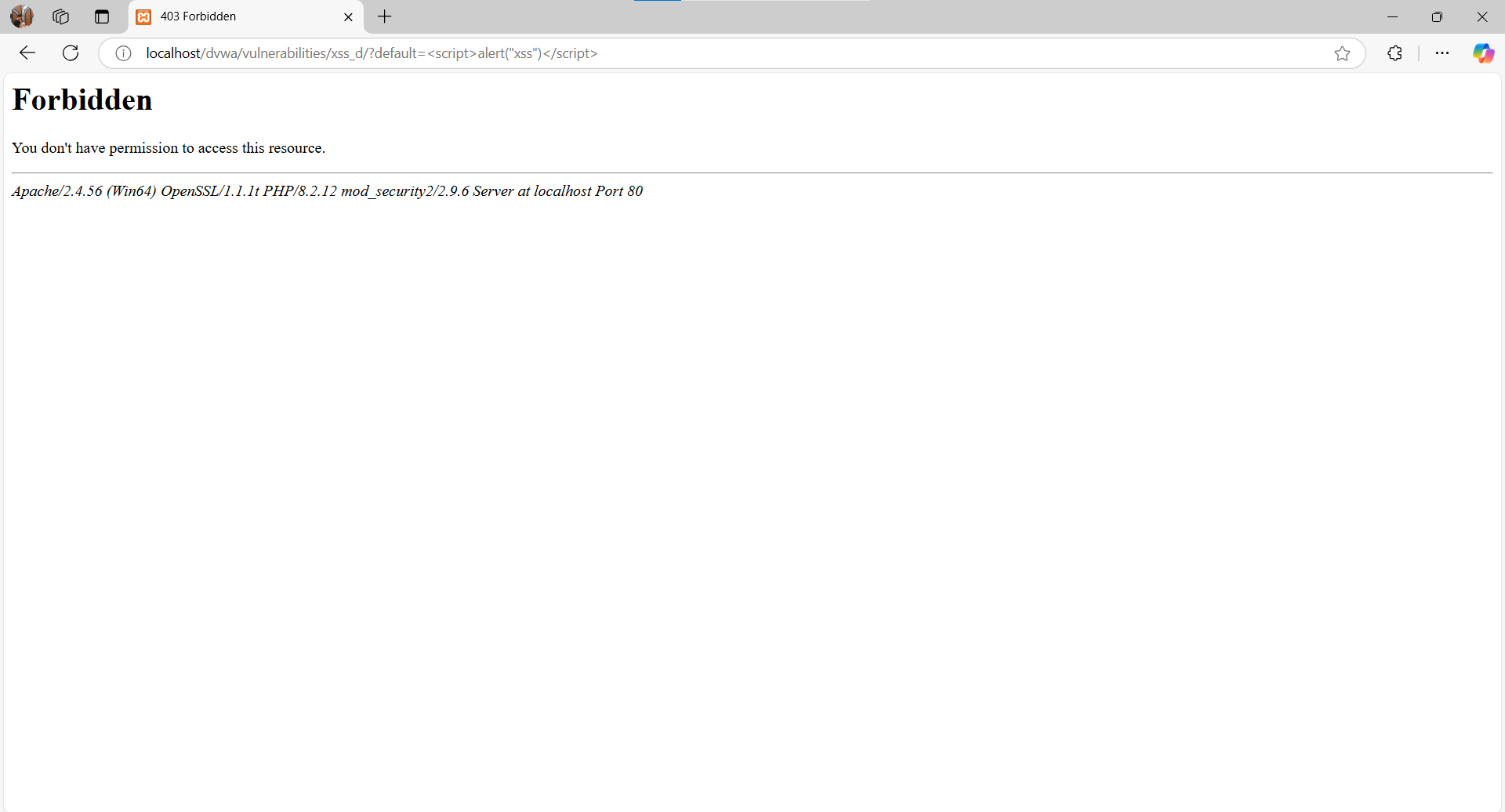


Hình 16- Thử nghiệm tấn công SQLi khi đã cài đặt Mod Security

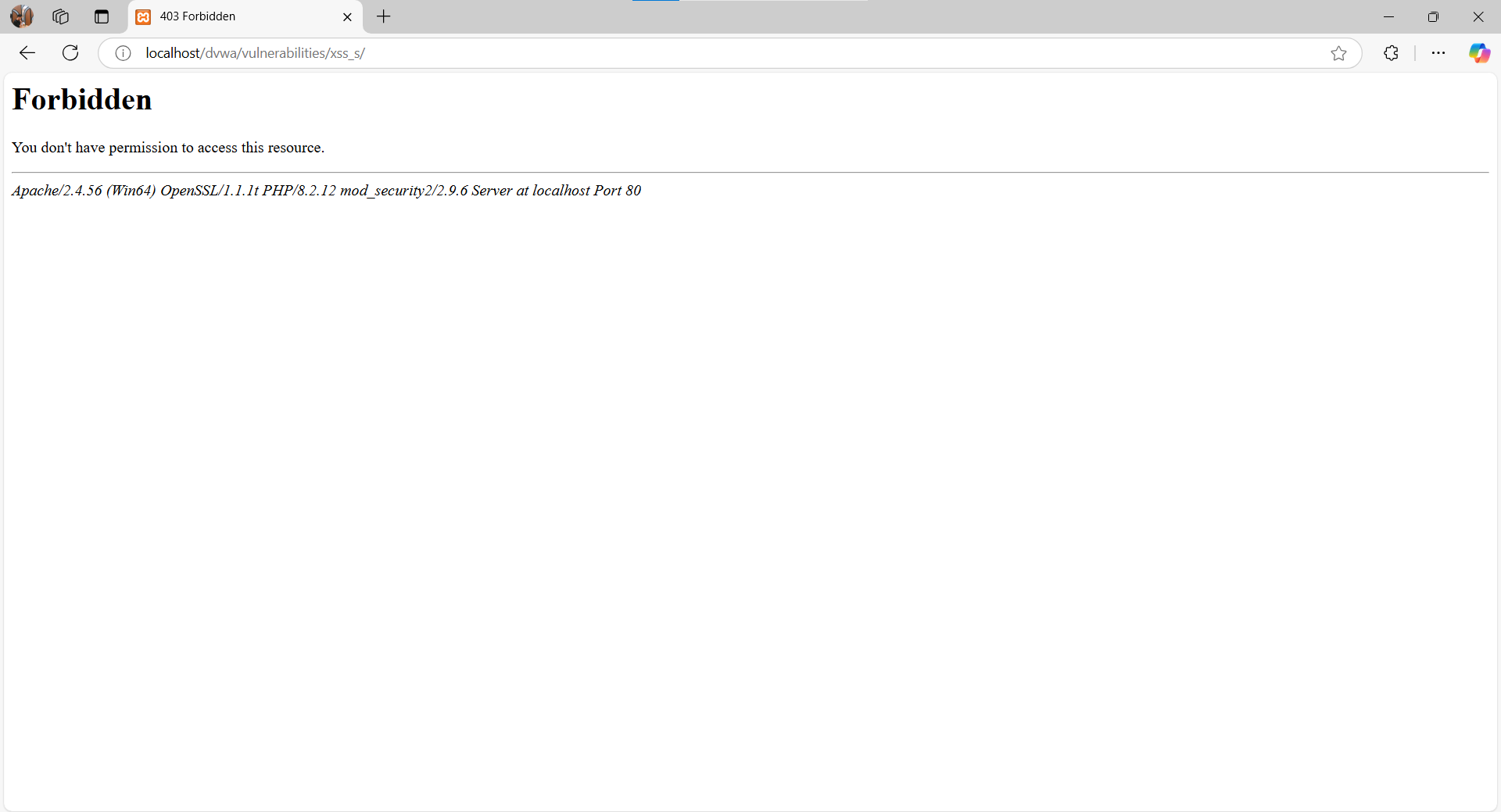
Thực hiện tấn công vào XSS DOM, XSS Reflected, XSS Store cũng có kết quả tương tự, đều nhận được thông báo không thể truy cập và yêu cầu bị chặn bởi Mod Security



Hình 17- Thử nghiệm tấn công XSS Reflected khi đã cài đặt Mod Security



Hình 18- Thử nghiệm tấn công XSS DOM khi đã cài đặt Mod Security



Hình 19- Thử nghiệm tấn công XSS Store khi đã cài đặt Mod Security

Kiểm tra file modsec\_audit.log, ta nhận được các thông báo lỗi như sau:



Hình 20- Kiểm tra log của Mod Security

* 1. Thử nghiệm bypass qua Mod Security

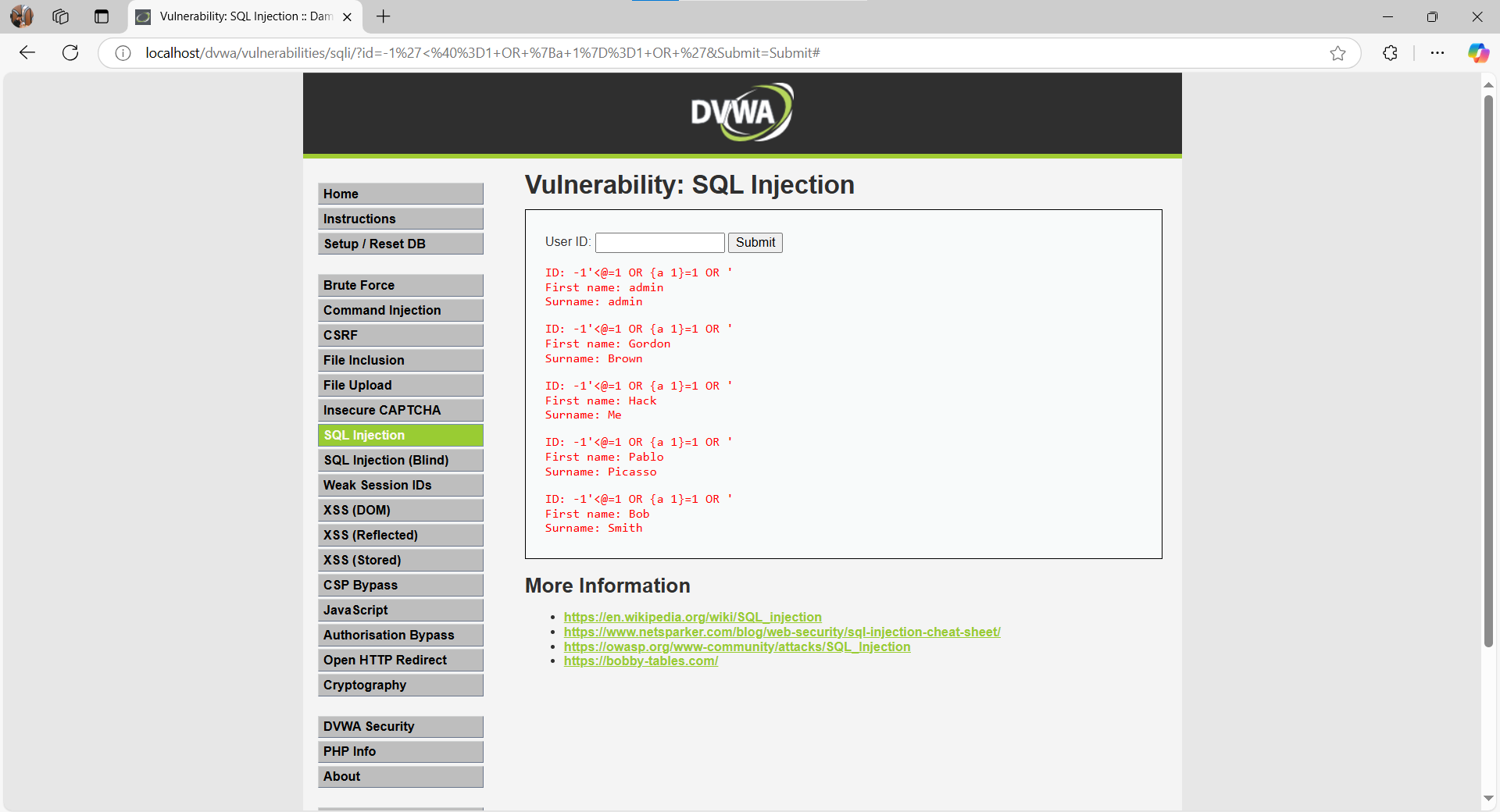
Tuy Mod Security là một tưởng lửa được sử dụng để chặn những cuộc tấn công phổ biến, nhưng vì dựa trên những luật được người quản trị cấu hình và cài đặt nên vì thế vẫn có thể bypass được tường lửa này. Nhóm em trong quá trình nghiên cứu đã tìm ra một số biện pháp để có thể vượt qua Mod Security để khai thác XSS và SQLi.

Với tấn công SQLi, nhóm em đã tìm ra những câu lệnh sau để có thể vượt qua Mod Security để tấn công:

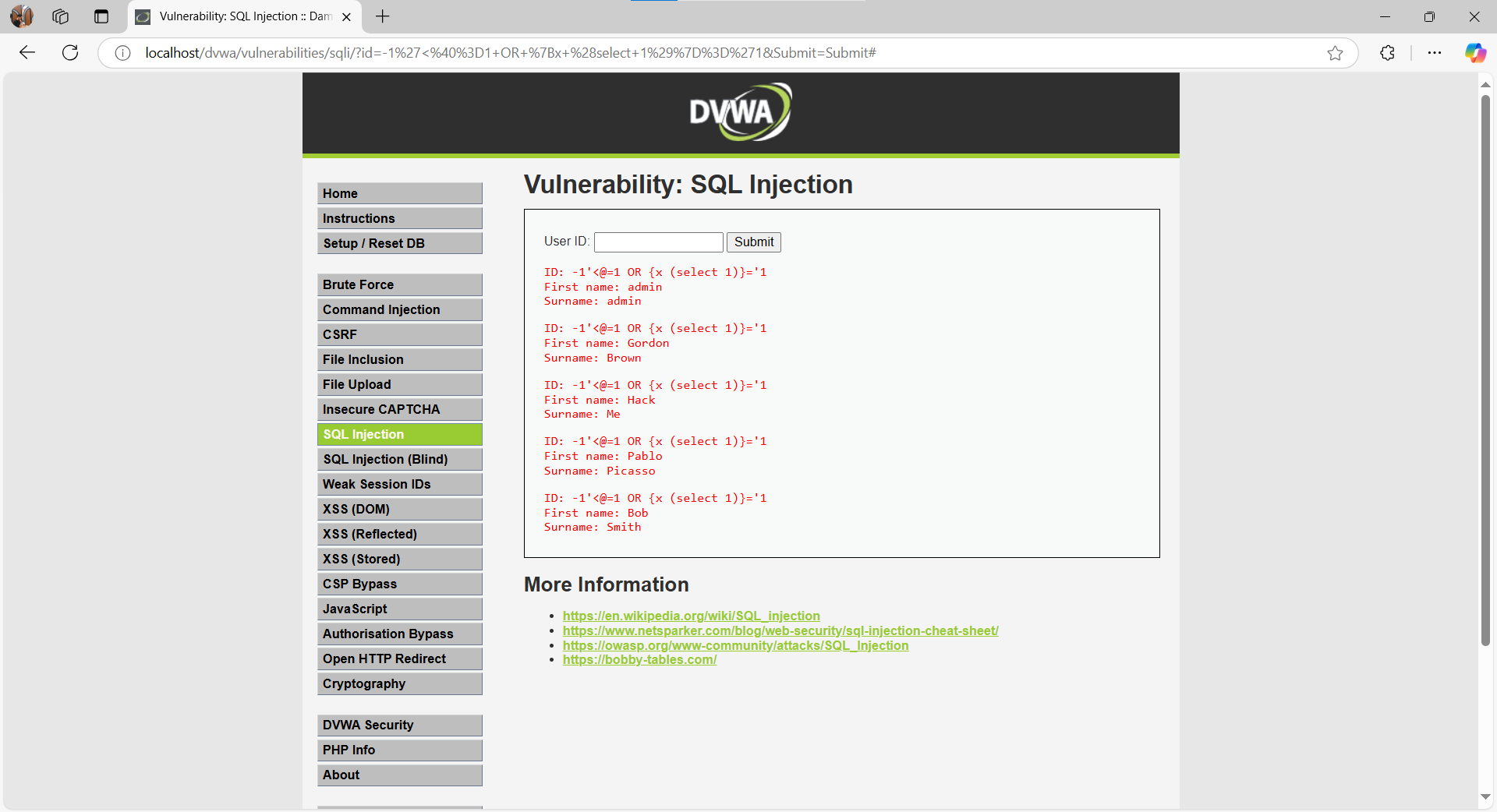
***-1'<@=1 OR {a 1}=1 OR '***

***-1'<@=1 OR {x (select 1)}='1***

Kết quả của các câu lệnh trên là đều thành công khai thác lỗ hổng SQLi và vượt qua sự ngăn chặn của Mod Security



Hình 21- Tấn công SQLi bypass qua Mod Security



Hình 22- Tấn công SQLi bypass qua Mod Security

Với tấn công XSS, nhóm em đã tìm ra được một số câu lệnh để khai thác như sau:

* ***<a href=jav&#x0D;ascript&colon;\u0061lert&#x28;document.domain&#x29;>XSS</a>***
* ***<a href=jav&#x0D;ascript&colon;\u0065val(\u0061tob("YWxlcnQoZG9jdW1lbnQuY29va2llKQ=="));>XSS</a>***

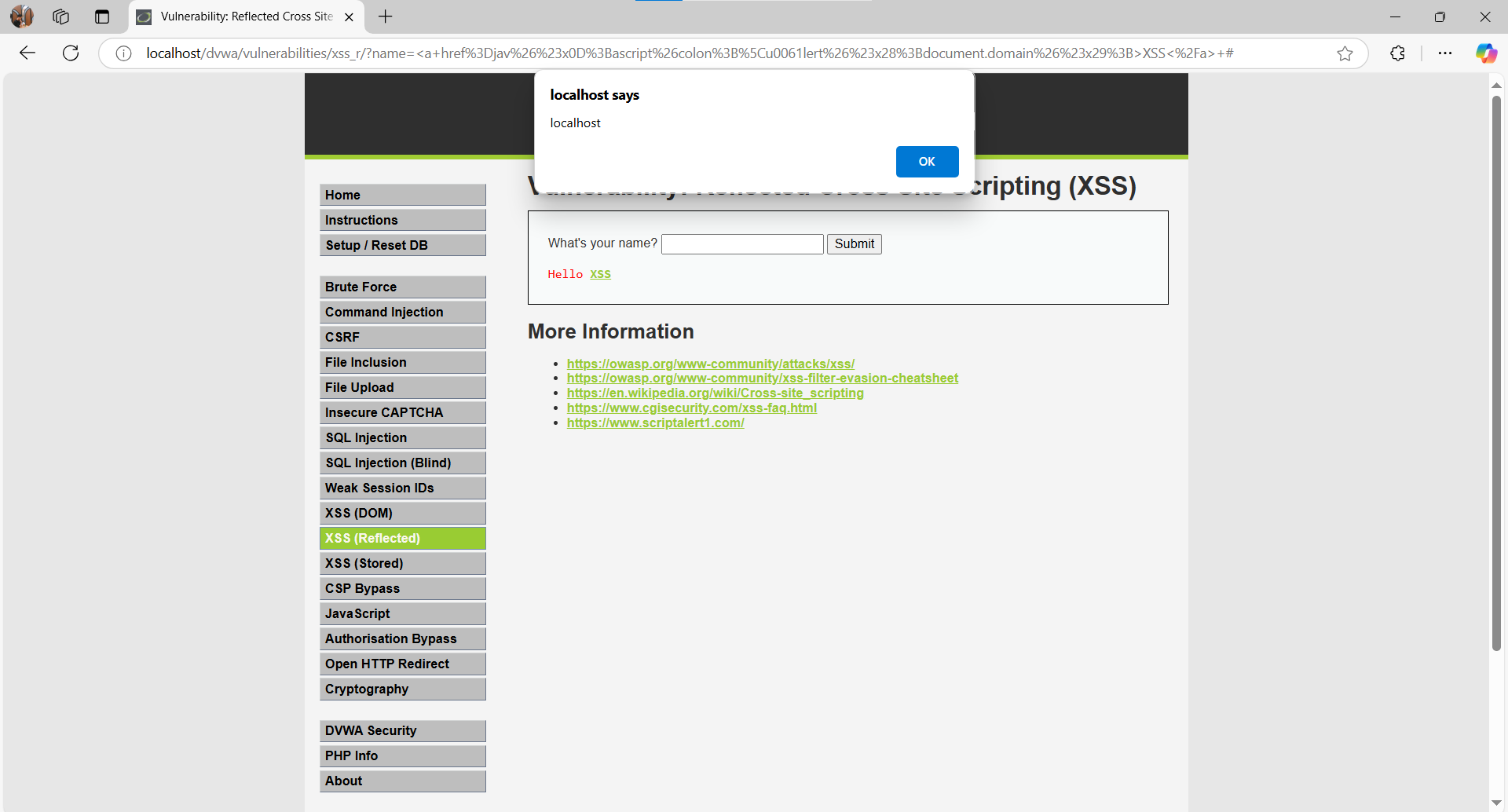
Cách tấn công trên hoạt động như sau:

* &#x0D; là mã hex đại diện cho ký tự "Carriage Return" (CR, ký tự dòng mới).
* &colon; là mã ký tự HTML cho dấu hai chấm (:).
* \u0061 là mã unicode đại diện cho ký tự ‘a’
* \u0065 là mã unicode đại diện cho ký tự ‘e’
* &#x28; là một thực thể ký tự HTML đại diện cho dấu ngoặc mở (
* &#x29; là một thực thể ký tự HTML đại diện cho dấu ngoặc đóng )
* YWxlcnQoZG9jdW1lbnQuY29va2llKQ== là mã base64 của giá trị document.cookie

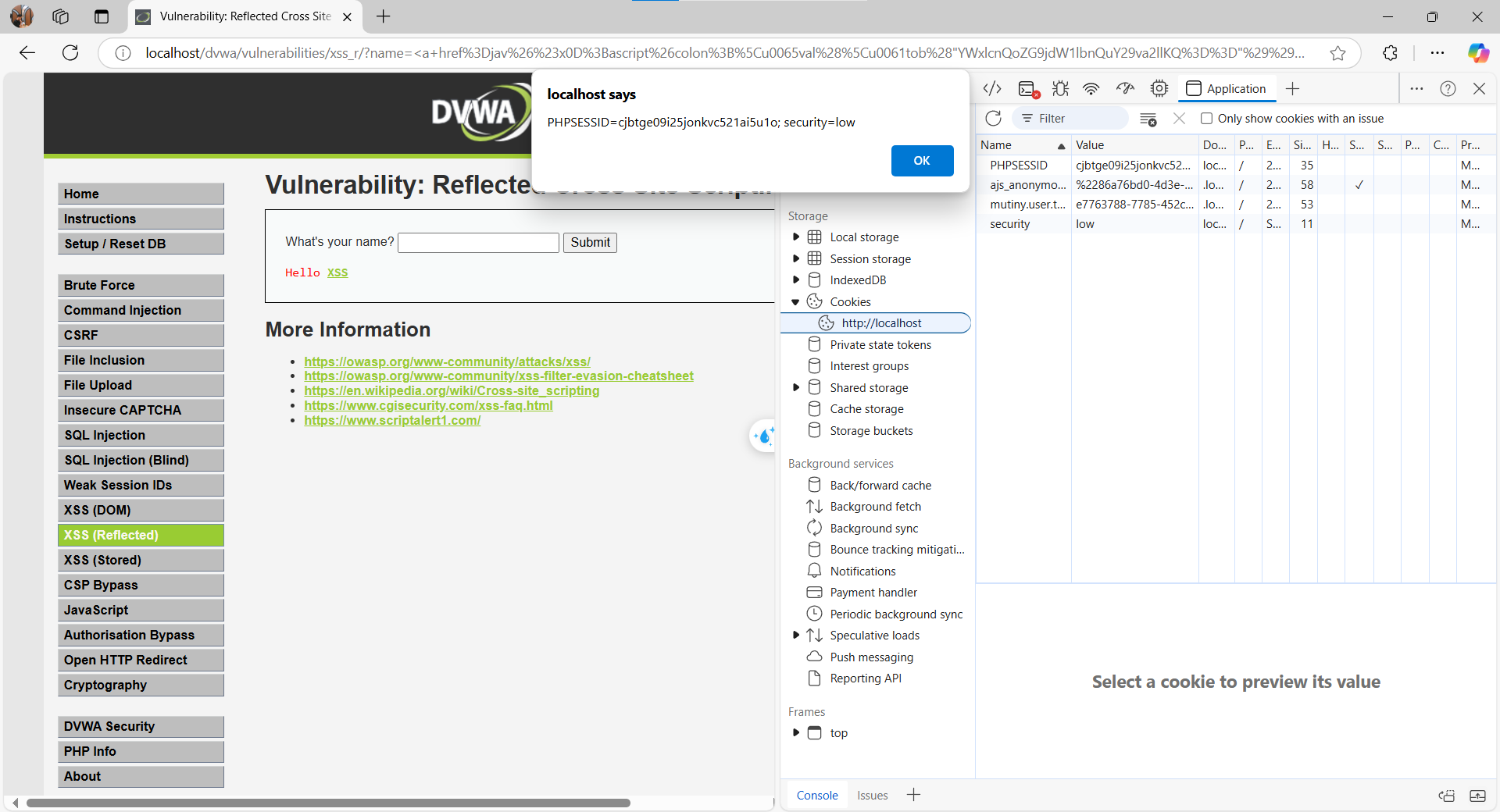
Dựa vào các cách viết khác của các ký tự và nội dung, câu lệnh đã thay đổi, sau khi được biên dịch và thực thi thì 2 câu lệnh trên được viết lại như sau

* ***<a href=javascript:alert(document.domain)>XSS</a>***
* ***<a href=javascript:eval(atob("YWxlcnQoZG9jdW1lbnQuY29va2llKQ=="));>XSS</a>***

Điểu này làm khi người dùng click vào XSS sẽ hiển thị domain hoặc cookie của người dùng. Kết quả khi chạy lệnh trên trong XSS Reflected như sau:



Hình 23- Tấn công XSS bypass qua Mod Security



Hình 24- Tấn công XSS bypass qua Mod Security

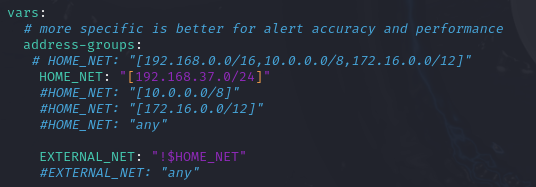
* 1. Thử nghiệm cài đặt hệ thống IDS hỗ trợ Mod Security phát hiện tấn công SQLi
     1. Cài đặt và cấu hình IDS (Suricata)

Ở đây nhóm sẽ cài đặt và cấu hình Suricata trên 1 máy Kali linux chạy trên chế độ IDS để theo dõi, phát hiện các cuộc tấn công SQLi vào trang web DVWA.

Chạy câu lệnh ***sudo apt install suricata -y*** để cài đặt suricata

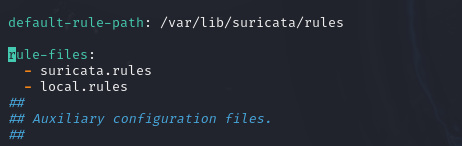
Sau khi cài đặt xong, chúng ta chỉnh sửa file cấu hình suricata bằng câu lệnh ***sudo nano /etc/suricata/suricata.yaml***

* Sửa đổi phần *HOME\_NET* thành dải địa chỉ của trang web DVWA

******

Hình 25- Cấu hình dải mảng trong Suricata

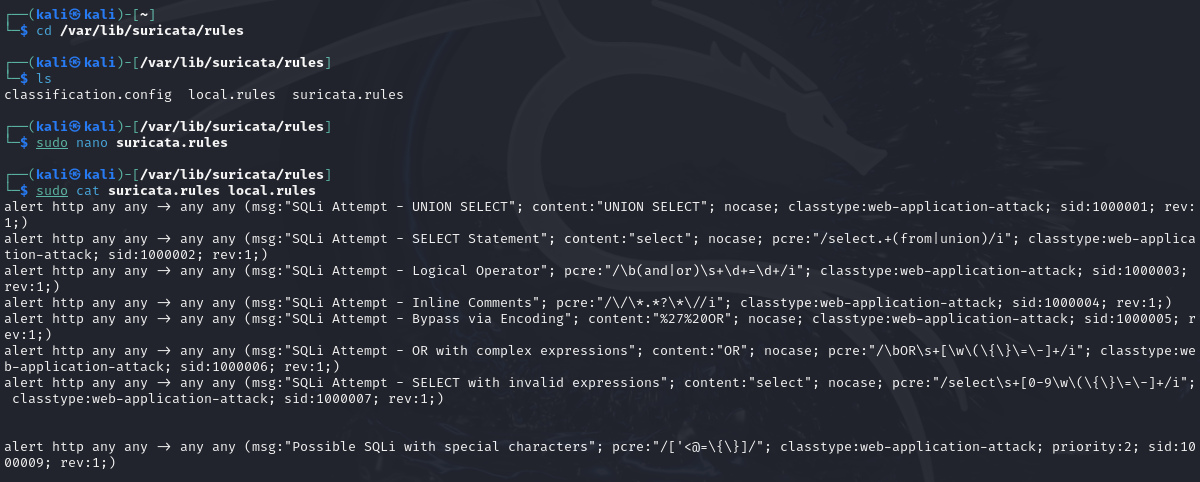
* Phần *rule-files* chỉnh sửa thêm các luật hỗ trợ phát hiện các cuộc tấn công

******

Hình 26- Cấu hình luật Suricata

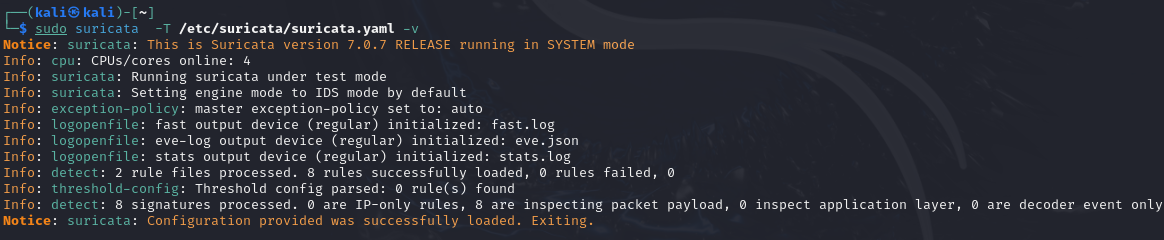
* Có thể sử dụng câu lệnh ***sudo suricata update*** để xem và cập nhật các luật mới nhất đến từ Emerging Threat và các nguồn khác

Sau đó, ta tiến vào thư mục ***/var/lib/suricata/ru***les để xem các luật, ở đây nhóm chỉ phát hiện tấn công SQLi bypass mod Security cơ bản nên chỉ đưa ra 2 file luật là ***suricata.rules*** và ***local.rules***



Hình 27- Thêm/sửa các luật Suricata

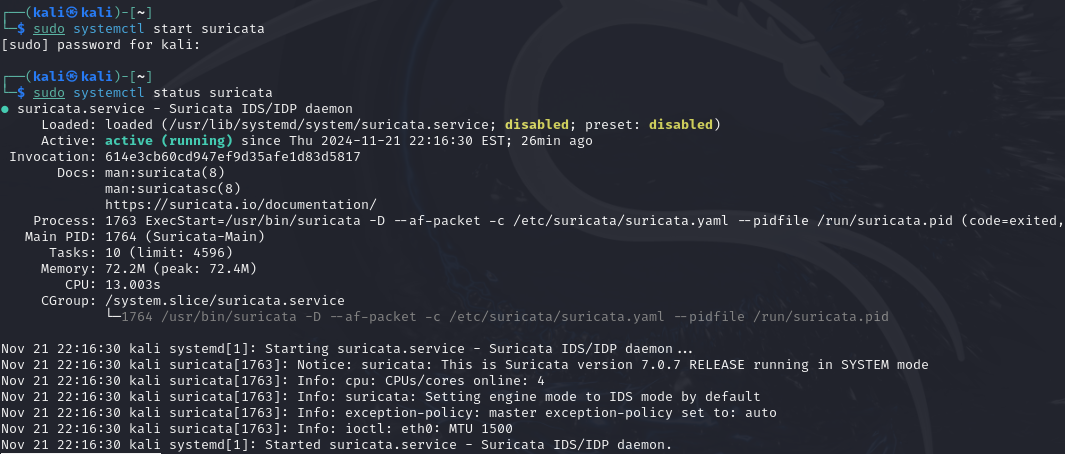
Ta có thể chạy lệnh ***sudo suricata -T /etc/suricata/suricata/yaml -v*** để xem lại các bước cấu hình và luật có gặp lỗi gì không



Hình 28- Kiểm tra cấu hình Suricata

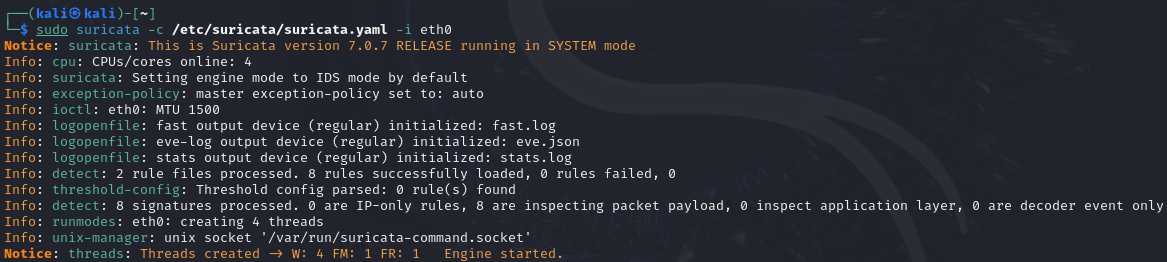
* + 1. Thử nghiệm

Đầu tiên ta khởi động suricata bằng lệnh ***sudo systemctl start suricata***



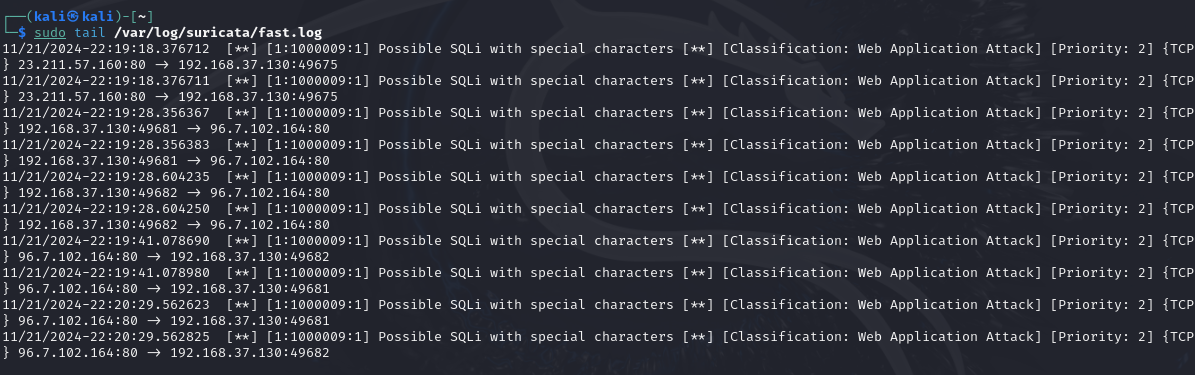
Hình 29- Khởi động Suricata

Tiếp đến, ta dùng lệnh ***sudo suricata -c /etc/suricata/suricata.yaml -i eth0*** để tiến hành giám sát



Hình 30- Sử dụng Suricata đẻ giám sát

Sau đó, ta vào DVWA và tiến hành chạy các cậu lệnh SQL bypass Mod Security mà ta đã thử nghiệm ở phần trên. Vì đang chạy ở chế độ IDS nên Suricata không thể ngăn chặn các câu lệnh này, nhưng ta đã cấu hình luật để nó đưa ra cảnh báo khi bị tấn công SQLi. Ta có thể xem log của suricata để tìm ra chúng



Hình 31- Kiểm tra log Suricata để phát hiện tấn công SQLi

* 1. Kết luận

Chỉ một vài bước đơn giản, chúng ta đã có thể chống lại tấn công SQL Injection và XSS mà không cần phải chỉnh sửa lại mã nguồn của trang web. Bằng việc kết hợp Mod Security và các bộ rules do tổ chức OWASP để chống lại nhiều loại tấn công phổ biến vào ứng dụng web. Cũng như có thể viết ra các rules phù hợp với mục đích sử dụng của riêng từng cá nhân và công ty. Chi phí đầu tư thấp, được hỗ trợ rộng rãi bởi cộng đồng bảo mật vì vậy giải pháp sử dụng Mod Security là một lựa chọn rất tốt trong việc bảo mật cho ứng dụng web. Tuy nhiên Mod Security vẫn có thể vượt qua vì Mod Security chỉ sử dụng luật để nhận diện các cuộc tấn công và kẻ tấn công có thể dựa vào luật đó để có thể vượt qua cuộc kiểm tra và tấn công được vào trang web nên cần các hệ thống khác như IDS/IPS để hỗ trợ.

kẾT LUẬN

Tường lửa ứng dụng web (WAF) đóng vai trò bảo vệ các lỗ hổng khỏi bị khai thác bằng cách cung cấp một lớp bảo mật mà tường lửa mạng thông thường không thể đạt được. WAF giải quyết vấn đề bằng cách cung cấp phương tiện lọc lưu lượng mạng trong khi vẫn cho phép các ứng dụng kết nối trực tiếp với internet. Thay vì tạo ra một bức tường giữa các tài nguyên mạng bên trong và bên ngoài, WAF hoạt động giống như những tấm màn, cho phép lưu lượng thân thiện đi qua nhưng chặn lưu lượng độc hại.

WAF là 1 phần không thể thiếu trong mô hình bảo mật, tuy rất cần thiết nhưng chỉ sử dụng mình WAF là chưa đủ, cần có thêm nhiều biện pháp bảo vệ nhiều lớp để đảm bảo an toàn cho ứng dụng web.

TÀI LIỆU THAM KHẢO

1. WAF introduction, <https://www.vmware.com/topics/web-application-firewall#an-introduction-to-a-web-application-firewall-or-waf>, truy cập tháng 11, 2023
2. OWASP WAF, <https://owasp.org/www-community/Web_Application_Firewall>, truy cập tháng 11, 2023
3. Cloudfare WAF, <https://www.cloudflare.com/learning/ddos/glossary/web-application-firewall-waf/>, truy cập tháng 11, 2023
4. BÀI GIẢNG AN TOÀN ỨNG DỤNG WEB VÀ CƠ SỞ DỮ LIỆU 2021, Hoàng Xuân Dậu