



Sonia Salido

---

# Índice

## **1.- DVWA en Debian**

- 1.1 Implementando DVWA en Debian
- 1.2 Archivo de configuración de DVWA
- 1.3 Configurando mySQL
- 1.4 Configurar PHP
- 1.5 Iniciamos DVWA desde el navegador

## **2.- DVWA en Docker**

- 2.1 Implementando DVWA en Docker
- 2.2 Iniciando sesión en DVWA
- 2.3 Creamos la Base de datos
- 2.4 Establecemos el nivel de dificultad

# 1.- DVWA en Debian

## 1.1 Implementando DVWA en Debian

Montaremos la plataforma a través de una máquina virtual debian 11.6.

```
sudo apt update  
sudo apt install -y apache2 mariadb-server mariadb-client php php-mysqli php-gd libapache2-mod-php
```

Accedemos a la carpeta de apache para instalar DVWA:

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

```
usuario@debian:/var/www$ sudo git clone https://github.com/digininja/DVWA  
Clonando en 'DVWA'...  
remote: Enumerating objects: 4221, done.  
remote: Total 4221 (delta 0), reused 0 (delta 0), pack-reused 4221  
Recibiendo objetos: 100% (4221/4221), 1.86 MiB | 2.23 MiB/s, listo.  
Resolviendo deltas: 100% (2008/2008), listo.  
usuario@debian:/var/www$ █
```

Ponemos los permisos adecuados para poder acceder al contenido:

```
sudo chmod -Rf 777 DVWA/
```

## 1.2 Archivo de configuración de DVWA

El archivo predeterminado de la configuración de DVWA se encuentra en el directorio /config/config.inc.php.dist. Renombramos este fichero →

```
cd DVWA
sudo cp config/config.inc.php.dist config/config.inc.php
```

```
usuario@debian:/var/www$ cd DVWA/
usuario@debian:/var/www/DVWA$ sudo cp config/config.inc.php.dist config/config.inc.php
usuario@debian:/var/www/DVWA$ █
```

De este fichero de configuración debemos ajustar algunos parámetros. Modificamos y guardamos:

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

```

usuario@debian: /var/www/DVWA
GNU nano 5.4 config/config.inc.php *
# 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' ] = '127.0.0.1';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ] = 'user';
$_DVWA[ 'db_password' ] = 'passWord';
$_DVWA[ 'db_port' ] = '3306';
```

## 1.3 Configurando mySQL

Arrancamos Apache:

```
sudo service mysql start
```

```
usuario@debian:/var/www/DVWA$ sudo service mysql start
usuario@debian:/var/www/DVWA$ █
```

Accedemos a mysql:

```
sudo mysql -u root -p
```

---

```
usuario@debian:/var/www/DVWA$ sudo mysql -u root -p
```

```
Enter password:
```

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
```

```
Your MariaDB connection id is 30
```

```
Server version: 10.5.18-MariaDB-0+deb11u1 Debian 11
```

```
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)]> █
```

Creamos la BD:

```
create database dvwa;
```

Creamos un usuario de base de datos con el nombre 'user' y la contraseña 'password' en nuestro servidor localhost en '127.0.0.1' usando el siguiente comando:

```
create user 'user'@'127.0.0.1' identified by 'password';
```

```
MariaDB [(none)]> create user 'user'@'127.0.0.1' identified by 'password';  
Query OK, 0 rows affected (0,005 sec)
```

```
MariaDB [(none)]> █
```

Otorgamos privilegios de usuario en toda la base de datos 'dvwa'. Ejecutamos el siguiente comando. Debe proporcionar el nombre de usuario, la contraseña y la información del servidor nuevamente.

```
grant all privileges on dvwa.* to 'user'@'127.0.0.1' identified by 'password';
```

```
MariaDB [(none)]> grant all privileges on dvwa.* to 'user'@'127.0.0.1' identified by 'password';  
Query OK, 0 rows affected (0,007 sec)
```

```
MariaDB [(none)]> █
```

Cerramos la conexión con la base de datos con el comando exit.

## 1.4 Configurar PHP

Debemos modificar el fichero php.ini para hacer cambios en la función de PHP allow\_url\_include para habilitarla:

```
sudo nano /etc/php/7.4/apache2/php.ini
```



```
GNU nano 5.4 /etc/php/7.4/apache2/php.ini *
;;;;;;;;;;;;;;;;;;;;;;;;
; Fopen wrappers ;
;;;;;;;;;;;;;;;;;;;;;;;;

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

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

Guardamos y cerramos el documento.

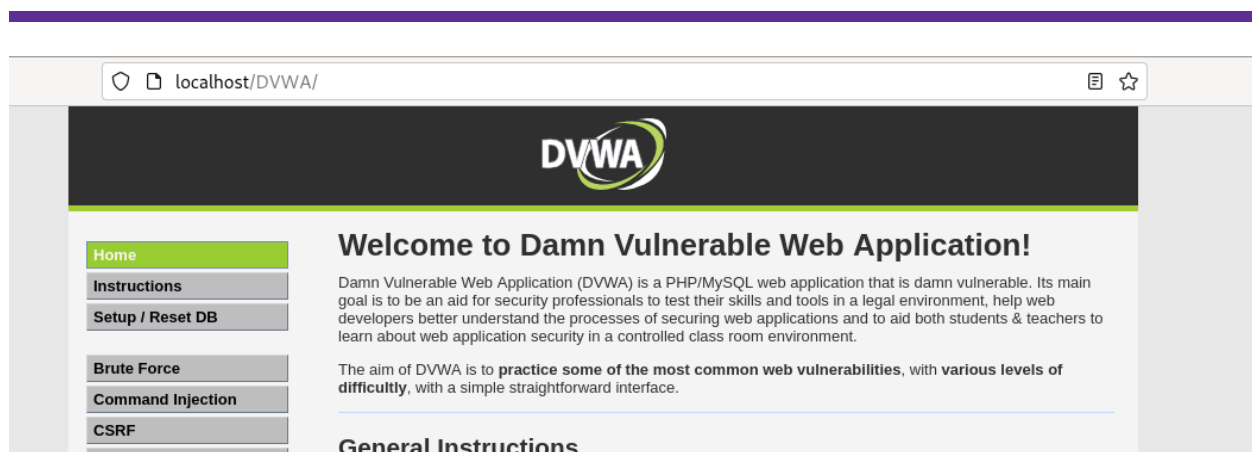
## 1.5 Iniciamos DVWA desde el navegador

Reiniciamos el servicio de Apache y accedemos en un navegador a la url:

```
sudo systemctl restart apache2.service
sudo service mysql restart
```

```
usuario@debian:/var/www/DVWA$ sudo systemctl restart apache2.service
usuario@debian:/var/www/DVWA$ sudo service mysql restart
usuario@debian:/var/www/DVWA$ █
```

Accedemos en el navegador a la url: localhost/DVWA/ → Login to DVWA with default credential → admin / password.



## 2.- DVWA en Docker

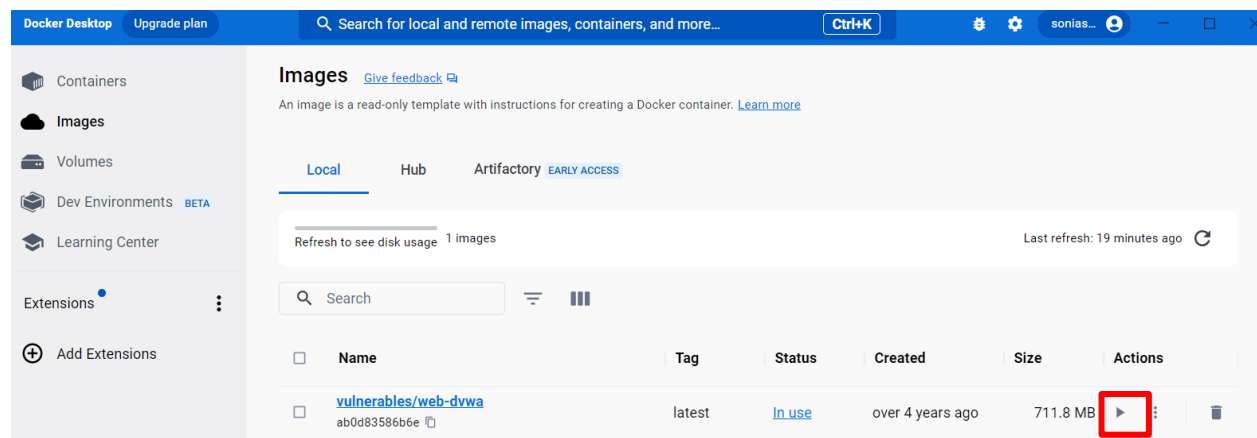
### 2.1 Implementando DVWA en Docker

Montaremos la plataforma a través de un contenedor Docker en Windows. Descargamos la imagen: <https://hub.docker.com/r/vulnerables/web-dvwa/>

```
docker pull vulnerables/web-dvwa
```

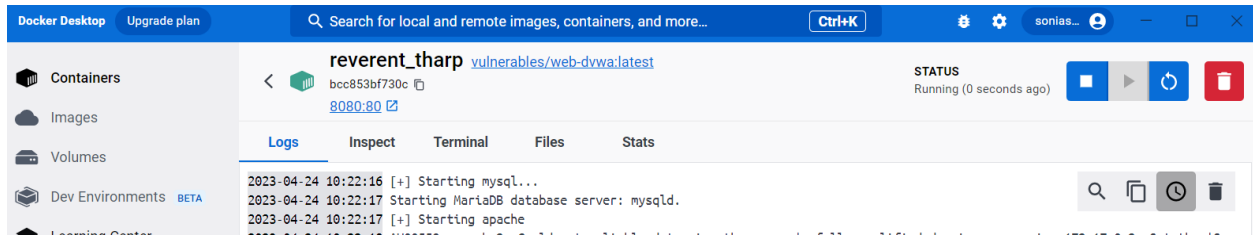
```
MINGW64:/c/Users/Usuario/Documents
Usuario@Laptop-Soina MINGW64 ~/Documents
$ docker pull vulnerables/web-dvwa
Using default tag: latest
latest: Pulling from vulnerables/web-dvwa
3e17c6eae66c: Pulling fs layer
0c57df616dbf: Pulling fs layer
eb05d18be401: Pulling fs layer
e9968e5981d2: Pulling fs layer
2cd72dba8257: Pulling fs layer
6c5ff5f35147f: Pulling fs layer
098c5ff43466: Pulling fs layer
b3d64a33242d: Pulling fs layer
```

Una vez descargado la imagen, la ejecutaremos haciendo click en el icono de Ejecutar:





Vemos cómo se inicia el contenedor:



## 2.2 Iniciando sesión en DVWA

En un navegador web escribimos la url → localhost:numeroPuerto



Username

Password

Login

Iniciamos sesión con las credenciales predeterminadas:

Nombre de usuario: admin

contraseña: password

## 2.3 Creamos la Base de datos

Creamos la Base de datos haciendo clic en el botón Create/Reset Database →

[User: www-data] Writable folder /var/www/html/config: **Yes**

**Status in red**, indicate there will be an issue when trying to complete some modules.

If you see disabled on either `allow_url_fopen` or `allow_url_include`, set the following in your `php.ini` file and restart Apache.


```
allow_url_fopen = On
allow_url_include = On
```

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them.

Create / Reset Database

First time using DVWA.  
Need to run 'setup.php'.

Termina la creación de las BD y nos regresa a la página de inicio para volver a iniciar sesión. Introducimos los credenciales y vemos la plataforma →



[Home](#)  
[Instructions](#)  
[Setup / Reset DB](#)  
  
[Brute Force](#)  
[Command Injection](#)  
[CSRF](#)  
[File Inclusion](#)  
[File Upload](#)  
[Insecure CAPTCHA](#)  
[SQL Injection](#)

## 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.

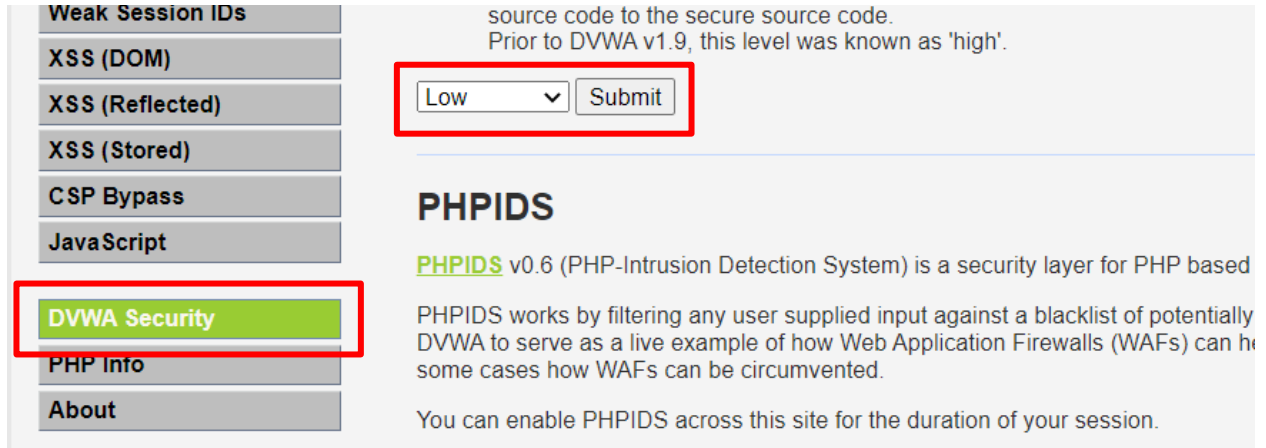
---

### 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.

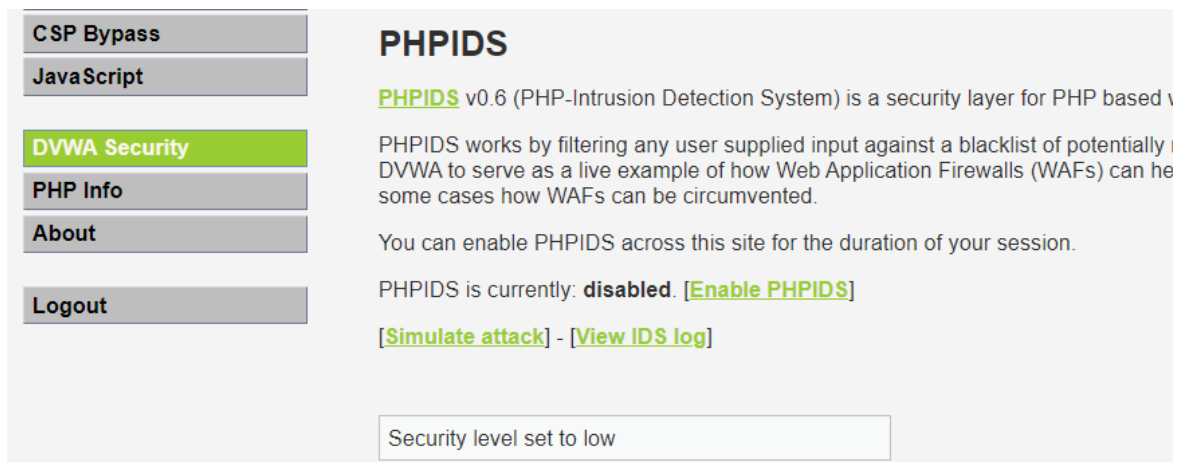
## 2.4 Establecemos el nivel de dificultad

Vamos a establecer el nivel de dificultad a Low →



The screenshot shows the DVWA Security page. On the left is a sidebar with menu items: Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript, DVWA Security (highlighted in green), PHP Info, and About. The main content area has a heading "source code to the secure source code. Prior to DVWA v1.9, this level was known as 'high'." Below this is a form with a dropdown menu set to "Low" and a "Submit" button, both enclosed in a red box. Further down is the "PHPIDS" section, which describes the PHP-Intrusion Detection System and mentions that it can be enabled across the site for the duration of the session.

Verificamos el cambio:



The screenshot shows the DVWA Security page after the change. The sidebar menu is the same, but "DVWA Security" is highlighted in green. The main content area shows the "PHPIDS" section. It states that PHPIDS v0.6 is a security layer for PHP based on a blacklist of potentially dangerous input. It mentions that PHPIDS works by filtering any user supplied input against a blacklist of potentially dangerous input. It also states that PHPIDS is currently disabled and provides links to "Enable PHPIDS", "Simulate attack", and "View IDS log". At the bottom, there is a box that says "Security level set to low".