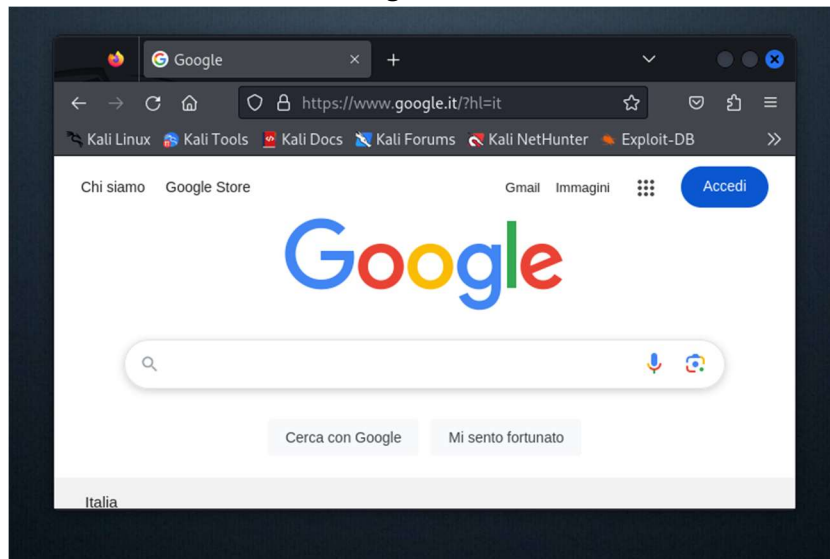


Il primo controllo che ho effettuato è stato verificare se Kali si connettesse ad internet dopo aver modificato la scheda di rete in “Scheda con bridge”:



### Configurazione di DVWA:

Mi sono spostato nella directory predefinita per rendere disponibili i file del server web Apache.

Successivamente sono andato a scaricare “DVWA” da GitHub

Poi ho modificato i permessi (chmod) della directory appena creata (/DVWA) i massimi permessi, ovvero di lettura, scrittura ed esecuzione (777)

Sono entrato nella cartella, ho visualizzato il contenuto e copiato il file \*.dist rinominandolo senza .dist.

E infine ho aperto con un modificatore di testo (nano) quel file.

```
root@kali: /var/www/html/DVWA/config
File Actions Edit View Help

(root@kali)-[/var/www/html/DVWA]
# cd /var/www/html

(root@kali)-[/var/www/html]
# git clone https://github.com/digininja/DVWA

(root@kali)-[/var/www/html]
# chmod -R 777 DVWA/

(root@kali)-[/var/www/html]
# cd DVWA/config

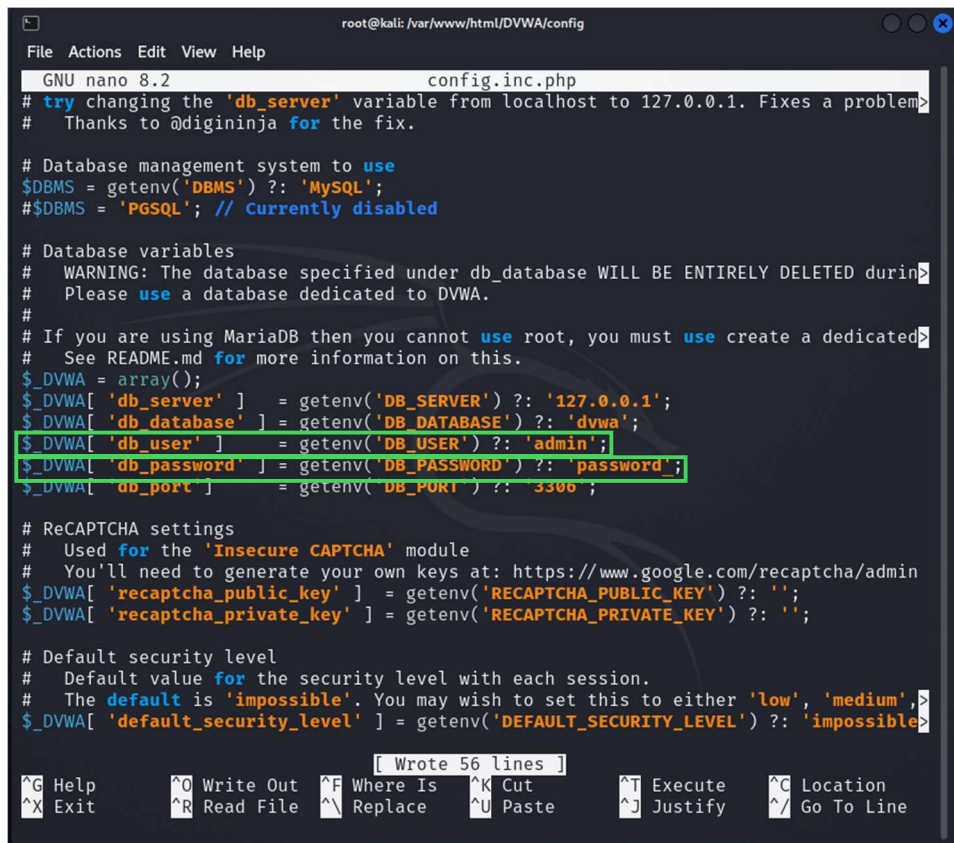
(root@kali)-[/var/www/html/DVWA/config]
# ls
config.inc.php.bak  config.inc.php.dist

(root@kali)-[/var/www/html/DVWA/config]
# cp config.inc.php.dist config.inc.php

(root@kali)-[/var/www/html/DVWA/config]
# nano config.inc.php
```

Dopo aver aperto il file, sono andato a modificare i parametri di username e password impostandoli in “admin” e “password”.

Salvato il file col comando “cmd+x”:



```
root@kali: /var/www/html/DVWA/config
File Actions Edit View Help
GNU nano 8.2 config.inc.php
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a problem>
# 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 durin>
# Please use a database dedicated to DVWA.

# If you are using MariaDB then you cannot use root, you must use create a dedicated>
# 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') ?: 'admin';
$_DVWA[ 'db_password' ] = getenv('DB_PASSWORD') ?: 'password';
$_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',>
$_DVWA[ 'default_security_level' ] = getenv('DEFAULT_SECURITY_LEVEL') ?: 'impossible'>

[ Wrote 56 lines ]
^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line
```

## MySQL

Successivamente ho avviato il database con il servizio MySQL all'interno della stessa directory dichiarando l'utente root come utente (-u) con l'inserimento della password che é la stessa che ho inserito nel file precedente:



```
root@kali: /var/www/html/DVWA/config
File Actions Edit View Help

(root@kali)-[/var/www/html/DVWA/config]
# service mysql start

(root@kali)-[/var/www/html/DVWA/config]
# mysql -u root -p
Enter password: password
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 11.4.2-MariaDB-4 Debian n/a

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

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> _
```

Adesso creo un utenza sul database usando il comando:

```
MariaDB [(none)]> create user 'kali'@'127.0.0.1' identified by 'kali';
```

Con la successiva assegnazione di privilegi:

```
MariaDB [(none)]> grant all privileges on dvwa.* to 'kali'@'127.0.0.1' identified by 'kali';
```

## Apache

Ora che il servizio MySQL è configurato, posso configurare il server web “apache”.

Lo avvio:

```
(root@kali)-[/var/www/html/DVWA/config]
# service apache2 start
```

E successivamente mi sposto nella cartella :

```
(root@kali)-[/var/www/html/DVWA/config]
# cd /etc/php/8.2/apache2
```

Andando a visualizzare i file all'interno, vado a modificare il file “php.ini”:

```
(root@kali)-[/etc/php/8.2/apache2]
# ls
conf.d  php.ini
# nano php.ini
```

Utilizzando “CTRL + F” ho trovato la stringa “allow\_url\_include” configurata su Off e l’ho modificata in “On”:

```
; 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
```

Adesso posso avviare il servizio Apache2 usando il comando iniziale.

## DVWA

Ora posso aprire una scheda del browser e digitando “127.0.0.1/DVWA/setup.php” si apre la schermata di configurazione.

Nella parte inferiore c'è il collegamento alla pagina di configurazione del livello di sicurezza::

Setup :: Damn Vulnerable x

127.0.0.1/DVWA/setup.php

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

### DVWA

**Setup DVWA**

**Instructions**

**About**

## Database Setup

Click on the 'Create / Reset Database' button below to create or reset your database.  
If you get an error make sure you have the correct user credentials in: `/var/www/html/DVWA/config/config.inc.php`

If the database already exists, it will be cleared and the data will be reset.  
You can also use this to reset the administrator credentials ("admin // password") at any stage.

### Setup Check

Web Server SERVER\_NAME: 127.0.0.1

Operating system: \*nix

PHP version: 8.2.21  
PHP function display\_errors: Disabled  
PHP function display\_startup\_errors: Disabled  
PHP function allow\_url\_include: Enabled  
PHP function allow\_url\_fopen: Enabled  
PHP module gd: Missing - Only an issue if you want to play with captchas  
PHP module mysql: Installed  
PHP module pdo\_mysql: Installed

Backend database: MySQL/MariaDB  
Database username: admin  
Database password: \*\*\*\*\*  
Database database: dvwa  
Database host: 127.0.0.1  
Database port: 3306

reCAPTCHA key: Missing

Writable folder /var/www/html/DVWA/hackable/uploads/: Yes  
Writable folder /var/www/html/DVWA/config: Yes

**Status is 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**

127.0.0.1/DVWA/index.php

Kali Tools

Kali Docs

Kali Forums

Kali NetHunter

Exploit-DB

Google Hacking DB

OffSec

DVWA

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

DVWA Security

PHP Info

About

Logout

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.

Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). 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.

More Training Resources

DVWA aims to cover the most commonly seen vulnerabilities found in today's web applications. However there are plenty of other issues with web applications. Should you wish to explore any additional attack vectors, or want more difficult challenges, you may wish to look into the following other projects:

Nella schermata di sinistra sono presenti tutte le vulnerabilità e andando su DVWA security imposto la sicurezza in “low”.

DVWA

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

DVWA Security

PHP Info

About

Logout

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

Low

Medium

High

Impossible