

DVWA- Setup

Damn Vulnerable Web Application (DVWA) is a widely used PHP/MySQL web application designed to simulate a variety of security vulnerabilities, making it an invaluable educational tool for security professionals, students, and developers. It features a comprehensive range of common web vulnerabilities, such as SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), command injection, and file inclusion, among others. DVWA's adjustable security levels (low, medium, high, and impossible) allow users to progressively test and understand how different security measures impact an application's security. It serves as a realistic training ground, enabling users to practice exploiting vulnerabilities in a safe and legal environment, often in conjunction with tools like Burp Suite, OWASP ZAP, and SQLmap. Widely supported by an active community, DVWA is commonly used in academic settings, training programs, and self-study for certifications like CEH and OSCP. Its ease of installation on platforms like XAMPP, LAMP, or Docker ensures accessibility, making it an essential resource for those aiming to master web application security.

Steps: -

1. Firstly, move to this directory.

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

2. Now, git clone the dvwa tool.

```
(root@kali)-[/var/www/html]  
# git clone https://github.com/digininja/DVWA.git  
Cloning into 'DVWA' ...  
remote: Enumerating objects: 4590, done.  
remote: Counting objects: 100% (140/140), done.  
remote: Compressing objects: 100% (102/102), done.  
remote: Total 4590 (delta 58), reused 102 (delta 37), pack-reused 4450  
Receiving objects: 100% (4590/4590), 2.31 MiB | 5.90 MiB/s, done.  
Resolving deltas: 100% (2169/2169), done.
```

3. Now move to the DVWA directory and change the permission to 777 i.e. read, write and execute.

```
(root@kali)-[/var/www/html]  
# ls  
DVWA  index.html  index.nginx-debian.html  
  
(root@kali)-[/var/www/html]  
# chmod -R 777 DVWA
```

4. Now change to DVWA directory.

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

5. Now move to config directory.

```
(root@kali)-[/var/www/html/DVWA]  
# ls  
about.php  COPYING.txt  dvwa  index.php  phpinfo.php  README.fa.md  README.md  robots.txt  setup.php  
CHANGELOG.md  database  external  instructions.php  php.ini  README.fr.md  README.pt.md  SECURITY.md  tests  
compose.yml  Dockerfile  favicon.ico  login.php  README.ar.md  README.id.md  README.tr.md  security.php  vulnerabilities  
config  docs  hackable  logout.php  README.es.md  README.ko.md  README.zh.md  security.txt  
  
(root@kali)-[/var/www/html/DVWA]  
# cd config
```

6. Now change the name of config.inc.php.dist to config.inc.php.

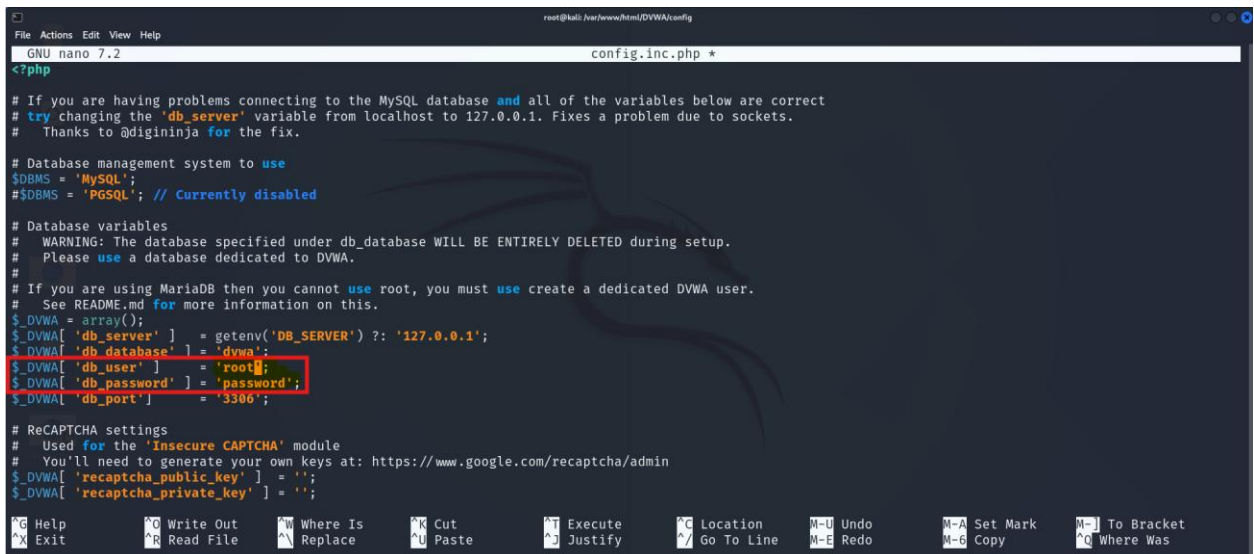
```
(root@kali)-[/var/www/html/DVWA/config]
# ls
config.inc.php.dist

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

7. Now we have to edit this file. To do so use nano editor.

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

8. Now set the user and password.



```
GNU nano 7.2 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 = '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'] = 'dvwa';
$DVWA['db_user'] = 'root';
$DVWA['db_password'] = 'password';
$DVWA['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'] = '';
$DVWA['recaptcha_private_key'] = '';

Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark M-I To Bracket
Exit Read File Replace Paste Justify Go To Line M-E Redo M-6 Copy M-Q Where Was
```

9. Now start the mysql service using following commands.

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

10. Now use the ID and Password to login into the mysql server.

```
(root@kali)-[/var/www/html/DVWA/config]
# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.11.6-MariaDB-2 Debian n/a

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)]> █
```

11. Now create a user inside the Database.

```
MariaDB [(none)]> create user 'user'@'127.0.0.1' identified by 'pass';
Query OK, 0 rows affected (0.014 sec)
```

12. Then we grant this user all the privileges over the database.

```
MariaDB [(none)]> grant all privileges on dvwa.* to 'user'@'127.0.0.1' identified by 'pass';
Query OK, 0 rows affected (0.009 sec)
```

13. Now now to apache2 directory.

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

14. Now edit the php.init file.

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

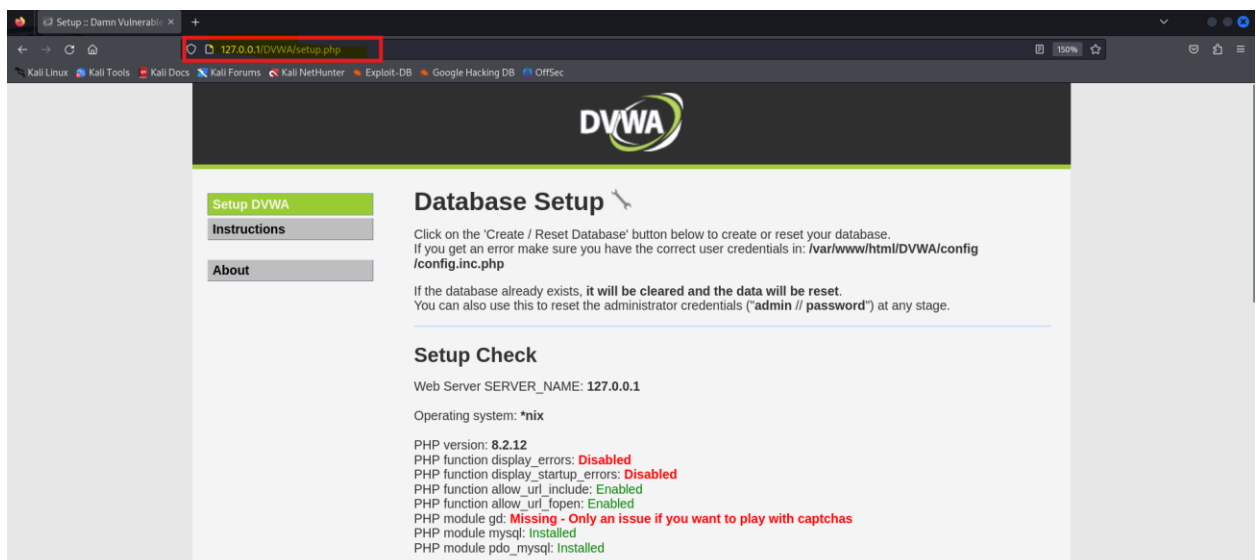
15. Now change the following settings and save it.

```
;;;;;;;;;;;;;;  
; 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
```

16. After this start the apache2 server.

```
(root@kali)-[/etc/php/8.2/apache2]  
# service apache2 start
```

17. Now we will search the local host address in the browser.



18. Now click on the create database button. It will redirect it to the login page.

reCAPTCHA key: **Missing**

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

19. Login to the DVWA using your credentials.



Username

admin

Password


••••••••

Login

You have logged out

The default login is

20. Here we have the dashboard of 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

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 vulnerability** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be seen as an extension for more advanced users)!

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.