**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO**

**NGHIÊN CỨU KHOA HỌC SINH VIÊN NĂM 2024**

**Đề tài:**

**Nghiên cứu giải pháp bảo mật ứng dụng web trong doanh nghiệp sử dụng WAF (Web Application Firewall).**

|  |  |
| --- | --- |
| **Sinh viên thực hiện:** | **Phạm Minh Trung** |
|  | **Đỗ Thị Linh** |
| **Lớp:** | **CNTT K20L** |
| **Giảng viên hướng dẫn:** | **ThS. Nguyễn Anh Chuyên** |

***Thái Nguyên, năm 2024***

**MỤC LỤC**

[MỤC LỤC 1](#_Toc174960614)

[DANH MỤC BẢNG 3](#_Toc174960615)

[LỜI NÓI ĐẦU 4](#_Toc174960616)

[LỜI CẢM ƠN 5](#_Toc174960617)

[CHƯƠNG I: TỔNG QUAN VỀ ỨNG DỤNG WEB VÀ CÁC GIẢI PHÁP BẢO ĐẢM AN TOÀN 6](#_Toc174960618)

[1.1. Tổng quan về Website 6](#_Toc174960619)

[1.2. Tổng quan về dịch vụ Website ( Website Service) 7](#_Toc174960620)

[1.2.1. Lợi ích khi sử dụng Web Services 11](#_Toc174960621)

[1.2.2. Ưu và nhược điểm của Web Services 13](#_Toc174960622)

[1.3. Các nguy cơ đối với ứng dụng website 14](#_Toc174960623)

[1.3.1. Tấn công Injection 14](#_Toc174960624)

[1.3.2. Tấn công thực thi các tập tin độc hại 17](#_Toc174960625)

[1.4. Các giải pháp đảm bảo an toàn ứng dụng website 21](#_Toc174960626)

[1.4.1. Kỹ thuật lập trình an toàn 21](#_Toc174960627)

[1.4.2. Bảo vệ hệ thống tệp tin 24](#_Toc174960628)

[1.4.3. Bảo vệ cơ sở dữ liệu 25](#_Toc174960633)

[1.5. Sử dụng các thiết bị và phần mềm bảo mật cho ứng dụng website 27](#_Toc174960644)

[1.5.1. Giải pháp sử dụng tường lửa 28](#_Toc174960645)

[1.5.2. Sử dụng thiết bị phát hiện và phòng chống xâm nhập (IDS/ IPS) 30](#_Toc174960652)

[Kết luận chương I 32](#_Toc174960663)

[CHƯƠNG II: TÌM HIỂU MODSECURITY 33](#_Toc174960664)

[2.1. Mod Security là gì ? 33](#_Toc174960665)

[2.2. Tính năng và vai trò của Mod Security 33](#_Toc174960666)

[2.2.1. Giám sát thời gian thực và phát hiện tấn công 33](#_Toc174960667)

[2.2.2. Chống tấn công và vá lỗi 33](#_Toc174960668)

[2.2.3. Khả năng linh hoạt của Core Rule Set 34](#_Toc174960669)

[2.3. Khi chưa tích hợp Mod Security có thể bị SQL Injection và tấn công XSS 34](#_Toc174960670)

[2.3.1. Thử nghiệm SQL Injection 34](#_Toc174960671)

[2.3.2. Thử nghiệm XSS 35](#_Toc174960672)

[2.4. Cách hoạt động của Mod Security 36](#_Toc174960674)

[2.5. Kiến trúc và thành phần của Mod Security 37](#_Toc174960675)

[2.5.1. Tổng quan về Mod Security core ruler 37](#_Toc174960676)

[2.5.2. Nội dung của core ruler 37](#_Toc174960677)

[2.5.3. Cấu trúc của core ruler 37](#_Toc174960678)

[2.6. Rules 38](#_Toc174960679)

[2.7. Logging 41](#_Toc174960757)

[2.7.1. Debug Log 41](#_Toc174960758)

[2.7.2. Audit logging 42](#_Toc174960759)

[2.7.3. Tùy biến thông tin log 42](#_Toc174960760)

[2.8. Xây dựng chính sách trên Mod Security chống lại một số tấn công 42](#_Toc174960761)

[2.8.1. SQL Injection 42](#_Toc174960762)

[2.8.2. XSS Attack 43](#_Toc174960763)

[CHƯƠNG III: TRIỂN KHAI HỆ THỐNG WAF MODSECURITY 45](#_Toc174960764)

[3.1 Cài đặt Mod Security 45](#_Toc174960765)

[3.2.Thử nghiệm phòng chống tấn công với ModSecurity 47](#_Toc174960772)

[3.2.1. Thử nghiệm phòng chống tấn công SQL Injection 47](#_Toc174960773)

[3.2.2. Thử nghiệm phòng chống tấn công XSS 49](#_Toc174960781)

**DANH MỤC BẢNG**

[Bảng 1: Các ký tự tên mã hóa để ngăn chặn tấn công XSS 43](#_Toc174960599)

[Bảng 2: Các script XSS và biểu thức chính quy 44](#_Toc174960600)

# LỜI NÓI ĐẦU

Ngày nay, ứng dụng web trong doanh nghiệp và cơ quan chính phủ phải đối mặt với hai thách thức lớn là: giảm thiểu nguy cơ bảo mật và bảo đảm quy trình trong công nghiệp hoặc những quy định chính phủ. May mắn thay khi tồn tại một giải pháp an toàn thông tin sẵn sàng hỗ trợ các tổ chức CNTT đạt được cả hai tiêu chí trên tại cùng một thời điểm. OWASP cho phép các chuyên gia an ninh CNTT giảm thiểu được các cuộc tấn công bằng các chủ động và liên tục củng cố các cấu hình an ninh của OS, ứng dụng web và Web Application Firewall. Đồng thời, các dự án thuộc chuẩn OWASP cho phép các kiểm soát viên giám sát việc tuân thủ các chính sách bắt buộc trong tổ chức, doanh nghiệp.

ModSecurity là một sản phẩm thuộc dự án OWASP, cho phép người dùng cấu hình tùy chỉnh các phương thức phát hiện tấn công vào web server. Phiên bản ModSecurity hiện tại đã hỗ trợ Apache, NginX và IIS. Cùng với dự án ModSecurity Core Rule Set thì việc triển khai hệ thống WAF càng dễ dàng hơn cho nhân viên hệ thống cũng như các chuyên viên bảo mật…

Vì vậy em đã chọn đề tài “Xây dựng **g**iải pháp bảo mật cho hệ thống website sử dụng tường lửa Web Mod\_Security cài đặt và triển khai trên máy chủ Linux server”, một phần mềm mã nguồn mở với nhiều chức năng mạnh mẽ cho phép quản lý các thiết bị, dịch vụ trong hệ thống mạng. Với mục tiêu nghiên cứu, tìm hiểm về giải pháp giúp cho mọi người có cái nhìn tổng quan về một hệ thống giám sát mạng hoàn chỉnh, đồng thời đưa ra một giải pháp cụ thể đối với một hệ thống mạng dành cho doanh nghiệp.

Nội dung nghiên cứu gồm 3 chương với nội dung tóm tắt như sau:

**Chương I:** Trình bày tổng quan về ứng dụng web và các giải pháp đảm bảo an toàn thông tin, cụ thể nêu nổi bật dịch vụ web và các nguy cơ tấn công tới dịch vụ web. Ngoài ra, còn trình bày chuyên sâu về các giải pháp đảm bảo an toàn, sử dụng các thiết bị và phần mềm bảo mật cho ứng dụng web. Và nêu sơ lược về Web Application Firewall.

**Chương II:** Giới thiệu ModSecurity

**Chương III:** Kịch bản Demo

# LỜI CẢM ƠN

Để hoàn thành nghiên cứu khoa học, trước hết chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến giảng viên, người đã tận tình hướng dẫn chúng em trong suốt quá trình nghiên cứu vừa qua. Trong thời gian được thầy hướng dẫn, chúng em không những tiếp thu thêm nhiều kiến thức bổ ích mà còn học tập được tinh thần, thái độ làm việc nghiêm túc, hiệu quả. Đây là điều rất cần thiết cho em trong quá trình học tập và làm việc sau này.

Xin kính chúc tất cả mọi người dồi dào sức khỏe và tràn đầy nhiệt huyết trong cuộc sống cũng như công việc của mình.

|  |  |
| --- | --- |
|  | **NHÓM SINH VIÊN THỰC HIỆN NGHIÊN CỨU** |

# CHƯƠNG I: TỔNG QUAN VỀ ỨNG DỤNG WEB VÀ CÁC GIẢI PHÁP BẢO ĐẢM AN TOÀN

## **Tổng quan về Website**

Website là tập hợp của rất nhiều trang website - một loại siêu văn bản (tập tin dạng HTML hoặc XHTML) trình bày thông tin trên mạng Internet - tại một địa chỉ nhất định để người xem có thể truy cập vào xem. Trang web đầu tiên người xem truy cập từ tên miền thường được gọi là trang chủ (homepage), người xem có thể xem các trang khác thông qua các siêu liên kết (hyperlinks).

Đặc điểm tiện lợi của Website: thông tin dễ dàng cập nhật, thay đổi, khách hàng có thể xem thông tin sản phẩm ngay tức khắc, ở bất kỳ nơi nào, tiết kiệm chi phí in ấn, gửi bưu điện, fax, thông tin không giới hạn (muốn đăng bao nhiêu thông tin cũng được, không giới hạn thông tin, hình ảnh…) và không giới hạn phạm vi khu vực sử dụng.

Một website thông thường chia làm hai phần: giao diện người dùng (front-end) và các chương trình được lập trình để website hoạt động (backend). Giao diện người dùng là định dạng trang web được trình bày trên màn hình của máy tính của người xem được xem bằng các phần mềm trình duyệt web như Internet Explorer, FireFox,…Tuy nhiên người xem ngày nay có thể xem website từ các thiết bị điện tử khác như: điện thoại di động, PDA,…Việc trình bày website phải đảm bảo các yếu tố về thẩm mỹ đẹp, ấn tượng, bố cục đơn giản, để dễ hiểu và dễ sử dụng, các chức năng tiện lợi cho người xem. Đặc biệt ngày nay, website trở nên sống động với những hiệu ứng đa dạng của hình ảnh và chữ kết hợp kết hợp với âm thanh. Phần backend là phần lập trình của website lưu trữ trên máy chủ (Server). Sự khác nhau ở phần lập trình backend của website làm phân ra hai hai loại website: website tĩnh và website động.

Website động (Dynamic website) là website có cơ sở dữ liệu, được cung cấp công cụ quản lý website (Admin Tool) để có thể cập nhật thông tin thường xuyên, quản lý các thành phần trên website. Loại website này thường được viết bằng các ngôn ngữ lập trình PHP, ASP.NET, Ruby hoặc Python,…Quản trị cơ sở dữ liệu bằng SQL hoặc MySQL.

**Website tĩnh (Static Website)** là thuật ngữ được sử dụng để chỉ những trang web có nội dung cố định và ít thay đổi. Trong web tĩnh là các trang web được tạo ra trước và lưu trữ dưới dạng các tập tin HTML, CSS và các tệp tin tĩnh khác trên máy chủ web. Khi người dùng truy cập vào trang web tĩnh, trình duyệt chỉ cần hiển thị các tệp tin đã được tạo sẵn trên máy chủ, mà không cần phải truy xuất cơ sở dữ liệu hay thực hiện xử lý dữ liệu động. Web tĩnh thường được sử dụng cho các trang web có nội dung không thay đổi thường xuyên, ví dụ về web tĩnh như trang giới thiệu công ty, trang thông tin sản phẩm, trang tin tức không cần cập nhật liên tục. Các trang web tĩnh đơn giản, dễ triển khai và tốn ít tài nguyên máy chủ.

Tuy nhiên, với những trang web đòi hỏi tính tương tác cao, cập nhật thường xuyên hay có nhiều chức năng phức tạp, thì website động thường là sự lựa chọn phù hợp hơn. Hiện nay, việc sử dụng website tĩnh ít phổ biến hơn và thường chỉ tồn tại trong các công ty chuyên về thiết kế website. Công ty này sử dụng website tĩnh vì họ có kiến thức và khả năng dễ dàng chỉnh sửa nội dung khi cần thiết. Với website tĩnh, việc thay đổi nội dung trang web đòi hỏi sự can thiệp trực tiếp vào mã nguồn và cập nhật thủ công từ phía người quản trị. Tuy nhiên, website tĩnh thường không phù hợp cho các trang web đòi hỏi tính tương tác cao, cập nhật thường xuyên và có nhiều chức năng phức tạp. Trong những trường hợp này, website động thường được sử dụng để cung cấp tính linh hoạt hơn trong việc cập nhật và tương tác với người dùng.

## **Tổng quan về dịch vụ Website ( Website Service)**

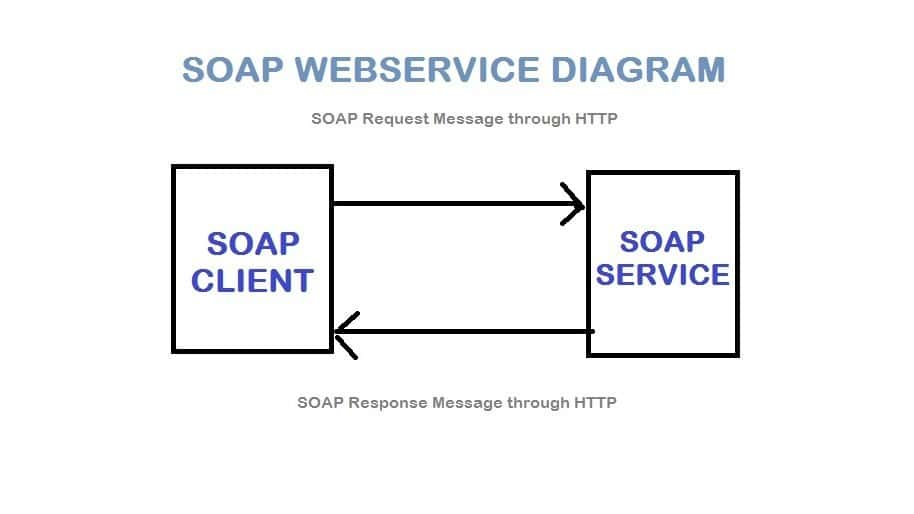
Web Service là tập hợp các giao thức và tiêu chuẩn mở được sử dụng để trao đổi dữ liệu giữa các ứng dụng hoặc giữa các hệ thống.



Hiểu đơn giản, Web Services là sự tích hợp giữa hai máy tính, giúp hai máy tính có thể tương tác hiệu quả với nhau qua mạng. Khi đó, web services sẽ cung cấp các chức năng của nó cho máy khách để người dùng đạt được các mục tiêu sử dụng nhất định. Ví dụ, ứng dụng java có thể tương tác với các ứng dụng Java, .Net và PHP. Vì vậy, web services là một cách độc lập về ngôn ngữ giao tiếp.

Hiện nay, có hai loại Web Services chính: SOAP Web Service và Restful Web Service.

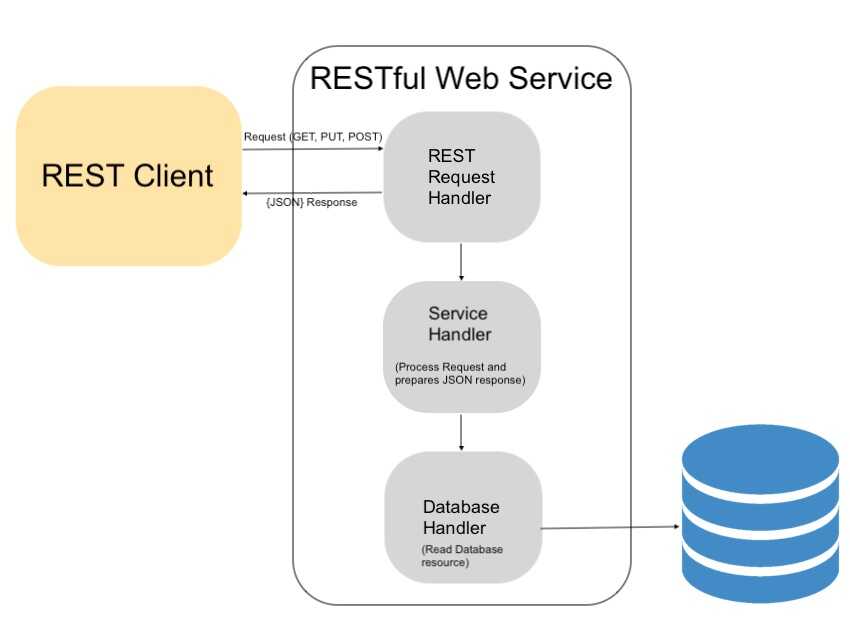
SOAP Web Service, SOAP là viết tắt của Simple Object Access Protocol. Nó là một giao thức dựa trên XML để truy cập các web services.



SOAP được khuyến cáo bởi W3C cho giao tiếp giữa hai ứng dụng.

Đó là nền tảng độc lập và ngôn ngữ độc lập. Vì dựa trên XML nên SOAP là một giao thức không phụ thuộc platform cũng như bất kì ngôn ngữ lập trình nào. Chúng ta có thể viết bằng Java, PHP, .NET, … và triển khai trên Window, Linux,…

**RESTful Web Service,** REST là viết tắt của REpresentational State Transfer, là một loại kiến trúc phần mềm (architectural style), không phải là một protocol.



RESTful web service nhanh vì không có đặc tả nghiêm ngặt như SOAP. Nó chiếm ít băng thông và tài nguyên hơn. Nếu tính theo số dịch vụ mạng sử dụng, REST đã nổi lên trong vài năm qua như là một mô hình thiết kế dịch vụ chiếm ưu thế. RESTful web service cho phép định dạng dữ liệu khác nhau như Plain Text, HTML, XML và JSON.

Nền tảng web services cơ bản là XML HTTP. Tất cả các web services chuẩn đều hoạt động bằng các thành phần sau:

**SOAP (Simple Object Access Protocol):**

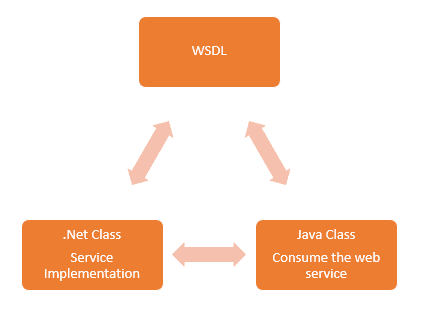
* SOAP là một giao thức dựa trên XML đơn giản cho phép các ứng dụng trao đổi thông tin qua HTTP.
* Sử dụng để gửi thông điệp giữa các ứng dụng với nhau. SOAP đã được thiết kế thông qua qua internet, và nó đơn giản, dễ mở rộng.

**UDDI (Universal Description, Discovery and Integration):**

* UDDI là một tiêu chuẩn dựa trên XML để mô tả, xuất bản và tìm kiếm các dịch vụ web.
* UDDI là một open framework và là nền tảng độc lập. UDDI có thể giao tiếp qua SOAP, CORBA và Giao thức RMI Java.

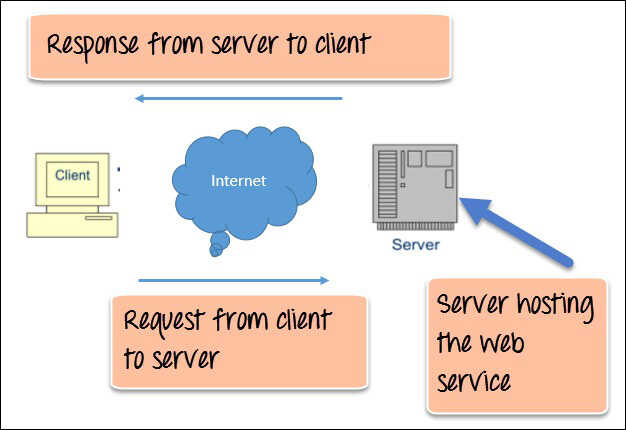
**WSDL (Web Services Description Language) – ngôn ngữ định nghĩa web service.**

* WSDL là một tài liệu xml chứa thông tin về các web services như tên phương thức, tham số phương thức và cách truy cập nó.
* Một WSDL hợp lệ gồm hai phần: phần giao diện (mô tả giao diện và phương thức kết nối) và phần thi hành mô tả thông tin truy xuất CSDL.



Một ứng dụng Web Service bao gồm 2 thành phần: Client và Server giao tiếp với nhau qua giao thức HTTP.

* Client gửi yêu cầu qua các lời gọi hàm thông qua HTTP Request đến Server
* Server gửi các kết quả được thực thi các ở hàm thông qua HTTP Request



**Cách thức hoạt động có thể mô tả như sau:**

Đầu tiên, Applicantion Client cần truy vấn các mẫu tin UDDI theo 1 thông tin nào đó(chẳng hạn tên loại) để xác định WebService cần tìm.

Khi đã xác định được WebService cần cho ứng dụng, Client có thế lấy thông tin về địa chỉ của tài liệu WSDL của WebService này dựa trên mẫu tin UDDI. Client tạo ra một Proxy cục bộ cho dịch vụ từ xa. Proxy chuyển một phương tiện khởi động phương thức của đối tượng thành một thông báo XML và ngược lại.

Tạo thông báo SOAP – Client tạo ra những gói tin SOAP/XML và gửi đến địa chỉ URL được xác định trong tệp WSDL để liên lạc với Server.

SOAP Listener nhận cuộc gọi và diễn dịch

Dịch vụ Web thực hiện các chức năng của mình và trả kết quả về cho client, thông qua listener và proxy.

## **Lợi ích khi sử dụng Web Services**

**Khả năng tương tác**

* Web services cho phép các ứng dụng khác nhau giao tiếp với nhau và chia sẻ dữ liệu, dịch vụ với nhau.
* Nhiều ứng dụng khác cũng có thể sử dụng các dịch vụ của Web services. Các dịch vụ web được sử dụng sẽ giúp cho, nền tảng ứng dụng và công nghệ trở nên độc lập hơn.
* Ví dụ, một ứng dụng VB hoặc .NET có thể giao tiếp với các dịch vụ web Java và ngược lại.

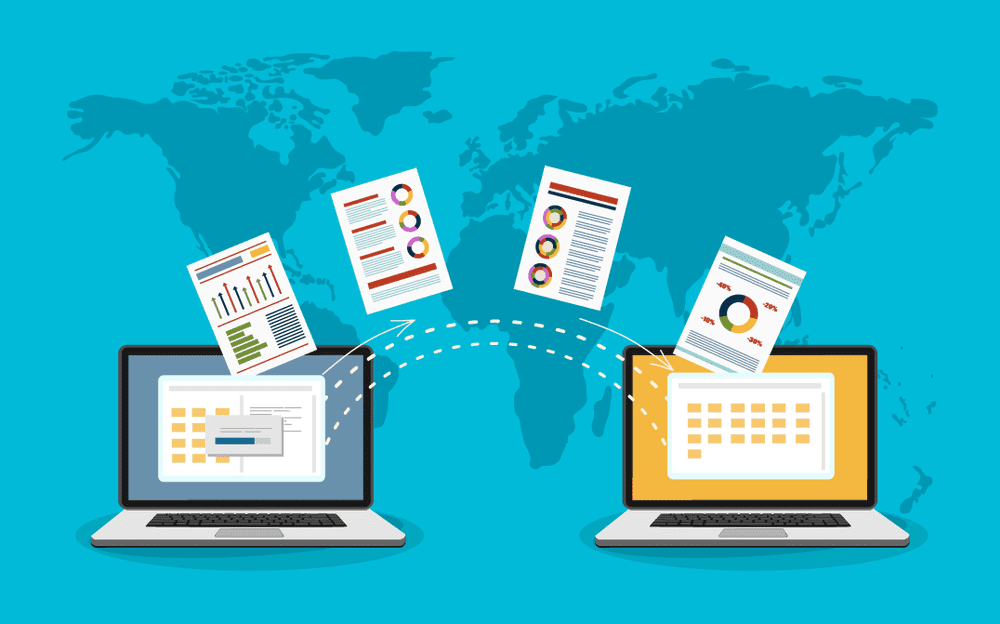


**Truyền thông chi phí thấp**

* So với các giải pháp độc quyền như EDI/B2B thì giải pháp này ít tốn kém hơn nhiều bởi các dịch vụ web sử dụng SOAP qua giao thức HTTP, vì vậy bạn có thể sử dụng mạng internet hiện có của mình để triển khai các dịch vụ web.
* Đồng thời các dịch vụ web cũng có thể được triển khai, trên các cơ chế truyền tải đáng tin cậy khác như FTP.

**Giao thức chuẩn hóa**

* Phương thức giao tiếp của Web services đều được chuẩn hóa, 4 lớp của web gồm: truyền tải dịch vụ, nhắn tin XML, mô tả dịch vụ, các layer khám phá dịch vụ.
* Việc chuẩn hóa stack giao thức này mang lại cho doanh nghiệp nhiều lợi thế như có nhiều lựa chọn, giảm chi phí do cạnh tranh và tăng chất lượng.



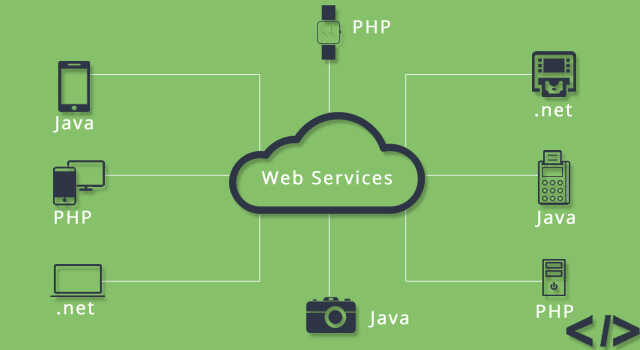
**Hiển thị chức năng hiện có trên mạng**

* Một dịch vụ web là một đơn vị mã được quản lý có thể được gọi từ xa bằng cách sử dụng HTTP. Được kích hoạt bằng cách sử dụng các yêu cầu HTTP.
* Dịch vụ web cho phép bạn hiển thị chức năng của mã hiện có qua mạng, khi được hiển thị trên mạng các ứng dụng khác có thể sử dụng chức năng của chương trình.

## **Ưu và nhược điểm của Web Services**

**Ưu điểm của Web Services**

* Web services cung cấp khả năng hoạt động rộng lớn với các ứng dụng phần mềm khác nhau chạy trên những nền tảng khác nhau.
* Dịch vụ Web khá đơn giản vì chỉ sử dụng URL.
* Làm việc với những giao thức chuẩn Web như XML, HTTP và TCP/IP.
* Sự an toàn của máy chủ cơ sở dữ liệu luôn được bảo mật một cách chắc chắn.
* Web service làm giảm giá thành cho việc tích hợp các hệ thống khác nhau



**Nhược điểm của Web Service**s

* Trong trường hợp sự cố xảy ra, Web Services không có khả năng khôi phục đủ tin cậy nhằm đảm bảo các giao dịch có thể trở về trạng thái ban đầu.
* Dịch vụ Web phụ thuộc vào tốc độ đường truyền Internet.
* Có quá nhiều chuẩn cho Web services khiến người dùng khó nắm bắt.
* Hiệu suất tối ưu của Web Services sẽ bị ảnh hưởng đáng kể bởi số lượng các ứng dụng cộng tác cùng hoạt động.
* Phải quan tâm nhiều hơn đến vấn đề an toàn và bảo mật.

## **Các nguy cơ đối với ứng dụng website**

Ứng dụng website hiện đại đóng vai trò quan trọng trong nhiều lĩnh vực, từ thương mại điện tử, dịch vụ tài chính đến truyền thông xã hội. Tuy nhiên, chúng cũng đối mặt với nhiều nguy cơ bảo mật đáng lo ngại. Một trong những nguy cơ phổ biến nhất là tấn công SQL injection, nơi kẻ tấn công chèn mã độc vào các câu lệnh SQL để truy cập trái phép vào cơ sở dữ liệu. Cross-site scripting (XSS) là một nguy cơ khác, cho phép kẻ tấn công chèn mã JavaScript độc hại vào trang web, có thể đánh cắp thông tin người dùng hoặc chiếm quyền điều khiển trình duyệt. Ngoài ra, các ứng dụng web cũng có thể bị tấn công từ chối dịch vụ (DDoS), làm cho website không thể truy cập được bởi người dùng hợp pháp. Các lỗ hổng bảo mật trong mã nguồn hoặc cấu hình server cũng là một nguy cơ tiềm ẩn, dễ bị khai thác để chiếm quyền điều khiển hoặc đánh cắp dữ liệu.

### **Tấn công Injection**

SQL (SQLi) là một lỗ hổng bảo mật web cho phép kẻ tấn công can thiệp vào các truy vấn mà ứng dụng thực hiện đối với cơ sở dữ liệu của nó. Điều này có thể cho phép kẻ tấn công xem dữ liệu mà thông thường chúng không thể truy xuất được. Điều này có thể bao gồm dữ liệu thuộc về người dùng khác hoặc bất kỳ dữ liệu nào khác mà ứng dụng có thể truy cập. Trong nhiều trường hợp, kẻ tấn công có thể sửa đổi hoặc xóa dữ liệu này, gây ra những thay đổi liên tục đối với nội dung hoặc hành vi của ứng dụng.

Trong một số trường hợp, kẻ tấn công có thể nâng cấp cuộc tấn công SQL để xâm phạm máy chủ cơ bản hoặc cơ sở hạ tầng phụ trợ khác. Nó cũng có thể cho phép họ thực hiện các cuộc tấn công từ chối dịch vụ.

Một cuộc tấn công tiêm nhiễm SQL thành công có thể dẫn đến truy cập trái phép vào dữ liệu nhạy cảm, chẳng hạn như:

Mật khẩu.

Chi tiết thẻ tín dụng.

Thông tin cá nhân của người dùng.

Các cuộc tấn công tiêm nhiễm SQL đã được sử dụng trong nhiều vụ vi phạm dữ liệu nghiêm trọng trong nhiều năm qua. Những điều này đã gây ra thiệt hại về danh tiếng và các khoản tiền phạt theo quy định. Trong một số trường hợp, kẻ tấn công có thể có được một cửa hậu liên tục xâm nhập vào hệ thống của tổ chức, dẫn đến sự xâm phạm lâu dài mà có thể không được chú ý trong một thời gian dài.

Bạn có thể phát hiện việc chèn SQL theo cách thủ công bằng cách sử dụng một bộ thử nghiệm có hệ thống đối với mọi điểm vào trong ứng dụng. Để làm điều này, bạn thường sẽ gửi:

Ký tự trích dẫn đơn ' và tìm kiếm lỗi hoặc các điểm bất thường khác.

Một số cú pháp dành riêng cho SQL đánh giá giá trị cơ sở (gốc) của điểm nhập và một giá trị khác, đồng thời tìm kiếm sự khác biệt mang tính hệ thống trong các phản hồi của ứng dụng.

Các điều kiện Boolean như OR 1=1 và OR 1=2, đồng thời tìm kiếm sự khác biệt trong phản hồi của ứng dụng.

Tải trọng được thiết kế để kích hoạt độ trễ thời gian khi được thực thi trong truy vấn SQL và tìm kiếm sự khác biệt về thời gian cần thiết để phản hồi.

Tải trọng OAST được thiết kế để kích hoạt tương tác mạng ngoài băng tần khi được thực thi trong truy vấn SQL và giám sát mọi tương tác phát sinh.

Ngoài ra, bạn có thể tìm thấy phần lớn các lỗ hổng SQL một cách nhanh chóng và đáng tin cậy bằng cách sử dụng Burp Scanner.

Hầu hết các lỗ hổng SQL xảy ra trong mệnh đề WHERE của truy vấn SELECT. Hầu hết những người thử nghiệm có kinh nghiệm đều quen thuộc với kiểu chèn SQL này.

Tuy nhiên, lỗ hổng chèn SQL có thể xảy ra ở bất kỳ vị trí nào trong truy vấn và trong các loại truy vấn khác nhau. Một số vị trí phổ biến khác phát sinh việc tiêm SQL là:

Trong câu lệnh UPDATE, trong các giá trị được cập nhật hoặc mệnh đề WHERE.

Trong câu lệnh INSERT, bên trong các giá trị được chèn.

Trong câu lệnh SELECT, trong tên bảng hoặc cột.

Trong câu lệnh SELECT, trong mệnh đề ORDER BY.

Hãy tưởng tượng một ứng dụng mua sắm hiển thị các sản phẩm theo các danh mục khác nhau. Khi người dùng nhấp vào danh mục Quà tặng, trình duyệt của họ sẽ yêu cầu URL: https://insecure-website.com/products?category=Gifts

Điều này khiến ứng dụng thực hiện truy vấn SQL để truy xuất thông tin chi tiết về các sản phẩm có liên quan từ cơ sở dữ liệu: CHỌN \* TỪ sản phẩm WHERE danh mục = 'Quà tặng' VÀ phát hành = 1. Truy vấn SQL này yêu cầu cơ sở dữ liệu trả về:

tất cả chi tiết (\*)

từ bảng sản phẩm

trong đó danh mục là Quà tặng

và phát hành là 1.

Hạn chế được phát hành = 1 đang được sử dụng để ẩn các sản phẩm chưa được phát hành. Chúng ta có thể giả sử đối với các sản phẩm chưa được phát hành, đã phát hành = 0.

Ứng dụng không triển khai bất kỳ biện pháp phòng vệ nào trước các cuộc tấn công tiêm nhiễm SQL. Điều này có nghĩa là kẻ tấn công có thể xây dựng cuộc tấn công sau, ví dụ: https://insecure-website.com/products?category=Gifts'-

Điều này dẫn đến truy vấn SQL: CHỌN \* TỪ sản phẩm WHERE danh mục = 'Quà tặng'--' VÀ phát hành = 1

Điều quan trọng cần lưu ý rằng -- là một chỉ báo nhận xét trong SQL. Điều này có nghĩa là phần còn lại của truy vấn được hiểu là một nhận xét, loại bỏ nó một cách hiệu quả. Trong ví dụ này, điều này có nghĩa là truy vấn không còn bao gồm AND đã phát hành = 1. Kết quả là tất cả các sản phẩm đều được hiển thị, bao gồm cả những sản phẩm chưa được phát hành.

Bạn có thể sử dụng một cuộc tấn công tương tự để khiến ứng dụng hiển thị tất cả các sản phẩm trong bất kỳ danh mục nào, bao gồm cả những danh mục mà chúng không biết:

https://insecure-website.com/products?category=Gifts'+OR+1=1--

Điều này dẫn đến truy vấn SQL: CHỌN \* TỪ sản phẩm WHERE danh mục = 'Quà tặng' HOẶC 1=1--' VÀ phát hành = 1

Truy vấn đã sửa đổi trả về tất cả các mục trong đó danh mục là Quà tặng hoặc 1 bằng 1. Vì 1=1 luôn đúng nên truy vấn trả về tất cả các mục.

Trong trường hợp ứng dụng phản hồi bằng kết quả của truy vấn SQL, kẻ tấn công có thể sử dụng lỗ hổng chèn SQL để lấy dữ liệu từ các bảng khác trong cơ sở dữ liệu. Bạn có thể sử dụng từ khóa UNION để thực hiện truy vấn SELECT bổ sung và nối kết quả vào truy vấn ban đầu.

Ví dụ: nếu một ứng dụng thực hiện truy vấn sau có chứa Quà tặng đầu vào của người dùng: CHỌN tên, mô tả TỪ sản phẩm WHERE danh mục = 'Quà tặng' Kẻ tấn công có thể gửi đầu vào:

' UNION CHỌN tên người dùng, mật khẩu TỪ người dùng--

Điều này khiến ứng dụng trả về tất cả tên người dùng và mật khẩu cùng với tên và mô tả sản phẩm.

Blind SQL injection là một kiểu tấn công SQL injection truy vấn cơ sở dữ liệu sử dụng các mệnh đề để đoán biết. Cách tấn công này thường được sử dụng khi mà một ứng dụng (web, apps) được cấu hình để chỉ hiển thị những thông báo lỗi chung chung, không hiển thị ra lỗi của SQL. Blind SQL Injection đôi khi được gọi là SQL Inferential SQL.

Có hai biến thể chính để thực hiện Blind SQL injection: Blind SQL injejction dựa vào nội dung phản hồi và Blind SQL injejction dựa vào độ trễ của thời gian phản hồi.

Các Blind SQL injection thường được sử dụng để xây dựng lược đồ cơ sở dữ liệu và lấy tất cả dữ liệu trong cơ sở dữ liệu. Điều này được thực hiện bằng cách sử dụng các kỹ thuật brute force và yêu cầu nhiều requests nhưng có thể được tự động hóa bởi những kẻ tấn công sử dụng các công cụ SQL injection như: SQLmap ..v..v..v..

Đối với SQL injection thông thường cho phép chúng ta khai thác thông tin trực tiếp từ cơ sở dữ liệu. Đối với Blind, chúng ta phải đoán thông tin dần dần dựa theo kết quả trả về là TRUE FALSE. Do đó sẽ dẫn tới một vấn để đó là nếu thông tin chúng ta cần lấy. ví dụ: tên cơ sở dữ liệu, tên bảng, tên user quá dài thì việc đoán sẽ gặp trở ngại lớn về thời gian.

### **Tấn công thực thi các tập tin độc hại**

Tấn công thực thi các tập tin độc hại là một dạng tấn công mà kẻ tấn công sử dụng các tập tin chứa mã độc (malware) để xâm nhập vào hệ thống máy tính của nạn nhân. Các tập tin này có thể được gửi qua email, tải về từ internet, hoặc thông qua các phương pháp khác.

Tấn công thực thi các tập tin độc hại (malicious file execution attack) là một phương thức mà kẻ tấn công cố gắng thực thi mã độc trên hệ thống máy tính của nạn nhân thông qua các tập tin được tạo ra để lợi dụng lỗ hổng bảo mật trong phần mềm hoặc hệ điều hành.

Khi thiết bị nhiễm Malware, bạn có thể nhận thấy các dấu hiệu sau:

Máy tính chạy chậm, tốc độ xử lý của hệ điều hành giảm cho dù bạn đang điều hướng Internet hay chỉ sử dụng các ứng dụng cục bộ.

Bạn bị làm phiền bởi quảng cáo pop-up, mà cụ thể hơn là Adware.

Hệ thống liên tục gặp sự cố, bị đóng băng hoặc hiển thị BSOD – màn hình xanh (đối với Windows).

Dung lượng ổ cứng giảm bất thường.

Hoạt động Internet của hệ thống tăng cao không rõ nguyên nhân.

Tài nguyên hệ thống tiêu hao bất thường, quạt máy tính hoạt động hết công suất.

Trang chủ của trình duyệt mặc định thay đổi mà không có sự cho phép của bạn. Các liên kết bạn nhấp vào sẽ chuyển hướng bạn đến các trang không mong muốn.

Các thanh công cụ, tiện ích mở rộng hoặc plugin mới được thêm vào trình duyệt.

Các chương trình anti-virus ngừng hoạt động và không cập nhật được.

Bạn nhận được thông báo đòi tiền chuộc từ Malware, nếu không dữ liệu của bạn sẽ bị xóa.

Trong quá trình sử dụng Internet, những thao tác sau có thể khiến bạn bị nhiễm Malware:

Truy cập các trang web độc hại, tải trò chơi, file nhạc nhiễm Malware, cài đặt thanh công cụ/phần mềm từ nhà cung cấp lạ, mở tệp đính kèm email độc hại (malspam) hoặc các dữ liệu tải xuống không được quét bởi phần mềm bảo mật.

Tải nhầm các ứng dụng độc hại ngụy trang dưới dạng các ứng dụng hợp pháp, các thông báo cảnh báo khi cài đặt ứng dụng, đặc biệt khi ứng dụng yêu cầu quyền truy cập email hoặc thông tin cá nhân.

Tải ứng dụng ở các nguồn không đáng tin cậy.

Vô tình cài đặt các phần mềm bổ sung đi kèm với ứng dụng (potentially unwanted program) chứa Malware.

Ngoài ra, việc không sử dụng các chương trình bảo mật cũng là lý do khiến Malware xâm nhập dễ dàng hơn.

**Các loại Malware phổ biến:**

**Virus**

Loại chương trình này vô cùng nguy hiểm vì có khả năng sinh sôi, lây lan ra khắp hệ thống phần mềm, gây thiệt hại phần cứng,… với tốc độ rất nhanh. Nếu không khắc phục kịp thời, mọi thông tin, dữ liệu, thậm chí là thiết bị đều sẽ mất kiểm soát.

**Worm**

Hay còn được hiểu với nghĩa là **con sâu** và chương trình này còn độc hại hơn cả virus. Bởi Worm có thể tự sinh sôi, hoạt động mà không chịu bất kỳ sự tác động, điều khiển nào đến từ con người cả. Thậm chỉ khi đã bị “tiêu diệt” rồi thì vẫn có khả năng tự tái tạo, hoạt động lại như bình thường. Nghe khá giống với kiểu AI – trí tuệ nhân tạo.

**Trojan**

Một phần mềm được xây dựng như một chương trình chính chủ, hợp pháp và uy tín. Được quảng cáo và sở hữu chức năng bảo vệ, giúp máy tính tránh khỏi sự xâm nhập, tấn công của Virus. Thực chất Trojan giống như một cánh cổng mở ra và cho phép hàng triệu loại Virus khác nhau tiến công, gây hại cho máy tính. Mặc dù Trojan không có chức năng sao chép dữ liệu những lại có khả năng “hủy diệt” rất kinh khủng.



*Các Malware phổ biến*

**Spyware**

Spyware hoàn toàn không có chức năng hủy hoại dữ liệu nhưng lại là chuyên gia theo dõi, sao chép và quan sát hoạt động của người dùng. Bất kỳ dữ liệu nào được nhập, xuất ra khỏi thiết bị đều được Spyware ghi nhận, cung cấp lại cho những kẻ gian mà không ai hay biết.

**Rootkit**

Kể từ khi người dùng cài đặt phần mềm này vào thiết bị, Rootkit ngay lập tức tấn công và tước quyền quản trị. Khi này các tin tắc có thể tự do truy cập trái phép, vượt qua được bất cứ “bức tường bảo vệ” nào một cách dễ dàng. Đánh cấp dữ liệu, theo dõi hành vi người dùng một cách ung dung mà không có bất kỳ cảnh báo lỗi hệ thống nào diễn ra.

**Ransomware**

Ngăn bạn truy cập vào thiết bị và mã hóa dữ liệu, sau đó buộc bạn phải trả tiền chuộc để lấy lại chúng. Ransomware được xem là vũ khí của tội phạm mạng vì nó thường dùng các phương thức thanh toán nhanh chóng bằng tiền điện tử.

**Cách phòng tránh Malware**

Bạn nên cảnh giác với các web có domain kết thúc bằng tập hợp các chữ cái riêng lẻ, và có đuôi không giống như bình thường (.com, .vn hay .org,... ).

Bạn nên chú ý đến các dấu hiệu nhiễm Malware của máy tính bạn ngay từ đầu để ngăn chặn sự xâm nhập.

Bạn nên tránh nhấp vào các quảng cáo pop-up khi bạn lướt web.

Không nên mở các file lạ có đính kèm trên email.

Không nên tải các phần mềm, ứng dụng ở trên các website không đáng tin cậy.



Bạn nên thường xuyên cập nhật hệ điều hành, ứng dụng hay plugin.

Chỉ nên tải các app có lượt tải lớn và thứ hạng cao từ Google Play hay Apple Store,...

Không nên tải ứng dụng từ các nguồn bên thứ 3, và nếu như bạn sử dụng thiết bị [Android](https://www.thegioididong.com/dtdd?g=android) thì bạn nên vào: Cài đặt > Bảo mật > Tắt ứng dụng không xách định để tránh cài đặt phải các ứng dụng từ bên thứ 3.

Không nên nhấp vào các liên kết lạ, các liên kết không xác định ở trong email hay văn bản và tin nhắn.

## **Các giải pháp đảm bảo an toàn ứng dụng website**

Đảm bảo an toàn cho ứng dụng website là một yếu tố quan trọng để bảo vệ dữ liệu và duy trì sự tin cậy của người dùng. Các giải pháp an ninh có thể được áp dụng bao gồm việc sử dụng tường lửa ứng dụng web (WAF) để ngăn chặn các cuộc tấn công SQL Injection, Cross-Site Scripting (XSS), và các mối đe dọa phổ biến khác. Mã hóa dữ liệu cả trong quá trình truyền tải lẫn lưu trữ giúp bảo vệ thông tin nhạy cảm khỏi bị truy cập trái phép. Xác thực đa yếu tố (MFA) cung cấp một lớp bảo mật bổ sung bằng cách yêu cầu người dùng xác nhận danh tính qua nhiều bước. Ngoài ra, việc thường xuyên cập nhật và vá lỗi phần mềm, kết hợp với kiểm tra bảo mật định kỳ, giúp phát hiện và khắc phục kịp thời các lỗ hổng bảo mật. Các giải pháp này, khi được thực hiện đồng bộ và nhất quán, sẽ tạo ra một môi trường an toàn cho ứng dụng web và bảo vệ người dùng khỏi các nguy cơ tiềm ẩn.

### **Kỹ thuật lập trình an toàn**

Kỹ thuật lập trình an toàn bao gồm nhiều phương pháp và kỹ thuật khác nhau để đảm bảo rằng phần mềm được phát triển và duy trì một cách bảo mật. Dưới đây là một số cách phổ biến.

Kiểm tra và Xác thực Dữ liệu Đầu vào:

Sử dụng các hàm kiểm tra và lọc dữ liệu để ngăn chặn các cuộc tấn công như SQL injection và cross-site scripting (XSS).

# Ví dụ về kiểm tra và xác thực dữ liệu đầu vào trong Python

def login(username, password):

# Kiểm tra xem dữ liệu đầu vào có chứa ký tự đặc biệt không

if any(char in username or char in password for char in ['<', '>', '$', '@', '&']):

return "Dữ liệu đầu vào không hợp lệ"

else:

# Xác thực thông tin đăng nhập

# (Đoạn này có thể sử dụng hàm hash cho mật khẩu)

if username == "admin" and password == "password":

return "Đăng nhập thành công"

else:

return "Tên đăng nhập hoặc mật khẩu không đúng"

Đảm bảo rằng dữ liệu đầu vào chỉ chứa những ký tự hoặc định dạng hợp lệ.

Sử dụng Mã hóa:

Mã hóa dữ liệu nhạy cảm cả trong quá trình truyền tải và khi lưu trữ.

Sử dụng các giao thức mã hóa mạnh mẽ như SSL/TLS cho truyền tải dữ liệu.

# Ví dụ về mã hóa dữ liệu trong Python sử dụng thư viện hashlib

import hashlib

def encrypt\_password(password):

# Mã hóa mật khẩu bằng thuật toán SHA-256

hashed\_password = hashlib.sha256(password.encode()).hexdigest()

return hashed\_password

Quản lý Phiên Làm Việc:

Sử dụng các mã token duy nhất cho mỗi phiên làm việc để ngăn chặn các cuộc tấn công đánh cắp phiên (session hijacking).

# Ví dụ về quản lý phiên làm việc trong Python sử dụng Flask

from flask import Flask, session, redirect, url\_for, request

app = Flask(\_\_name\_\_)

app.secret\_key = 'your\_secret\_key'

@app.route('/')

def index():

# Kiểm tra xem người dùng đã đăng nhập hay chưa

if 'username' in session:

return 'Đã đăng nhập với tên: ' + session['username']

return 'Bạn chưa đăng nhập'

@app.route('/login', methods=['GET', 'POST'])

def login():

if request.method == 'POST':

# Kiểm tra thông tin đăng nhập

if request.form['username'] == 'admin' and request.form['password'] == 'password':

session['username'] = request.form['username']

return redirect(url\_for('index'))

else:

return 'Sai tên đăng nhập hoặc mật khẩu'

return '''

<form method="post">

<p><input type="text" name="username">

<p><input type="password" name="password">

<p><input type="submit" value="Đăng nhập">

</form>

'''

@app.route('/logout')

def logout():

# Xóa phiên làm việc khi người dùng đăng xuất

session.pop('username', None)

return redirect(url\_for('index'))

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

Hạn chế thời gian sống của các phiên làm việc và yêu cầu xác thực lại sau một khoảng thời gian nhất định.

Xác thực và Ủy quyền:

Sử dụng các phương pháp xác thực mạnh mẽ, như xác thực hai yếu tố (2FA).

Kiểm tra quyền truy cập của người dùng đối với từng chức năng và dữ liệu cụ thể trong hệ thống.

Quản lý Lỗi và Ngoại lệ:

Xử lý lỗi một cách an toàn, không tiết lộ thông tin nhạy cảm qua thông báo lỗi.

Ghi nhật ký (logging) các lỗi và sự kiện an ninh để theo dõi và phân tích.

Bảo vệ Mã nguồn:

Sử dụng các công cụ quản lý mã nguồn như Git với các cấu hình bảo mật phù hợp.

Thường xuyên kiểm tra mã nguồn để phát hiện sớm các lỗ hổng bảo mật.

Kiểm tra Bảo mật và Đánh giá Lỗ hổng:

Sử dụng các công cụ kiểm tra bảo mật tự động để phát hiện các lỗ hổng trong phần mềm.

Thực hiện các bài kiểm tra xâm nhập (penetration testing) để đánh giá khả năng chống đỡ trước các cuộc tấn công.

Cập nhật và Vá lỗi Thường xuyên:

Theo dõi và cài đặt các bản vá lỗi và cập nhật bảo mật từ các nhà cung cấp phần mềm và thư viện bên thứ ba.

Sử dụng các hệ thống quản lý bản vá (patch management) để đảm bảo tất cả các thành phần phần mềm luôn được cập nhật.

Đào tạo và Nhận thức Bảo mật:

Đào tạo lập trình viên và nhân viên về các nguyên tắc và thực hành bảo mật.

Thường xuyên tổ chức các buổi hội thảo và khóa học về bảo mật để cập nhật kiến thức mới.

Thiết kế Bảo mật Ngay từ Đầu:

Áp dụng các nguyên tắc thiết kế bảo mật ngay từ giai đoạn đầu của dự án phát triển phần mềm.

Sử dụng các mô hình thiết kế như bảo mật theo lớp (defense in depth) và tối thiểu hóa quyền truy cập (principle of least privilege).

### **Bảo vệ hệ thống tệp tin**

Hệ thống tệp tin là một thành phần quan trọng trong cơ sở hạ tầng công nghệ thông tin của một tổ chức, nơi lưu trữ các dữ liệu quan trọng và nhạy cảm. Việc bảo vệ hệ thống tệp tin là cần thiết để đảm bảo an toàn thông tin và tránh các rủi ro mất mát hoặc rò rỉ dữ liệu. Một số giải pháp để bảo mật hệ thống tệp tin:

* Kiểm soát Truy cập (Access Control): Việc phân quyền là quan trọng trong hệ thống, kiểm soát và giới hạn truy cập vào tài nguyên giúp tránh được rủi ro về dữ liệu.
* Mã hóa tệp tin: Trong một số trường hợp kẻ truy cập có được quyền truy cập vào tài nguyên hạn chế, khả năng dữ liệu bị đánh cắp cho mục đích cá nhân là vô cùng nguy hiểm. Lúc này, mã hóa các tệp tin sang dạng không thể đọc được sẽ bảo vệ các tệp an toàn
* Sao lưu và Khôi phục: Các bản sao lưu định kỳ sẽ giúp phục hồi lại các đổ vỡ sau một cuộc tấn công phá hoại hoặc các vấn đề từ bên trong như mất mát hoặc hư hỏng.
* Giám sát và Ghi Log: Theo dõi và ghi lại các hoạt động truy cập tệp tin để phát hiện và phản ứng kịp thời với các hành vi bất thường hoặc tấn công.

### **Bảo vệ cơ sở dữ liệu**

* Triển khai bảo mật vật lý:

Tấn công vật lý là một trong những cuộc tấn công an ninh mạng thường gặp khiến các trung tâm dữ liệu hoặc máy chủ mất quyền kiểm soát truy cập. Từ đó, dẫn đến dữ liệu bị lấy cắp, làm hỏng hoặc chèn virus độc hại để giành quyền truy cập từ xa. Được biết, đây là một trong những cách thức tấn công rất khó phát hiện bởi chúng vượt qua giao thức bảo mật kỹ thuật số.

Để khắc phục cũng như tăng cường an toàn hiệu quả cho cơ sở dữ liệu, cần tìm chọn các dịch vụ, công ty có những thành tích tốt về các vấn đề bảo mật. Không nên sử dụng các dịch vụ lưu trữ miễn phí hay thiếu những đảm bảo về yếu tố bảo mật. Trường hợp sở hữu hệ thống hạ tầng riêng, có thể ứng dụng thêm các biện pháp an ninh vật lý như camera, khóa và nhân viên an ninh. Đồng thời, quán triệt mọi truy cập vào máy chủ vật lý phải được ghi lại cũng như chỉ cấp quyền cho những người cụ thể đảm nhiệm.

* Tách biệt các máy chủ cơ sở dữ liệu:

Nếu như việc đặt CSDL trên cùng một máy chủ web sẽ dễ bị tấn công bởi các nền tảng dịch vụ trực tuyến. Việc đưa ra các biện pháp bảo mật riêng biệt cho các CSDL sẽ đảm bảo an toàn và giảm đáng kể rủi ro khi bị tấn công.

Nên tách biệt các máy chủ cơ sở dữ liệu ra khỏi các dịch vụ khác. Cùng với đó, kết hợp sử dụng thông tin bảo mật theo thời gian thực, theo dõi sự kiện truy cập vào cơ sở dữ liệu một cách độc lập. Khi ấy, hoàn toàn có thể làm chủ tình thế một cách nhanh chóng nếu có phát hiện vi phạm.

* Thiết lập máy chủ Proxy HTTPS:

Máy chủ Proxy là loại máy chuyên đánh giá các yêu cầu được gửi từ máy trạm trước khi truy cập vào máy chủ cơ sở dữ liệu. Bằng một cách nào đó, máy chủ Proxy hoạt động như một người “gác cổng” trung gian, ngăn chặn các yêu cầu không được phép.

Chỉ cần thiết lập máy chủ Proxy HTTPS, mọi dữ liệu đi qua sẽ được mã hóa và trang bị thêm “lớp áo giáp” bảo mật bổ sung. Từ đó, ngăn chặn và hạn chế tối đa tình trạng xâm nhập an ninh mạng.

* Tránh sử dụng cổng mạng mặc định

Khi thiết lập các CSDL, đơn vị triển khai thường sử dụng các cổng mạng mặc định như CSDL Oracle là cổng 1521 kết nối và truy cập CSDL. Trong khi các cổng này lại được sử dụng rất nhiều trong những cuộc tấn công như một hình thức “mở đường” cho các cuộc xâm nhập CSDL nhanh gọn

Người dùng không nên sử dụng cổng mạng mặc định. Lúc này kẻ tấn công sẽ phải thử và mất nhiều thời gian, công sức hơn để dò tìm trên nhiều cổng khác nhau truy cập vào máy chủ

* Liên tục giám sát cơ sở dữ liệu theo thời gian thực

Việc thường xuyên theo dõi, giám sát CSDL truy tìm các truy cập lạ sẽ giúp tăng cường bảo mật và cho phép phản ứng nhanh chóng với các cuộc tấn công. Có thể ứng dụng phần mềm Oracle Audit Vault để ghi lại, phát hiện và cảnh báo tất cả các hành động thực hiện trên máy chủ CSDL. Ngoài ra, cũng có thể thiết lập giao thức leo thang (escalate) để dễ dàng xử lý các cuộc tấn công mạng, tăng tính an toàn cho dữ liệu nhạy cảm.

* Dùng tường lửa cơ sở dữ liệu
* Thiết lập các giao thức mã hóa dữ liệu

Việc thiết lập và cài đặt các giao thức mã hóa cho dữ liệu sẽ góp phần ngăn chặn hiệu quả các nguy cơ các cuộc tấn công, xâm phạm/đánh cắp dữ liệu. Khi đó, việc triển khai các giao thức mã hóa sẽ giúp kẻ xấu không đọc và khai thác được ngay cả khi đã lấy được dữ liệu từ hệ thống.

* Sao lưu cơ sở dữ liệu đều đặn

Ngoài việc giám sát, theo dõi chặt chẽ CSDL, để giảm thiểu nguy cơ rò rỉ, phá hoại dữ liệu, doanh nghiệp nên thường xuyên sao lưu các dữ liệu với thời gian lên lịch hợp lý và đảm bảo bản sao lưu có thể dùng để khôi phục khi có trường hợp xấu xảy ra. Cũng cần thực hiện song song phương pháp mã hóa và lưu trữ trên máy chủ khác cho bản sao lưu để đảm bảo CSDL được an toàn và có thể khôi phục khi dữ liệu chính bị đánh cắp.

* Cập nhật phiên bản mới nhất

Dựa trên khảo sát cho thấy, trên 10 ứng dụng được lưu hành thì có tới 9 ứng dụng không được cập nhật bản mới. Điều này ảnh hưởng nghiêm trọng tới vấn đề bảo mật bởi kẻ xấu có thể xâm nhập vào bất cứ thành phần nào trong hệ thống bao gồm cả các phần mềm phụ, Plugin hay ứng dụng bên thứ ba. Doanh nghiệp nên thường xuyên cập nhật các phiên bản mới được cung cấp bởi các hãng nổi tiếng cho phần mềm CSDL và các phần mềm ứng dụng khác có trên hệ thống

* Phương thức xác thực người dùng mạnh:

Phương pháp để bảo mật dữ liệu là xác thực người dùng mạnh. Dựa theo nghiên cứu mới đây nhất của Verizon, có tới 80% các vụ xâm nhập đánh cắp dữ liệu là qua mật khẩu. Theo đó, hầu hết người dùng thường sử dụng các mật khẩu dễ đoán và dễ bị đánh cắp. Người dùng nên bổ sung thêm lớp bảo mật khác để tăng tính bảo mật CSDL. Cách khác, doanh nghiệp có thể thiết lập địa chỉ IP cho phép truy cập vào CSDL để giảm nguy cơ xâm phạm tiềm ẩn

## **Sử dụng các thiết bị và phần mềm bảo mật cho ứng dụng website**

Trong bối cảnh các mối đe dọa an ninh mạng ngày càng gia tăng, việc bảo vệ các ứng dụng website trở nên cấp thiết. Sử dụng các thiết bị và phần mềm bảo mật đóng vai trò then chốt trong chiến lược an ninh của tổ chức. Các thiết bị bảo mật như tường lửa (firewall), hệ thống phát hiện và ngăn chặn xâm nhập (IDS/IPS), cùng với phần mềm bảo mật như phần mềm chống virus và hệ thống quản lý sự kiện và thông tin bảo mật (SIEM) giúp ngăn chặn, phát hiện và phản ứng với các mối đe dọa. Việc triển khai những công nghệ này tạo ra một lớp bảo vệ đa tầng, tăng cường khả năng phòng thủ và đảm bảo an toàn cho ứng dụng website. Phần này sẽ xem xét chi tiết các giải pháp bảo mật hàng đầu và cách triển khai chúng hiệu quả.

Dưới đây là bảng so sánh giữa Web Application Firewall (WAF) và Intrusion Prevention System (IPS):

|  |  |  |
| --- | --- | --- |
| Tiêu chí | Web Application Firewall (WAF) | Intrusion Prevention System (IPS) |
| Mục đích | Bảo vệ ứng dụng web khỏi các tấn công dựa trên HTTP/HTTPS | Bảo vệ hệ thống mạng khỏi các cuộc tấn công mạng tổng quát |
| Phạm vi bảo vệ | Ứng dụng web | Toàn bộ mạng, bao gồm tất cả các thiết bị và giao thức |
| Phương pháp hoạt động | Phân tích và lọc lưu lượng HTTP/HTTPS | Phân tích và kiểm tra lưu lượng mạng để phát hiện và ngăn chặn tấn công |
| Loại tấn công ngăn chặn | SQL Injection, XSS, CSRF, và các lỗ hổng ứng dụng web khác | DoS, DDoS, malware, và các cuộc tấn công mạng khác |
| Vị trí triển khai | Giữa ứng dụng web và người dùng | Trên toàn mạng, thường là giữa firewall và mạng nội bộ |
| Tính năng nổi bật | Bảo vệ khỏi các tấn công web đặc thù, khả năng ghi nhật ký chi tiết | Bảo vệ toàn diện khỏi các tấn công mạng, khả năng tích hợp với các hệ thống bảo mật khác |
| Khả năng tích hợp | Tích hợp tốt với các ứng dụng web và nền tảng đám mây | Tích hợp tốt với các thiết bị mạng và hệ thống bảo mật khác |

Bảng so sánh trên giúp bạn có cái nhìn tổng quan về sự khác biệt và lợi ích của WAF và IPS trong việc bảo vệ hệ thống mạng và ứng dụng web.

### **Giải pháp sử dụng tường lửa**

Tường lửa (firewall) là công cụ bảo mật quan trọng giúp bảo vệ các ứng dụng website bằng cách kiểm soát lưu lượng truy cập vào và ra khỏi mạng, ngăn chặn các truy cập trái phép và bảo vệ hệ thống trước các mối đe dọa. Tường lửa có thể lọc gói tin, theo dõi trạng thái kết nối và giám sát lưu lượng ở cấp độ ứng dụng để ngăn chặn các cuộc tấn công như DDoS, SQL injection và XSS. Việc thiết lập các quy tắc bảo mật chặt chẽ, giám sát và cập nhật thường xuyên, cùng kết hợp với các biện pháp bảo mật khác sẽ giúp tổ chức nâng cao mức độ bảo vệ cho ứng dụng website và duy trì tính toàn vẹn, sẵn sàng của hệ thống.

*Tường lửa của CheckPoint*

Check Point là một trong những nhà cung cấp hàng đầu về các giải pháp bảo mật mạng, nổi bật với sản phẩm tường lửa tiên tiến. Tường lửa của Check Point cung cấp một loạt các tính năng bảo mật mạnh mẽ và linh hoạt, được thiết kế để bảo vệ toàn diện cho hệ thống mạng và ứng dụng website của tổ chức

* Chức năng và Tính năng Nổi bật: Như Kiểm soát truy cập toàn diện giúp đảm bảo kết nối hợp lệ và quản lý quyền hiệu quả. Bảo vệ ứng dụng giúp ngăn chặn các cuộc tấn công và các lỗ hổng bảo mật, khả nnawg kiểm tra nội dung gói tin. Bảo vệ chống lại các cuộc đe dọa mới và quản lý giám sát tập chung.
* Có nhiều lợi ích như bảo vệ toàn diện và hiệu quả. Dễ dàng triển khai và quản lý cũng như tính linh hoạt cao cho phép mở rộng cho phù hợp với hệ thống.

Việc triển khai tường lửa Check Point bao gồm cấu hình các chính sách bảo mật chi tiết dựa trên nhu cầu tổ chức, định kỳ xem xét và cập nhật các chính sách bảo mật, giám sát hoạt động tường lửa để phát hiện và phản ứng kịp thời với các mối đe dọa. Đồng thời, tích hợp tường lửa Check Point với các giải pháp bảo mật khác như IDS/IPS, VPN và DLP tạo ra một hệ thống bảo mật đa lớp toàn diện, nâng cao mức độ bảo vệ cho hệ thống mạng và ứng dụng website.

*Giải pháp với thiết bị tường lửa ứng dụng web của Imperva*

Imperva là một nhà cung cấp nổi tiếng trong lĩnh vực bảo mật, đặc biệt với các giải pháp tường lửa ứng dụng web (WAF). Thiết bị tường lửa ứng dụng web của Imperva giúp bảo vệ các ứng dụng web khỏi nhiều loại tấn công mạng, bao gồm các lỗ hổng bảo mật phổ biến như SQL injection, cross-site scripting (XSS), và các cuộc tấn công DDoS. Dưới đây là các đặc điểm và lợi ích của giải pháp này:

* Chức năng và tính năng nổi bật như khả năng nhận diện đáng ngờ, kiểm soát truy cập giống như phân quyền, và đặc biệt là nó bảo vệ DDOS và khả năng bảo vệ web luôn sẵn sàng ngay cả khi bị tấn công.
* Lợi ích mà nó mang lại có thể kể đến như ngăn chặn các cuộc tấn công phức tạp. Dễ dàng triển khai và quản lý cũng như nó liên tục cập nhật các lỗ hổng mới.

Tường lửa ứng dụng web của Imperva là một giải pháp bảo mật mạnh mẽ, cung cấp sự bảo vệ toàn diện cho các ứng dụng web khỏi các cuộc tấn công mạng phức tạp và đa dạng. Sự dễ dàng trong triển khai và quản lý, cùng với khả năng cập nhật bảo mật liên tục, giúp các tổ chức duy trì an toàn và hiệu quả cho các ứng dụng web.

*Sử dụng phần mềm tường lửa*

Phần mềm tường lửa đóng vai trò quan trọng trong việc bảo vệ các ứng dụng web và hệ thống mạng trước các mối đe dọa từ bên ngoài. Không giống như các thiết bị phần cứng, phần mềm tường lửa có thể được cài đặt trực tiếp trên máy chủ hoặc các thiết bị đầu cuối, mang lại sự linh hoạt và dễ dàng trong việc quản lý và cấu hình:

* Chức năng truy cập linh hoạt hơn so với các thiết bị phần cứng, có thể tự do cấu hình theo yêu cầu. Các bản ghi log sẽ liên tục cấp cho quản lý, phân tích và sử lý yêu cầu tổng quát
* Lợi ích mà nó đem lại đầu tiên là cài đặt linh hoạt và tiện lợi, nó có thể được triển khai nhanh chóng trên các thiết bị đầu cuối mà không cần phần mềm chuyên dụng. Cập nhật dễ dàng cũng là một trong tính tiện lợi mà nó đem lại cũng như khả năng tương tác với các giải pháp bảo mật khác.

Sử dụng phần mềm tường lửa là một giải pháp bảo mật linh hoạt và hiệu quả, giúp bảo vệ các ứng dụng web và hệ thống mạng khỏi các mối đe dọa từ bên ngoài. Với khả năng triển khai nhanh chóng, dễ dàng cập nhật và tích hợp với các giải pháp bảo mật khác, phần mềm tường lửa mang lại sự an toàn và hiệu quả cao cho các tổ chức.

### **Sử dụng thiết bị phát hiện và phòng chống xâm nhập (IDS/ IPS)**

Thiết bị phát hiện và phòng chống xâm nhập (IDS/IPS) là một thành phần quan trọng trong hệ thống bảo mật mạng, giúp giám sát và bảo vệ hệ thống khỏi các cuộc tấn công và xâm nhập trái phép. IDS (Intrusion Detection System) và IPS (Intrusion Prevention System) không chỉ phát hiện các mối đe dọa mà còn có khả năng ngăn chặn chúng trước khi gây hại cho hệ thống.

* Thiết bị Phát hiện Xâm nhập (IDS) : IDS là hệ thống giám sát mạng hoặc hệ thống máy tính để phát hiện các hoạt động bất thường hoặc trái phép. IDS có thể được chia thành hai loại chính:
  + Network-based IDS (NIDS): Giám sát lưu lượng mạng để phát hiện các hành vi xâm nhập trái phép. NIDS thường được đặt tại các điểm chiến lược trong mạng, như gần tường lửa hoặc các router.
  + Host-based IDS (HIDS): Giám sát các hoạt động trên từng máy tính cá nhân để phát hiện các dấu hiệu xâm nhập. HIDS thường được cài đặt trên từng thiết bị và theo dõi các file log, các thay đổi hệ thống và các hoạt động ứng dụng.
* Thiết bị Phòng chống Xâm nhập (IPS): IPS là hệ thống không chỉ phát hiện mà còn ngăn chặn các hành vi xâm nhập trái phép. IPS thường được xem là bước tiến hóa của IDS với khả năng phản ứng tự động. IPS cũng được chia thành hai loại chính:
  + Network-based IPS (NIPS): Giám sát và kiểm soát lưu lượng mạng để ngăn chặn các hành vi xâm nhập. NIPS có khả năng chặn hoặc hạn chế lưu lượng mạng đáng ngờ.
  + Host-based IPS (HIPS): Tương tự như HIDS nhưng có khả năng ngăn chặn các hoạt động xâm nhập trái phép ngay tại thiết bị cá nhân.

Chức năng chính của IDS/IPS

* Phát hiện các mối đe dọa: IDS/IPS sử dụng các cơ chế phát hiện dựa trên chữ ký, hành vi, hoặc các phương pháp phát hiện bất thường để nhận diện các mối đe dọa.
* Cảnh báo và thông báo: Khi phát hiện một hành vi xâm nhập, IDS sẽ cảnh báo quản trị viên mạng, trong khi IPS có thể thực hiện các hành động ngay lập tức để ngăn chặn mối đe dọa.
* Phân tích và điều tra: Cung cấp các thông tin chi tiết về các sự kiện bảo mật để giúp các quản trị viên phân tích và điều tra nguồn gốc của các cuộc tấn công.
* Phòng thủ chủ động: IPS có khả năng tự động chặn các kết nối mạng đáng ngờ hoặc giới hạn quyền truy cập để bảo vệ hệ thống.

IDS và IPS là hai công cụ quan trọng trong việc bảo vệ an ninh mạng, giúp phát hiện và ngăn chặn các hành vi xâm nhập trái phép. Mặc dù có những thách thức nhất định, việc triển khai IDS/IPS sẽ tăng cường khả năng bảo vệ hệ thống mạng khỏi các mối đe dọa an ninh.

## **Kết luận chương I**

Chương I đã cung cấp một cái nhìn tổng quan về dịch vụ website, đi sâu vào các nguy cơ mà các ứng dụng website phải đối mặt, và đề xuất những giải pháp bảo mật toàn diện. Trong bối cảnh internet ngày càng phát triển và trở thành một phần không thể thiếu trong cuộc sống hàng ngày, việc bảo vệ ứng dụng website khỏi các mối đe dọa tiềm tàng là một nhiệm vụ cực kỳ quan trọng.

Trước tiên, chúng ta đã tìm hiểu tổng quan về dịch vụ website, hiểu rõ hơn về sự phát triển và vai trò quan trọng của các ứng dụng web trong việc cung cấp dịch vụ cho người dùng. Các dịch vụ này không chỉ mang lại sự tiện lợi mà còn mở ra nhiều cơ hội kinh doanh mới, đồng thời cũng đặt ra những thách thức về an ninh mạng mà chúng ta phải đối mặt.

Các nguy cơ đối với ứng dụng website, như tấn công Injection và thực thi các tập tin độc hại, đã được phân tích chi tiết. Injection Attack là một trong những phương thức tấn công phổ biến và nguy hiểm nhất, cho phép kẻ tấn công chèn mã độc vào hệ thống thông qua các đầu vào không được kiểm tra kỹ lưỡng. Tấn công thực thi các tập tin độc hại, mặt khác, lợi dụng các lỗ hổng bảo mật để tải và thực thi các phần mềm gây hại trên máy chủ hoặc thiết bị của người dùng.

Để đối phó với những mối đe dọa này, các giải pháp đảm bảo an toàn cho ứng dụng website đã được đề xuất. Kỹ thuật lập trình an toàn là nền tảng, giúp giảm thiểu các lỗ hổng bảo mật từ giai đoạn thiết kế và phát triển. Bảo vệ hệ thống tệp tin và cơ sở dữ liệu là những bước quan trọng tiếp theo, ngăn chặn việc truy cập trái phép và bảo vệ dữ liệu nhạy cảm khỏi bị đánh cắp hoặc hư hại.

Sử dụng các thiết bị và phần mềm bảo mật cho ứng dụng website cũng là một phương thức hiệu quả. Tường lửa đóng vai trò như một lá chắn, kiểm soát và lọc lưu lượng truy cập vào và ra khỏi hệ thống. Trong đó, tường lửa của CheckPoint và thiết bị tường lửa ứng dụng web của Imperva được đánh giá cao về khả năng bảo vệ mạnh mẽ và linh hoạt. Việc sử dụng phần mềm tường lửa, kết hợp với các thiết bị phát hiện và phòng chống xâm nhập (IDS/IPS), giúp phát hiện sớm và ngăn chặn các cuộc tấn công trước khi chúng gây ra thiệt hại nghiêm trọng.

Tổng hợp lại, chương I đã xây dựng một nền tảng kiến thức vững chắc về bảo mật ứng dụng website, từ việc nhận diện các nguy cơ tiềm tàng đến việc áp dụng các giải pháp bảo vệ hiệu quả. Những kiến thức này không chỉ giúp nâng cao nhận thức về an ninh mạng mà còn cung cấp các công cụ và phương pháp cụ thể để bảo vệ các ứng dụng web khỏi các mối đe dọa không ngừng biến đổi. Qua đó, đảm bảo sự ổn định và an toàn cho các dịch vụ trực tuyến, góp phần phát triển một môi trường internet an toàn và tin cậy hơn cho tất cả mọi người.

# CHƯƠNG II: TÌM HIỂU MODSECURITY

* + 1. Mod Security là gì ?

ModSecurity, hay còn được biết đến với tên gọi Modsec, là một tường lửa ứng dụng cho web [mã nguồn mở](https://vinahost.vn/ma-nguon-mo-la-gi/) (WAF).

Ban đầu, nó được phát triển như một mô đun cho [máy chủ Apache](https://vinahost.vn/apache-la-gi/) HTTP. Sau đó, Mod Security đã được mở rộng để cung cấp chức năng lọc phản hồi và yêu cầu giao thức truyền văn bản, cùng với nhiều tính năng bảo mật khác.

Mod Security có khả năng hoạt động trên nhiều nền tảng khác nhau như máy chủ Apache HTTP, [Microsoft IIS](https://vinahost.vn/iis-la-gi/), và [Nginx](https://vinahost.vn/nginx-la-gi/).

* + 1. Tính năng và vai trò của Mod Security
    2. Giám sát thời gian thực và phát hiện tấn công

Modsecurity không chỉ cung cấp khả năng ghi log mà còn có khả năng theo dõi luồng dữ liệu HTTP theo thời gian thực để phát hiện các cuộc tấn công. Do đó, hệ thống chạy Modsecurity được xem là một công cụ phát hiện xâm nhập web, cho phép phản ứng đối với các sự kiện đáng ngờ diễn ra trên[hệ thống web](https://vinahost.vn/website-la-gi/).

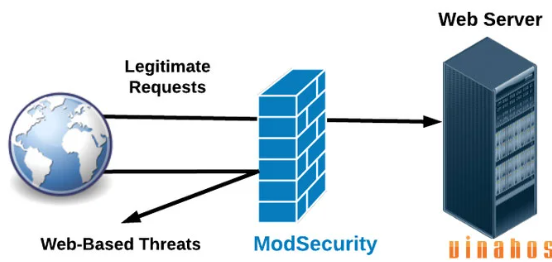
* + 1. Chống tấn công và vá lỗi

Modsecurity cũng có khả năng thực hiện hành động ngay lập tức để đối mặt với các cuộc tấn công vào các ứng dụng web. Có 03 phương pháp thông thường được áp dụng:

**Sử dụng mô hình bảo mật NSM (Negative Security Model):** Theo dõi những yêu cầu không bình thường, các hành vi khác thường nhắm vào ứng dụng web. Modsecurity ghi lại các điểm bất thường này trong mỗi yêu cầu, bao gồm [địa chỉ IP](https://vinahost.vn/ip-la-gi/), phiên ứng dụng, và tài khoản người dùng. Các yêu cầu có điểm bất thường cao sẽ được log hoặc loại bỏ.

**Mô hình Bảo mật PSM (Positive Security Model):** Với ưu điểm là khi triển khai, chỉ có những yêu cầu hợp lệ mới được chấp nhận, và tất cả những thứ khác sẽ bị loại bỏ. Phương pháp tiếp cận này thường đem lại hiệu quả tốt nhất đối với các ứng dụng được sử dụng phổ biến và ít khi được cập nhật.

**Phương pháp Known weaknesses and vulnerabilities:** Modsecurity trở thành một công cụ lý tưởng cho quá trình sửa chữa. Thường thì nhà phát triển sẽ mất nhiều tuần để sửa chữa các lỗ hổng trong ứng dụng. Nhưng với Modsecurity, ứng dụng có thể được chắp vá từ bên ngoài mà không cần can thiệp vào mã nguồn của ứng dụng (thậm chí không yêu cầu truy cập đến nó), làm cho hệ thống vẫn an toàn cho đến khi lỗi của ứng dụng được khắc phục.

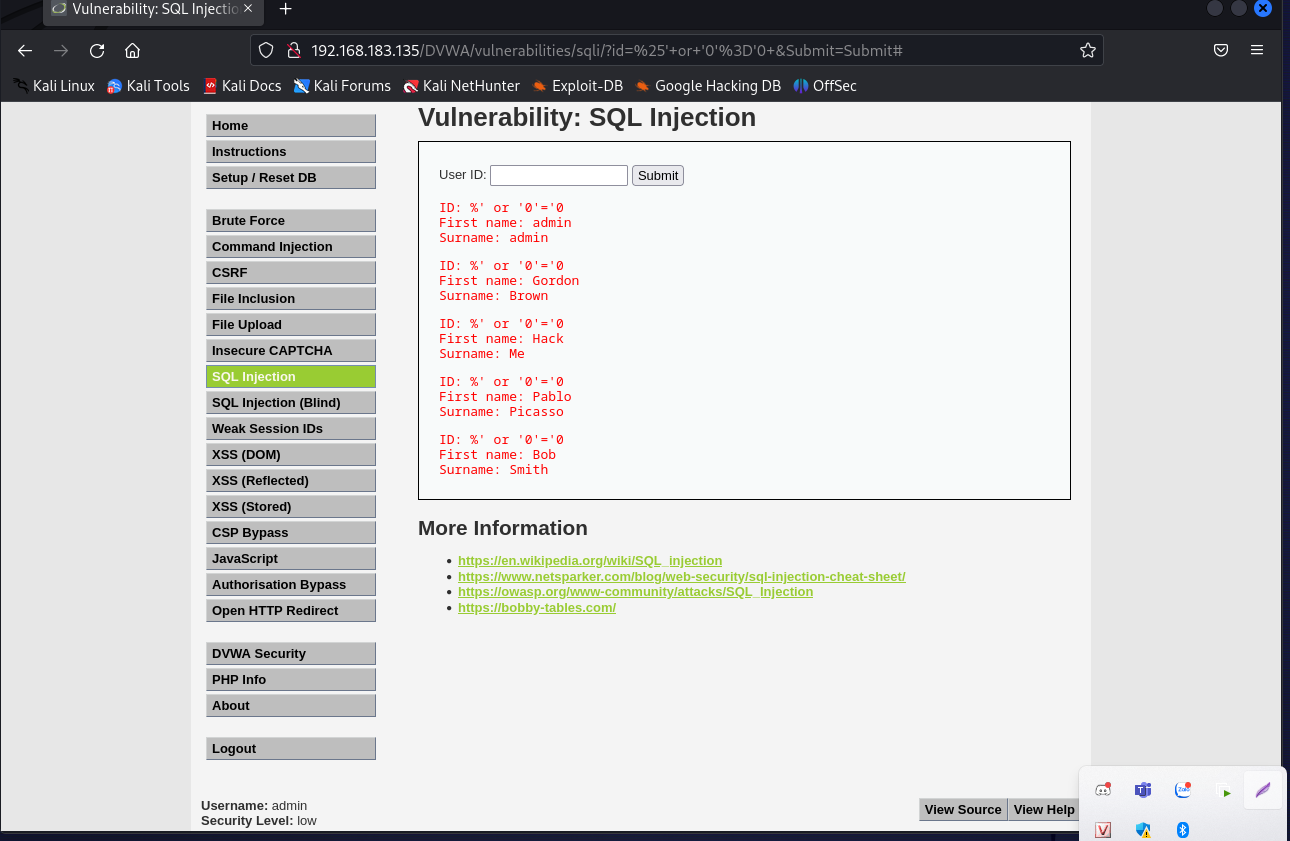


* + 1. Khả năng linh hoạt của Core Rule Set

ModSecurity sử dụng một bộ quy tắc được gọi là Core Rule Set (CRS), một tập hợp các quy tắc được thiết kế để ngăn chặn các mô hình tấn công phổ biến.

* + 1. Khi chưa tích hợp Mod Security có thể bị SQL Injection và tấn công XSS
    2. Thử nghiệm SQL Injection

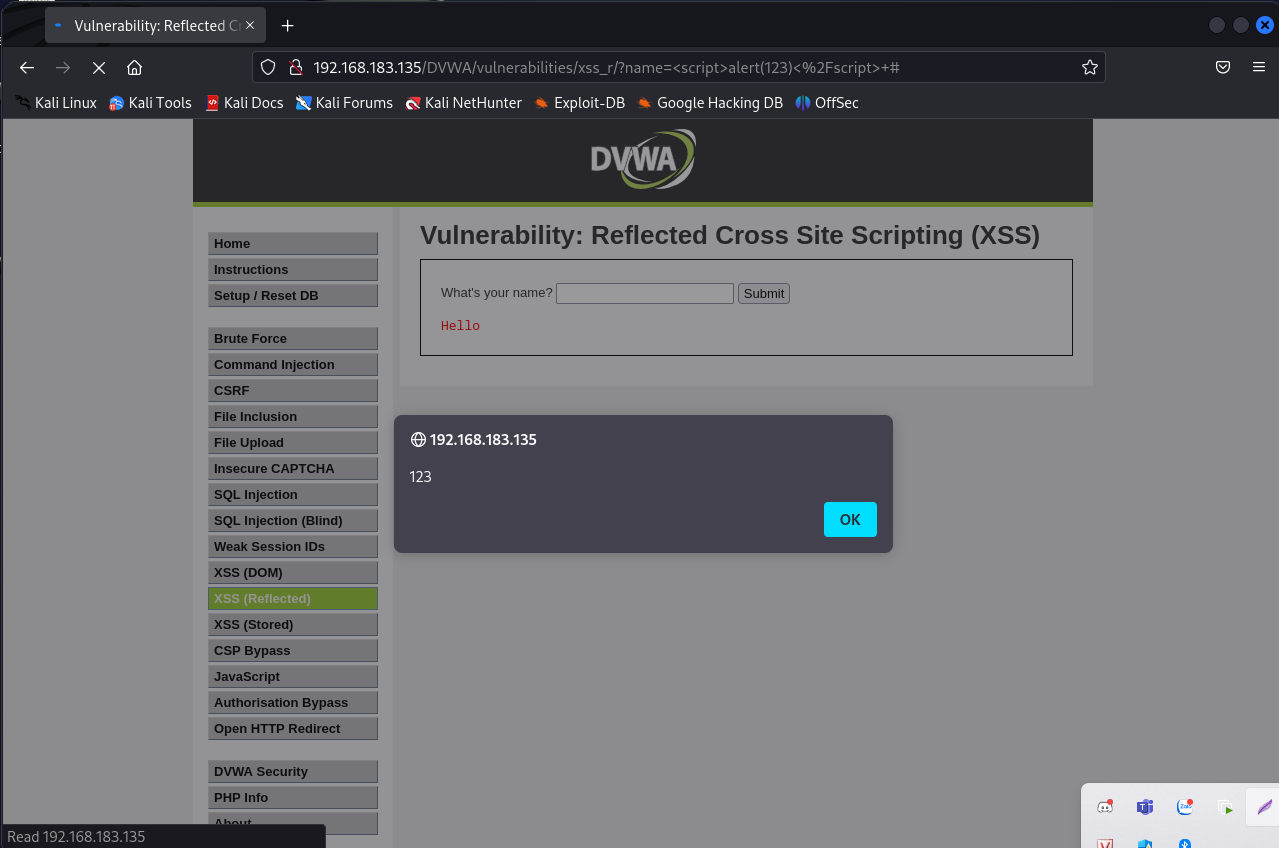
Thực hiện SQL Injection bằng cách nhập **%’ or ‘0’=’0** và nhấn **submit**.



Kết quả sẽ trả về thông tin của tất cả người dùng có trong cơ sở dữ liệu, vì câu truy vấn tại thời điểm này được hiểu là: **SELECT \* FROM users WHERE user\_id = ‘%’ or ‘0’=’0′**

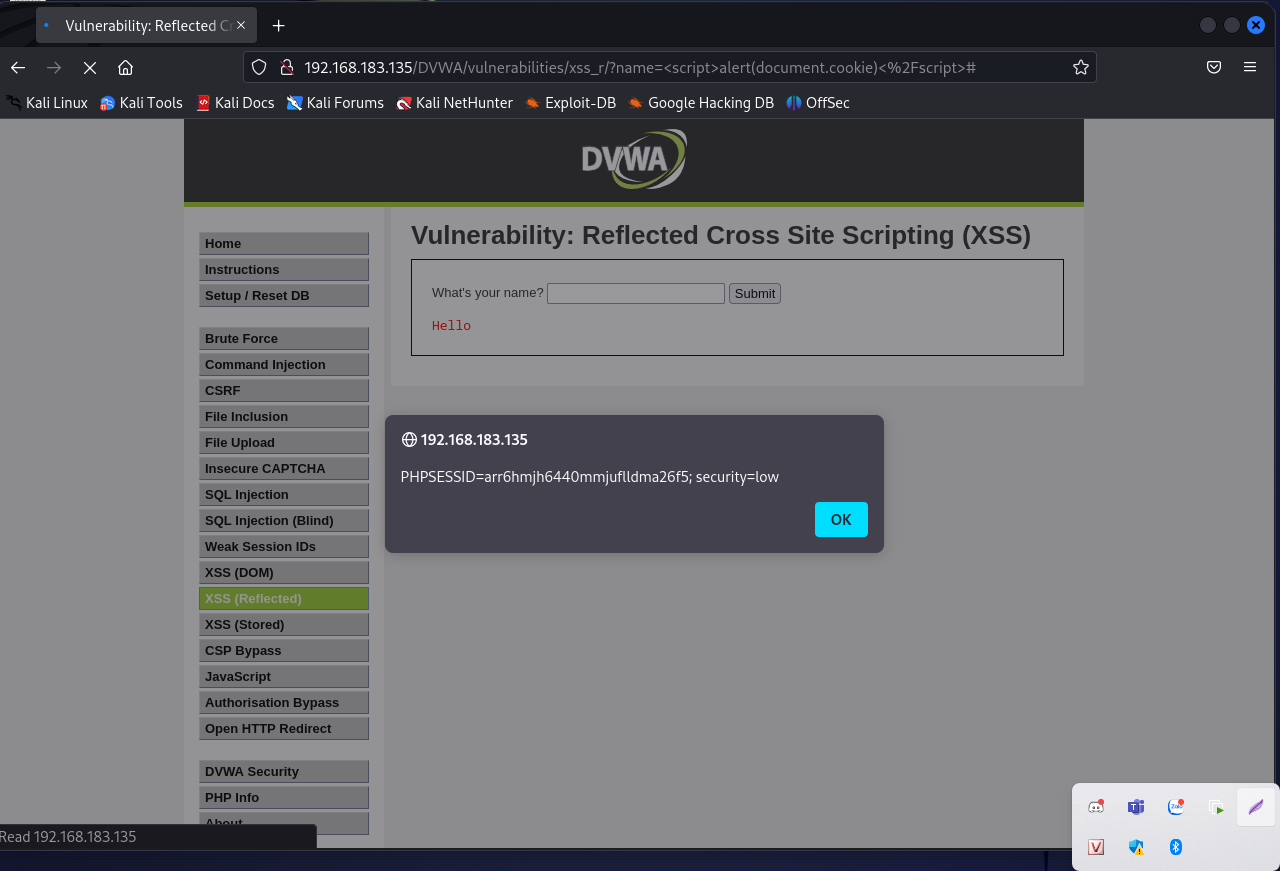
* + 1. Thử nghiệm XSS

Trang DVWA <http://localhost/DVWA/vulnerabilities/xss_r/> bị ảnh hưởng bởi XSS được phản ánh trong tham số **name**. Điều này có thể được nhìn thấy trong hình bên dưới khi chúng ta tiêm mã JavaScript **<script>alert(123)</script>** và nó được phản ánh và thực thi trong trang response.



## **XSS Attack: Chiếm phiên người dùng**

Session cookie là thông tin nhạy cảm, nếu bị xâm nhập, có thể cho phép kẻ tấn công mạo danh người dùng hợp pháp và có quyền truy cập vào phiên web hiện tại của anh ta. Cuộc tấn công này được gọi là tấn công phiên **(session hijacking)**. Mã JavaScript đang chạy trong trình duyệt có thể truy cập session cookie (khi chúng thiếu HTTPOnly) bằng cách gọi document.cookie. Vì vậy, nếu chúng ta tiêm khối mã sau: [**<script>alert(document.cookie)</script**](http://localhost:81/DVWA/vulnerabilities/xss_r/?name=%3cscript%3ealert(document.cookie)%3c/script)**>** vào tham số **name**, trang dễ bị tấn công sẽ hiển thị giá trị cookie hiện tại trong khối cảnh báo:



* + 1. Cách hoạt động của Mod Security

Modsecurity hoạt động như một hệ thống phát hiện và ngăn chặn xâm nhập (IDPS – Intrusion Detection and Prevention System). Trong quá trình phát hiện xâm nhập, Modsecurity giám sát luồng dữ liệu HTTP, HTTTPS khi đi vào (Request) hoặc ra khỏi [Web Server](https://vinahost.vn/web-server-la-gi/) (Response), phân tích chúng để tìm kiếm các dấu hiệu của sự kiện bất thường.

Điều này bao gồm việc phát hiện các vi phạm hoặc đe dọa sắp xảy ra đối với chính sách an ninh, chính sách truy cập, hoặc hoạt động không tuân thủ chính sách chuẩn của máy chủ web. Sự kiện bất thường có thể xuất phát từ nhiều nguyên nhân, bao gồm cả việc kẻ tấn công lợi dụng lỗ hổng SQL injection để truy cập hệ thống một cách trái phép và người dùng hợp pháp lạm dụng quyền hoặc cố gắng thêm vào các quyền mà họ không được phép.

Modsecurity thực hiện phát hiện tấn công thông qua hai phương pháp:

* Dựa vào các mẫu
* Dựa vào các dấu hiệu bất thường

Một trong những cách để phát hiện các cuộc tấn công là sử dụng phương pháp dựa trên mẫu (Pattern) và dấu hiệu (Signature) trong request gửi đến Web server. Các dấu hiệu và mẫu nguy hiểm của các loại tấn công được cấu hình sẵn trong rule. Khi luồng dữ liệu HTTP được kiểm tra và phát hiện chứa các dấu hiệu, mẫu nguy hiểm này, nó sẽ bị cấm. Các hành động kèm theo, như ghi log và mô tả rõ ràng về loại tấn công của request này cũng được thực hiện.

Các mẫu này cụ thể là các ký tự trong các gói tin request hoặc Response, chẳng hạn như request URI, host, user-agent, access-encoding, và cookie. Hầu hết các dạng tấn công đã biết có thể dễ dàng thu thập mẫu thông qua quá trình Capture. Những mẫu đặc biệt trong các kiểu tấn công này sẽ được tích hợp vào các rule.

Từ các thông báo lỗi, người quản trị cũng có thể nhanh chóng khắc phục với sự hỗ trợ của Modsecurity. Các lỗi sẽ hiện ra màn hình dưới sự kiểm soát của Modsecurity, do đó, hacker gặp khó khăn khi cố gắng biết và khai thác các hệ thống này.

* + 1. Kiến trúc và thành phần của Mod Security
    2. Tổng quan về Mod Security core ruler

Modsecurity là một tường lửa ứng dụng dành cho website, do đó, tự nó không có nhiều khả năng phòng chống tấn công mặc định và cũng không được tích hợp các quy tắc cấu hình cẩn thận. Để tận dụng được ưu điểm từ các tính năng có sẵn của Modsecurity, các nhà phát triển đã thiết lập một bộ luật chặt chẽ, đầy đủ và hoàn toàn miễn phí.

Khác với các hệ thống dò tìm và phát hiện mối nguy hiểm xâm nhập khác, những core rule này có khả năng cung cấp sự bảo vệ chống lại những yếu điểm chưa được phát hiện trong các ứng dụng web.

* + 1. Nội dung của core ruler

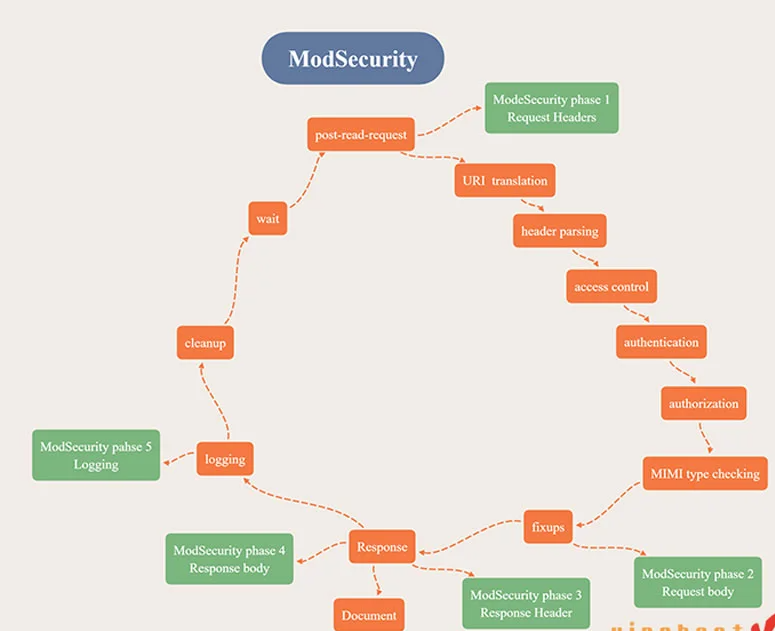
Để đảm bảo sự bảo vệ toàn diện cho website, core rule bao gồm các thành phần sau:

* **Thành phần bảo vệ luồng dữ liệu HTTP:**Chức năng của thành phần này là phát hiện các hành vi vi phạm giao thức HTTP và thực hiện các quy tắc xử lý đã được xác định theo core rule.
* **Phòng chống các cuộc tấn công phổ biến vào Web server:** Tính năng này giúp phát hiện các cuộc tấn công vào ứng dụng bảo mật của web server.
* **Một số tính năng khác:** Tự động phát hiện Bots, ngăn chặn và phát hiện truy cập từ Trojan horses, nhận diện lỗi tấn công và các thông báo từ web server.
  + 1. Cấu trúc của core ruler

Cấu hình của Rule modsecurity bao gồm các thông tin khác nhau, yêu cầu đặc biệt về nội dung. Dưới đây là một số cấu trúc của Rule modsecurity:

* Cấu trúc chứa các chỉ thị, hàm chuyển đổi, biến và signature, cùng với các action tương ứng cho phép hoặc không cho phép.
* Thông tin về logic để phát hiện các cuộc tấn công.
* Thông tin thiết lập chính sách, quyết định hành động xử lý cụ thể khi phát hiện cuộc tấn công.

Modsecurity cho phép xử lý từng thành phần tấn công khác nhau. Core rule được xây dựng dựa trên mẫu có sẵn và có thể tự thay đổi, thêm hoặc giảm bớt các core rule.



* + 1. Rules
    2. Cấu trúc của rules

SecRule<TARGET\_VARIABLE>"<PATTERN>""id:<UNIQUE\_ID>,phase:<PH ASE\_NUMBER>,deny,status:<STATUS\_CODE>,msg:'<LOG\_MESSAGE>'"

Trong đó:

* <TARGET\_VARIABLE>: biến mục tiêu, như ARGS, ARGS\_NAMES, REQUEST\_HEADERS, REQUEST\_BODY,…
* <PATTERN>: biểu thức chính quy để phát hiện các mẫu ký tự đáng ngờ.
* <UNIQUE\_ID>: ID duy nhất cho quy tắc
* <PH ASE\_NUMBER>: giai đoạn thực thi quy tắc (1-5)
* <STATUS\_CODE>: mã trạng thái HTTP trả về khi từ chối yêu cầu
* <LOG\_MESSAGE>: thông báo ghi nhật ký khi quy tắc được kích hoạt.

VD: SecRule ARGS "('|--|;|\/\\*|\\*\/)" "id:1,phase:2,deny,status:403,msg:'Phat Hien Cac Ky Tu Dac Biet'"

* + 1. Variables
* SecRule ARGS divty
* Có thể dung 1 hay nhiều variables
* SecRule ARGS|REQUEST\_HEADERS: User-Agent divty
* 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 (2006)
* TIME\_MON : Tháng hiện tại (2)
* 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: 20060220144530 : 20/02/2006 14h 45' 30"
* API\_VERSION
* THE\_REQUEST : dòng đầu tiên của request. vd: GET/ HTTP/1.1
  + 1. Collections
* Một variables có thể bao gồm 1 hay nhiều phần dữ liệu. Khi variable có nhiều hơn 1 giá trị thì ta gọi nó là collection
* Ví dụ: với variable ARGS ta có 2 thông số p, và q
* SecRule ARGS:p dirty
* SecRule ARGS:q dirty
  + 1. Operators
* Sử dụng @ để chỉ ra đây là một operation SecRule ARGS "@rx dirty"
* Sử dụng !@ để chỉ ra một operation negation
* SecRule &ARGS"!@rx ^OS"
* Ở đây chúng ta đề cập đến một operation là rx (regular expression).
* RX quy định các thể hiện
* [Jj]oy : thể hiện mọi chuỗi có chứa Joy hoặc joy
* [0-9] : mọi số từ 0 tới 9
* [a-zA-Z] : mọi chữ từ a đến z cả chủ thường lẫn in hoa
* ^:1
* A : bắt đầu một chuỗi
* S : kết thúc chuỗi
* abcS : chuỗi chỉ bao gồm từ abc
* . mọi kĩ tự
* p.t : ví dụ như pat.pet, pzt....
  + 1. Actions

Khi request vi phạm một rule nào đó thì mod\_security sẽ thực thi một hành động (action). Khi action không được chỉ rõ trong rule thì rule đó sẽ sử dụng default action . Có 3 loại actions :

* Primary Actions

Primary actions sẽ quyết định cho phép request tiếp tục hay không. Mỗi rule chỉ có một primary action. Có 5 primary actions :

+ deny : Request sẽ bị ngắt, mod\_security sẽ trả về HTTP status code 500 hoặc là status code của bạn thiết lập trong chỉ thị status:

+ pass : Cho phép request tiếp tục được xử lý ở các rules tiếp theo

+ Allow. Cho phép truy cập ngay lập tức và bỏ qua các phases khác (trừ phases logging). Nếu muốn chỉ cho qua phase hiện tại thì cần chỉ rõ allow:phase Khi đó sẽ vẫn được kiểm tra bởi các luật tại các phases sau. Chỉ cho phép truy cập tới các request phases: allow:request, nó sẽ cho qua phase 1,2 và vẫn kiểm tra ở phase 3 trở đi

+ redirect : Redirect một request đến một url nào đó.

* Secondary Actions

Secondary actions sẽ bổ sung cho Primary actions, một rule có thể có nhiều Secondary actions

+ status : n khi một Request vi phạm một rule nào đó thì mod\_security có thể trả về các HTTP status code n thay vì status code 500 mặc định.

+ exec : thực thi một lệnh nào đó nếu một request vi phạm

+ log : ghi log những request vi phạm rule

+ nolog : không ghi log

pause : n mod\_security sẽ đợi một thời gian n ms rồi mới trả về kết quả

+ Flow Actions

chain: kết nối 2 hay nhiều rules lại với nhau

skipnext:n mod\_security sẽ bỏ qua n rules theo sau nó

* Default Action

Khi một rule không chỉ rõ action thì rule đó sẽ dùng default action được thiết lập trong SecDefaultAction.

Ví dụ : SecDefaultAction "phase:2,deny,log,status:403"

* + 1. Logging
    2. Debug Log

Sử dụng SecDebugLog directive lựa chọn file để ghi lại các thông tin debug

*SecDebugLog logs/modsec-debug.log*

Bạn có thể thay đổi mức độ chi tiết các thông tin được log thông qua directive :

**SecDebugLogLevel 4**

Giá trị log có thể thay đổi từ 0-9 :

0 - no logging.

1 - errors (intercepted requests) only.

2 - warnings.

3 - notices.

4 - details of how transactions are handled.

5 - as above, but including information about each piece of information handled.

9 - log everything, including very detailed debugging information

* + 1. Audit logging

Apache log ít thông tin vì thế nó không cho phép chúng ta có thể lần ngược các bước của kẻ tấn công. Mod\_security hỗ trợ audit loging với đầy đủ thông tin và từ đó có thể lần ngược lại quá trình của kẻ tấn công. cũng như là chỉnh sửa các rules cho hợp lý tránh bị “false positive”. Có 2 directives:

*Sec.AuditEngine On* : bật audit log lên

*SecAuditLog logs/audit.log*: chỉ ra file lưu trữ log chính

Ngoài ra còn có

*Sec4uditLog2 logs/audit2.log* :chi ra file lưu trữ log phụ

* + 1. Tùy biến thông tin log

SecAuditEngine chấp nhận 3 giá trị sau :

On – log tất cả các requests

Off – không log

RelevantOnly – chỉ log những gì được sinh ra bởi các bộ lọc

Ngoài ra mod\_security còn hỗ trợ log dựa vào status code, ví dụ bạn cần log lại những requests gây ra lỗi 5xx :

*SecAuditLogRelevant Status ^5*

* + 1. Xây dựng chính sách trên Mod Security chống lại một số tấn công
    2. SQL Injection
* Các từ khóa chính thường sử dụng trong tấn công SQL Injection và các regular expressions tương ứng:

UNION SELECT union\s+select

UNION ALL SELECT union\s+all\s+select

INTO OUTFILE into\s+outfile

DROP TABLE drop\s+table

ALTER TABLE alter\s+table

LOAD\_FILE load\_file

SELECT \* select\s+\*

\s : được định nghĩa trong PCRE là một regular expression cho phép phát hiện mọi khoảng trắng và cả các mã thay thế (%20)

* Để chống lại tấn công SQL Injection, ta dựa vào các đặc điểm trên từ đó đưa ra rule sau:

*SecRule ARGS "union\s+select" "t:lowercase, deny,msg: 'SQL Injection""*

*SecRule ARGS "union\s+all\s+select" "t:lowercase, deny,msg:"SQL Injection""*

*SecRule ARGS "into\s+outfile" "t:lowercase, deny,msg:"SQL Injection""*

*SecRule ARGS "drop\s+table" "t: lowercase, deny, msg: 'SQL Injection""*

*SecRule ARGS "alter\s+table" "t: lowercase, deny, msg: 'SQL Injection""*

*SecRule ARGS "load\_file" "t:lowercase, deny, msg: 'SQL Injection""*

*SecRule ARGS "select\s+from" "t:lowercase, deny, msg: 'SQL Injection""*

* + 1. XSS Attack

Đề ngăn chặn tấn công XSS, chúng ta phải đảm bảo tất cả dữ liệu mà người dùng gởi lên đều được cản lọc. Cụ thể, chúng ta có thể thay thế hoặc loại bỏ các ký tự, các chuỗi thường được dùng trong tấn công XSS như đấu ngoặc góc (< và >), script...

Dưới đây là danh sách các ký tự nên mã hoả khi được client cung cấp để lưu vào cơ sở dữ liệu.

Bảng 1: Các ký tự tên mã hóa để ngăn chặn tấn công XSS

|  |  |
| --- | --- |
| Ký tự | Mã hóa HTML |
| < | &lt; |
| > | &gt; |
| ( | &#40 |
| ) | &#41 |
| # | &#35 |
| & | &amp; |
| “ | &#39 |

Nếu chúng ta muốn ngăn chặn tấn công với ModSecurity, dưới đây là các đoạn script XSS phổ biến và các biểu thức chính quy để ngăn chặn người dung request chứa các chuỗi này:

Bảng 2: Các script XSS và biểu thức chính quy

|  |  |
| --- | --- |
| Các script | Biểu thức chính quy |
| <script | <script |
| eval ( | eval\s\* ( |
| onMouseOver | onmouseover |
| onMouseOut | onmouseout |
| onMouseDown | onmousedown |
| onMouseMove | onmousemove |
| onClick | onclick |

Sau đây là rule thực hiện:

*SecRule ARGS "alert\s+\(" "t: lowercase, deny, msg: 'XSS""*

*SecRule ARGS "&\{+}" "t: lowercase, deny, msg: 'XSS"""*

*SecRule ARGS "<+>" "t:lowercase, deny,msg: 'XSS""*

*SecRule ARGS "javascript:" "t:lowercase, deny,msg: 'XSS""*

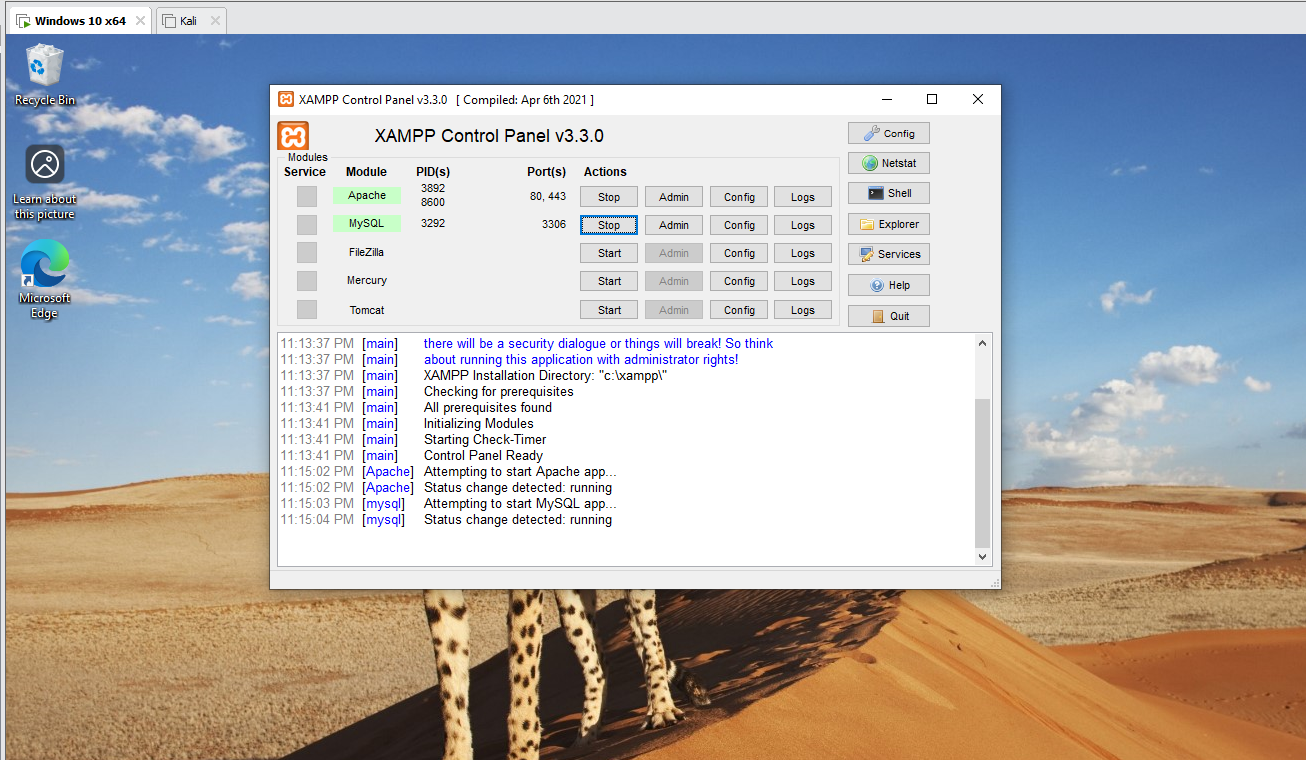
*SecRule ARGS "vbscript:" "t:lowercase, deny,msg: 'XSS""*

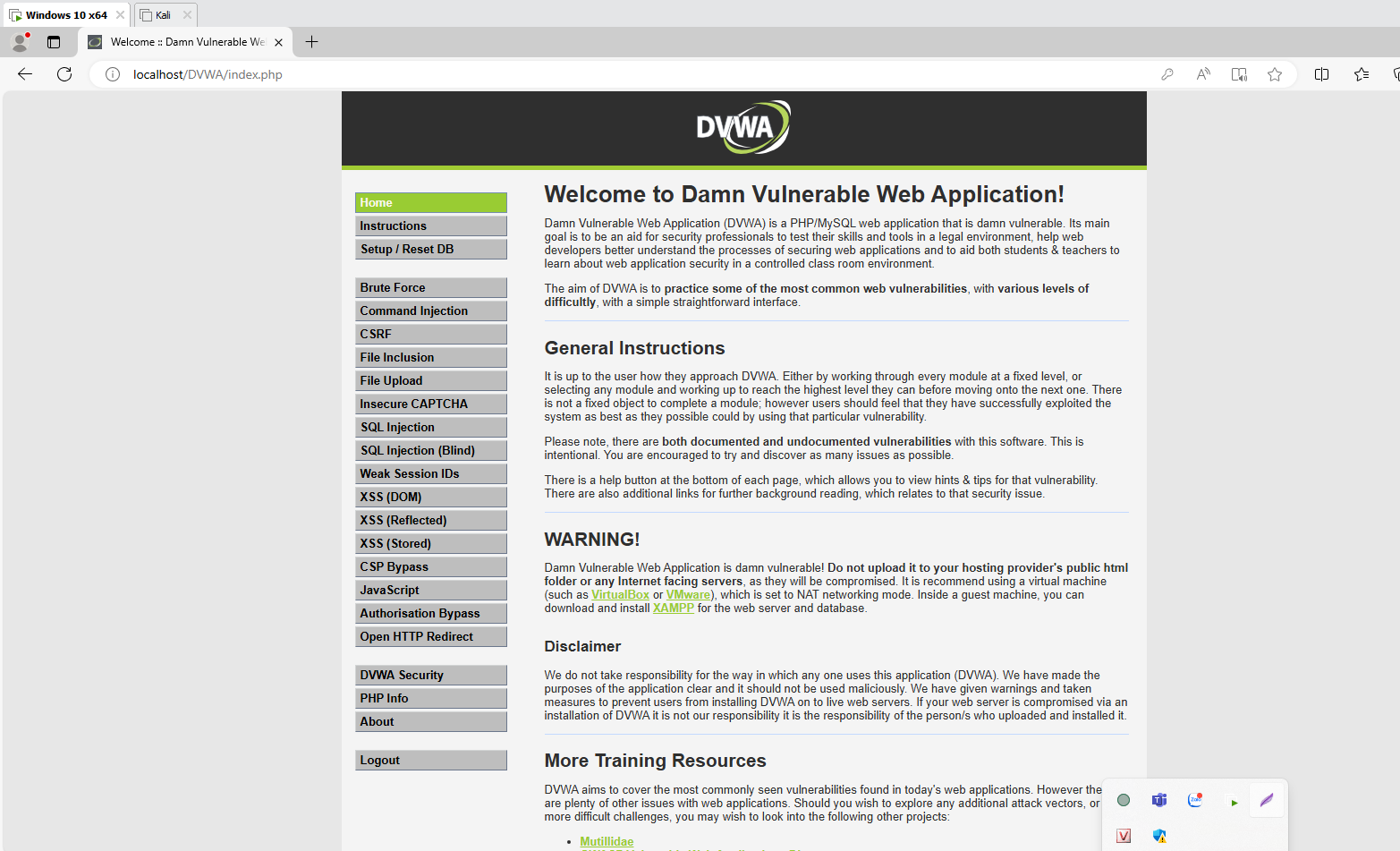
**CHƯƠNG III: TRIỂN KHAI HỆ THỐNG WAF MODSECURITY**

**3.1 Cài đặt Mod Security**

**3.1.1 Cài đặt ModSecurity trên Apache server**

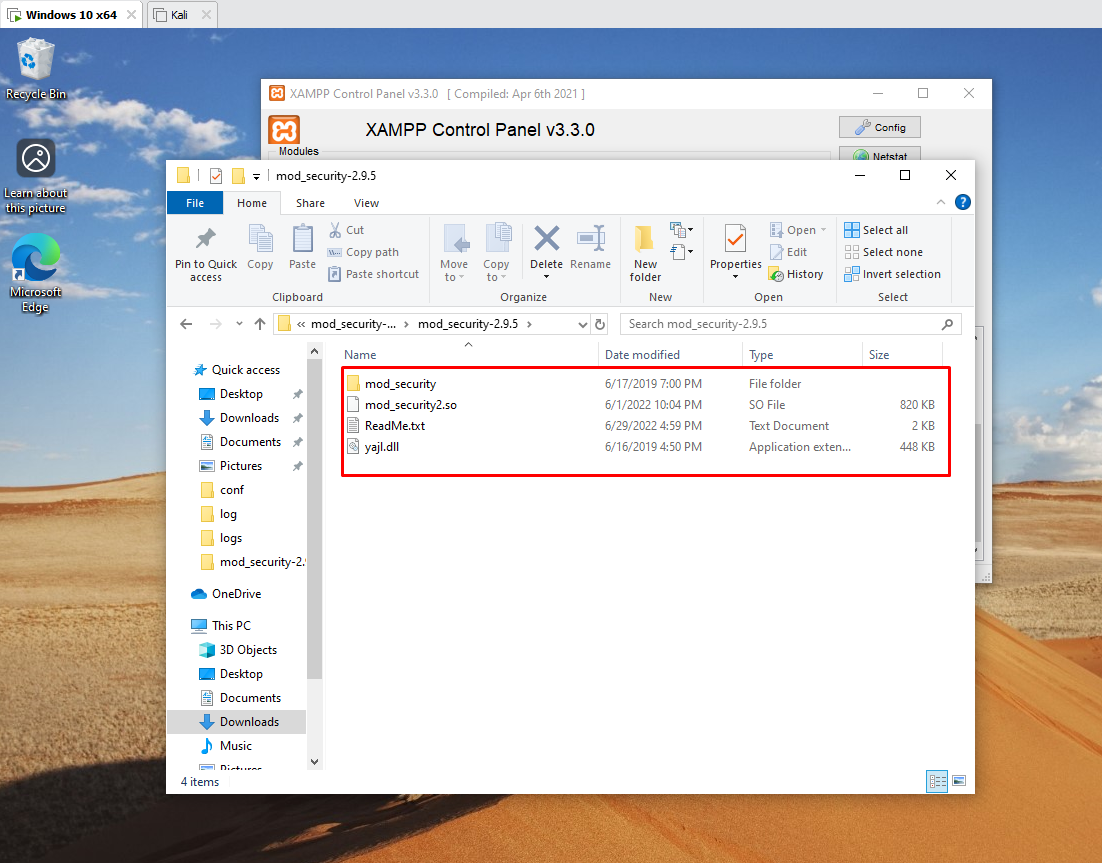
Bước 1: Cài Xampp trên Windows 10, cài và cấu hình DVWA



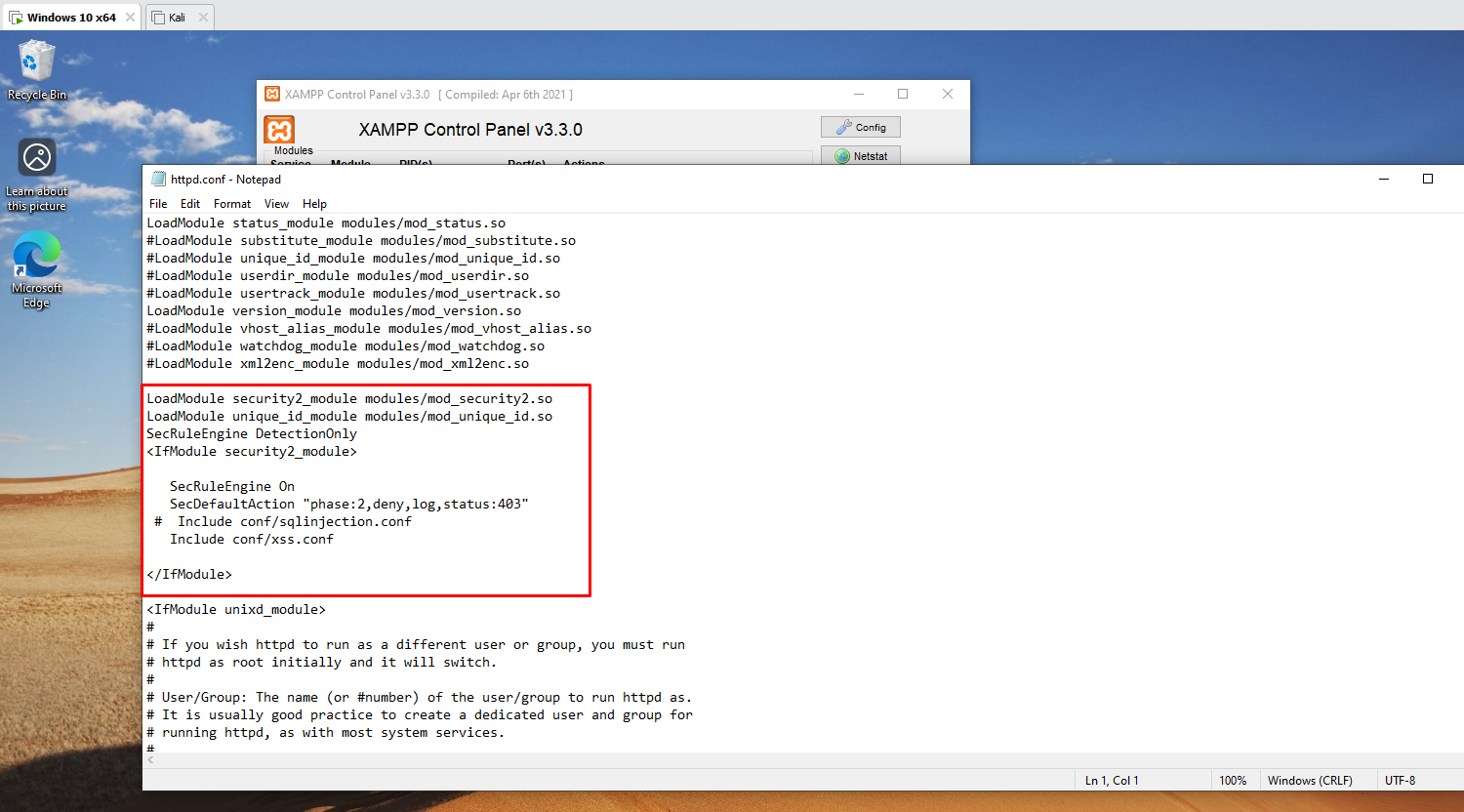


Bước 2: Cài và cấu hình ModSecurity trên Apache

* Tải xuống mô đun WAF ModSecurity theo đường dẫn: <https://www.apachelounge.com/download/VS17/modules/mod_security-2.9.7-win64-VS17.zip> tiến hành giải nén:



* Sao chép tệp yajl.dll tới đường dẫn: *C:\xampp\apache\bin*
* Sao chép mô đun ModSecurity tới Apache bằng cách di chuyển tệp mod\_security2.so đến thư mục chứa các mô đun của Apache theo đường dẫn *C:\xampp\apache\modules*
* Thêm đoạn mã sau vào vào file cấu hình chính Apache httpd.conf để thực hiện tải mô đun ModSecurity và bật tính năng kiểm tra và lọc các request cùng cấu hình hành động mặc định cho các rule:



**3.2.Thử nghiệm phòng chống tấn công với ModSecurity**

**3.2.1. Thử nghiệm phòng chống tấn công SQL Injection**

- Công thức tổng quát để xây dựng các quy tắc:

SecRule<TARGET\_VARIABLE>"<PATTERN>""id:<UNIQUE\_ID>,phase:<PHASE\_NUMBER>,deny,status:<STATUS\_CODE>,msg:'<LOG\_MESSAGE>'"

+ Phát hiện các ký tự đặc biệt thường dùng trong SQL Injection

SecRule ARGS “(‘|\”| -- |;|\/\\*|\\*\/)” “id:1001,phase:2,deny,status:403,msg: ‘Da phat hien các ky tu dac biet thuong duoc su dung trong SQL Injection’”

+ Phát hiện các từ khóa SQL thông dụng

SecRuleARGS“(?i:select|union|insert|update|delete|drop|exec|truncate|alter)"“id:1002,phase:2,deny,status:403,msg:’tu khoa sql duoc phat hien’”

+ Phát hiện các chuỗi SQL Injection thông dụng

SecRule ARGS "(?i:(?:')|(?:--

)|(\bOR\b.+\bLIKE\b)|(?:\/\\*)|(?:\\*\/)|(\bUNION\b.+\bSELECT\b))" "id:1003,phase:2,deny,status:403,msg:'Detected common SQL Injection patterns'"

+ Phát hiện các biểu thức điều kiện thường dùng

SecRule ARGS "(?i:\bAND\b.+\b=\b)" "id:1004,phase:2,deny,status:403,msg:'Detected boolean expressions commonly used in SQL Injection'"

+ Phát hiện các mẫu tấn công dựa trên số học

SecRule ARGS "(?i:\b1\s\*=\s\*1\b|\b0\s\*=\s\*0\b)" "id:1008,phase:2,deny,status:403,msg:'Detected common arithmetic expressions used in SQL Injection'"

+ Phát hiện các truy vấn tham số hóa

SecRule ARGS "(?i:\bEXEC\b|\bEXECUTE\b)" "id:1007,phase:2,deny,status:403,msg:'Detected execution of parameterized queries'"

+ Phát hiện mã hex

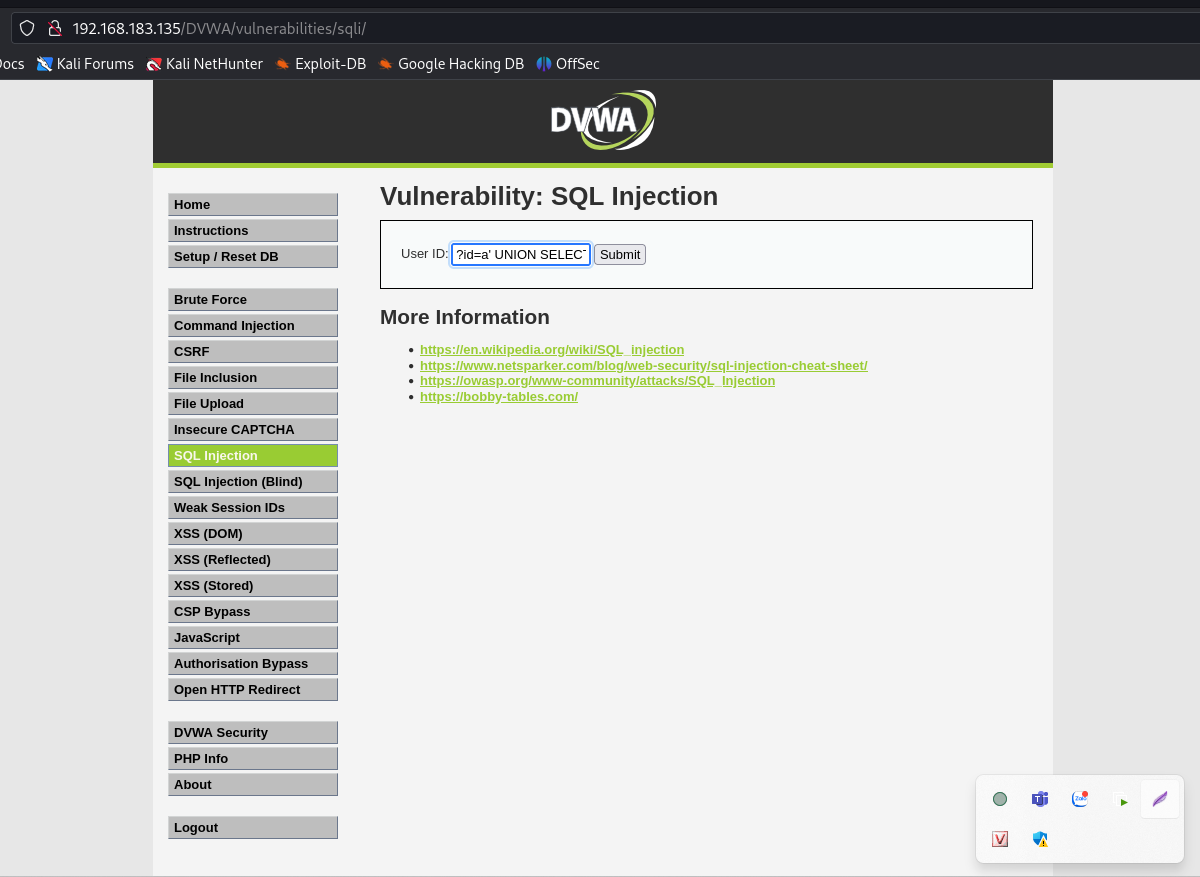
SecRule ARGS "[\x27\x22\x3d\x2f]" "id:3,phase:2,deny,status:403,msg:'Phat Hien Ma Hex'"

+ Phát hiện comments

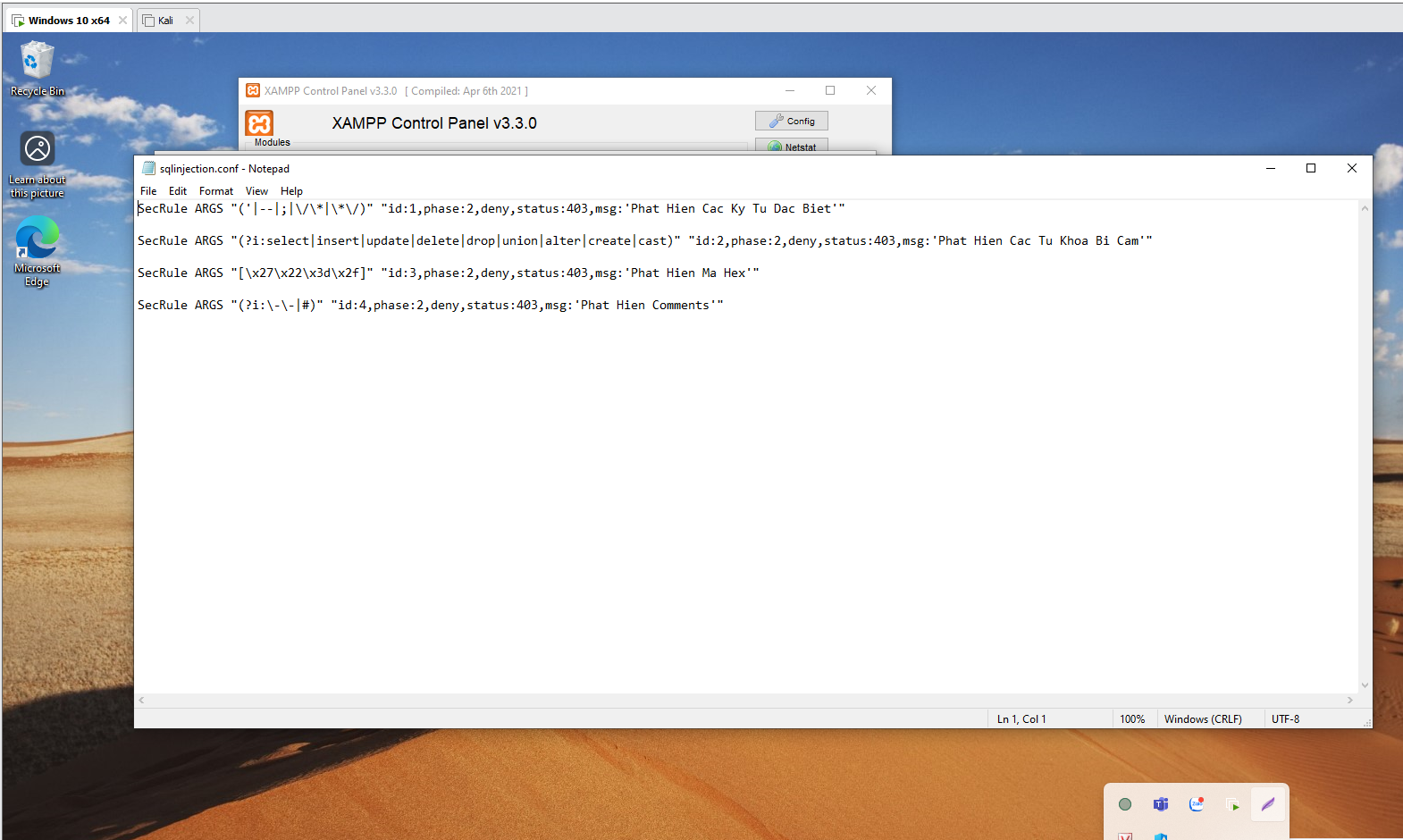
SecRule ARGS "(?i:\-\-|#)" "id:4,phase:2,deny,status:403,msg:'Phat Hien Comments'"

* Dùng chuỗi sau tấn công SQL Injection:

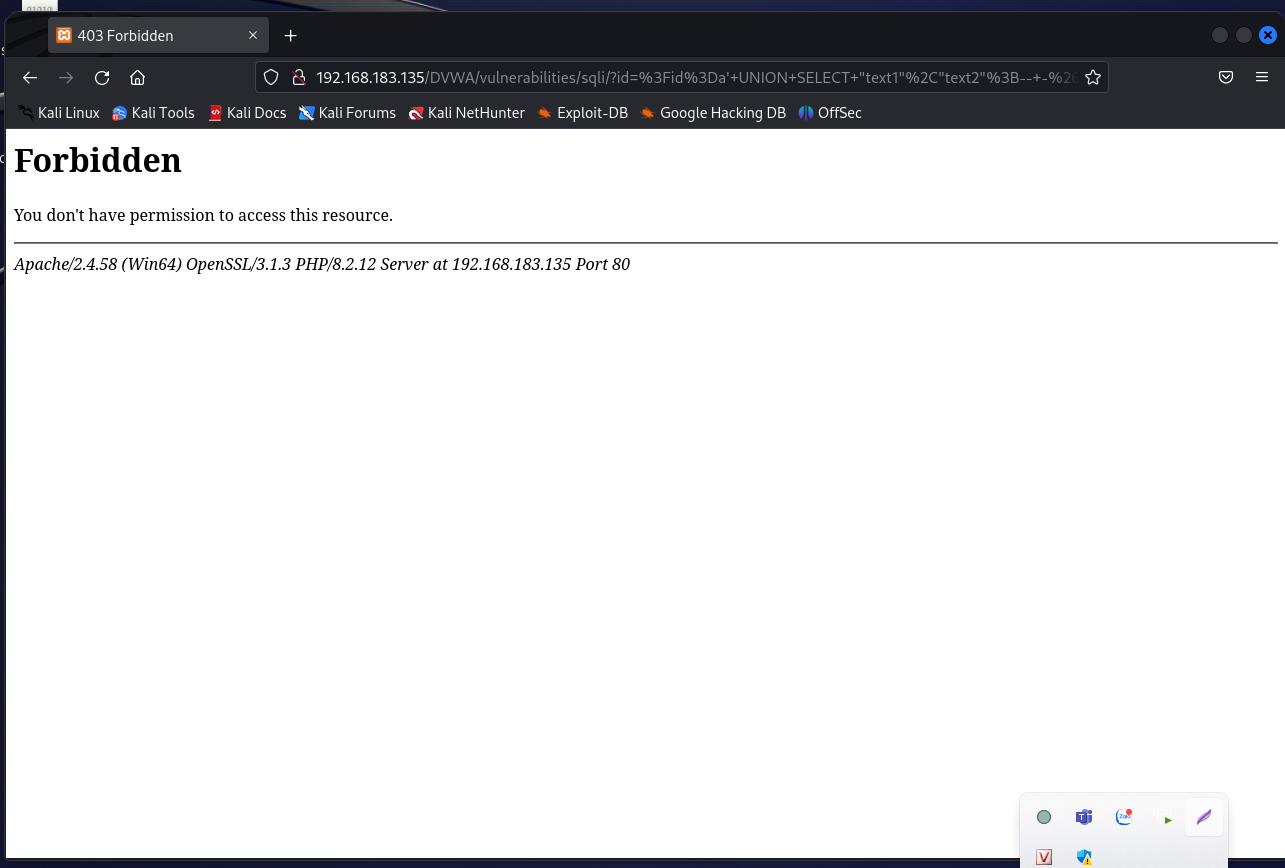
?id=a' UNION SELECT "text1","text2";-- -&Submit=Submit



* Sử dụng quy tắc sau:



* Tập luật của ModSecurity được kích hoạt và chặn cuộc tấn công này.



* Log ghi lại nhật ký



**3.2.2. Thử nghiệm phòng chống tấn công XSS**

- Công thức tổng quát xây dựng quy tắc chống tấn công XSS:

SecRule<TARGET\_VARIABLE>"<PATTERN>""id:<UNIQUE\_ID>,phase:<PHASE\_NUMBER>,deny,status:<STATUS\_CODE>,msg:'<LOG\_MESSAGE>'"

**Trong đó:**

+ SecRule: là từ khóa của ModSecurity để định nghĩa một quy tắc

+ <TARGET\_VARIABLE>: là biến mục tiêu, tức là phần nào của yêu cầu HTTP mà quy tắc sẽ kiểm tra. Một số ví dụ phổ biến bao gồm:

+ <PATTERN>: là biểu thức chính quy mà quy tắc sẽ sử dụng để so khớp. Đây là mẫu mà bạn muốn phát hiện trong biến mục tiêu, chẳng hạn như:

* "(?i:document\.cookie)": Biểu thức chính quy không phân biệt chữ hoa chữ thường để phát hiện document.cookie.
* "('|--|;|\/\\*|\\*\/)": Phát hiện các ký tự đặc biệt thường dùng trong các cuộc tấn công SQL Injection.

+ id:<UNIQUE\_ID>: là ID duy nhất của quy tắc này. Mỗi quy tắc cần một ID duy nhất để có thể được quản lý và theo dõi một cách dễ dàng

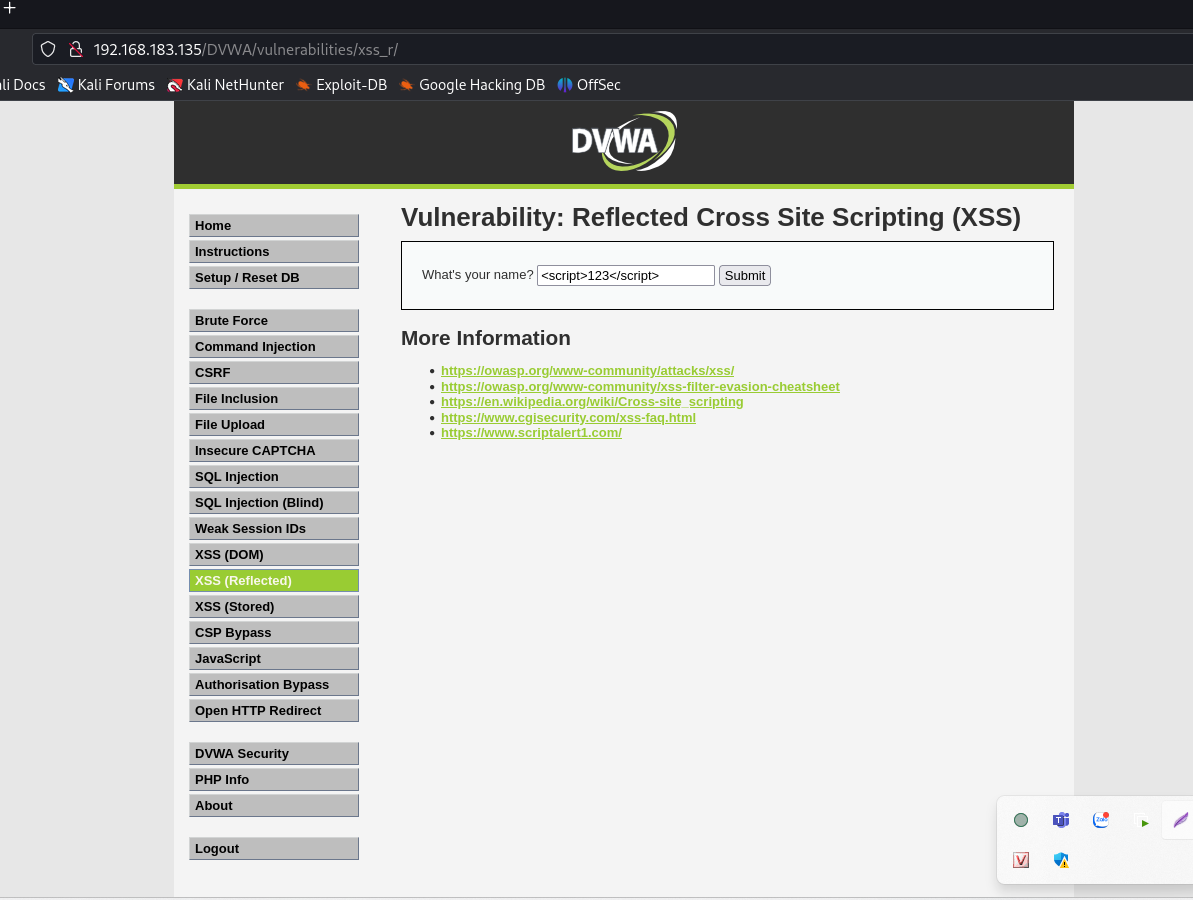
+ phase:<PHASE\_NUMBER>: xác định giai đoạn của xử lý yêu cầu HTTP mà quy tắc sẽ được thực hiện

+ deny: là hành động thực hiện khi quy tắc khớp, yêu cầu sẽ bị từ chối

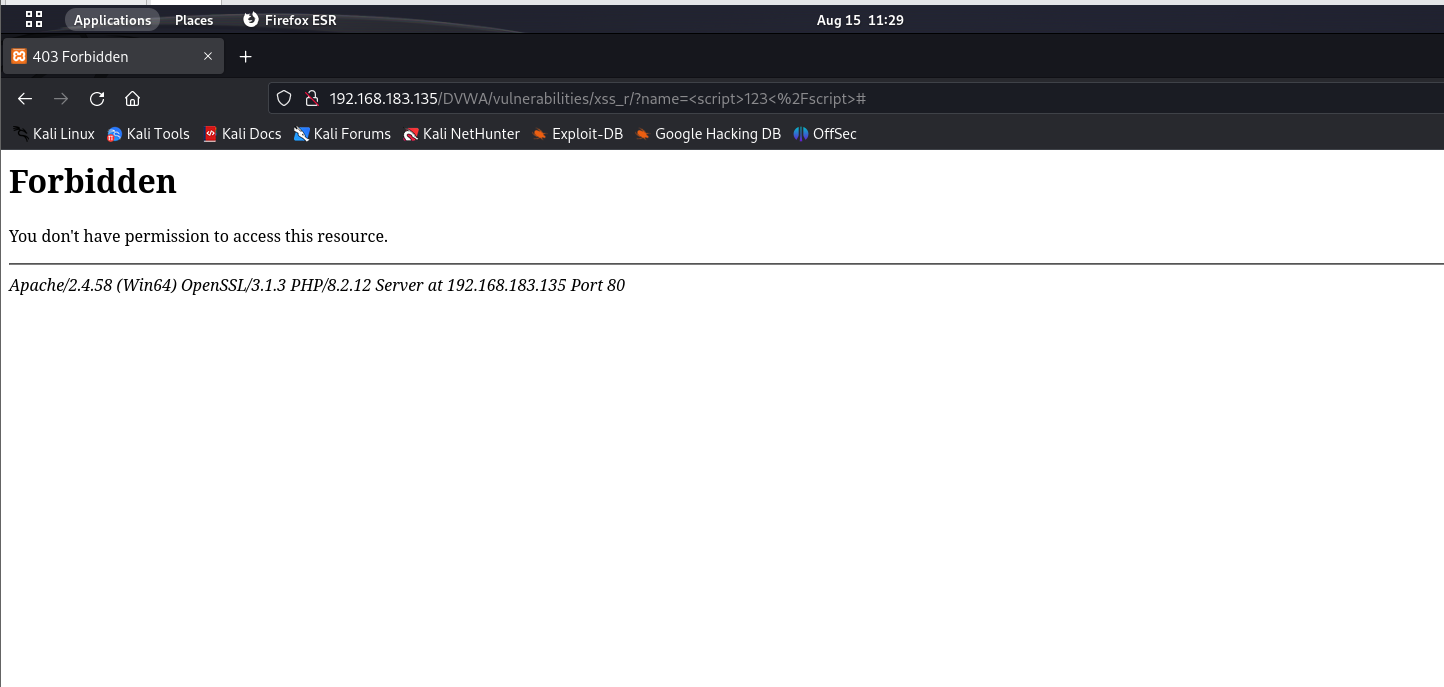
+ status:<STATUS\_CODE>: là mã trạng thái HTTP trả về khi từ chối yêu cầu

+ msg:'<LOG\_MESSAGE>': là thông báo ghi nhật ký khi quy tắc được kích hoạt

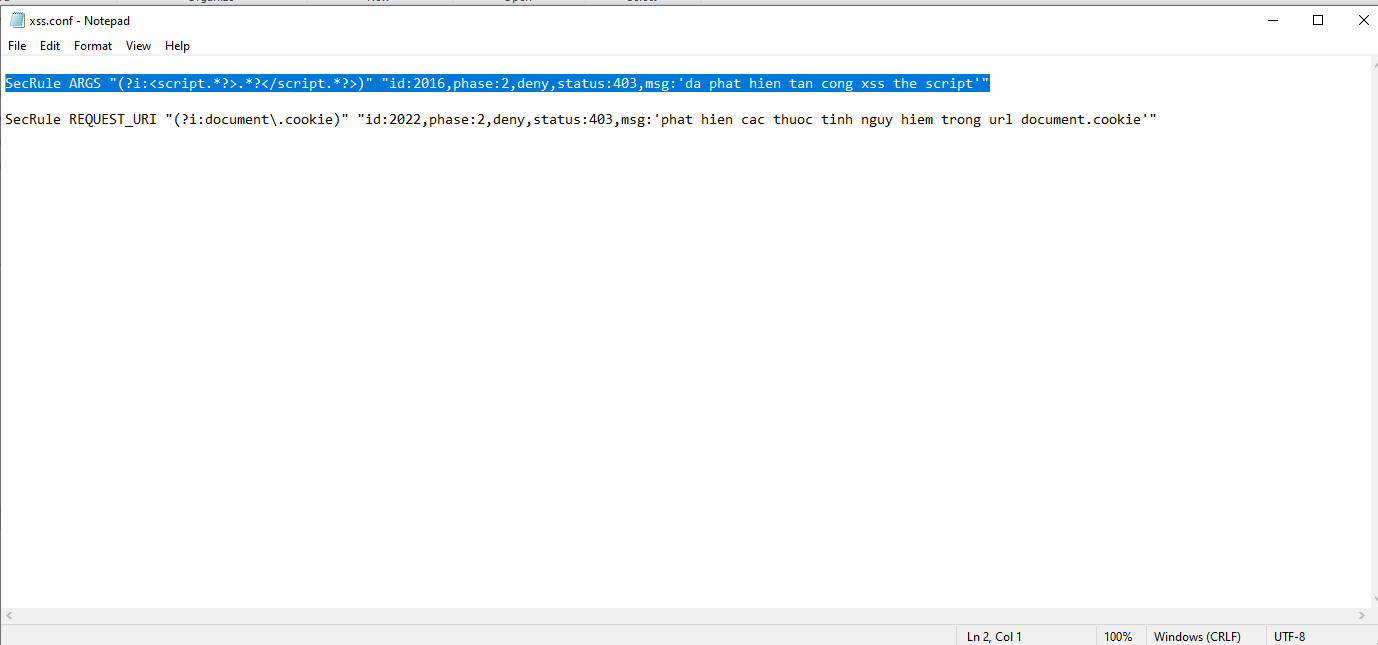
- Đoạn mã độc chứa thẻ script



* Tập luật của ModSecurity được kích hoạt và chặn cuộc tấn công này.



* Sử dụng quy tắc sau:

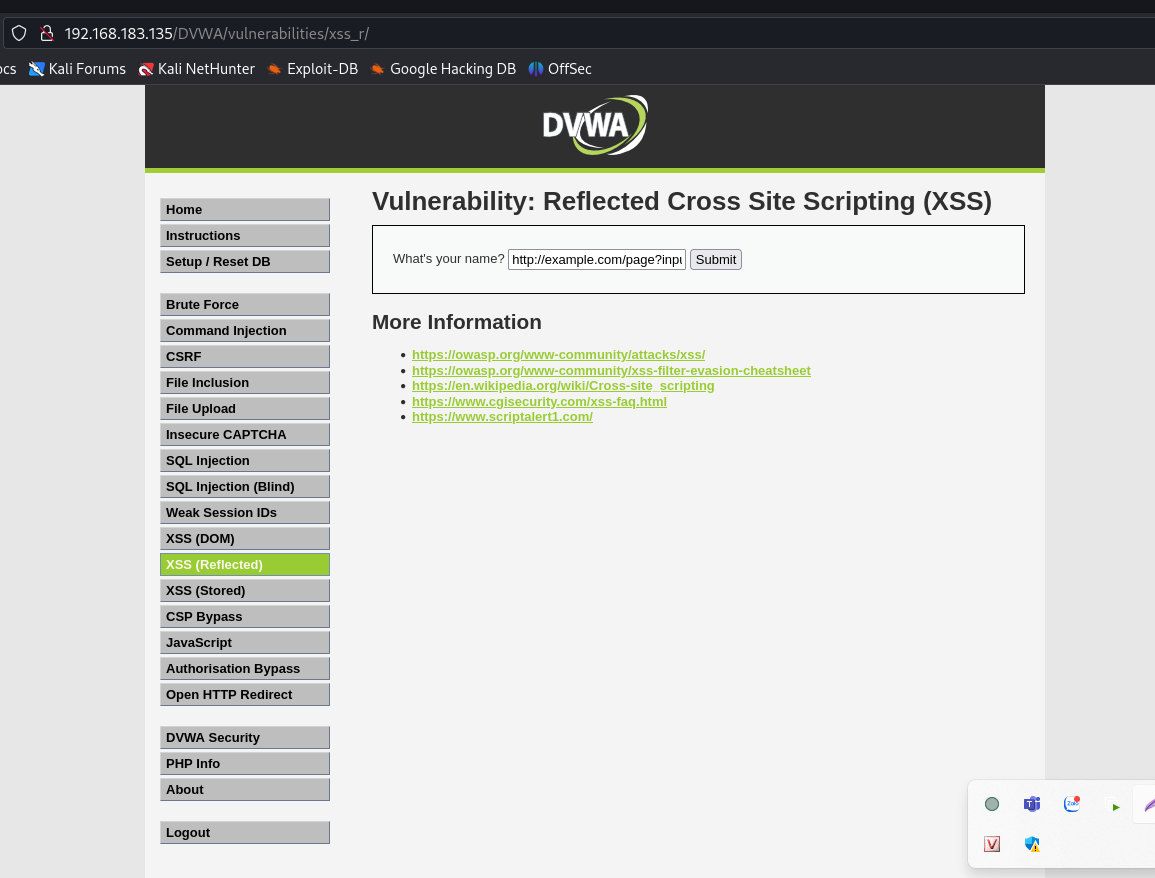


* Log thông báo lỗi ghi lại nhật ký

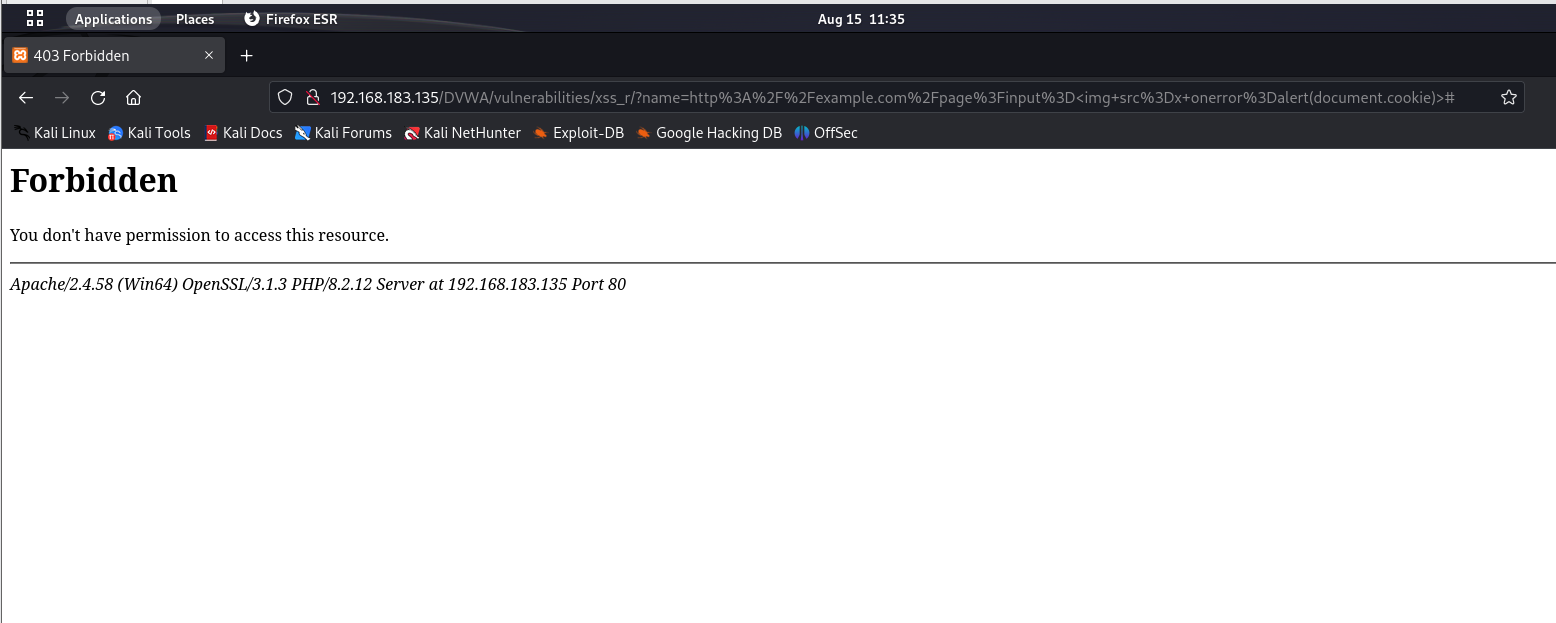


* Đoạn mã độc được đặt trong một tham số GET của URL:

http://example.com/page?input=<img src=x onerror=alert(document.cookie)>



* ModSecurity đã chặn cuộc tấn công này



* Sử dụng quy tắc sau:



* Log ghi lại nhật ký báo lỗi

