# Practical No. 5

## #Setting up DVWA on Kali Linux

### Step 1: Download Damn Vulnerable Web Application (DVWA)

To get started, we will need to clone the DVWA GitHub into our /var/www/html directory. That is the location where Localhost files are stored in Linux systems. Launch the Terminal and change our directory to the /var/www/html directory with the command below. After cloning, we can rename the DVWA folder to dvwa. That is not mandatory, but it makes work easier when executing multiple commands.

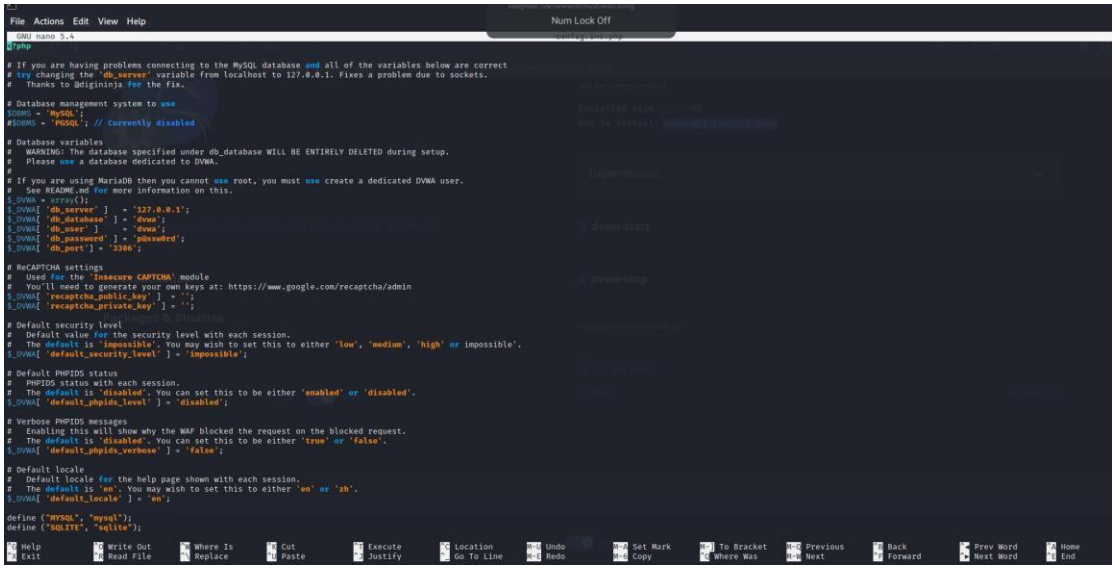

### Step 2: Configure DVWA

After downloading cloning DVWA in our /var/www/html directory, we still need to do some minor configurations. To get started, let's set read, write, and execute permissions to the DVWA directory. Execute the command below.

After successfully executing the command, we need to set up the user and password required to access the database. Change directory to point to the config directory with the command below. When you run the ls command to view the files inside the directory, you will see the config.inc.php.dist file. That is the original file containing the default configurations. We won't edit it.

Instead, we will create a copy of this file called config.inc.php and the original config.inc.php.dist file will act as our backup in case things go wrong. Run the command below to open the newly created file with nano editor and make the necessary changes, as shown in the image below.


We will set db_user as user and db_password as pass. Feel free to use a different username or password. Save the file (Ctrl + O, then Enter) and Exit (Ctrl + X). That's it! We are done configuring the DVWA Web application. Let's move on and configure the database (MySQL).




**Step 3: Install MySQL on Kali Linux**

By default, MySQL comes pre-installed on Kali Linux. If that's not the case for you or maybe you messed up with MySQL, we can go ahead and install it manually. If you have worked with Debian-based distributions, MySQL comes in two packages:

- ★ mysql-server
- ★ mysql-client

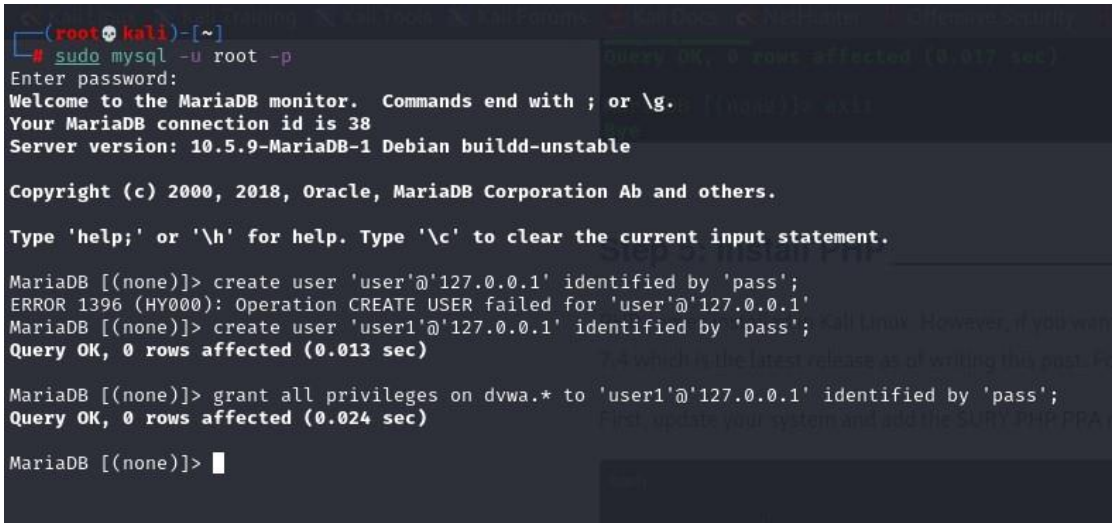
In our case, we will need to install the mysql-server. However, there is a catch. If you try using the command apt install mysql-server you will most likely get the error "Package mysql-server is not available, but is referred to by another package. E: Package 'mysql-server' has no installation candidate." That's because the package mysql-server is referred to default-mysqlserver in Kali Linux and also in the latest release of Debian (Debian 10). Therefore, use the command below:


**Step 4: Configure MySQL Database**

Start the Mysql service with the command below. You can check whether the service is running using the systemctl status command below. Login to the MySQL database using the command below as root. If you have another name set for the superuser in your system, use it instead of root. Login to the MySQL database using the command below as root. If you have another name set for the superuser in your system, use it instead of root. We will create a new user with the username and password set in our DVWA application configuration file. In my case, the username was ‘user,' and the password was ‘pass.’ The server we are using is Localhost (127.0.0.1). Use the command below.

We need to grant this new user privilege over the dvwa database. Execute the command below.

Up to this point, we are through with configuring both the DVWA application and the MySQL database. Type exit to close the database.

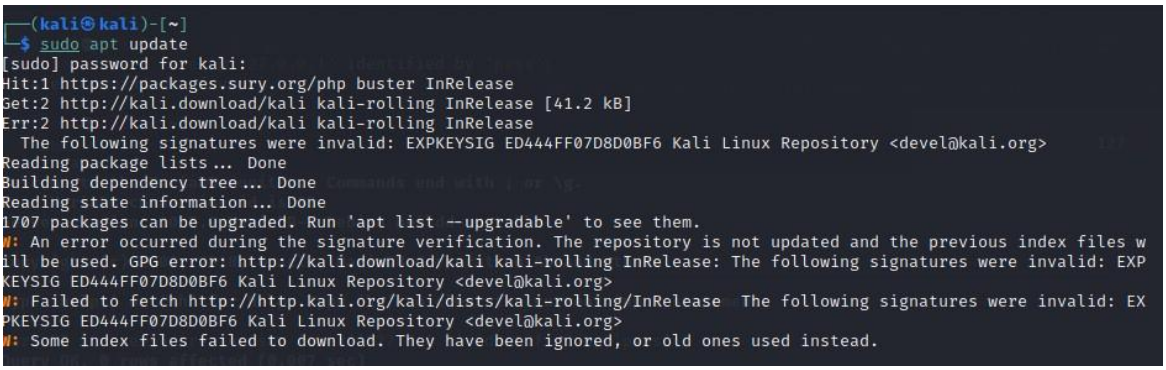


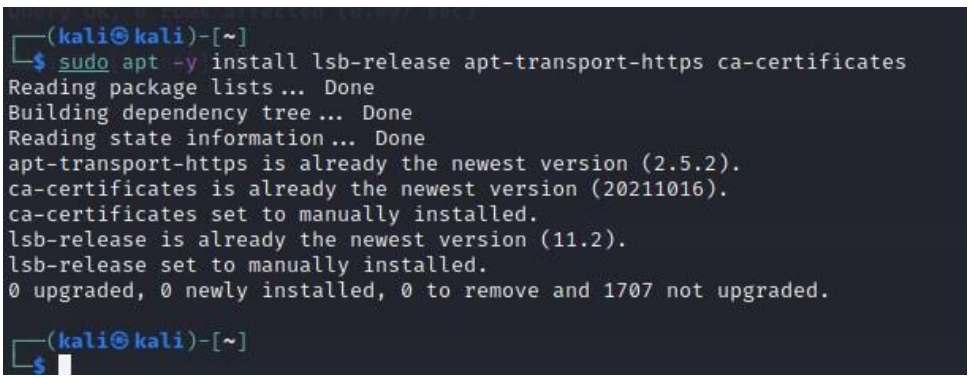

**Step 5: Install PHP**

PHP comes installed in Kali Linux. However, if you want to install a particular version, you can do it manually from the Terminal. In this post, we will install PHP 7.4 which is the latest release as of writing this post.

Follow the steps below. First, update your system and add the SURY PHP PPA repository by executing the commands below. After successfully adding the repository, use the command below to install PHP 7.4.

To install additional PHP extensions, use the syntax below where xxx stands for the extension name.



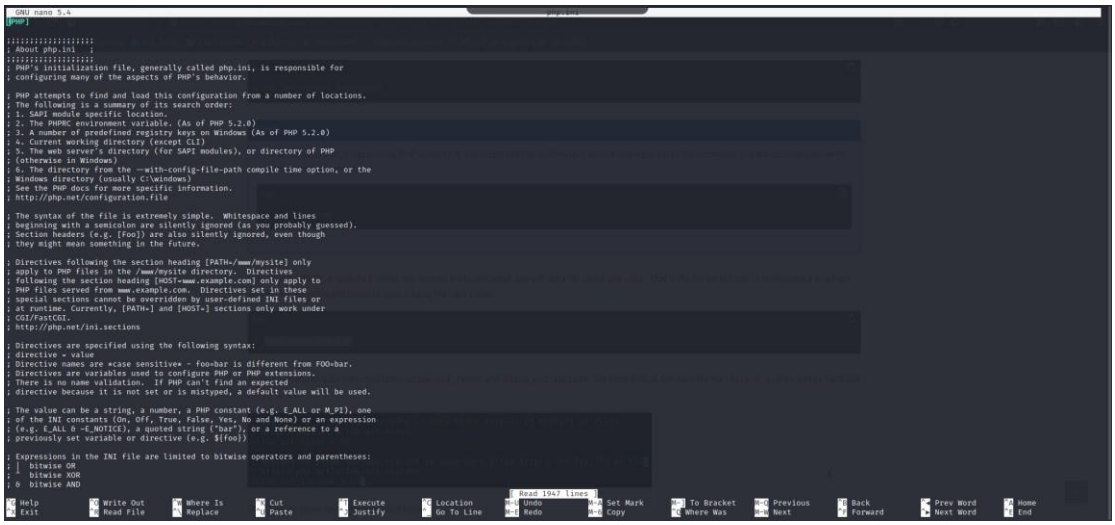
```
┌──(kali㉿kali)-[~]
└─$ sudo wget -O /etc/apt/trusted.gpg.d/php.gpg https://packages.sury.org/php/apt.gpg
--2023-09-09 02:25:35--  https://packages.sury.org/php/apt.gpg
Resolving packages.sury.org (packages.sury.org)... 103.180.115.10
Connecting to packages.sury.org (packages.sury.org)|103.180.115.10|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1769 (1.7K) [application/octet-stream]
Saving to: '/etc/apt/trusted.gpg.d/php.gpg'

/etc/apt/trusted.gpg.d/php.gp 100%[===================>]   1.73K  --.-KB/s    in 0s

2023-09-09 02:25:35 (94.0 MB/s) - '/etc/apt/trusted.gpg.d/php.gpg' saved [1769/1769]
```

```
┌──(kali㉿kali)-[~]
└─$ echo "deb https://packages.sury.org/php/ buster main" | sudo tee /etc/apt/sources.list.d/php.list~
deb https://packages.sury.org/php/ buster main

┌──(kali㉿kali)-[~]
└─$ sudo apt update
Hit:1 https://packages.sury.org/php buster InRelease
Get:2 http://kali.download/kali kali-rolling InRelease [41.2 kB]
Err:2 http://kali.download/kali kali-rolling InRelease
  The following signatures were invalid: EXPKEYSIG ED444FF07D8D0BF6 Kali Linux Repository <devel@kali.org>
Fetched 41.2 kB in 1s (42.5 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1707 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: An error occurred during the signature verification. The repository is not updated and the previous index files will be u
G ED444FF07D8D0BF6 Kali Linux Repository <devel@kali.org>
W: Failed to fetch http://http.kali.org/kali/dists/kali-rolling/InRelease  The following signatures were invalid: EXPKEYSIG
W: Some index files failed to download. They have been ignored, or old ones used instead.

┌──(kali㉿kali)-[~]
└─$ sudo apt install php7.4 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be upgraded:
  php7.4
1 upgraded, 0 newly installed, 0 to remove and 1706 not upgraded.
Need to get 51.7 kB of archives.
After this operation, 2,048 B of additional disk space will be used.
Get:1 https://packages.sury.org/php buster/main amd64 php7.4 all 1:7.4.33-8+0~20230904.88+debian10~1.gbp87c414 [51.7 kB]
Fetched 51.7 kB in 0s (110 kB/s)
(Reading database ... 271627 files and directories currently installed.)
Preparing to unpack .../php7.4_1%3a7.4.33-8+0~20230904.88+debian10~1.gbp87c414_all.deb ...
Unpacking php7.4 (1:7.4.33-8+0~20230904.88+debian10~1.gbp87c414) over (1:7.4.30-6+0~20220918.70+debian10~1.gbpcf2025) ...
Setting up php7.4 (1:7.4.33-8+0~20230904.88+debian10~1.gbp87c414) ...

┌──(kali㉿kali)-[~]
└─$
```

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install php7.4-{cli,json,imap,bcmath,bz2,intl,gd,mbstring,mysql,zip}
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.
The following information may help to resolve the situation:

The following packages have unmet dependencies:
 php7.4-common : Depends: libffi6 (≥ 3.0.10~rc8) but it is not installable
E: Unable to correct problems, you have held broken packages.
```

**Step 6: Configure Apache Server**

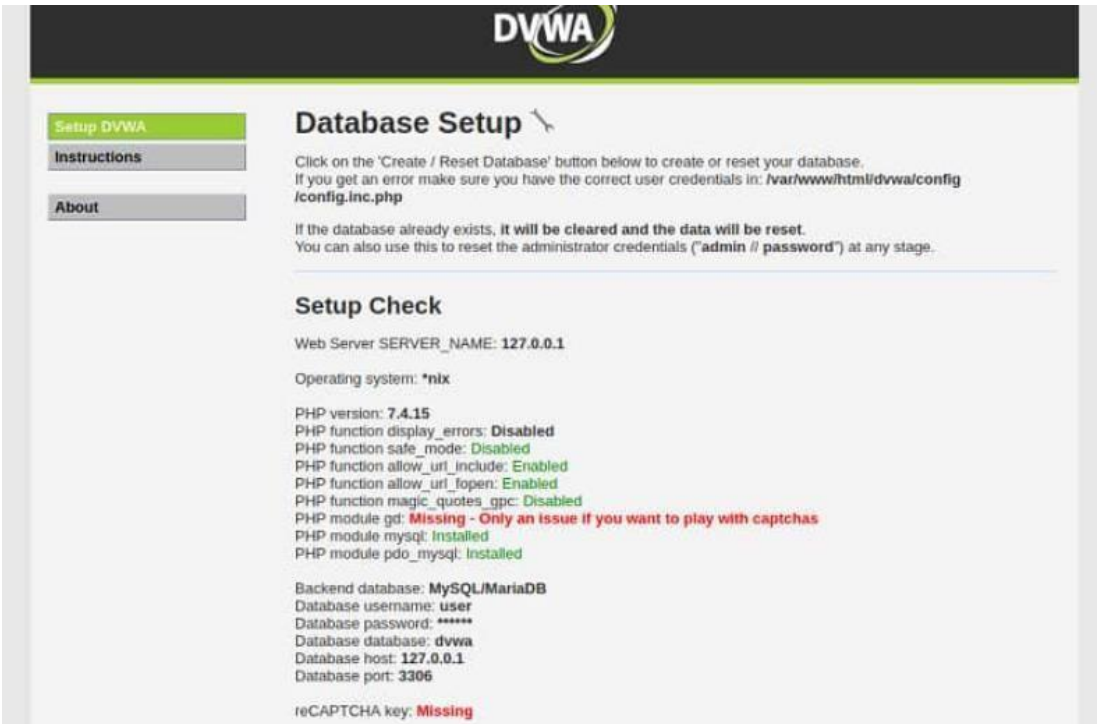
Now, we need to configure the server. Use the command below to change your location on the Terminal to point to /etc/php/7.3/apache2 directory. In the /etc/php/7.4/apache2, when you execute the ls command, you will see a file called php.ini. That is the file we will edit to configure our localhost server. Use the command below to open it using the nano editor. Scroll down and look for these two lines: allow_url_fopen and allow_url_include. Set them both as On. Save the file (Ctrl + O, then Enter) and Exit (Ctrl + X). Start Apache server using the command below. To check whether the service started successfully, use the status command.

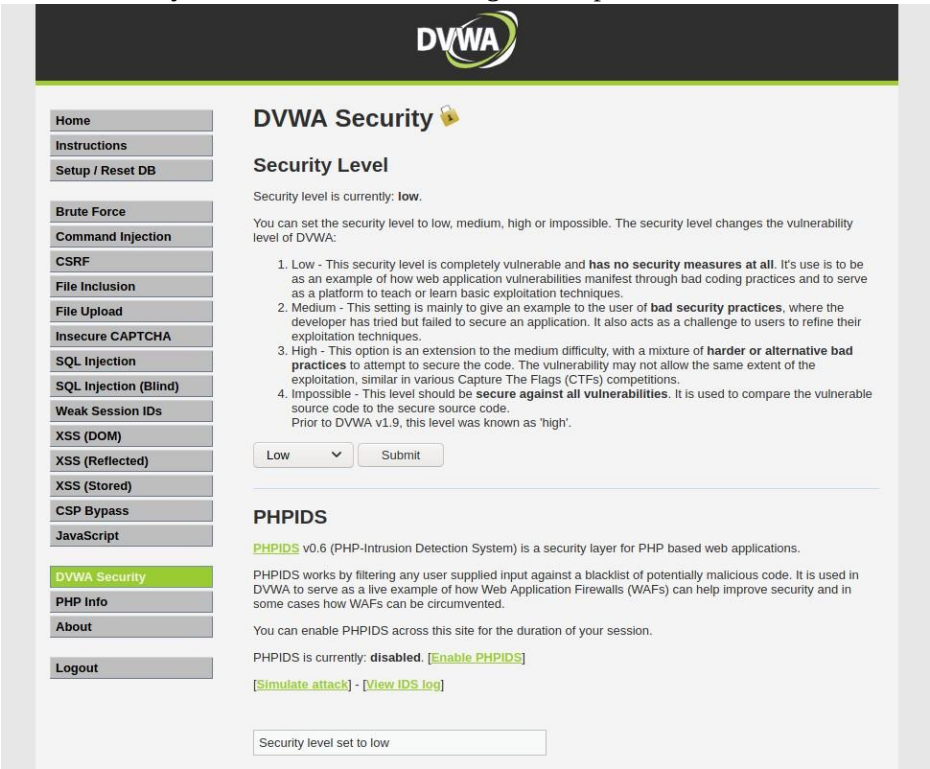

**Step 7: Access DVWA on Your Browser**

That's it! We now have everything configured, and we can proceed to launch DVWA. Open your browser and enter the URL: That will open the setup.php web page as shown in the image: You might see the errors colored in red as in the image above. Don't panic! Scroll down and click the Create / Reset Database button. That will create and configure the database. After some time, you will be redirected to the DVWA login page. Log in with these credentials:

★ Username - admin
★ Password - password



Once logged in, you will see the DVWA main page. On the left panel, we have the different types of attacks you can exploit and the DVWA Security button that allows you to choose the desired security level - Low, Medium, High, or Impossible.



# #Implementing SQL Injection on DVWA

### Step 1: Setup DVWA for SQL Injection

After successfully installing DVWA, open your browser and enter the required URL 127.0.0.1/dvwa/login.php Log in using the username “admin” and password as “password”. These are the default DVWA login credentials. After a successful login, set the DVWA security to LOW then click on SQL Injection on the left-side menu.

### Step 2: Basic Injection

On the User ID field, enter “1” and click Submit. That is supposed to print the ID, First_name, and Surname on the screen as you can see below. The SQL syntax being exploited here is:


Interestingly, when you check the URL, you will see there is an injectable parameter which is the ID. Currently, my URL looks like this. Let’s change the ID parameter of the URL to a number like 1,2,3,4 etc. That will also return the First_name and Surname of all users as follows: If you were executing this command directly on the DVWA database, the query for User ID 3 would look like this:

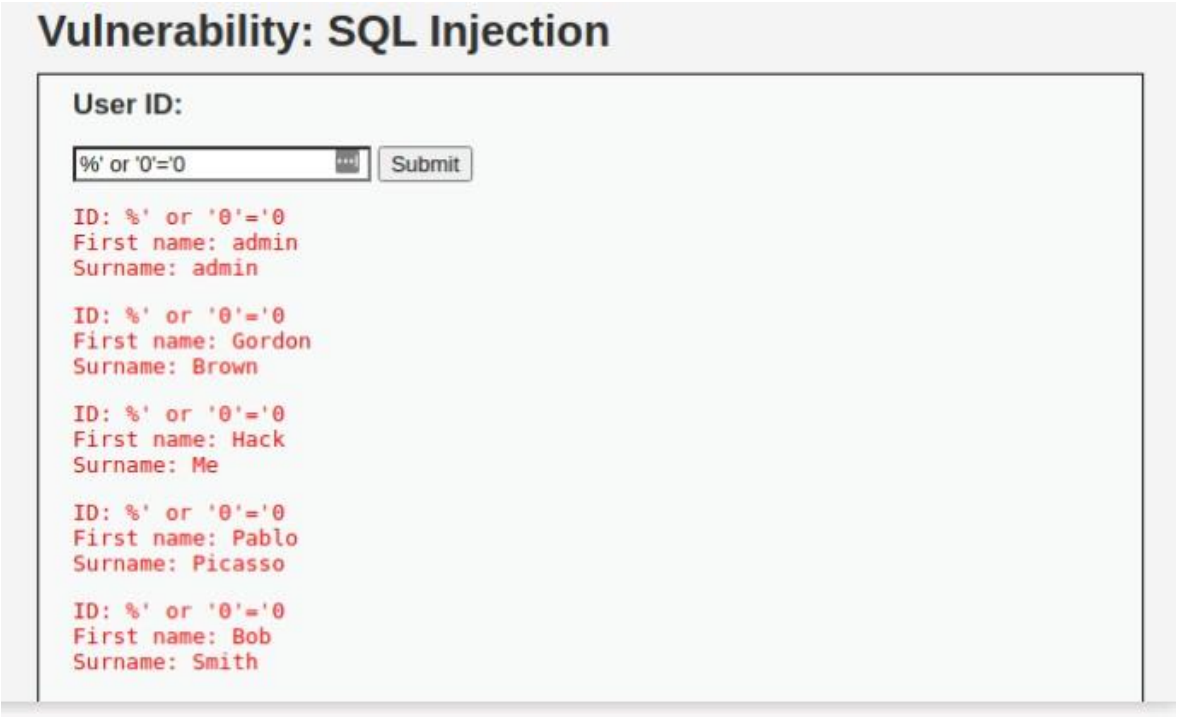
```
MariaDB [(none)]> \r
Connection id:    98
Current database: *** NONE ***

MariaDB [(none)]> \r dvwa
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Connection id:    99
Current database: dvwa

MariaDB [dvwa]>
```

```
Connection id:    99
Current database: dvwa

MariaDB [dvwa]> SELECT first_name, last_name FROM users WHERE user_id = '3';
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Hack       | Me        |
+------------+-----------+
1 row in set (0.000 sec)

MariaDB [dvwa]>
```

```
MariaDB [dvwa]> SELECT first_name, last_name FROM users WHERE user_id = '%' or '1'='1';
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| admin      | admin     |
| Gordon     | Brown     |
| Hack       | Me        |
| Pablo      | Picasso   |
| Bob        | Smith     |
+------------+-----------+
5 rows in set (0.002 sec)

MariaDB [dvwa]>
```

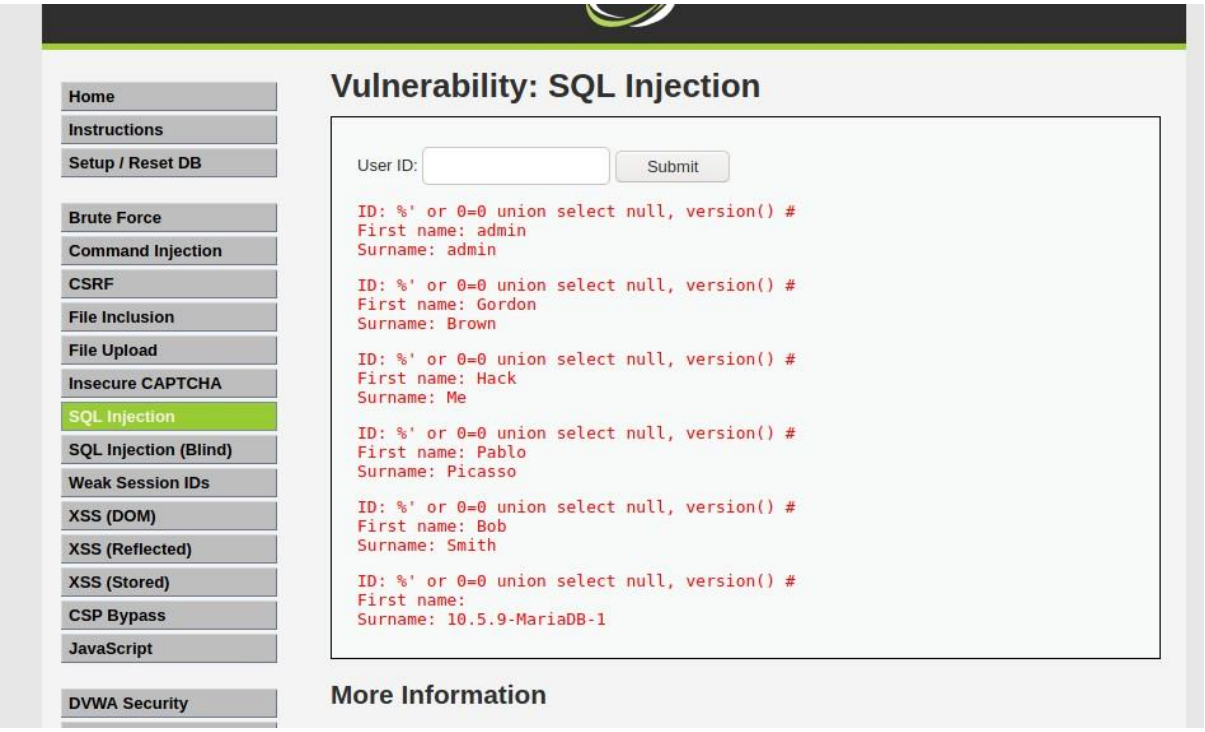
**Step 3: Always True Scenario**

An advanced method to extract all the First_names and Surnames from the database would be to use the input: %' or '1'='1'



**Step 4: Display Database Version**

To know the database version the DVWA application is running on, enter the text below in the User ID field.

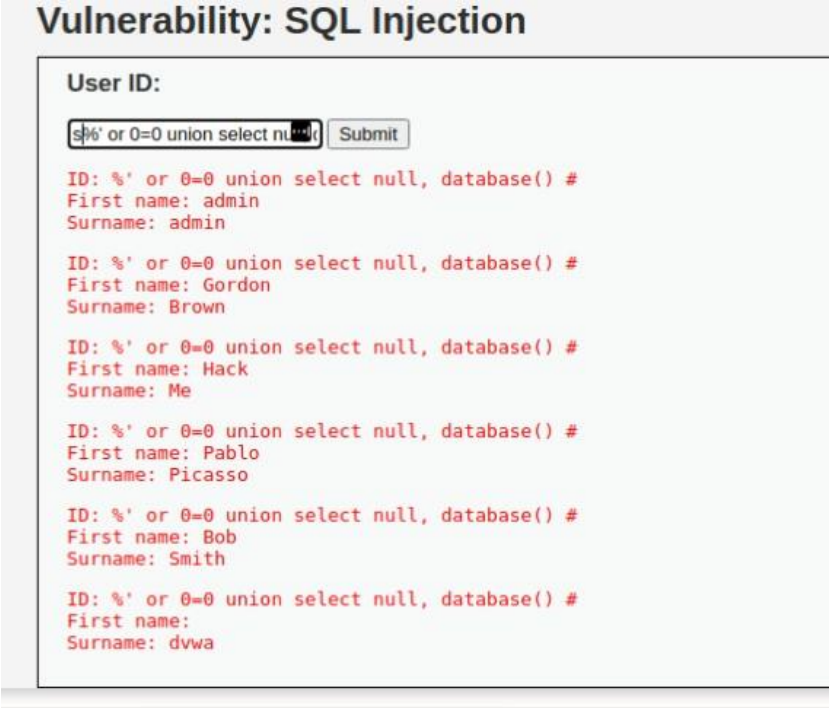



**Step 5: Display Database User**

To display the Database user who executed the PHP code powering the database, enter the text below in the USER ID field.

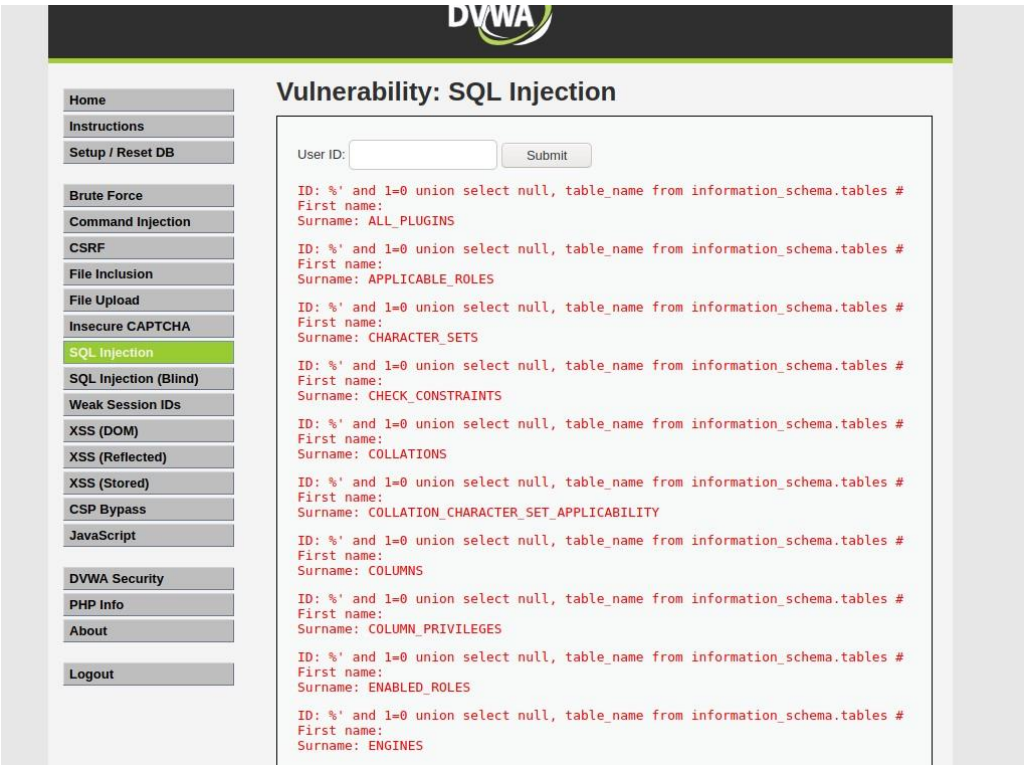
**Step 6: Display Database Name**

To display the database name, we will inject the SQL code below in the User ID field.




**Step 7: Display all tables in information_schema**

The Information Schema stores information about tables, columns, and all the other databases maintained by MySQL. To display all the tables present in the information_schema, use the text below.



**Step 8: Display all the user tables in information_schema**

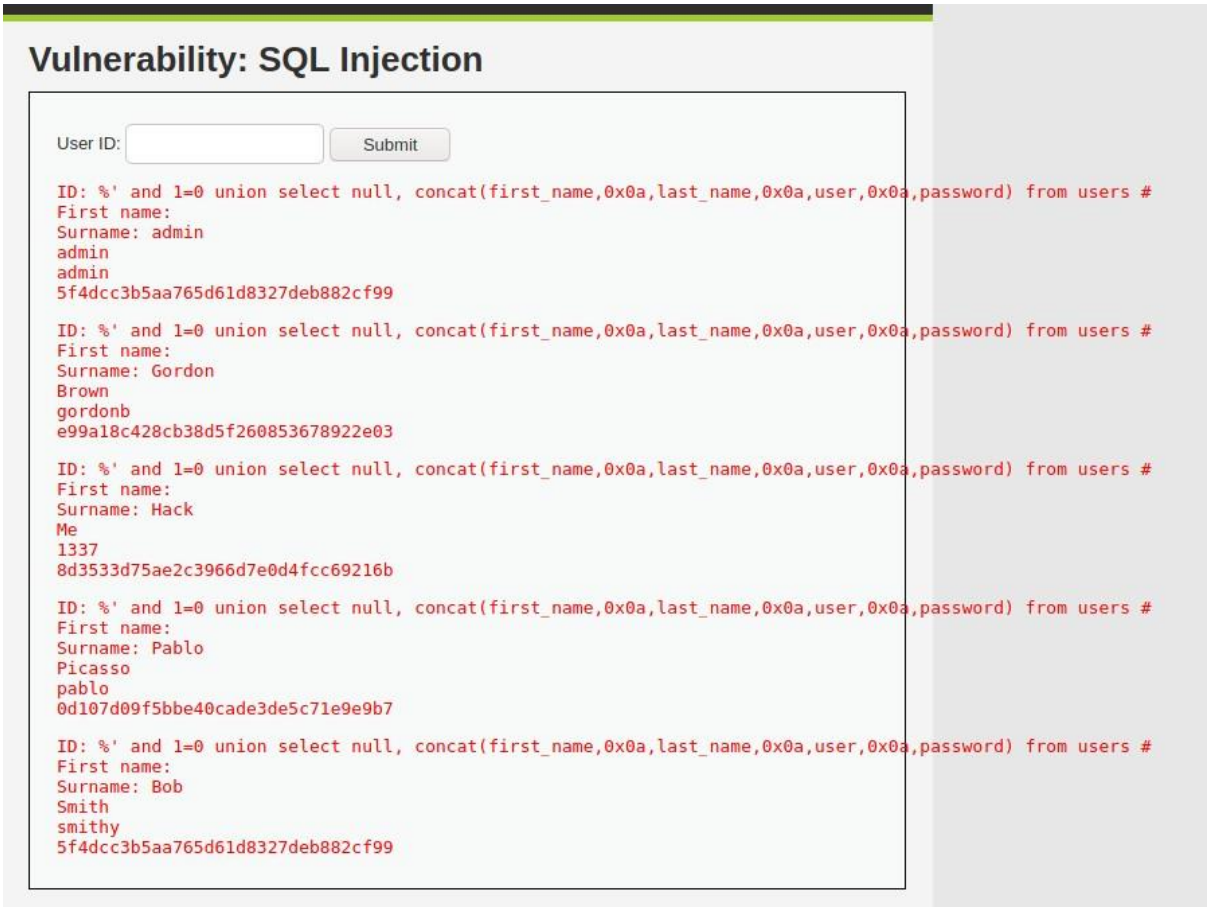
For this step, we will print all the tables that start with the prefix user as stored in the information_schema. Enter the SQL code below in the User ID.


**Step 9: Display all the columns fields in the information_schema user table**

We will print all the columns present in the users' table. This information will include column information like User_ID, first_name, last_name, user, and password. Enter the input in the User_ID field.

**Step 10: Display Column field contents**

To display all the necessary authentication information present in the columns as stored in the information_schema, use the SQL syntax below:




From the image above, you can see the password was returned in its hashed format. To extract the password, copy the MD5 hash and use applications like John the Ripper to crack it. There are also sites available on the internet where you can paste the hash and if lucky, you will be able to extract the password.