INTERNSHIP REPORT AT FUTURE INTERNS

Task 1: Web Application Security Testing

Report Prepared By: **ISSA HASSAN YOUSSOUF**

Internship period : 17/07/2025 to 17/08/2025
Company : Future Interns

INTRODUCTION

Web application security is a strategic priority in any digital environment today. Because web applications are often exposed to the internet, they are a prime target for cybercriminals. This task immerses you in the world of offensive security testing, also known as Pentest, with a focus on identifying common vulnerabilities such as SQL injection, XSS vulnerabilities, and authentication errors.

OBJECTIVE :

The purpose of this task is to perform a security test on a web application to identify, document, and propose solutions to multiple vulnerabilities. You'll learn how to exploit common vulnerabilities and generate a professional report that showcases your results.

This task builds on this framework to familiarize you with the most common vulnerabilities and teach you how to detect them effectively.

SKILLS DEVELOPED:

- In-depth understanding of web vulnerabilities (SQLi, XSS, CSRF, etc.)
- Mastery of security testing tools
- Technical vulnerability analysis and report writing
- Knowledge of OWASP best practices

TOOLS USED:

To accomplish this task, the following tools are highly recommended:

OWASP ZAP (Zed Attack Proxy)

- Open-source automatic scanner specialized in web application security testing.
- Helps identify common vulnerabilities such as XSS, CSRF, command injection, and more.
- Intuitive graphical interface suitable for beginners and advanced users alike.

Burp Suite

- Professional web penetration testing tool.
- Works as a proxy intercepting requests between the browser and the server.
- Offers powerful modules for mapping, fuzzing, passive and active scanning.

Kali Linux

- Linux distribution dedicated to cybersecurity.
- Contains hundreds of auditing and penetration testing tools (including Burp, ZAP, SQLMap, Nikto, etc.)
- Provides an ideal environment for testing in a secure, isolated setting.

1. TOOL INSTALLATION

You can install these tools on your local machine or through a virtual machine, but here we were virtualizing Kali Linux.

a. Installing Kali Linux on VMware Workstation

Downloading and Importing into VMware Workstation:

- Download the official ISO image: https://www.kali.org/get-kali/
- Go to the *Virtual Machines* section to download Kali pre-installed
- Opens *VMware Workstation*
- Click Open *a Virtual Machine*
- Selects the unzipped .vmx file
- The Kali VM is ready to be launched!

Recommended Requirements:

- RAM: 2 to 4 GB
- CPU: 2 cores minimum
- Drive: 20 GB or more
- Enables *virtualization* in the BIOS

Useful optimizations:

- Installs *VMware Tools* to:
    - Better screen resolution
    - Drag and drop between host and VM
    - Folder sharing
- Creates snapshots before each test for easy rewinding

The very complete official guide on https://www.kali.org/docs/virtualization/install-vmware-guest-vm/ or this https://oleks.ca/2024/09/26/installation-de-kali-linux-sur-vmware-workstation/ if you want an illustrated version.

The Burp tool is pre-installed on Kali Linux, where we will configure DVWA to launch our tests (SQLi, XSS, CSRF, etc.) and ZAP to scan and identify vulnerabilities on the web.

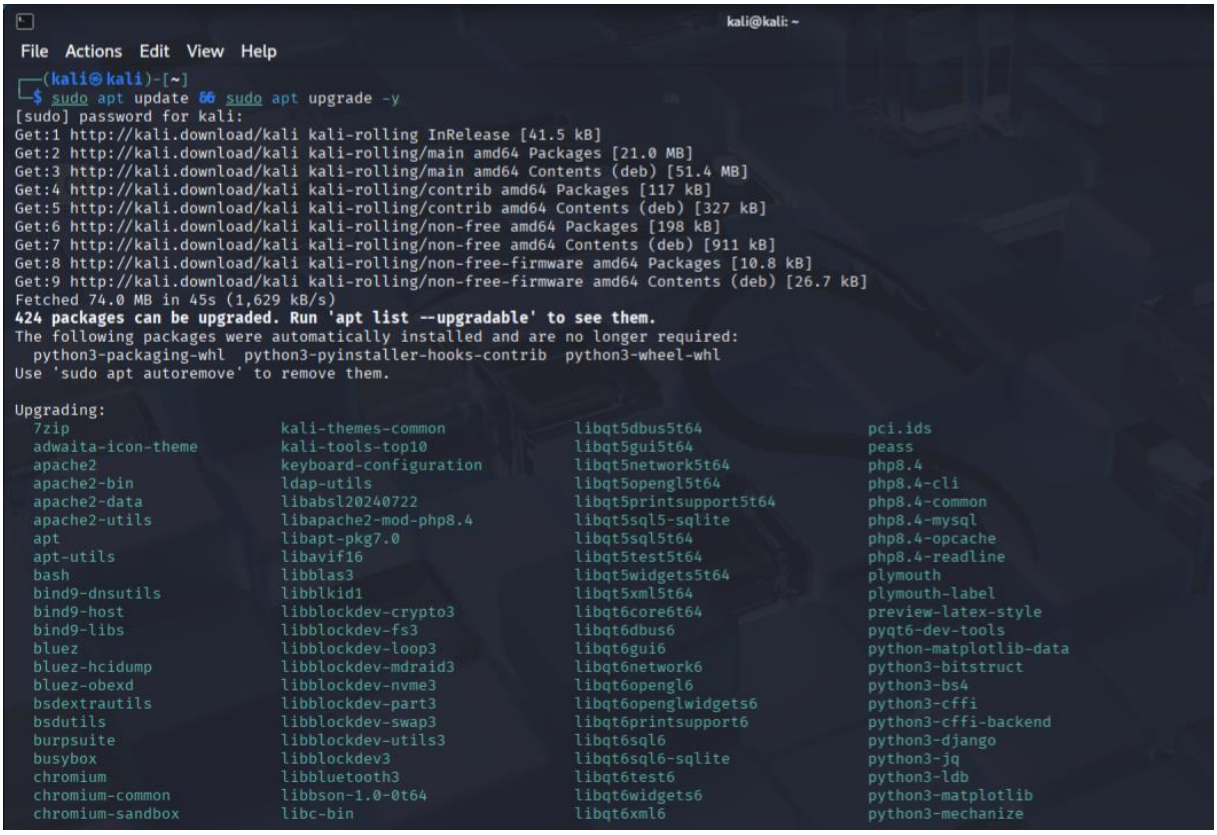
b. Installing, configuring, and using Damn Vulnerable Web Application (DVWA)

Installation objective:

- Install **DVWA** on Kali Linux.
- Configure Apache, MySQL and PHP web server.
- Access the DVWA web interface.
- Exploit vulnerabilities.

Update Kali Linux

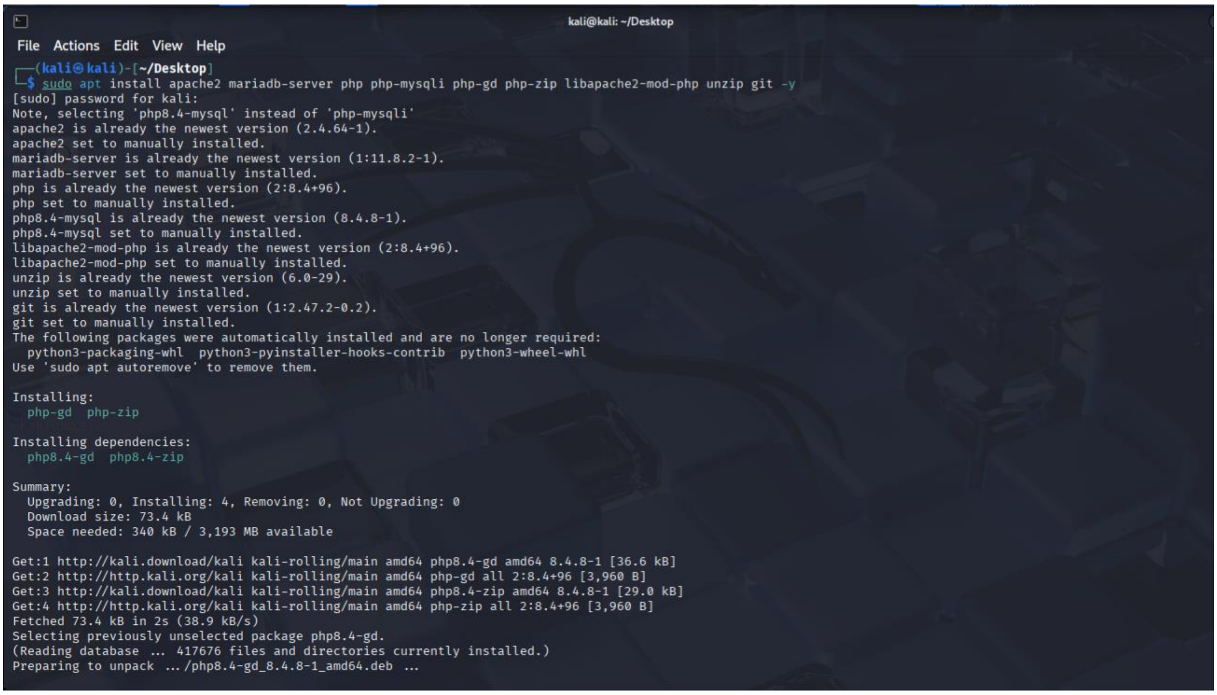
Open Terminal: `sudo apt update & sudo apt upgrade -y`



```
┌──(kali㉿kali)-[~]
└─$ sudo apt update && sudo apt upgrade -y
[sudo] password for kali:
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [21.0 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [51.4 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [117 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [327 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [198 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [911 kB]
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [10.8 kB]
Get:9 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [26.7 kB]
Fetched 74.0 MB in 45s (1,629 kB/s)
424 packages can be upgraded. Run 'apt list --upgradable' to see them.
The following packages were automatically installed and are no longer required:
  python3-packaging-whl  python3-pyinstaller-hooks-contrib  python3-wheel-whl
Use 'sudo apt autoremove' to remove them.

Upgrading:
  7zip                       kali-themes-common              libqt5dbus5t64                  pci.ids
  adwaita-icon-theme         kali-tools-top10                libqt5gui5t64                   peass
  apache2                    keyboard-configuration          libqt5network5t64               php8.4
  apache2-bin                ldap-utils                      libqt5opengl5t64                php8.4-cli
  apache2-data               libabsl20240722                 libqt5printsupport5t64          php8.4-common
  apache2-utils              libapache2-mod-php8.4           libqt5sql5-sqlite               php8.4-mysql
  apt                        libapt-pkg7.0                   libqt5sql5t64                   php8.4-opcache
  apt-utils                  libavif16                       libqt5test5t64                  php8.4-readline
  bash                       libblas3                        libqt5widgets5t64               plymouth
  bind9-dnsutils             libblkid1                       libqt5xml5t64                   plymouth-label
  bind9-host                 libblockdev-crypto3             libqt6core6t64                  preview-latex-style
  bind9-libs                 libblockdev-fs3                 libqt6dbus6                     pyqt6-dev-tools
  bluez                      libblockdev-loop3               libqt6gui6                      python-matplotlib-data
  bluez-hcidump              libblockdev-mdraid3             libqt6network6                  python3-bitstruct
  bluez-obexd                libblockdev-nvme3               libqt6opengl6                   python3-bs4
  bsdextrautils              libblockdev-part3               libqt6openglwidgets6            python3-cffi
  bsdutils                   libblockdev-swap3               libqt6printsupport6             python3-cffi-backend
  burpsuite                  libblockdev-utils3              libqt6sql6                      python3-django
  busybox                    libblockdev3                    libqt6sql6-sqlite               python3-jq
  chromium                   libbluetooth3                   libqt6test6                     python3-ldb
  chromium-common            libbson-1.0-0t64                libqt6widgets6                  python3-matplotlib
  chromium-sandbox           libc-bin                        libqt6xml6                      python3-mechanize
```

After a few minutes the updates is complete, necessary packets are installed of which now we will install Apache, MySQL, PHP and git with the following command:

```
sudo apt install apache2 mariadb-server php-mysqli php-gd php-zip libapache2-mod-php unzip git -y
```



```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo apt install apache2 mariadb-server php php-mysqli php-gd php-zip libapache2-mod-php unzip git -y
[sudo] password for kali:
Note, selecting 'php8.4-mysql' instead of 'php-mysqli'
apache2 is already the newest version (2.4.64-1).
apache2 set to manually installed.
mariadb-server is already the newest version (1:11.8.2-1).
mariadb-server set to manually installed.
php is already the newest version (2:8.4+96).
php set to manually installed.
php8.4-mysql is already the newest version (8.4.8-1).
php8.4-mysql set to manually installed.
libapache2-mod-php is already the newest version (2:8.4+96).
libapache2-mod-php set to manually installed.
unzip is already the newest version (6.0-29).
unzip set to manually installed.
git is already the newest version (1:2.47.2-0.2).
git set to manually installed.
The following packages were automatically installed and are no longer required:
  python3-packaging-whl  python3-pyinstaller-hooks-contrib  python3-wheel-whl
Use 'sudo apt autoremove' to remove them.

Installing:
  php-gd  php-zip

Installing dependencies:
  php8.4-gd  php8.4-zip

Summary:
  Upgrading: 0, Installing: 4, Removing: 0, Not Upgrading: 0
  Download size: 73.4 kB
  Space needed: 340 kB / 3,193 MB available

Get:1 http://kali.download/kali kali-rolling/main amd64 php8.4-gd amd64 8.4.8-1 [36.6 kB]
Get:2 http://http.kali.org/kali kali-rolling/main amd64 php-gd all 2:8.4+96 [3,960 B]
Get:3 http://kali.download/kali kali-rolling/main amd64 php8.4-zip amd64 8.4.8-1 [29.0 kB]
Get:4 http://http.kali.org/kali kali-rolling/main amd64 php-zip all 2:8.4+96 [3,960 B]
Fetched 73.4 kB in 2s (38.9 kB/s)
Selecting previously unselected package php8.4-gd.
(Reading database ... 417676 files and directories currently installed.)
Preparing to unpack .../php8.4-gd_8.4.8-1_amd64.deb ...
```

COMPONENT	ROLE
`apache2`	HTTP Server
`mariadb-server`	Database Management System to manage the `dvwa database`
`PHP`	PHP interpreter
`php-mysqli`	Allows PHP to talk to MySQL
`php-gd`	Image management
`php-zip`	Manipulating compressed files
`libapache2-mod-php`	Connection between Apache and PHP
`Git`	To clone the DVWA Git repository

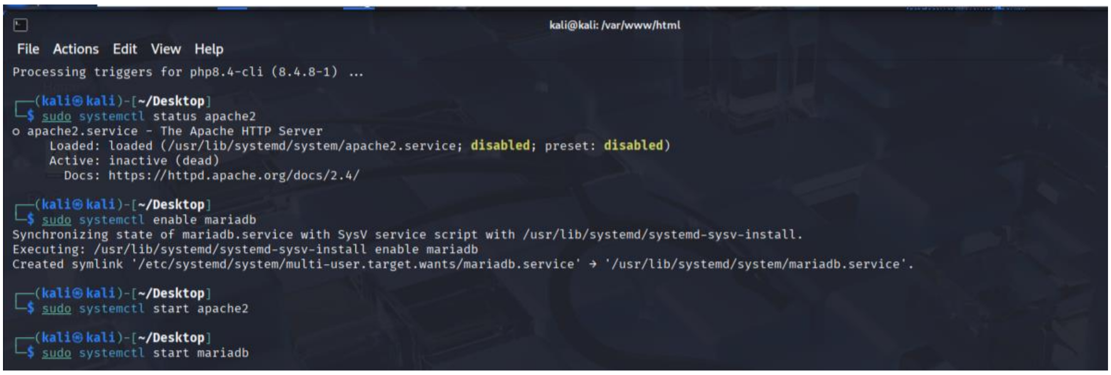
Check that everything is working after installation.

- Verify services with the following commands:

```
sudo systemctl status apache2
```

Make sure the service is running, but in our case the service is dead so we'll enable and start the services with the following commands:

```
sudo systemctl enable mariadb
```



```
Processing triggers for php8.4-cli (8.4.8-1) ...

┌──(kali㉿kali)-[~/Desktop]
└─$ sudo systemctl status apache2
○ apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; preset: disabled)
     Active: inactive (dead)
       Docs: https://httpd.apache.org/docs/2.4/

┌──(kali㉿kali)-[~/Desktop]
└─$ sudo systemctl enable mariadb
Synchronizing state of mariadb.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable mariadb
Created symlink '/etc/systemd/system/multi-user.target.wants/mariadb.service' → '/usr/lib/systemd/system/mariadb.service'.

┌──(kali㉿kali)-[~/Desktop]
└─$ sudo systemctl start apache2

┌──(kali㉿kali)-[~/Desktop]
└─$ sudo systemctl start mariadb
```

`sudo systemctl start apache2`

`sudo systemctl start mariadb`

- Installing DVWA

Go to the root folder of the web server and clone DVWA from GitHub with the following command:

```
cd /var/www/html/sudo git clone
https://github.com/digininja/DVWA.git
```

Rename with the following command: `sudo mv DVWA dvwa`

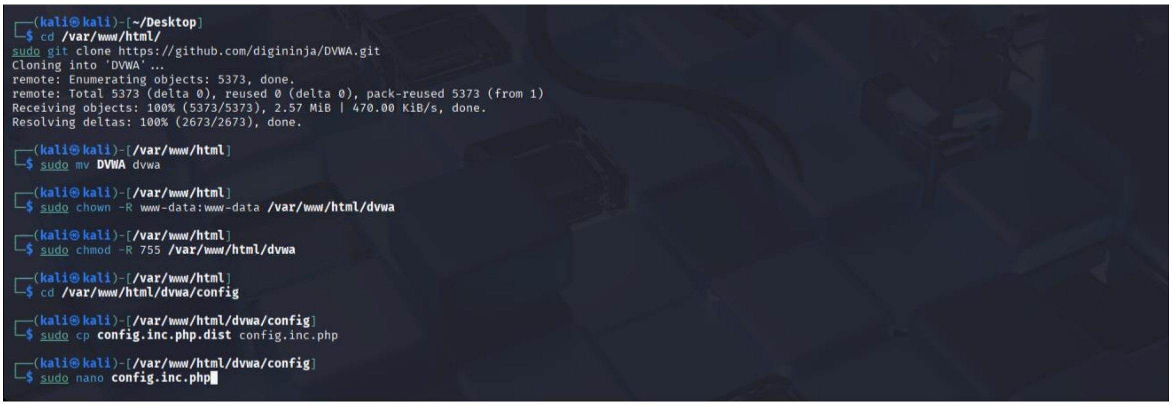
After naming the file, we'll give Apache rights and configure the `config.inc.php file`

- Giving the rights to Apache:

Here we type the following commands to give Apache rights:
`sudo chown -R www-data:www-data /var/www/html/dvwa`

`sudo chmod -R 755 /var/www/html/dvwa`

`www-data` is the user used by Apache to access the files.

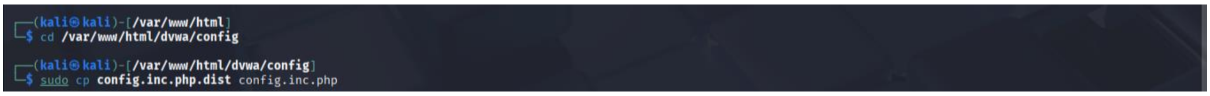


```
┌──(kali㉿kali)-[~/Desktop]
└─$ cd /var/www/html/
sudo git clone https://github.com/digininja/DVWA.git
Cloning into 'DVWA'...
remote: Enumerating objects: 5373, done.
remote: Total 5373 (delta 0), reused 0 (delta 0), pack-reused 5373 (from 1)
Receiving objects: 100% (5373/5373), 2.57 MiB | 470.00 KiB/s, done.
Resolving deltas: 100% (2673/2673), done.

┌──(kali㉿kali)-[/var/www/html]
└─$ sudo mv DVWA dvwa

┌──(kali㉿kali)-[/var/www/html]
└─$ sudo chown -R www-data:www-data /var/www/html/dvwa

┌──(kali㉿kali)-[/var/www/html]
└─$ sudo chmod -R 755 /var/www/html/dvwa

┌──(kali㉿kali)-[/var/www/html]
└─$ cd /var/www/html/dvwa/config

┌──(kali㉿kali)-[/var/www/html/dvwa/config]
└─$ sudo cp config.inc.php.dist config.inc.php

┌──(kali㉿kali)-[/var/www/html/dvwa/config]
└─$ sudo nano config.inc.php
```

- Configure the file and copy `config.inc.php` :

```
cd /var/www/html/dvwa/config
```

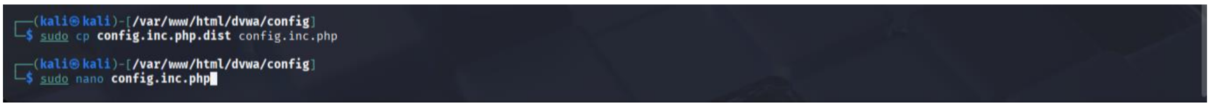
```
sudo cp config.inc.php.dist config.inc.php
```



```
┌──(kali㉿kali)-[/var/www/html]
└─$ cd /var/www/html/dvwa/config

┌──(kali㉿kali)-[/var/www/html/dvwa/config]
└─$ sudo cp config.inc.php.dist config.inc.php
```

- Edit File

```
sudo nano config.inc.php
```

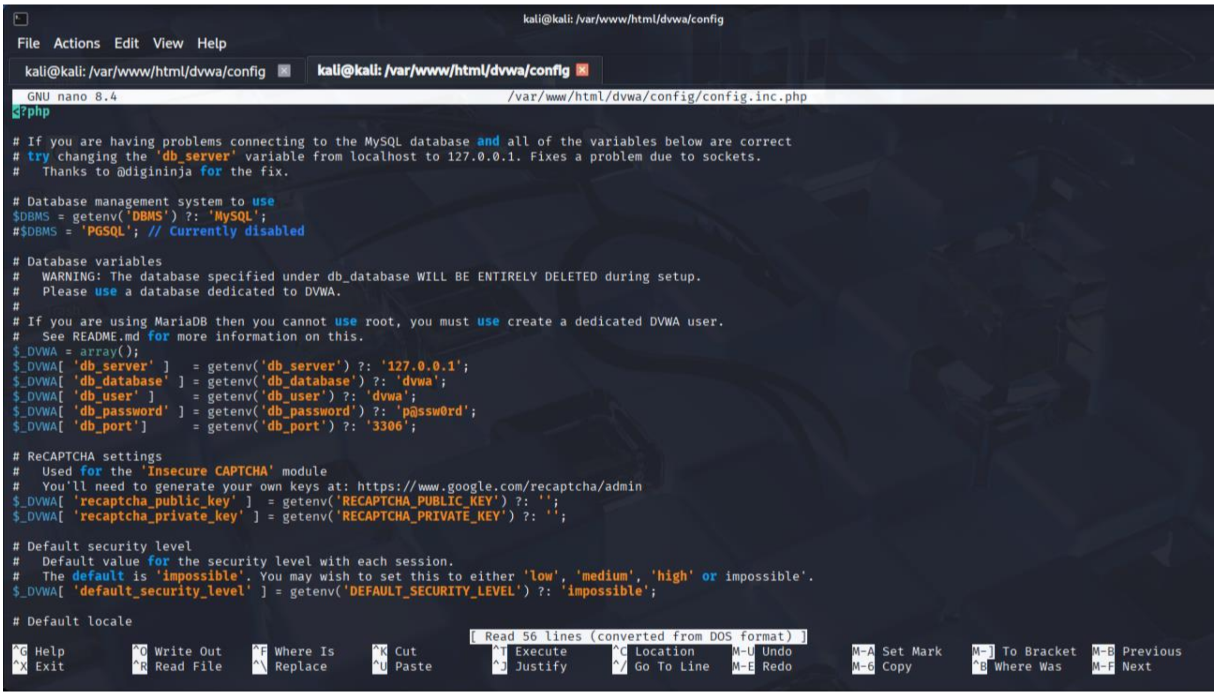


```
┌──(kali㉿kali)-[/var/www/html/dvwa/config]
└─$ sudo cp config.inc.php.dist config.inc.php

┌──(kali㉿kali)-[/var/www/html/dvwa/config]
└─$ sudo nano config.inc.php
```

Changes the following lines:

```
$_DVWA[ 'db_server' ] = getenv( 'db_server' ) ?: '127.0.0.1' ;
$_DVWA[ 'db_datebase' ] = getenv( 'db_base' ) ?: 'dvwa';
$_DVWA[ 'db_user' ] = getenv( 'db_user' ) ?: 'dvwa';
$_DVWA[ 'db_password' ] = getenv( 'db_password' ) ?: 'p@ssw0rd';
$_DVWA[ 'db_port' ] = getenv( 'db_port' ) ?: '3306';
```

Save with ***Ctrl* + *O***, then ***Enter*,** and exit with ***Ctrl* + *X*.**

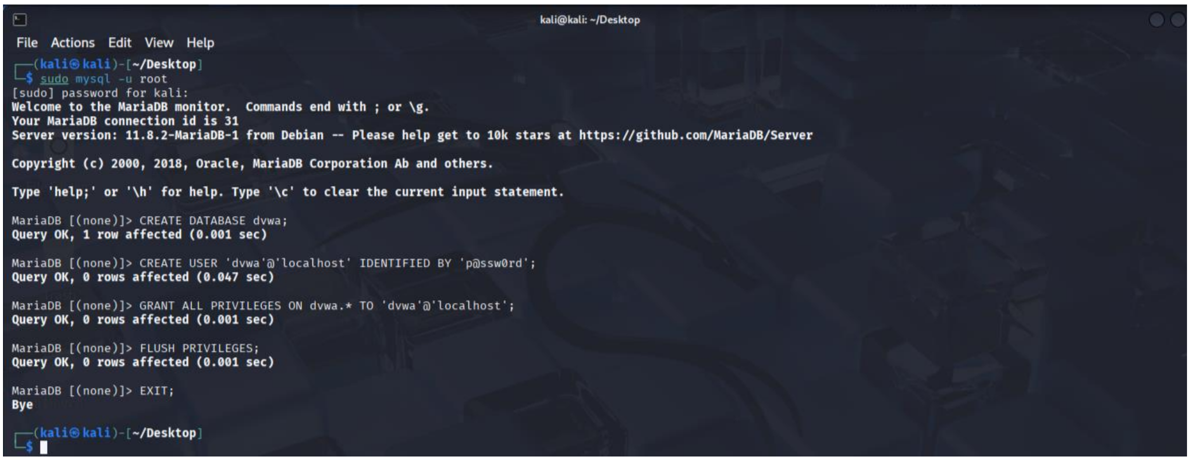

kali@kali: /var/www/html/dvwa/config
File Actions Edit View Help
kali@kali: /var/www/html/dvwa/config
kali@kali: /var/www/html/dvwa/config

```
  GNU nano 8.4                                    /var/www/html/dvwa/config/config.inc.php
<?php

# If you are having problems connecting to the MySQL database and all of the variables below are correct
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a problem due to sockets.
#   Thanks to @digininja for the fix.

# Database management system to use
$DBMS = getenv('DBMS') ?: 'MySQL';
#$DBMS = 'PGSQL'; // Currently disabled

# Database variables
#   WARNING: The database specified under db_database WILL BE ENTIRELY DELETED during setup.
#   Please use a database dedicated to DVWA.
#
# If you are using MariaDB then you cannot use root, you must use create a dedicated DVWA user.
#   See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ]   = getenv('db_server') ?: '127.0.0.1';
$_DVWA[ 'db_database' ] = getenv('db_database') ?: 'dvwa';
$_DVWA[ 'db_user' ]     = getenv('db_user') ?: 'dvwa';
$_DVWA[ 'db_password' ] = getenv('db_password') ?: 'p@ssw0rd';
$_DVWA[ 'db_port']      = getenv('db_port') ?: '3306';

# ReCAPTCHA settings
#   Used for the 'Insecure CAPTCHA' module
#   You'll need to generate your own keys at: https://www.google.com/recaptcha/admin
$_DVWA[ 'recaptcha_public_key' ]  = getenv('RECAPTCHA_PUBLIC_KEY') ?: '';
$_DVWA[ 'recaptcha_private_key' ] = getenv('RECAPTCHA_PRIVATE_KEY') ?: '';

# Default security level
#   Default value for the security level with each session.
#   The default is 'impossible'. You may wish to set this to either 'low', 'medium', 'high' or impossible'.
$_DVWA[ 'default_security_level' ] = getenv('DEFAULT_SECURITY_LEVEL') ?: 'impossible';

# Default locale
                                        [ Read 56 lines (converted from DOS format) ]
^G Help       ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket M-B Previous
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^B Where Was   M-F Next
```

- Create the dvwa user in MariaDB (or MySQL)

Open a terminal and run: `sudo mysql -u root` then in the MySQL shell, type this line by line:

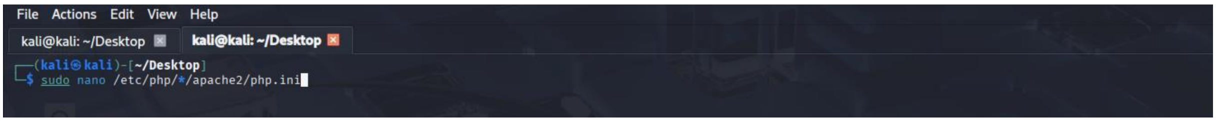
```
CREATE DATABASE dvwa;
CREATE USER 'dvwa'@'localhost' IDENTIFIED BY 'p@ssw0rd';
GRANT ALL PRIVILEGES ON dvwa.* TO 'dvwa'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```



```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo mysql -u root
[sudo] password for kali:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 11.8.2-MariaDB-1 from Debian -- Please help get to 10k stars at https://github.com/MariaDB/Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE dvwa;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> CREATE USER 'dvwa'@'localhost' IDENTIFIED BY 'p@ssw0rd';
Query OK, 0 rows affected (0.047 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON dvwa.* TO 'dvwa'@'localhost';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> EXIT;
Bye

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

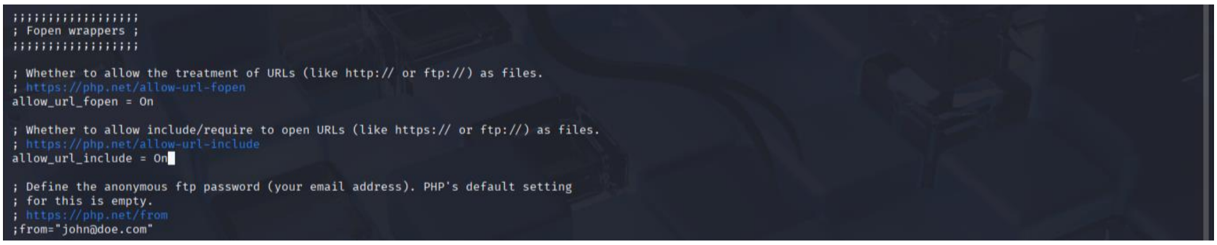
- Edit php.ini to enable `allow_url_include` and `display_errors`

```
sudo nano /etc/php/*/apache2/php.ini
```

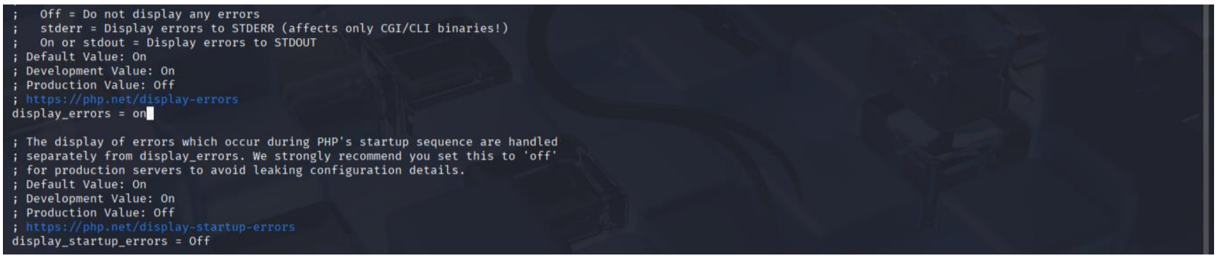


```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo nano /etc/php/*/apache2/php.ini
```

- Search `allow_url_include` : `allow_url_include = On`
- Search display_errors : `display_errors = On`



```
;;;;;;;;;;;;;;;;;;
; Fopen wrappers ;
;;;;;;;;;;;;;;;;;;

; Whether to allow the treatment of URLs (like http:// or ftp://) as files.
; https://php.net/allow-url-fopen
allow_url_fopen = On

; Whether to allow include/require to open URLs (like https:// or ftp://) as files.
; https://php.net/allow-url-include
allow_url_include = On

; Define the anonymous ftp password (your email address). PHP's default setting
; for this is empty.
; https://php.net/from
;from="john@doe.com"
```

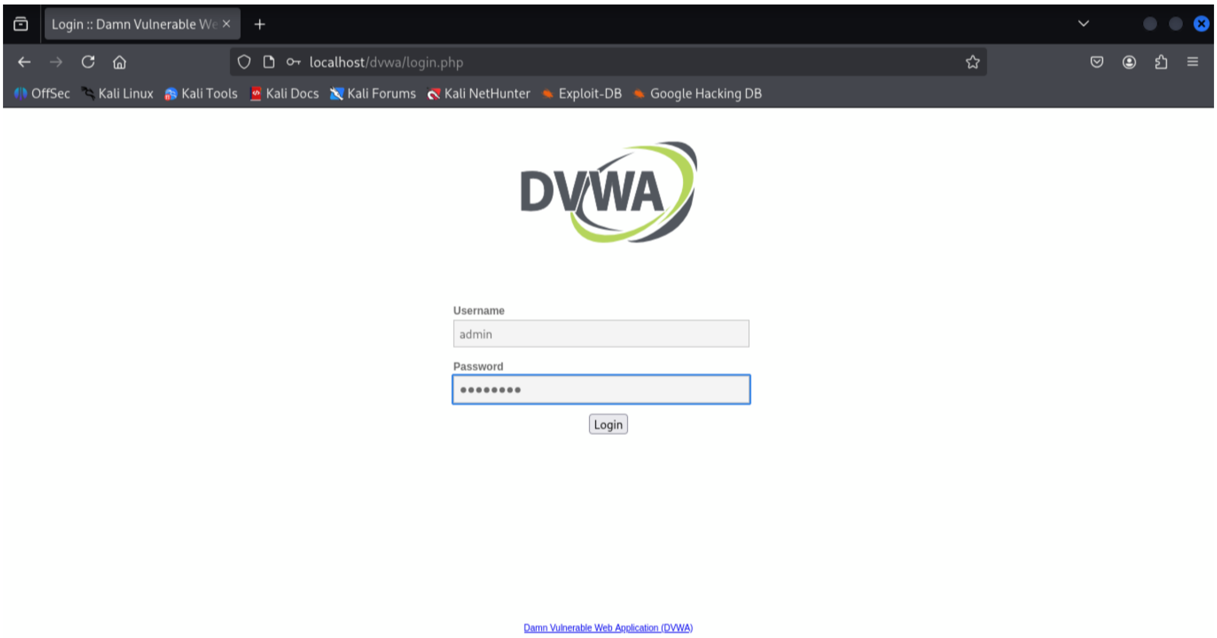


```
;   Off = Do not display any errors
;   stderr = Display errors to STDERR (affects only CGI/CLI binaries!)
;   On or stdout = Display errors to STDOUT
; Default Value: On
; Development Value: On
; Production Value: Off
; https://php.net/display-errors
display_errors = on

; The display of errors which occur during PHP's startup sequence are handled
; separately from display_errors. We strongly recommend you set this to 'off'
; for production servers to avoid leaking configuration details.
; Default Value: On
; Development Value: On
; Production Value: Off
; https://php.net/display-startup-errors
display_startup_errors = Off
```

Saves with ***Ctrl + O***, then ***Enter,*** then exits with ***Ctrl + X***

After activation, it is recommended to restart Apache.

Launch the browser and go to: http://localhost/dvwa/login.php



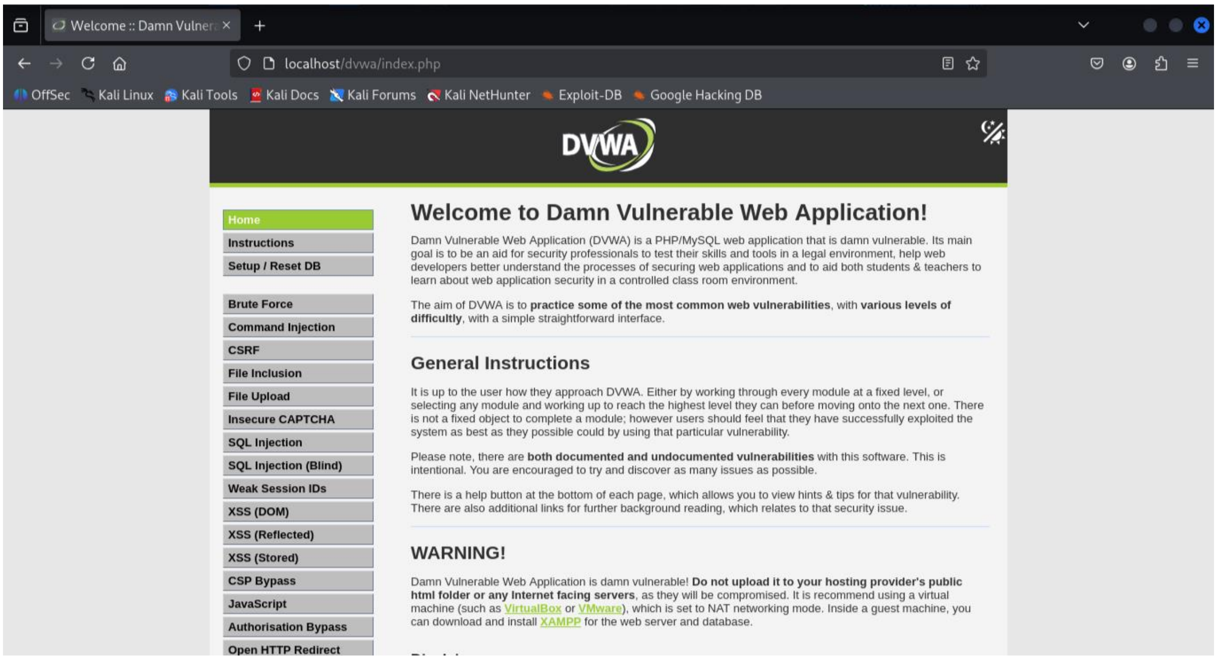
Username: admin

Password: ••••••••

Login ID:

- **Login** : admin
- **Password** : password

DVWA's Welcome Interface



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 difficultly**, 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.

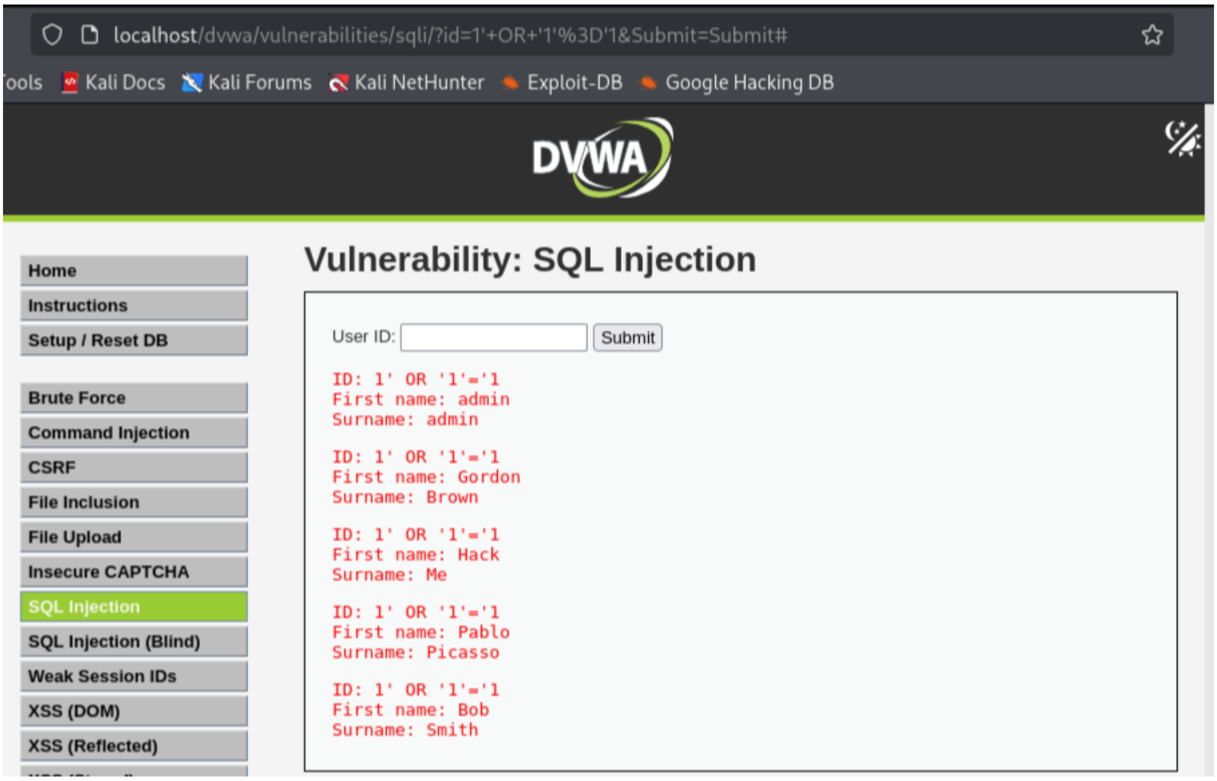
WARNING!

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.

2. LAUNCH OF TESTS (SQLI, XSS, CSRF...)

SQL injection

- Test Input : 1' OR '1'='1
- Result: The application is vulnerable to classic SQL Injection. Sensitive data was extracted by manipulating the SQL query.



localhost/dvwa/vulnerabilities/sqli/?id=1'+OR+'1'%3D'1&Submit=Submit#

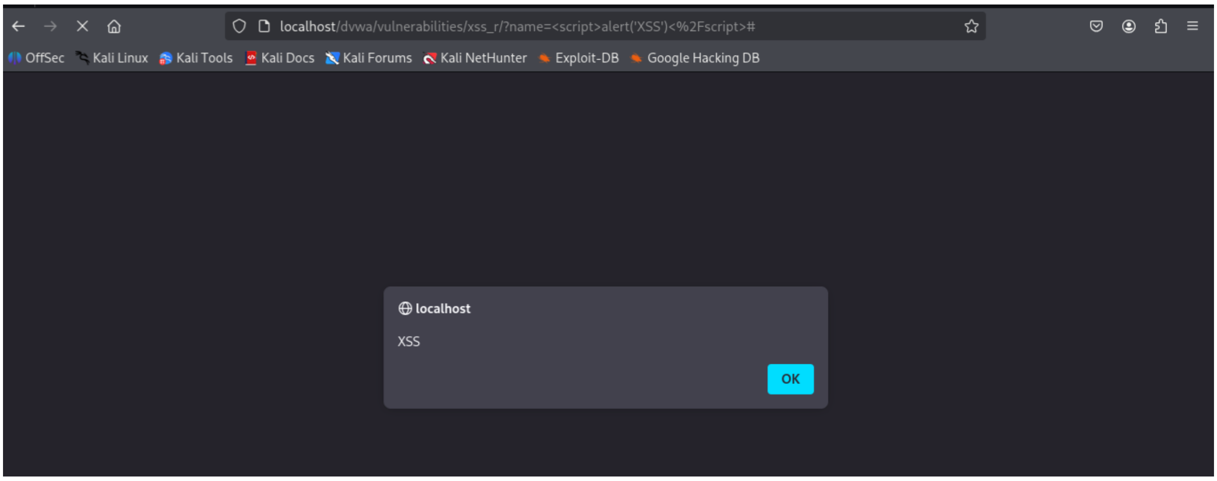
Vulnerability: SQL Injection

User ID:

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

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

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

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

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

Reflected XSS

- Test Input: <script>alert('XSS')</script>
- Result : Input was reflected without sanitization confirming Reflected XSS vulnerability.



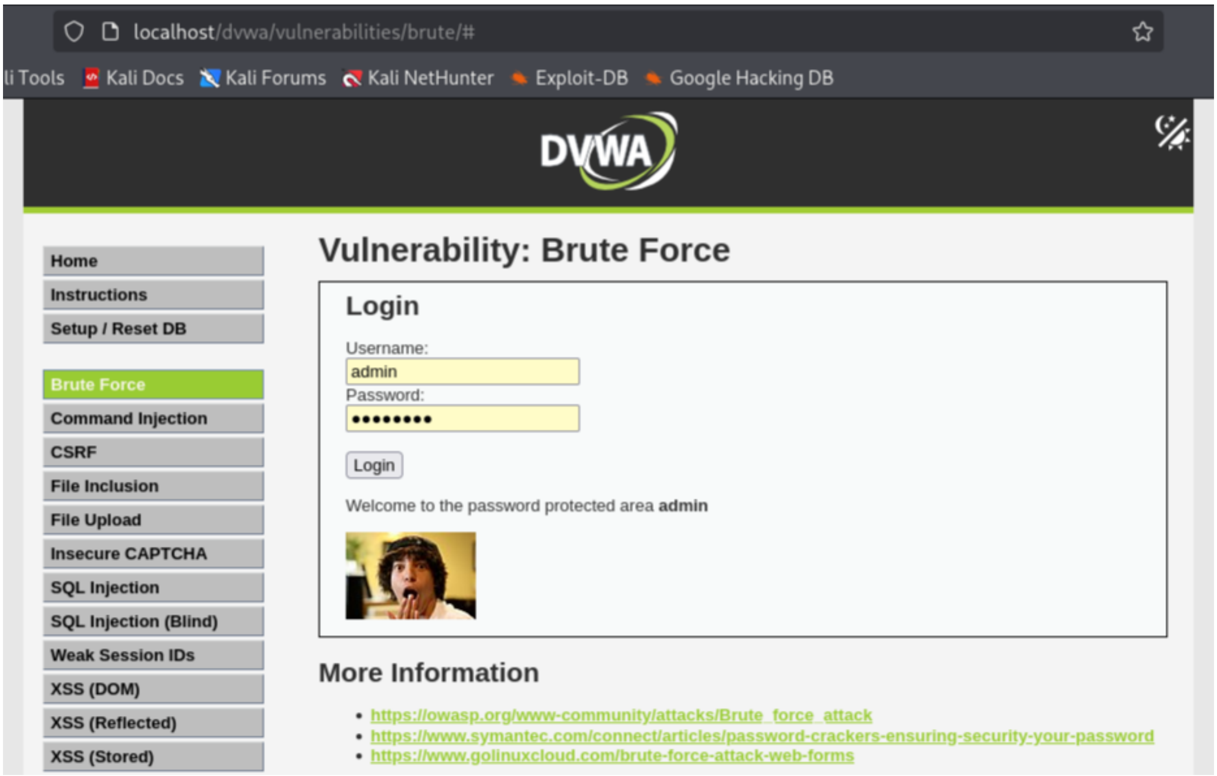
localhost/dvwa/vulnerabilities/xss_r/?name=<script>alert('XSS')<%2Fscript>#

XSS

Stored XSS

- Test Input: <script>alert(' Stored XSS')</script>
- Result : Stored XSS confirmed, malicious code persisted and impacted all users.

Brute Force



Vulnerability: Brute Force

Login

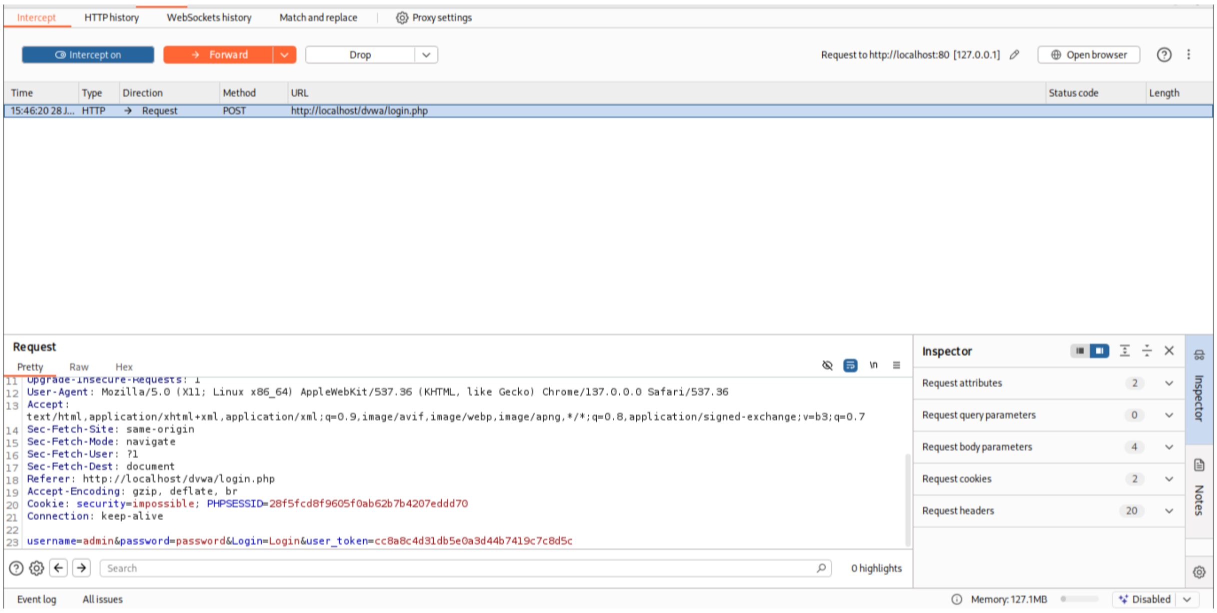
Username: admin

Password: ●●●●●●●●

Welcome to the password protected area **admin**

More Information

- https://owasp.org/www-community/attacks/Brute_force_attack
- https://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password
- https://www.golinuxcloud.com/brute-force-attack-web-forms



Request to http://localhost:80 [127.0.0.1]

Time	Type	Direction	Method	URL	Status code	Length
15:46:20 28 J...	HTTP	→ Request	POST	http://localhost/dvwa/login.php		

Request

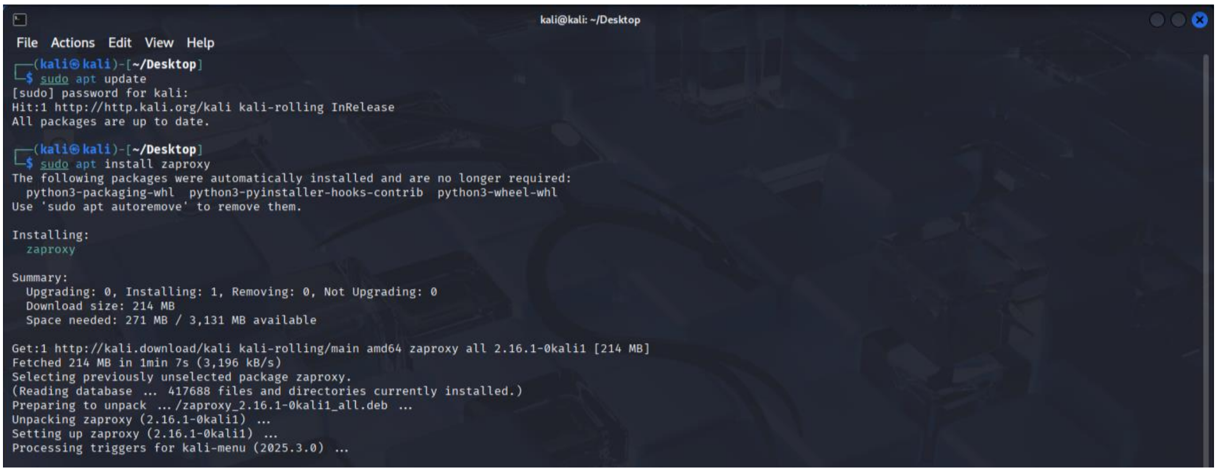
```
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost/dvwa/login.php
Accept-Encoding: gzip, deflate, br
Cookie: security=impossible; PHPSESSID=28f5fcd8f9605f0ab62b7b4207eddd70
Connection: keep-alive

username=admin&password=password&Login=Login&user_token=cc8a8c4d31db5e0a3d44b7419c7c8d5c
```

Inspector	
Request attributes	2
Request query parameters	0
Request body parameters	4
Request cookies	2
Request headers	20

- Install ZAP and run vulnerability scanning

From the official Kali repositories, open the terminal and type the following commands:

```
sudo apt update
sudo apt install zaproxy
```



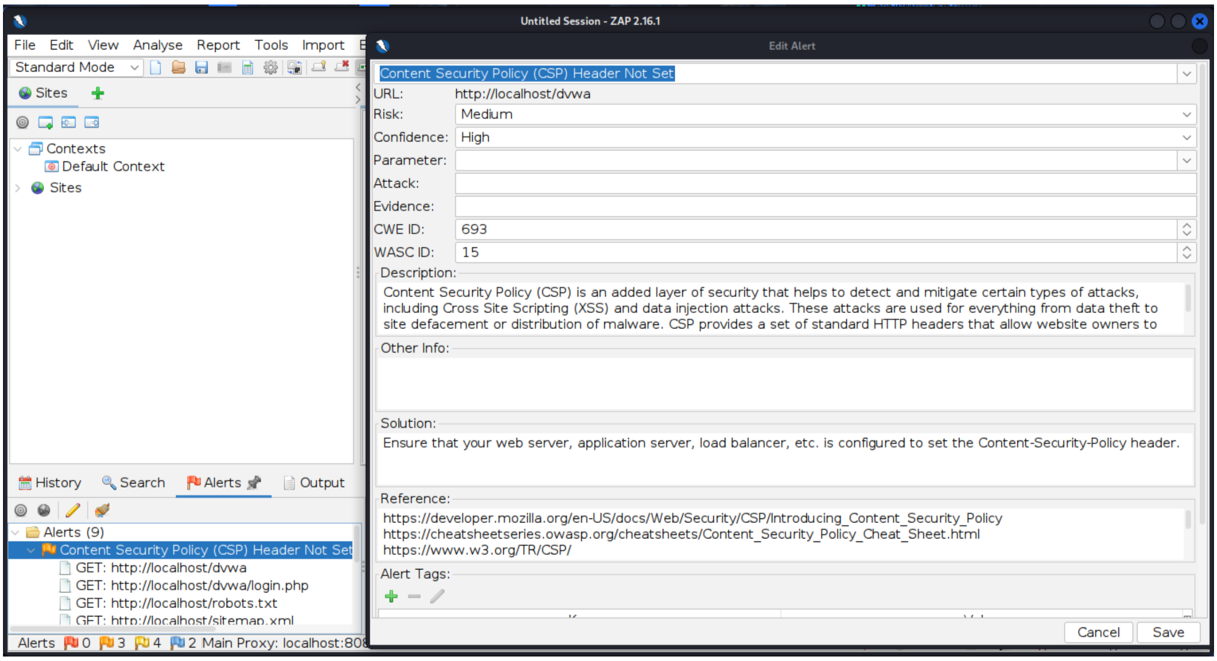
```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo apt update
[sudo] password for kali:
Hit:1 http://http.kali.org/kali kali-rolling InRelease
All packages are up to date.

┌──(kali㉿kali)-[~/Desktop]
└─$ sudo apt install zaproxy
The following packages were automatically installed and are no longer required:
  python3-packaging-whl  python3-pyinstaller-hooks-contrib  python3-wheel-whl
Use 'sudo apt autoremove' to remove them.

Installing:
  zaproxy

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 0
  Download size: 214 MB
  Space needed: 271 MB / 3,131 MB available

Get:1 http://kali.download/kali kali-rolling/main amd64 zaproxy all 2.16.1-0kali1 [214 MB]
Fetched 214 MB in 1min 7s (3,196 kB/s)
Selecting previously unselected package zaproxy.
(Reading database ... 417688 files and directories currently installed.)
Preparing to unpack .../zaproxy_2.16.1-0kali1_all.deb ...
Unpacking zaproxy (2.16.1-0kali1) ...
Setting up zaproxy (2.16.1-0kali1) ...
Processing triggers for kali-menu (2025.3.0) ...
```

ZAP will be installed in `/usr/share/zaproxy/`

To launch it: `zaproxy`

3. DETAILLED VULNERABILITY

- Content security Policy (CSP) Header Not Set


Untitled Session - ZAP 2.16.1
File Edit View Analyse Report Tools Import E
Standard Mode
Sites
Contexts
Default Context
Sites
History
Search
Alerts
Output
Alerts (9)
Alerts 0 3 4 2 Main Proxy: localhost:808
Edit Alert
Cancel
Save

Content Security Policy (CSP) Header Not Set
GET: http://localhost/dvwa
GET: http://localhost/dvwa/login.php
GET: http://localhost/robots.txt
GET: http://localhost/sitemap.xml

Content Security Policy (CSP) Header Not Set

URL: http://localhost/dvwa

Risk: Medium

Confidence: High

Parameter:

Attack:

Evidence:

CWE ID: 693

WASC ID: 15

Description:

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to

Other Info:

Solution:

Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

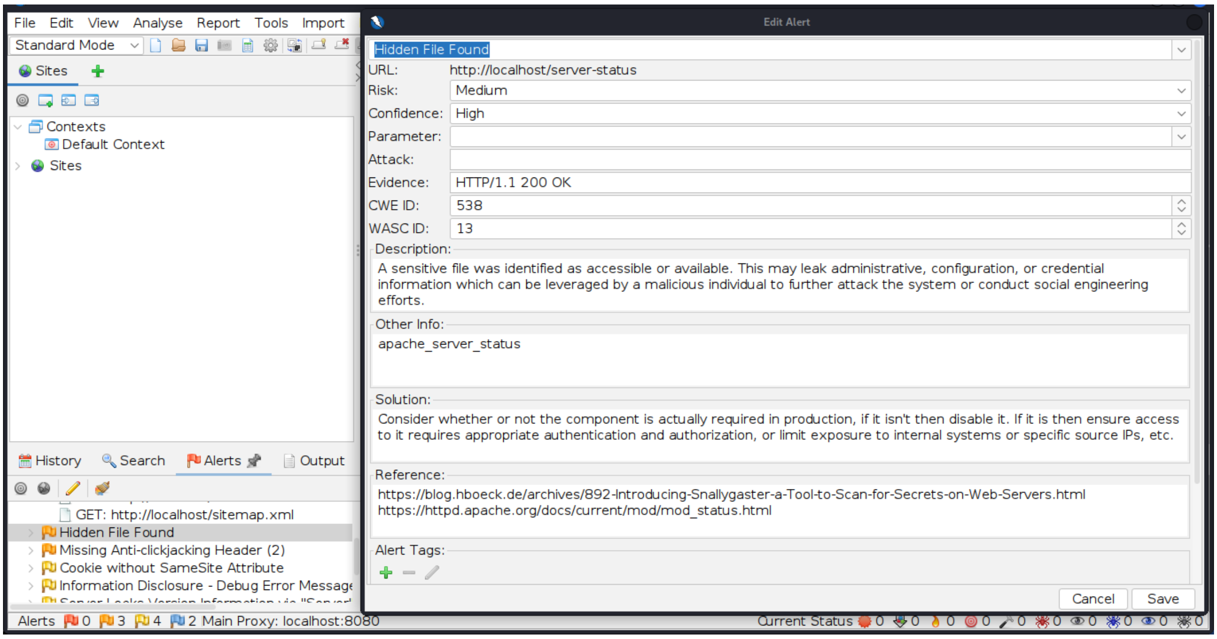
Reference:

https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy
https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
https://www.w3.org/TR/CSP/

Alert Tags:

Description: Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.

- Hidden File Found


File Edit View Analyse Report Tools Import
Standard Mode
Sites
Contexts
Default Context
Sites
History Search Alerts Output
Edit Alert
Cancel Save

GET: http://localhost/sitemap.xml
Hidden File Found
Missing Anti-clickjacking Header (2)
Cookie without SameSite Attribute
Information Disclosure - Debug Error Message

Hidden File Found
URL: http://localhost/server-status
Risk: Medium
Confidence: High
Parameter:
Attack:
Evidence: HTTP/1.1 200 OK
CWE ID: 538
WASC ID: 13

Description:
A sensitive file was identified as accessible or available. This may leak administrative, configuration, or credential information which can be leveraged by a malicious individual to further attack the system or conduct social engineering efforts.

Other Info:
apache_server_status

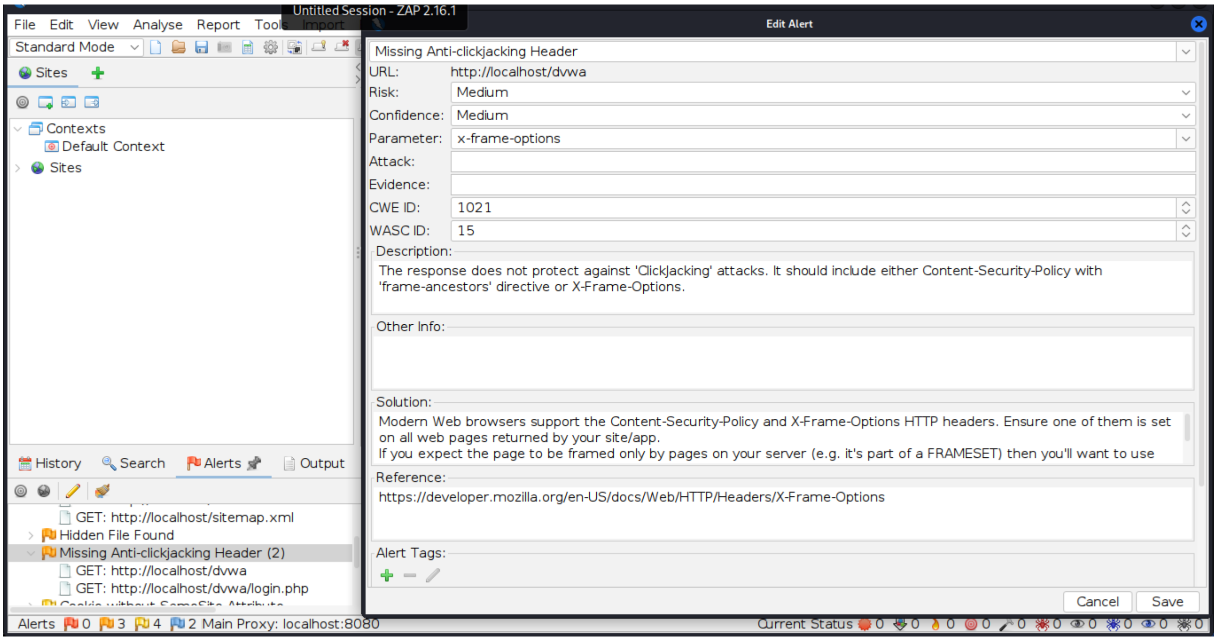
Solution:
Consider whether or not the component is actually required in production, if it isn't then disable it. If it is then ensure access to it requires appropriate authentication and authorization, or limit exposure to internal systems or specific source IPs, etc.

Reference:
https://blog.hboeck.de/archives/892-Introducing-Snallygaster-a-Tool-to-Scan-for-Secrets-on-Web-Servers.html
https://httpd.apache.org/docs/current/mod/mod_status.html

Alert Tags:

Alerts 0 3 4 2 Main Proxy: localhost:8080 Current Status 0 0 0 0 0 0 0 0 0 0

- Missing Anti-clikjacking Header (2)


Untitled Session - ZAP 2.16.1
File Edit View Analyse Report Tools Import
Standard Mode
Sites
Contexts
Default Context
Sites
History Search Alerts Output
Edit Alert
Cancel Save

GET: http://localhost/sitemap.xml
Hidden File Found
Missing Anti-clickjacking Header (2)
GET: http://localhost/dvwa
GET: http://localhost/dvwa/login.php

Missing Anti-clickjacking Header
URL: http://localhost/dvwa
Risk: Medium
Confidence: Medium
Parameter: x-frame-options
Attack:
Evidence:
CWE ID: 1021
WASC ID: 15

Description:
The response does not protect against 'ClickJacking' attacks. It should include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options.

Other Info:

Solution:
Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app.
If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use

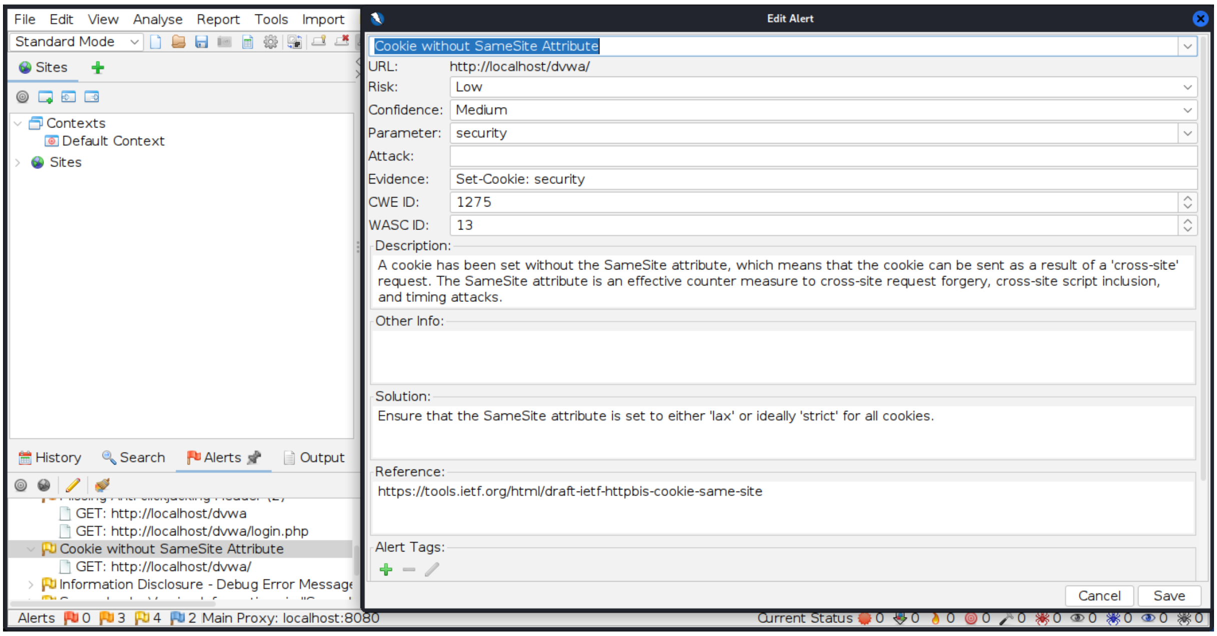
Reference:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

Alert Tags:

Alerts 0 3 4 2 Main Proxy: localhost:8080 Current Status 0 0 0 0 0 0 0 0 0 0

Solution: Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app.
If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.

- Cookie without SameSite Attribute

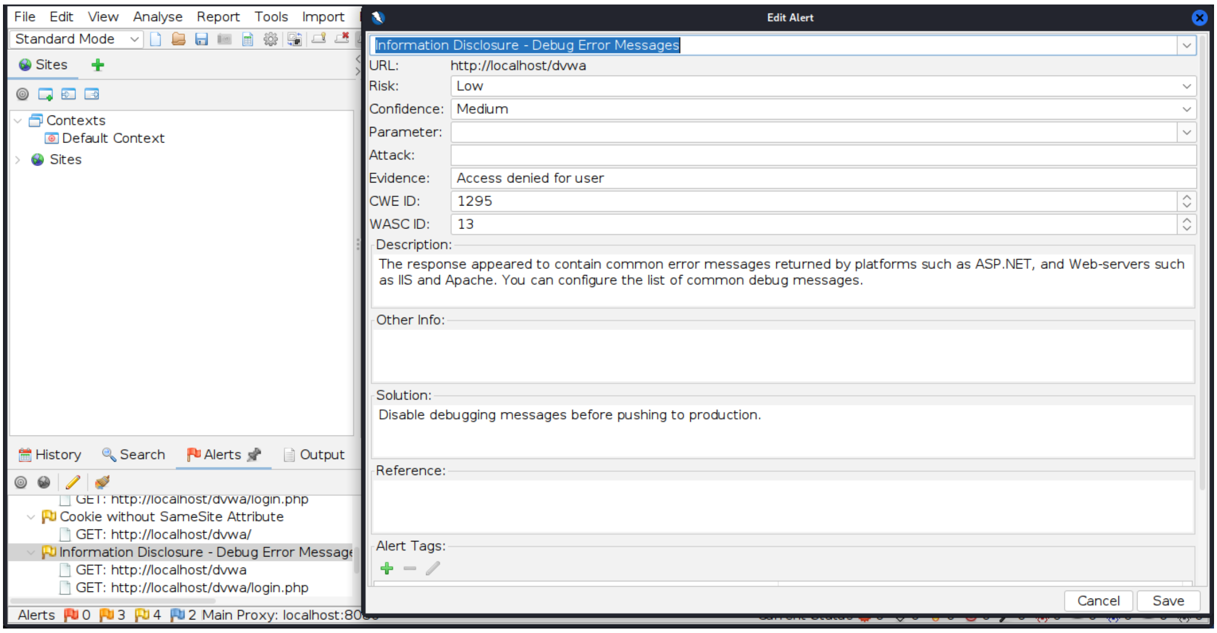

File Edit View Analyse Report Tools Import
Standard Mode
Sites
Contexts
Default Context
Sites
History Search Alerts Output
Edit Alert
Cancel
Save

GET: http://localhost/dvwa
GET: http://localhost/dvwa/login.php
Cookie without SameSite Attribute
GET: http://localhost/dvwa/
Information Disclosure - Debug Error Message

Cookie without SameSite Attribute
URL: http://localhost/dvwa/
Risk: Low
Confidence: Medium
Parameter: security
Attack:
Evidence: Set-Cookie: security
CWE ID: 1275
WASC ID: 13
Description:
A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.
Other Info:
Solution:
Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.
Reference:
https://tools.ietf.org/html/draft-ietf-httpbis-cookie-same-site
Alert Tags:

Alerts 0 3 4 2 Main Proxy: localhost:8080 Current Status 0 0 0 0 0 0 0 0 0 0

- Information Disclosure – Debug Error Messages

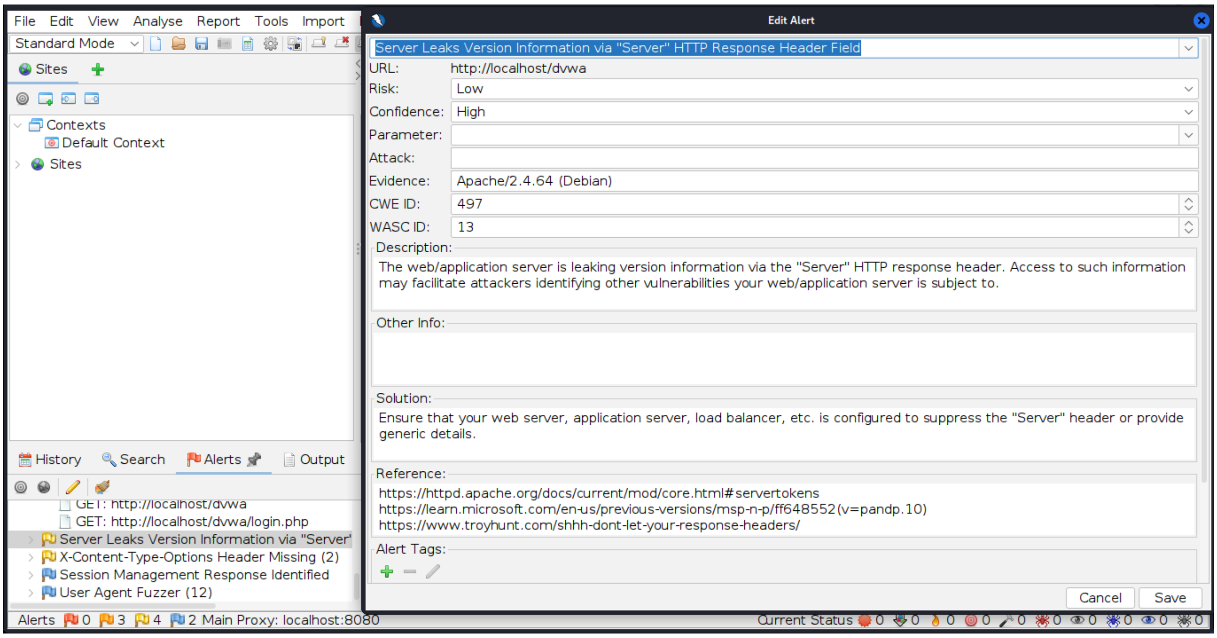

File Edit View Analyse Report Tools Import
Standard Mode
Sites
Contexts
Default Context
Sites
History Search Alerts Output
Edit Alert
Cancel
Save

GET: http://localhost/dvwa/login.php
Cookie without SameSite Attribute
GET: http://localhost/dvwa/
Information Disclosure - Debug Error Message
GET: http://localhost/dvwa
GET: http://localhost/dvwa/login.php

Information Disclosure - Debug Error Messages
URL: http://localhost/dvwa
Risk: Low
Confidence: Medium
Parameter:
Attack:
Evidence: Access denied for user
CWE ID: 1295
WASC ID: 13
Description:
The response appeared to contain common error messages returned by platforms such as ASP.NET, and Web-servers such as IIS and Apache. You can configure the list of common debug messages.
Other Info:
Solution:
Disable debugging messages before pushing to production.
Reference:
Alert Tags:

Alerts 0 3 4 2 Main Proxy: localhost:8080

- **Server Leaks Version Information via "Server" HTTP Response Header Field**


File Edit View Analyse Report Tools Import
Standard Mode
Sites
Contexts
Default Context
Sites
History Search Alerts Output
GET: http://localhost/dvwa
GET: http://localhost/dvwa/login.php
Server Leaks Version Information via "Server"
X-Content-Type-Options Header Missing (2)
Session Management Response Identified
User Agent Fuzzer (12)
Alerts 0 3 4 2 Main Proxy: localhost:8080
Current Status 0 0 0 0 0 0 0 0 0 0
Edit Alert
Cancel
Save

Server Leaks Version Information via "Server" HTTP Response Header Field

URL: http://localhost/dvwa

Risk: Low

Confidence: High

Parameter:

Attack:

Evidence: Apache/2.4.64 (Debian)

CWE ID: 497

WASC ID: 13

Description:

The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.

Other Info:

Solution:

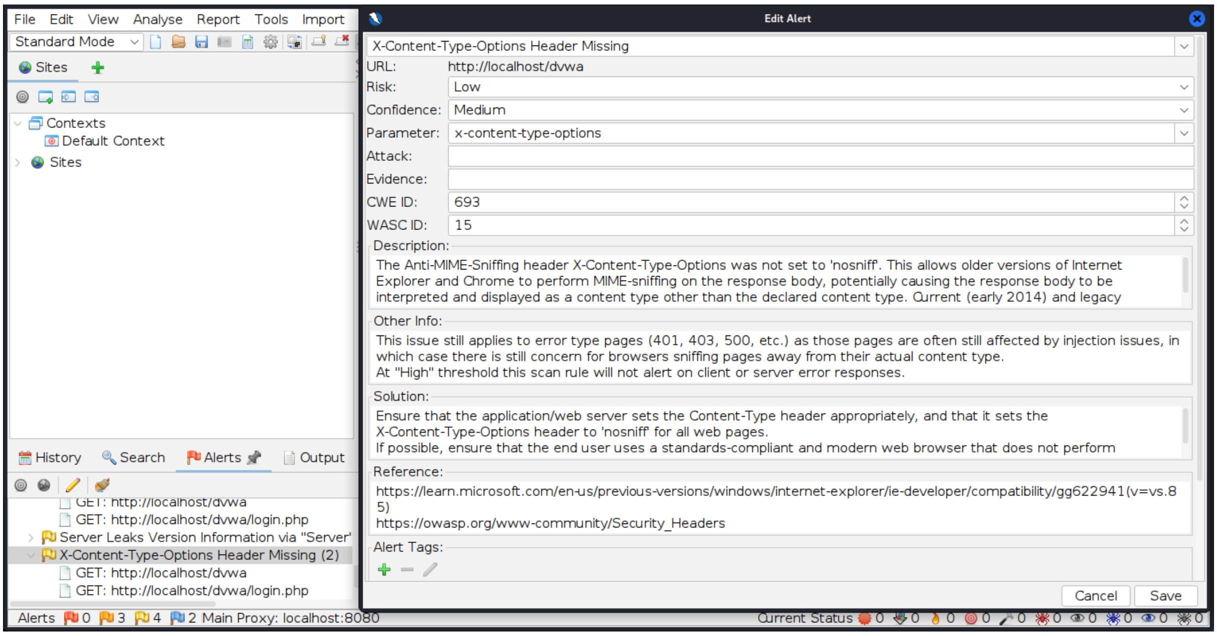
Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Reference:

https://httpd.apache.org/docs/current/mod/core.html#servertokens
https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ff648552(v=pandp.10)
https://www.troyhunt.com/shhh-dont-let-your-response-headers/

Alert Tags:

- **X-Content-Type-Options Header Missing**


File Edit View Analyse Report Tools Import
Standard Mode
Sites
Contexts
Default Context
Sites
History Search Alerts Output
GET: http://localhost/dvwa
GET: http://localhost/dvwa/login.php
Server Leaks Version Information via "Server"
X-Content-Type-Options Header Missing (2)
GET: http://localhost/dvwa
GET: http://localhost/dvwa/login.php
Alerts 0 3 4 2 Main Proxy: localhost:8080
Current Status 0 0 0 0 0 0 0 0 0 0
Edit Alert
Cancel
Save

X-Content-Type-Options Header Missing

URL: http://localhost/dvwa

Risk: Low

Confidence: Medium

Parameter: x-content-type-options

Attack:

Evidence:

CWE ID: 693

WASC ID: 15

Description:

The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy

Other Info:

This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type.
At "High" threshold this scan rule will not alert on client or server error responses.

Solution:

Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.
If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform

Reference:

https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/compatibility/gg622941(v=vs.85)
https://owasp.org/www-community/Security_Headers

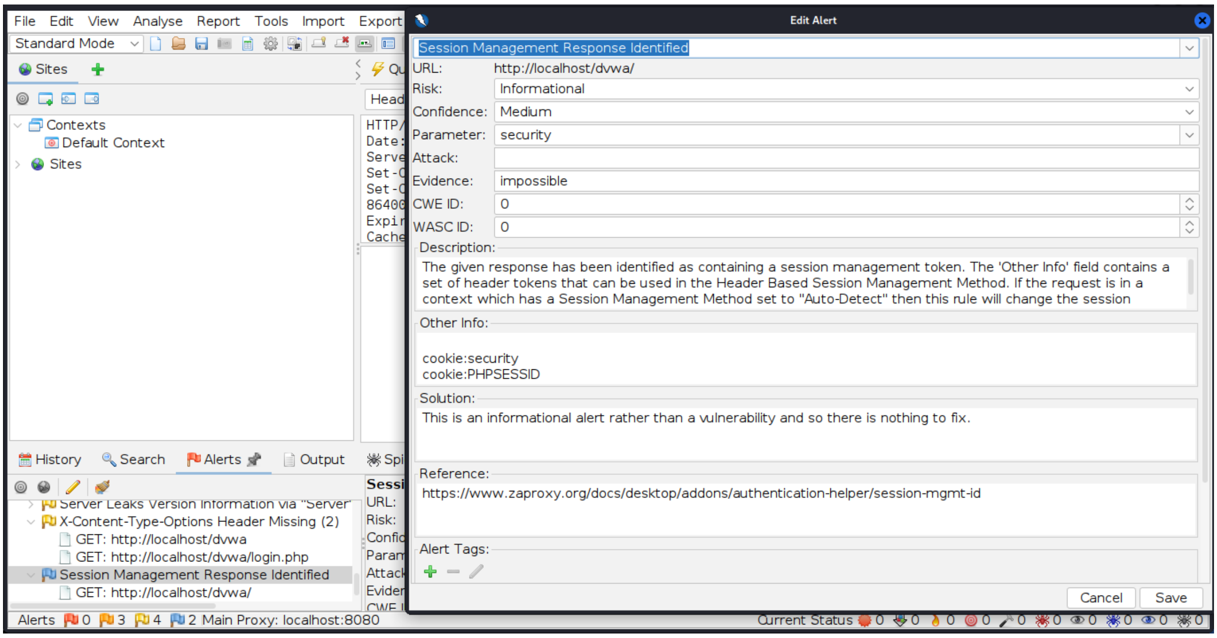
Alert Tags:

Description: The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.

Solution: Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.

If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.

- **Session Management Response Identified**


File
Edit
View
Analyse
Report
Tools
Import
Export
Standard Mode
Sites
Contexts
Default Context
Sites
History
Search
Alerts
Output
Server Leaks Version Information via "Server"
X-Content-Type-Options Header Missing (2)
GET: http://localhost/dvwa
GET: http://localhost/dvwa/login.php
Session Management Response Identified
GET: http://localhost/dvwa/
Alerts 0 3 4 2 Main Proxy: localhost:8080
Edit Alert
Cancel
Save
Current Status 0 0 0 0 0 0 0 0 0 0

Session Management Response Identified

URL: http://localhost/dvwa/

Risk: Informational

Confidence: Medium

Parameter: security

Attack:

Evidence: impossible

CWE ID: 0

WASC ID: 0

Description:

The given response has been identified as containing a session management token. The 'Other Info' field contains a set of header tokens that can be used in the Header Based Session Management Method. If the request is in a context which has a Session Management Method set to "Auto-Detect" then this rule will change the session

Other Info:

cookie:security
cookie:PHPSESSID

Solution:

This is an informational alert rather than a vulnerability and so there is nothing to fix.

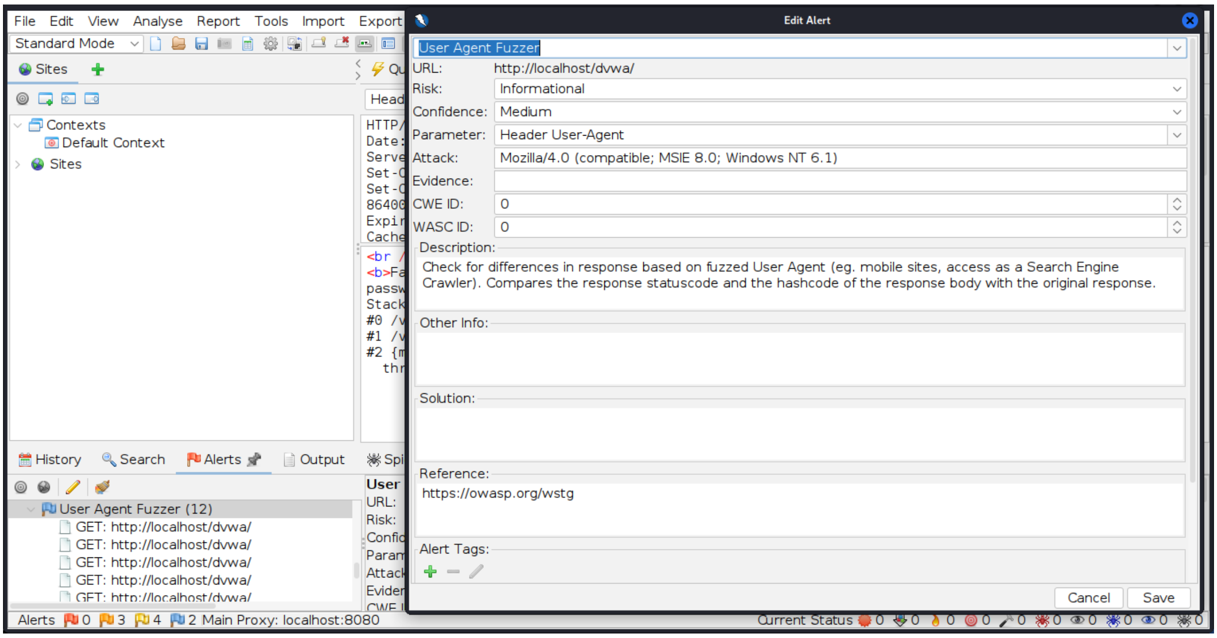
Reference:

https://www.zaproxy.org/docs/desktop/addons/authentication-helper/session-mgmt-id

Alert Tags:

Description: The given response has been identified as containing a session management token. The 'Other Info' field contains a set of header tokens that can be used in the Header Based Session Management Method. If the request is in a context which has a Session Management Method set to "Auto-Detect", then this rule will change the session management to use the tokens identified.

- User Agent Fuzzer



User Agent Fuzzer (12)
GET: http://localhost/dvwa/
GET: http://localhost/dvwa/
GET: http://localhost/dvwa/
GET: http://localhost/dvwa/
GET: http://localhost/dvwa/

User Agent Fuzzer
URL: http://localhost/dvwa/
Risk: Informational
Confidence: Medium
Parameter: Header User-Agent
Attack: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1)
Evidence:
CWE ID: 0
WASC ID: 0
Description: Check for differences in response based on fuzzed User Agent (eg. mobile sites, access as a Search Engine Crawler). Compares the response statuscode and the hashcode of the response body with the original response.
Other Info:
Solution:
Reference: https://owasp.org/wstg
Alert Tags:

Alerts 0 3 4 2 Main Proxy: localhost:8080
Current Status 0 0 0 0 0 0 0 0 0 0

4. RISK ANALYSIS

Risk Level	Count	Priority
Medium	3	Action to be corrected as soon as possible.
Low	4	Action to be corrected if possible, but low priority.
Informational	2	Action to be monitored, useful for attackers in the reconnaissance phase.

5. RECOMMENDATIONS

- Conduct a Comprehensive Code Review

Perform a thorough audit of the application's source code to identify and remediate insecure design patterns and potential security flaws.

- Regularly Update JavaScript Libraries

Maintain an up-to-date inventory of all JavaScript dependencies and ensure timely updates to mitigate risks from publicly known vulnerabilities (e.g., via tools like Snyk or npm audit).

- Implement Security Headers

Enable and correctly configure HTTP security headers, including:

  - Content Security Policy (CSP) to prevent XSS attacks
  - HTTP Strict Transport Security (HSTS) to enforce HTTPS
  - X-Frame-Options to protect against clickjacking attacks

- Strengthen Session and CSRF Protections
  - Enforce token-based CSRF protection mechanisms

Configure secure session policies, including appropriate timeouts, use of HttpOnly and Secure flags, and session regeneration after authentication.

CONCLUSION

This internship was a major formative experience for me in my cybersecurity learning journey. It allowed me to discover and practice concrete techniques for vulnerability analysis, penetration testing and security audits, in a supervised, ethical and professional framework.

Through the exploitation of environments such as Kali Linux, the analysis of vulnerable applications such as DVWA, the use of specialized tools (such as OWASP ZAP, Burp Suite), I acquired a solid technical foundation on:

- Identification and exploitation of common vulnerabilities (XSS, SQLi, CSRF, etc.); The methodology of a web security audit based on the framework
- OWASP Top 10;
- The implementation of a controlled, secure test environment that complies with the best practices of the field;
- The importance of clear, structured and professional documentation of results.

This internship also allowed me to understand the ethical and legal dimension of offensive cybersecurity, in particular in strict compliance with test environments and rules of engagement. This experience not only gave me real-world technical skills, but it also strengthened my motivation to evolve in the field of cybersecurity, continuing to learn, practice and respect the fundamental principles of security and responsibility.