

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



SECURITY PRO

# PROJECT # 3

D V W A ( D A M N V U L N E R A B L E W E B A P P L I C A T I O N )



SECURITY PRO



# INTRODUCTION

**Name:** Arslan Ali

**Qualification:** BS Software Engineering

**Cohort:** IEC Cybersecurity Cohort -5

**Instructor:** Shahzaib Ali Khan

**Project:** Project 3 Web penetration Test

**Web:** DVWA



SECURITY PRO



DVWA

Username: admin

Password: password

Login

```
o access official Ubuntu documentation, please visit:  
http://help.ubuntu.com/  
o mail.  
sfadmin@metasploitable:~$ ifconfig  
eth0      Link encap:Ethernet HWaddr 08:00:27:4b:b5:27  
          inet addr:192.168.0.101 Bcast:192.168.0.255 Mask:255.255.255.0  
          inet6 addr: fe80::a00:27ff:fe4b:b527/64 Scope:Link  
             UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
             RX packets:45 errors:0 dropped:0 overruns:0 frame:0  
             TX packets:71 errors:0 dropped:0 overruns:0 carrier:0  
             collisions:0 txqueuelen:1000  
             RX bytes:5805 (5.6 KB) TX bytes:7419 (7.2 KB)  
             Base address:0xd020 Memory:f0200000-f0220000  
  
o Link encap:Local Loopback  
    inet addr:127.0.0.1 Mask:255.0.0.0  
    inet6 addr: ::1/128 Scope:Host  
       UP LOOPBACK RUNNING MTU:16436 Metric:1  
       RX packets:92 errors:0 dropped:0 overruns:0 frame:0  
       TX packets:92 errors:0 dropped:0 overruns:0 carrier:0  
       collisions:0 txqueuelen:0  
       RX bytes:19393 (18.9 KB) TX bytes:19393 (18.9 KB)  
sfadmin@metasploitable:~$
```

# DAMN VULNERABLE WEB APPLICATION (DVWA)

**Username:** admin

**Password:** password

In this project, I installed the Virtual box on my laptop and then install the mirror of metasploitable 2. The default username and password of the metasploitable 2 is "msfadmin" and find ip address of this machine and open it in Firefox

[Home](#)[Instructions](#)[Setup](#)[Brute Force](#)[Command Execution](#)[CSRF](#)[File Inclusion](#)[SQL Injection](#)[SQL Injection \(Blind\)](#)[Upload](#)[XSS reflected](#)[XSS stored](#)[DVWA Security](#)[PHP Info](#)[About](#)

# Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are 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 aid teachers/students to teach/learn web application security in a class room environment.

## WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing [XAMPP](#) onto a local machine inside your LAN which is used solely for testing.

## Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

## General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

You have logged in as 'admin'



01

## XSS Attack(Reflected)

Cross-Site Scripting (xss) attacks  
with Javascript

02

## XSS Attack(Stored)

Cross-Site Scripting (xss) attacks  
with Javascript

03

## HTML Injection Stored

Type of Cross-Site Scripting (xss)  
attacks with HTML language

04

## HTML Injection Reflected

Type of Cross-Site Scripting (xss)  
attacks with HTML language

05

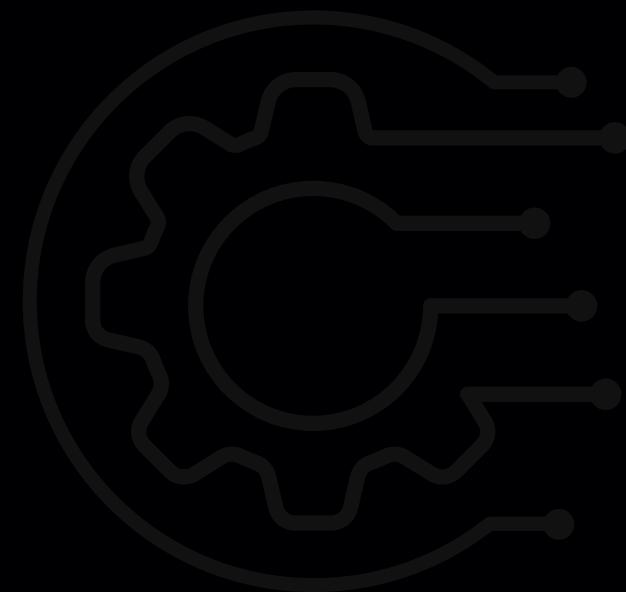
## SQL Injection

Attack on the database by using SQL  
language

06

## Blind SQL Injection

Attack on the database by using SQL  
language





07

## Command Line Injection

involves executing arbitrary commands on a host operating system (os)

08

## CSRF Attack

Cross-Site Request Forgery (CSRF)

09

## Malicious file upload Attack

Placing files onto a server or computer in such a way that they contain some form of backdoor code

10

## File inclusion Attack

Trick the application into exposing or running files on the server

11

## SQL Login Injection

Bypass login page with using SQL language trick

12

## Brute force Attack

'guessing' usernames and passwords to gain unauthorized access to a system



# XSS ATTACK(REFLECTED)



The screenshot shows a Firefox browser window on a Kali Linux desktop. The address bar shows the URL `192.168.1.5/dvwa/vulnerabilities/xss_r/?name=<script>alert(2)<%2Fscript>#`. The DVWA interface displays a form with the question "What's your name?" and a text input field containing "`<script>alert("It is a Vunlability")`". Below the input is a "Submit" button. The response "Hello" is displayed in red text below the input field. A sidebar on the left lists various security vulnerabilities, with "XSS reflected" highlighted in green.

The screenshot shows the same DVWA interface after the exploit was triggered. A modal dialog box appears with the message "192.168.1.5" and "It is a Vunlability". An "OK" button is visible in the bottom right corner of the dialog. The rest of the DVWA interface and desktop environment remain the same as in the first screenshot.

**Code:**

```
<script>alert("Hello i am error")</script>
```

**Severity:**

**High**

**Risk:**

- Steal sensitive information
- Steal cookies
- Modify the appearance of a website



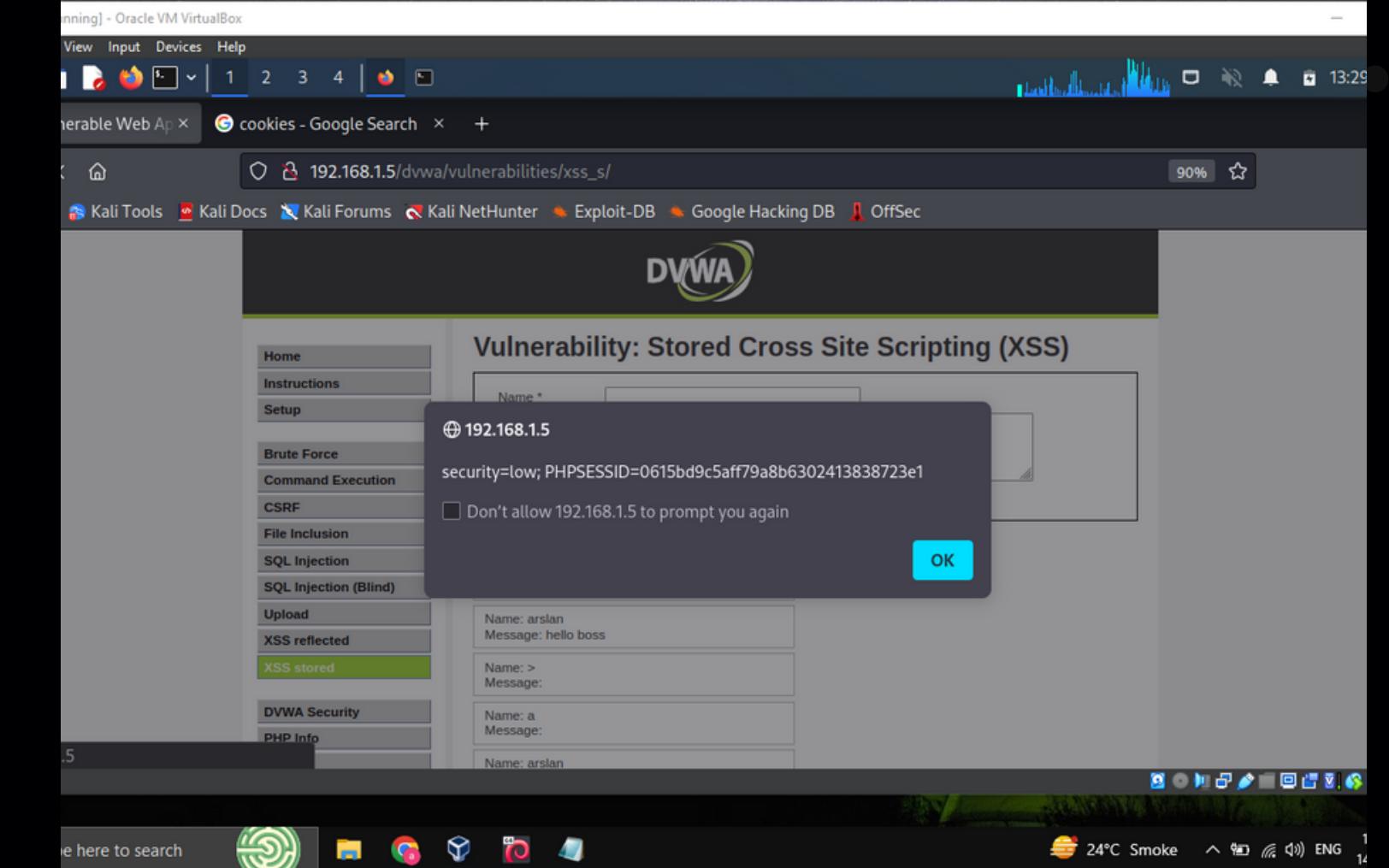
# XSS ATTACK(STORED)



The screenshot shows the DVWA application's XSS stored page. A user has injected the following code into the 'Message' field:

```
<script>alert(document.cookie)</script>
```

The message is displayed on the page, and an alert box is visible in the browser's developer tools, confirming the injection was successful.



**Code:**

```
<script>alert(document.cookie);</script>
```

**Severity:**

**High**

**Risk:**

- Session hijacking
- Malware distribution
- Steal cookies
- Website defacement



# SQL INJECTION



A screenshot of a web browser window. The address bar shows the URL `192.168.1.3/dvwa/vulnerabilities/sqli/?id=%27&Submit=Submit#`. The page content displays an error message: "You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '''' at line 1". Below the error message is a navigation menu with options like Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (the current page), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, and About.

A screenshot of a web browser window titled "Vulnerability: SQL Injection". The URL in the address bar is `192.168.0.101/dvwa/vulnerabilities/sqli/?id=admin' OR '1'='1&Submit=Submit#`. On the left, there is a sidebar with links to various exploit types. The main content area shows a user input field containing "admin' OR '1'='1" and a "Submit" button. Below the input field, several user records are listed, each with an ID, first name, and surname. All records show "ID: admin' OR '1'='1" in the ID field. The first record is "First name: admin Surname: admin". Subsequent records show variations: "First name: Gordon Surname: Brown", "First name: Hack Surname: Me", "First name: Pablo Surname: Picasso", and "First name: Bob Surname: Smith". A "More info" link is visible at the bottom of the content area.

**Code:**

`admin' OR '1'='1`

**Severity:**

**Critical**

**Risk:**

- Unauthorized data access
- Data manipulation or deletion
- Server compromise



# BLIND SQL INJECTION



Damn Vulnerable Web App (DVWA) - Not secure | 192.168.1.3/dvwa/vulnerabilities/sql\_injection/?id=%27&Submit=Submit#

User ID:  Submit

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/tctips/sql-injection.html>

Home  
Instructions  
Setup  
Brute Force  
Command Execution  
CSRF  
File Inclusion  
SQL Injection  
SQL Injection (Blind)  
Upload  
XSS reflected  
XSS stored  
DVWA Security  
PHP Info  
About  
Logout

Username: admin

Damn Vulnerable Web App - Oracle VM VirtualBox

User ID:  Submit

ID: 2' OR '1'='1  
First name: admin  
Surname: admin

ID: 2' OR '1'='1  
First name: Gordon  
Surname: Brown

ID: 2' OR '1'='1  
First name: Hack  
Surname: Me

ID: 2' OR '1'='1  
First name: Pablo  
Surname: Picasso

ID: 2' OR '1'='1  
First name: Bob  
Surname: Smith

Home  
Instructions  
Setup  
Brute Force  
Command Execution  
CSRF  
File Inclusion  
SQL Injection  
SQL Injection (Blind)  
Upload  
XSS reflected  
XSS stored  
DVWA Security  
PHP Info  
About

Code:

`admin' OR '1'='1`

Severity:

Critical

Risk:

- Unauthorized data access
- Data manipulation or deletion
- Server compromise



# HTML INJECTION (REF)



A screenshot of the DVWA Reflected XSS (XSS\_Reflected) page. The URL is `http://192.168.1.5/dvwa/vulnerabilities/xss_r/`. The page title is "Vulnerability: Reflected Cross Site Scripting (XSS)". On the left, there's a sidebar with various exploit categories like 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 with a placeholder "What's your name?" and an input field containing "`><h1>Hack this now</h1>`". Below the form, there's a "Submit" button and a "More info" section with links to XSS resources.

A screenshot of the DVWA Reflected XSS (XSS\_Reflected) page after the injection was submitted. The URL is `http://192.168.1.5/dvwa/vulnerabilities/xss_r/?name=><h1>Hack+this+now<%2Fh1>#`. The page title is "Vulnerability: Reflected Cross Site Scripting (XSS)". The main content area shows the injected code "`Hello > Hack this now`" displayed in red text, indicating it was reflected back to the user.

**Code:**

`><h1>Hack this now</h1>`

**Severity:**

**High**

**Risk:**

- Steal sensitive information
- Steal cookies
- Modify the appearance of a website



# HTML INJECTION (STORED)



Kali Linux [Running] - Oracle VM VirtualBox

Damn Vulnerable Web App | cookies - Google Search

192.168.1.5/dvwa/vulnerabilities/xss\_s/

DVWA

Vulnerability: Stored Cross Site Scripting (XSS)

Name \* ><b>1</b>  
</textarea><h1>I want to hack you</h1>

Message \*

Sign Guestbook

Name: test  
Message: This is a test comment.

Name: arslan  
Message: hello boss

Name: >

More info

<http://ha.ckers.org/xss.html>  
[http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting)  
<http://www.cgisecurity.com/xss-faq.html>

Kali Linux [Running] - Oracle VM VirtualBox

Damn Vulnerable Web App | cookies - Google Search

192.168.1.5/dvwa/vulnerabilities/xss\_s/

DVWA

Usernames: admin

Name: >  
Message:

I want to hack you

More info

<http://ha.ckers.org/xss.html>  
[http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting)  
<http://www.cgisecurity.com/xss-faq.html>

Code:

/textarea><h1>I want to hack You</h1>

Severity:

High

Risk:

- Session hijacking
- Malware distribution
- Steal cookies
- Website defacement



# FILE INCLUSION ATTACK



The screenshot shows a browser window titled "Damn Vulnerable Web App (DVWA)" with the URL "192.168.1.3/dvwa/vulnerabilities/fi/?page=/etc/passwd". The page content is a shell dump of the /etc/passwd file, revealing sensitive information such as user accounts and their home directories. The DVWA navigation menu on the left shows "File Inclusion" is selected. The operating system taskbar at the bottom includes icons for search, file explorer, and various system status indicators.

**Code:**

**<http://192.168.1.3/dvwa/vulnerabilities/fi/?page=/etc/passwd>**

**Severity:**  
**Critical**

**Risk:**

- Unauthorized access
- Information theft
- Malware distribution





# MALICIOUS FILE UPLOAD



The screenshot shows a Linux desktop environment with a Kali Linux desktop manager. A Firefox browser window is open to the DVWA 'File Upload' page at <http://192.168.0.101/dvwa/vulnerabilities/upload/>. The page displays a success message: `.../hackable/uploads/index.php successfully uploaded!`. The DVWA sidebar on the left shows the 'Upload' option is selected.

**Code:**

**upload a malicious file like php or python file instead a jpg or png file.**

**Severity:**

**Critical**

**Risk:**

- Server compromise
- Information theft
- Malware distribution



# COMMAND LINE INJECTION



The screenshot shows the DVWA Command Execution page. In the 'Ping for FREE' section, the user has entered the command `8.8.4.4 ; ls` into the input field and clicked the 'submit' button. The page displays the resulting output of the command execution.

The screenshot shows the DVWA Command Execution page. The user has entered the command `8.8.4.4 ; ls` and the output shows the directory listing of the target machine. Below the output, ping statistics are displayed.

**Code:**

**ping 8.8.4.4 and also ls**

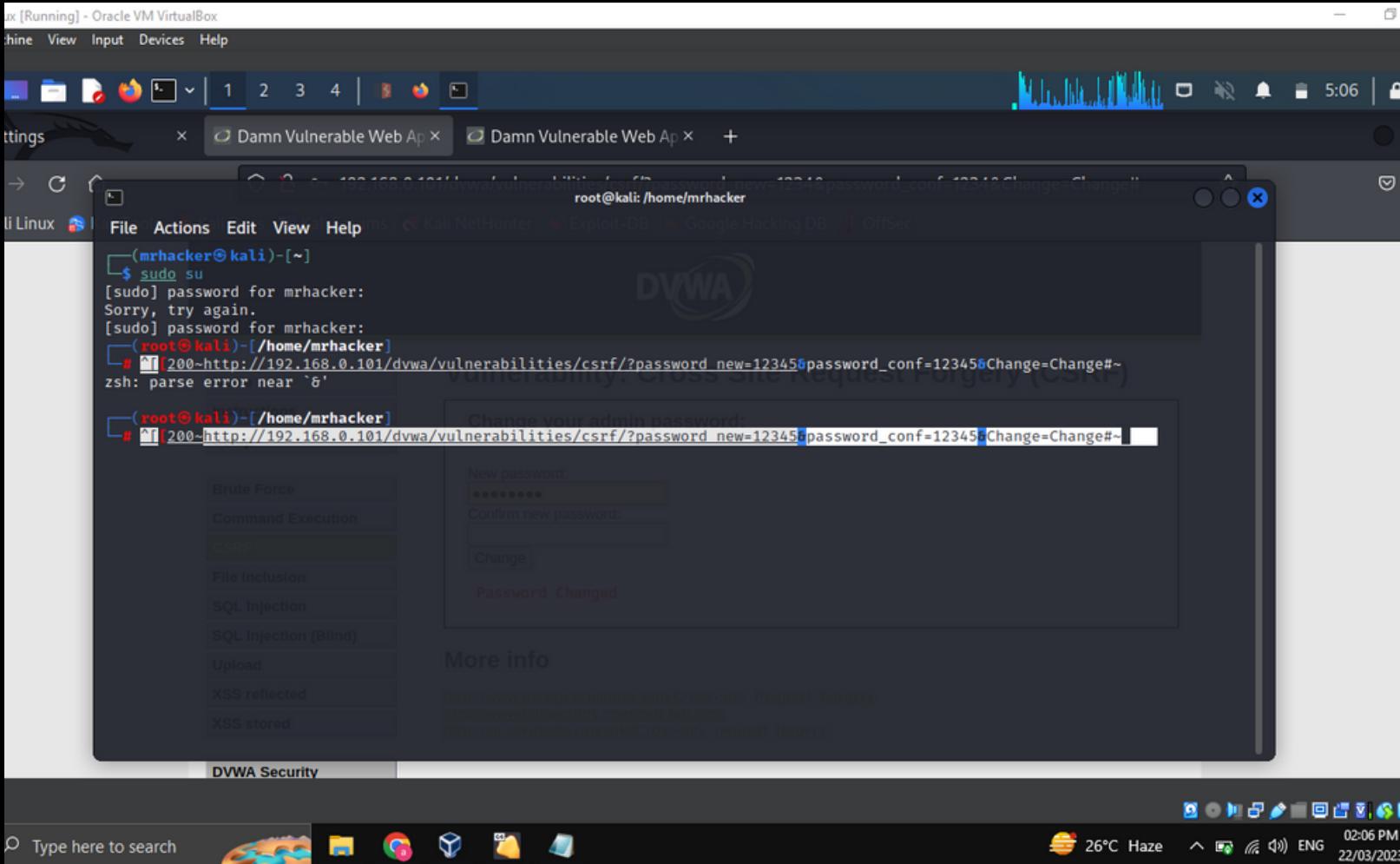
**"8.8.4.4 ; ls"**

**Severity:  
Critical**

**Risk:**

- Server compromise
- Information theft
- Malware distribution

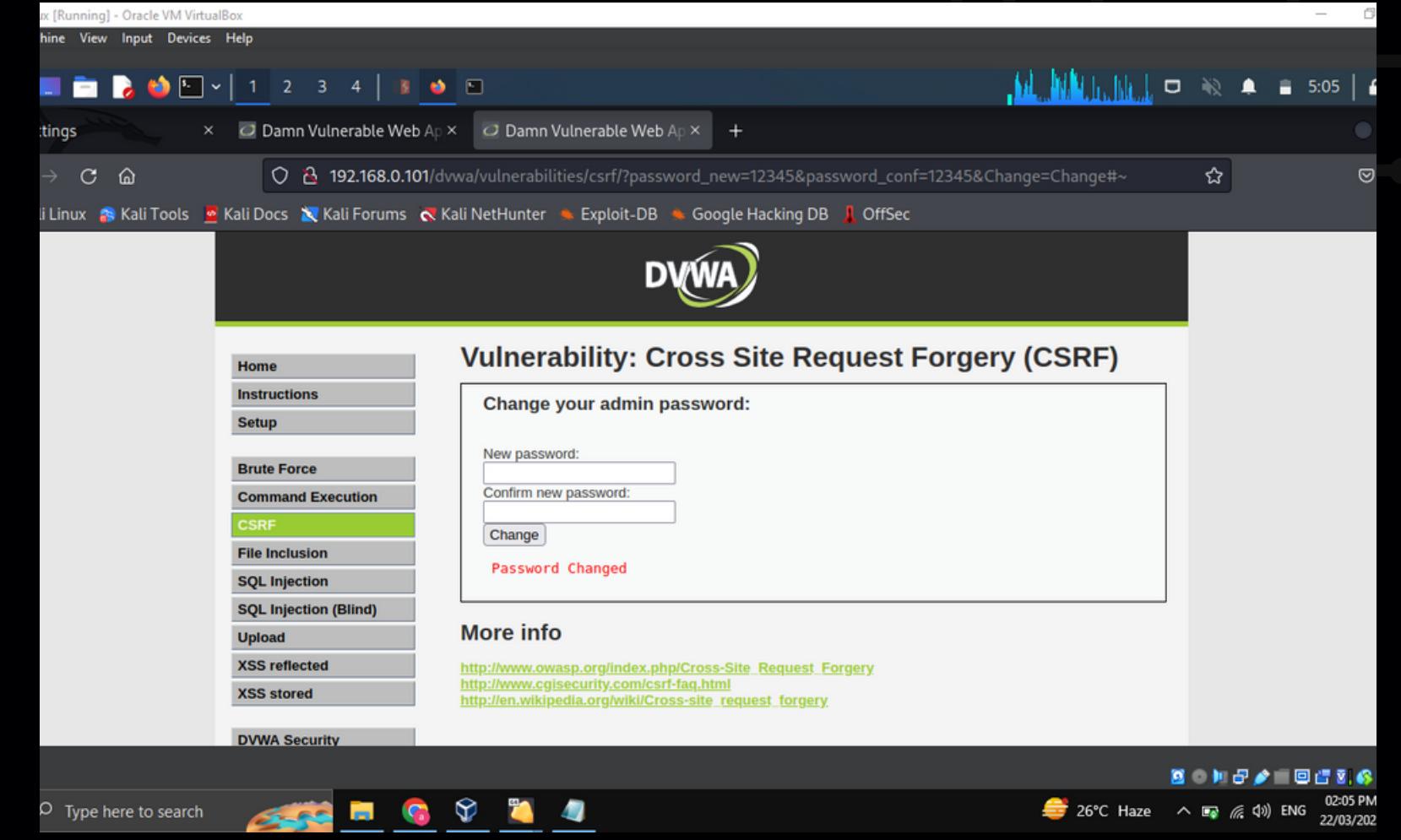
# CSRF ATTACK



A terminal window on a Kali Linux desktop showing a successful CSRF attack on the DVWA 'CSRF' vulnerability. The user has entered their password and confirmed it, but the password was changed without their knowledge due to the exploit.

```
(mrhacker㉿kali)-[~]
$ sudo su
[sudo] password for mrhacker:
Sorry, try again.
[sudo] password for mrhacker:
[root@kali]-[~/home/mrhacker]
# 200->http://192.168.0.101/dvwa/vulnerabilities/csrf/?password_new=12345&password_conf=12345&Change=Change#~
zsh: parse error near `&'`
```

The terminal also shows the DVWA CSRF page with the password changed message.



## Code:

**[http://192.168.0.101/dvwa/vulnerabilities/csrf/?password\\_current=password&password\\_new=1234&password\\_conf=1234&Change=Change#](http://192.168.0.101/dvwa/vulnerabilities/csrf/?password_current=password&password_new=1234&password_conf=1234&Change=Change#)**

## Risk:

- Unauthorized actions
- Information theft
- Reputation damage

**Severity:  
Low**

# SQL LOGIN INJECTION

This screenshot shows the DVWA Brute Force login screen. The URL is `192.168.0.101/dvwa/vulnerabilities/brute/?username=ghg&password=bv&Login=Login#`. In the 'Username' field, the value is set to `admin' OR '1='1`. The 'Login' button is visible below the fields. A red error message at the bottom states: "Username and/or password incorrect."

This screenshot shows the DVWA Brute Force login screen after a successful SQL injection. The URL is `192.168.0.101/dvwa/vulnerabilities/brute/?username=admin' OR '1='1&password=&Login=Login#`. The 'Username' field now contains `admin' OR '1='1`. The 'Login' button is visible. A success message at the bottom says: "Welcome to the password protected area admin' OR '1='1".

**Code:**

`admin' OR '1='1`

**Severity:**

**Critical**

**Risk:**

- Unauthorized actions
- Information theft
- Reputation damage



DVWA

# THANK YOU

S E C U R I T Y   P R O

