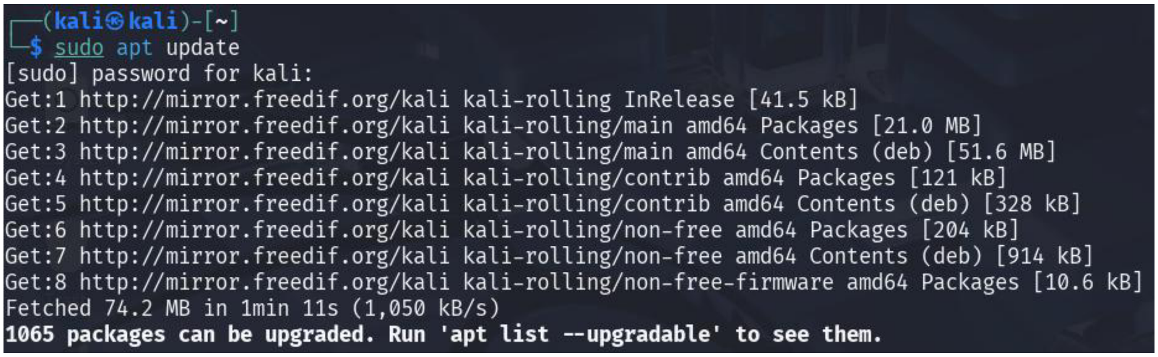
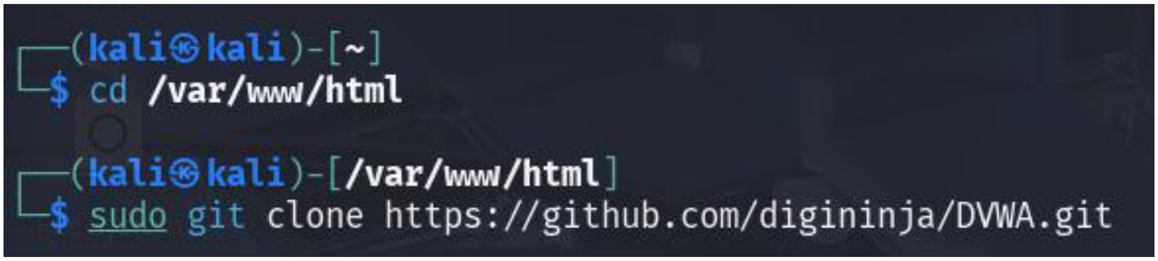
Local Deployment Notes -MagicDragonB

The first step is to make sure the Kali Linux has updated to the latest version. (Using sudo apt update command)



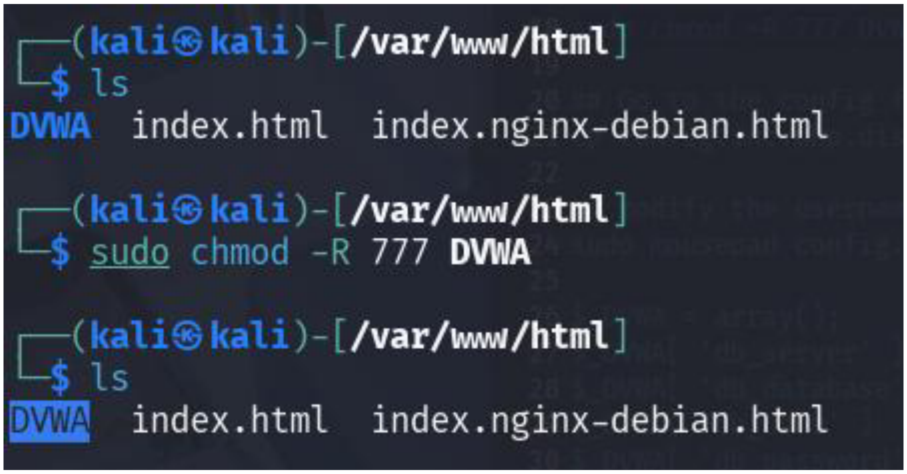
```
┌──(kali㉿kali)-[~]
└─$ sudo apt update
[sudo] password for kali:
Get:1 http://mirror.freedif.org/kali kali-rolling InRelease [41.5 kB]
Get:2 http://mirror.freedif.org/kali kali-rolling/main amd64 Packages [21.0 MB]
Get:3 http://mirror.freedif.org/kali kali-rolling/main amd64 Contents (deb) [51.6 MB]
Get:4 http://mirror.freedif.org/kali kali-rolling/contrib amd64 Packages [121 kB]
Get:5 http://mirror.freedif.org/kali kali-rolling/contrib amd64 Contents (deb) [328 kB]
Get:6 http://mirror.freedif.org/kali kali-rolling/non-free amd64 Packages [204 kB]
Get:7 http://mirror.freedif.org/kali kali-rolling/non-free amd64 Contents (deb) [914 kB]
Get:8 http://mirror.freedif.org/kali kali-rolling/non-free-firmware amd64 Packages [10.6 kB]
Fetched 74.2 MB in 1min 11s (1,050 kB/s)
1065 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Go to the Linux website default file road and download the DVWA GitHub resource.



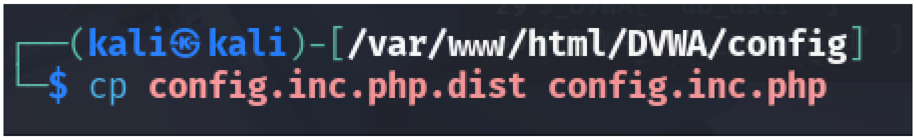
```
┌──(kali㉿kali)-[~]
└─$ cd /var/www/html

┌──(kali㉿kali)-[/var/www/html]
└─$ sudo git clone https://github.com/digininja/DVWA.git
```

Then check the file with the ls command. Then you can see the DVWA file. We should give this file the most privilege option (using the sudo chmod -R 777 DVWA command), then ls again. The DVWA file is now highlighted by the system.



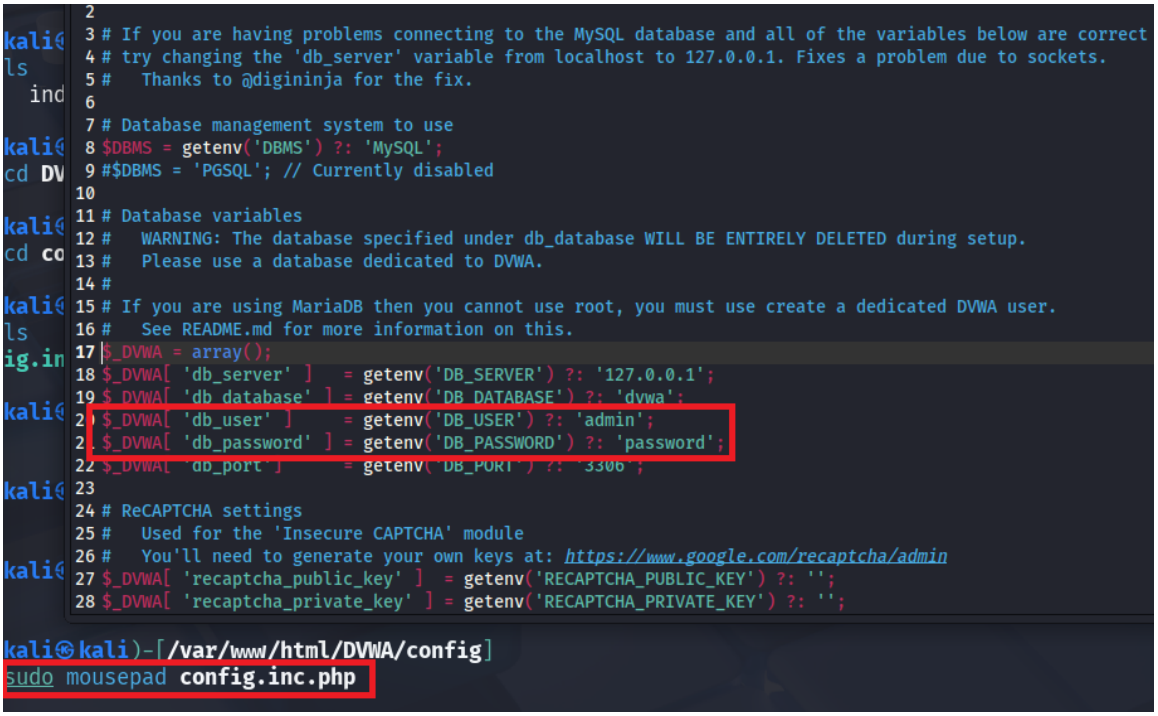
```
┌──(kali㉿kali)-[/var/www/html]
└─$ ls
DVWA  index.html  index.nginx-debian.html

┌──(kali㉿kali)-[/var/www/html]
└─$ sudo chmod -R 777 DVWA

┌──(kali㉿kali)-[/var/www/html]
└─$ ls
DVWA  index.html  index.nginx-debian.html
```

Now go into the DVWA file and go through the config file copy the config template



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

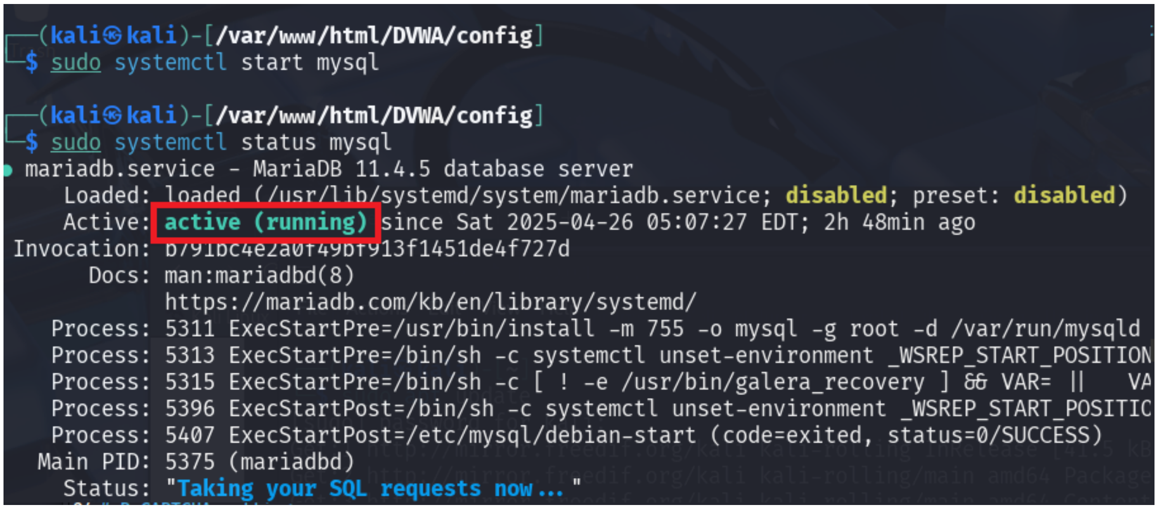
Use Mousepad to open the copied file, config.inc.php and modify the username and password to "admin" and "password."



```

# 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') ?: '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') ?: '';
```

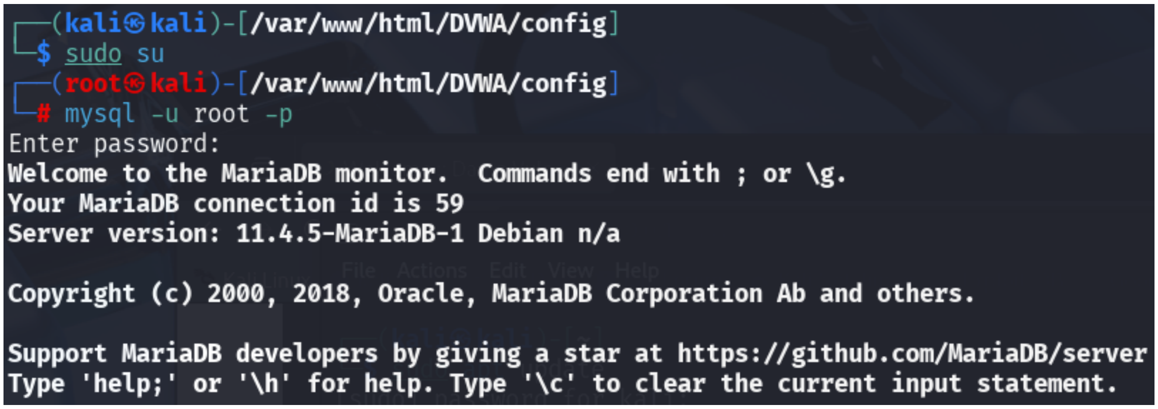
```
kali㉿kali)-[/var/www/html/DVWA/config]
sudo mousepad config.inc.php
```

Then start the MySQL and check the status



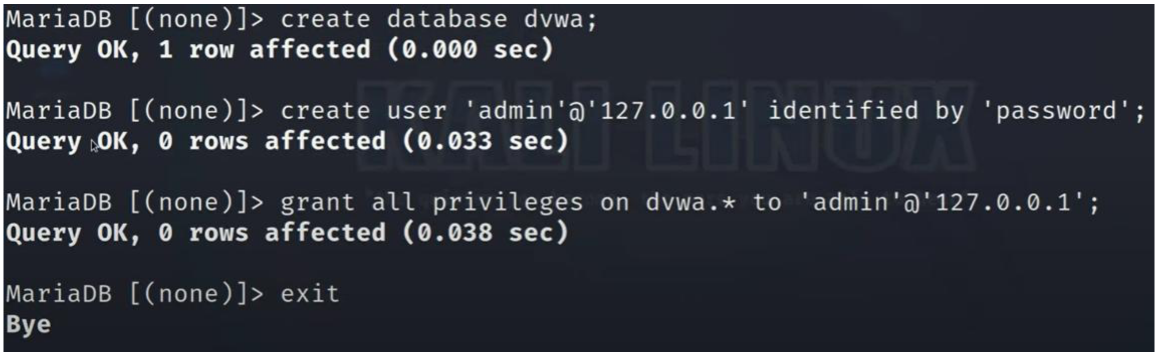
```
┌──(kali㉿kali)-[/var/www/html/DVWA/config]
└─$ sudo systemctl start mysql

┌──(kali㉿kali)-[/var/www/html/DVWA/config]
└─$ sudo systemctl status mysql
● mariadb.service - MariaDB 11.4.5 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; preset: disabled)
     Active: active (running) since Sat 2025-04-26 05:07:27 EDT; 2h 48min ago
 Invocation: b791bc4e2a0f49bf913f1451de4f727d
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 5311 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysqld
    Process: 5313 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION
    Process: 5315 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= ||    VA
    Process: 5396 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITIC
    Process: 5407 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
   Main PID: 5375 (mariadbd)
     Status: "Taking your SQL requests now..."
```

Now change the kali user to root (use sudo su command) and create a DVWA table. If not setting root password before just click enter to login in.



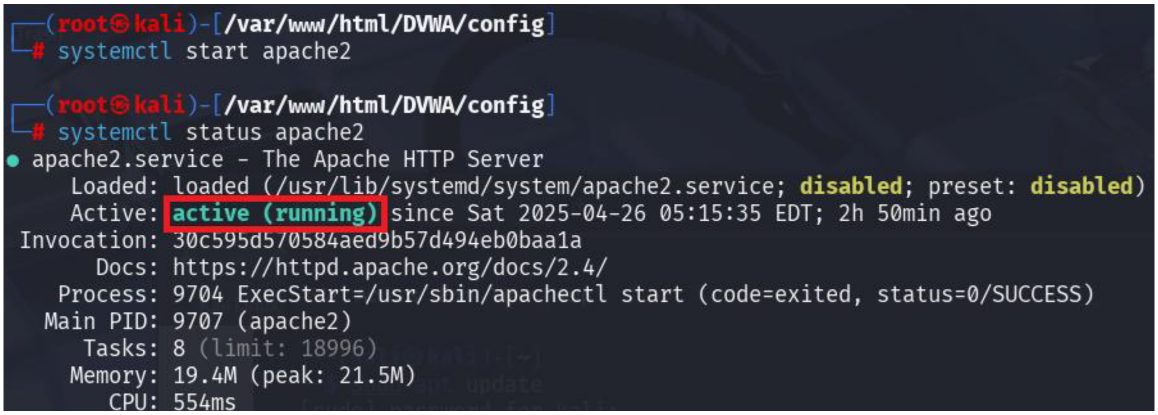
```
┌──(kali㉿kali)-[/var/www/html/DVWA/config]
└─$ sudo su
┌──(root㉿kali)-[/var/www/html/DVWA/config]
└─# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 59
Server version: 11.4.5-MariaDB-1 Debian n/a

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.
```

The command below: first, create a database (dvwa means create a table). Next, the command ‘admin’ and ‘password’ must match the previous settings. Then, grant the admin user all permissions, and then exit.



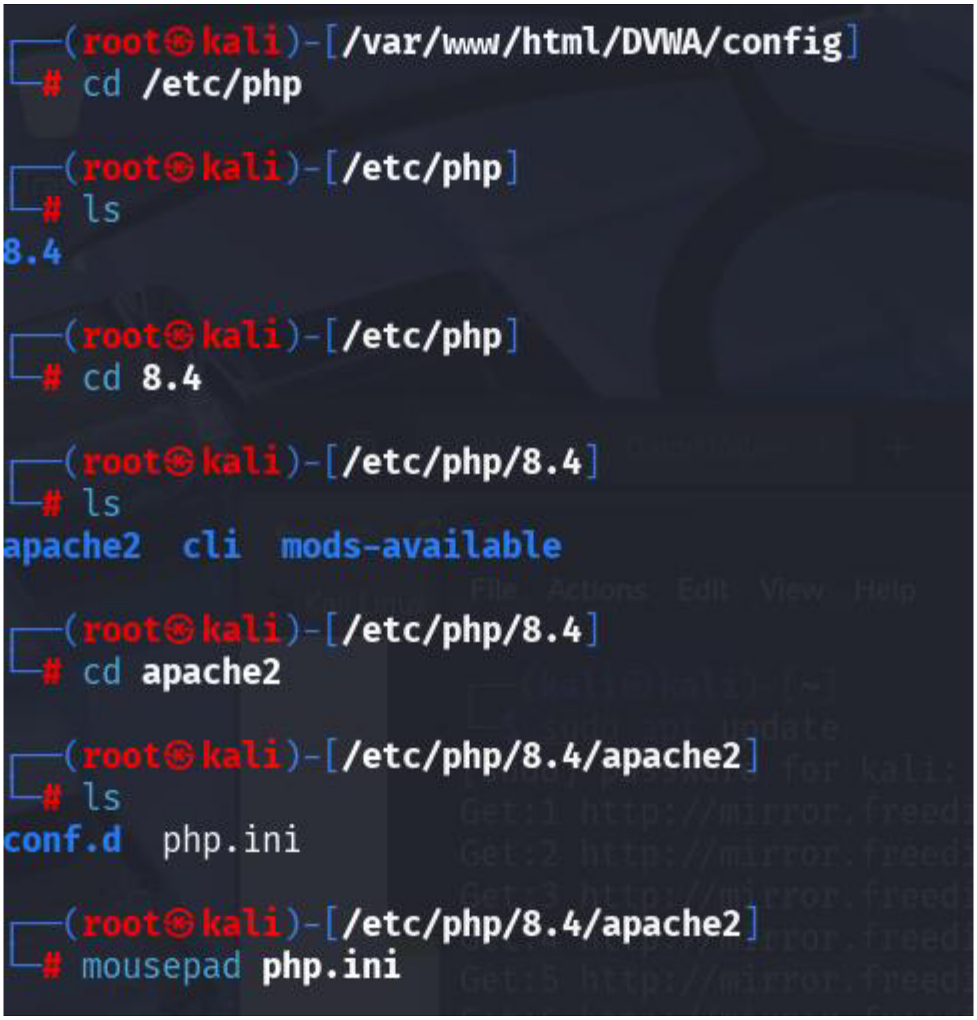
```
MariaDB [(none)]> create database dvwa;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> create user 'admin'@'127.0.0.1' identified by 'password';
Query OK, 0 rows affected (0.033 sec)

MariaDB [(none)]> grant all privileges on dvwa.* to 'admin'@'127.0.0.1';
Query OK, 0 rows affected (0.038 sec)

MariaDB [(none)]> exit
Bye
```

Then start the apache2 server and check the status.



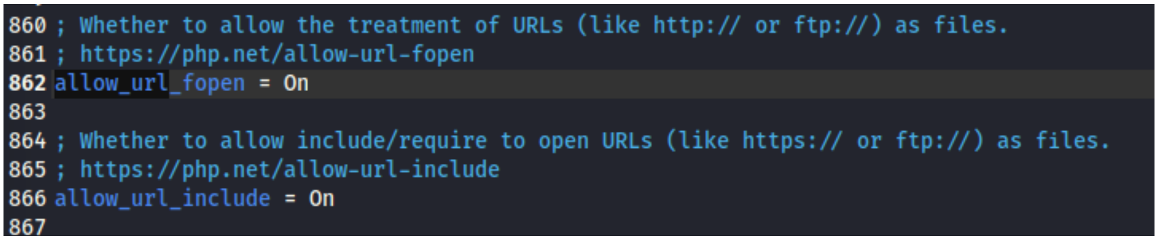
```
┌──(root㉿kali)-[/var/www/html/DVWA/config]
└─# systemctl start apache2

┌──(root㉿kali)-[/var/www/html/DVWA/config]
└─# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; preset: disabled)
     Active: active (running) since Sat 2025-04-26 05:15:35 EDT; 2h 50min ago
 Invocation: 30c595d570584aed9b57d494eb0baa1a
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 9704 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 9707 (apache2)
      Tasks: 8 (limit: 18996)
     Memory: 19.4M (peak: 21.5M)
        CPU: 554ms
```

Then go to the PHP file, check the version (mine is 8.4), and then go through the Apache2 file and use Mousepad to open the php.ini configuration.



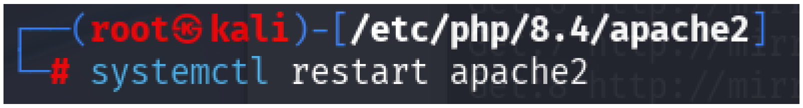
```
┌──(root㉿kali)-[/var/www/html/DVWA/config]
└─# cd /etc/php

┌──(root㉿kali)-[/etc/php]
└─# ls
8.4

┌──(root㉿kali)-[/etc/php]
└─# cd 8.4

┌──(root㉿kali)-[/etc/php/8.4]
└─# ls
apache2  cli  mods-available

┌──(root㉿kali)-[/etc/php/8.4]
└─# cd apache2

┌──(root㉿kali)-[/etc/php/8.4/apache2]
└─# ls
conf.d  php.ini

┌──(root㉿kali)-[/etc/php/8.4/apache2]
└─# mousepad php.ini
```

Make sure that the setting of allow_url_fopen and llow_url_include is on, if it shows off should modify it.



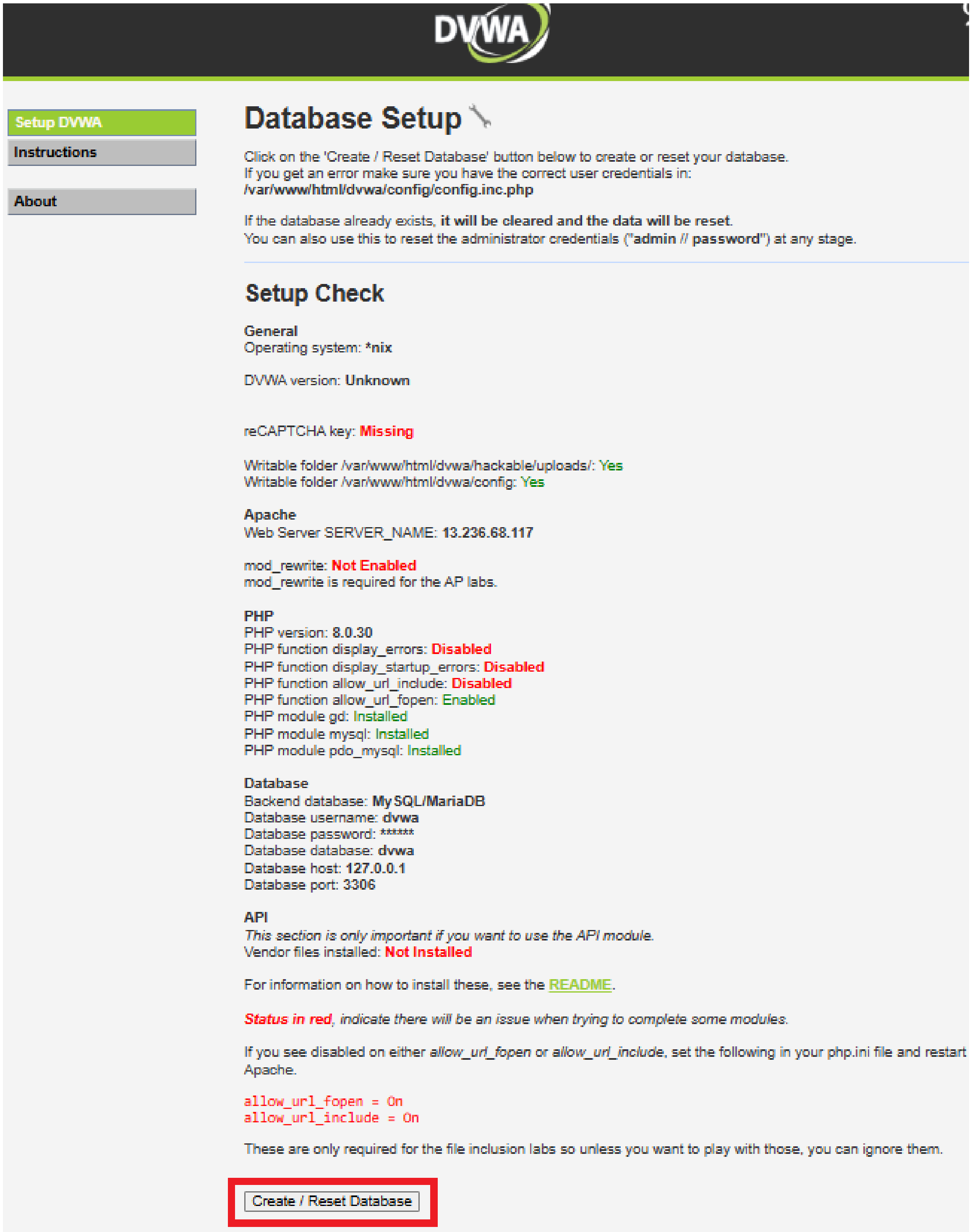
```
; Whether to allow the treatment of URLs (like http:// or ftp://) as files.
; https://php.net/allow-url-fopen
allow_url_fopen = On

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

```

Finally restart the apache2 server



```
┌──(root㉿kali)-[/etc/php/8.4/apache2]
└─# systemctl restart apache2
```

Then open the browser in Kali Linux and enter http://127.0.0.1/DVWA. The username is 'admin.' The password is 'password.' Once you log in, you will see the setup page; then click the Create/Reset Database button. After that, log in again, and you will see the Welcome to Damn Vulnerable Web Application!



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

General
Operating system: ***nix**

DVWA version: **Unknown**

reCAPTCHA key: **Missing**

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

Apache
Web Server SERVER_NAME: **13.236.68.117**

mod_rewrite: **Not Enabled**
mod_rewrite is required for the AP labs.

PHP
PHP version: **8.0.30**
PHP function display_errors: **Disabled**
PHP function display_startup_errors: **Disabled**
PHP function allow_url_include: **Disabled**
PHP function allow_url_fopen: Enabled
PHP module gd: Installed
PHP module mysql: Installed
PHP module pdo_mysql: Installed

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

API
This section is only important if you want to use the API module.
Vendor files installed: **Not Installed**

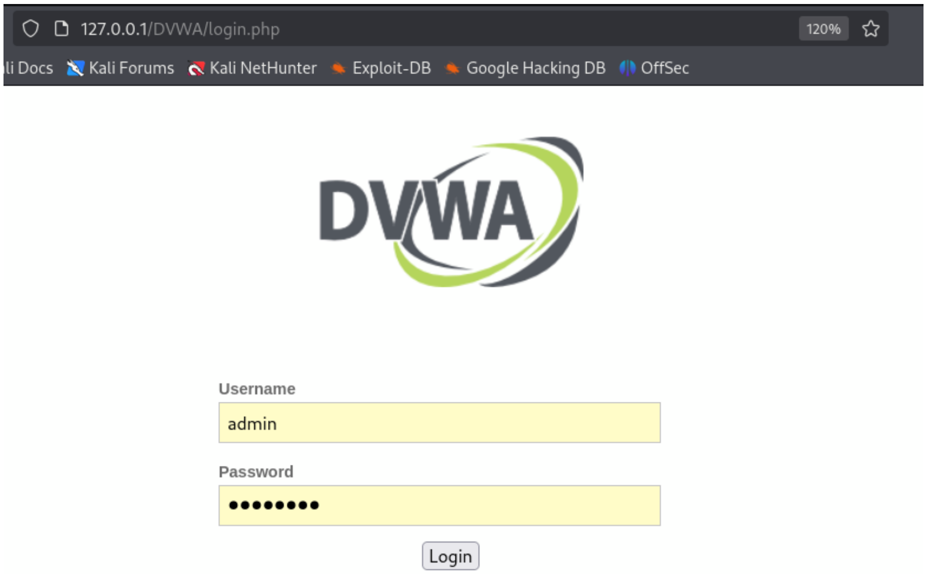
For information on how to install these, see the README.

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.

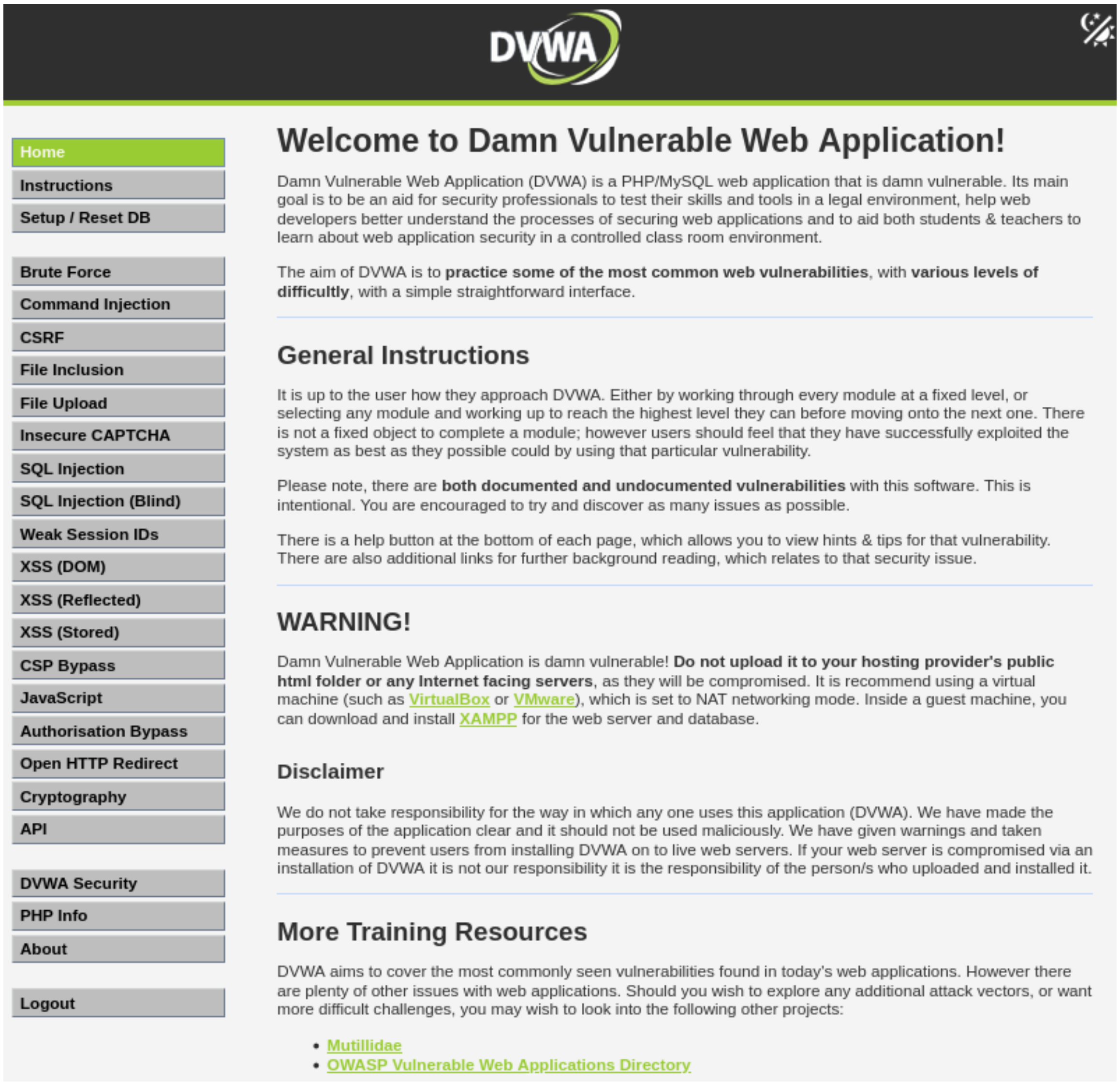


Username

admin

Password

••••••••



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.

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:

- Mutillidae
- OWASP Vulnerable Web Applications Directory