Configure Web Application Penetration Testing Lab

Table of Content

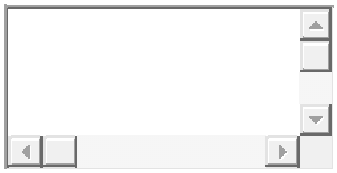
Requirement-Ubuntu 18.0

Web Application

A web application is a computer program that utilizes web browsers and web technology to perform tasks over the Internet. Web apps can be built for a wider uses which can be used by anyone; from an enterprise to an entity for a variety of reasons. Frequently used Web applications can include webmail.

DVWA

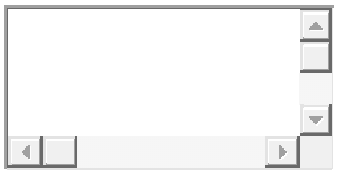
Let's start You should download and configure this web application only within the html directory for all web applications in the browser through localhost. Go to your Ubuntu terminal and move inside html directory by running the following command and then download dvwa lab from the given link.



1 cd /var/www/html
2 git clone https://github.com/ethicalhack3r/DVWA

```
root@ubuntu:/var/www/html# git clone https://github.com/ethicalhack3r/DVWA
Cloning into 'DVWA'...
remote: Enumerating objects: 2986, done.
remote: Total 2986 (delta 0), reused 0 (delta 0), pack-reused 2986
Receiving objects: 100% (2986/2986), 1.51 MiB | 823.00 KiB/s, done.
Resolving deltas: 100% (1322/1322), done.
root@ubuntu:/var/www/html# cd DVWA/
root@ubuntu:/var/www/html/DVWA# ls
about.php      config       docs  external     hackable     index.php         login.php
CHANGELOG.md   COPYING.txt  dvwa  favicon.ico  ids_log.php  instructions.php  logout.php
root@ubuntu:/var/www/html/DVWA# cd config
root@ubuntu:/var/www/html/DVWA/config# ls
config.inc.php.dist
root@ubuntu:/var/www/html/DVWA/config# mv config.inc.php.dist config.inc.php
root@ubuntu:/var/www/html/DVWA/config#
```

After the installation we will go inside the dvwa and there we will find a config folder, now we will move inside the config folder and there we will run the ls command to view all available folder, now, here you will see a config.inc.php.dist file. Now as you can see, we have moved **config.inc.php.dist file to config.inc.php**



```
1 cd /dvwa/config
2 mv config.inc.php.dist config.inc.php
```

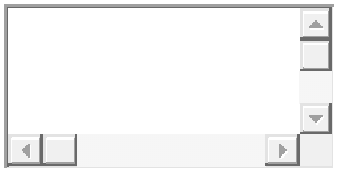
Now open the config file using nano; where you will find that db user is root and db password is password.

```
<?php

# If you are having problems connecting to the MySQL database and all of the variables below
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a problem due to
#   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 us
#   See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ]   = '127.0.0.1';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ]     = 'root';
$_DVWA[ 'db_password' ] = 'p@ssw0rd';

# Only used with PostgreSQL/PGSQL database selection.
$_DVWA[ 'db_port '] = '5432';

# 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' ] = '';

# Default security level
```

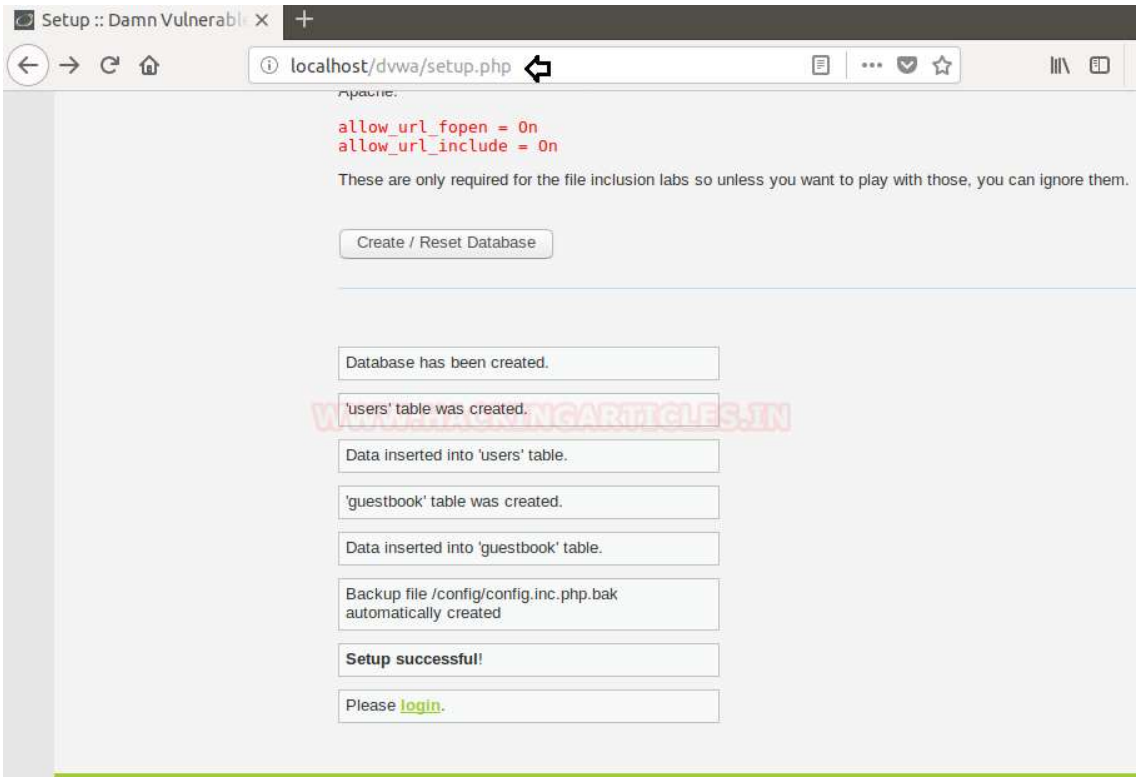
Here you need to make the changes and give access to the Ubuntu user as in our case we have written **raj** as db user and as our ubuntu password is **123** so we have written 123 as db password.

```
# 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 us
#   See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ]   = '127.0.0.1';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ]     = 'raj';
$_DVWA[ 'db_password' ] = '123';

# Only used with PostgreSQL/PGSQL database selection.
$_DVWA[ 'db_port '] = '5432';

# 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' ] = '';

# Default security level
```

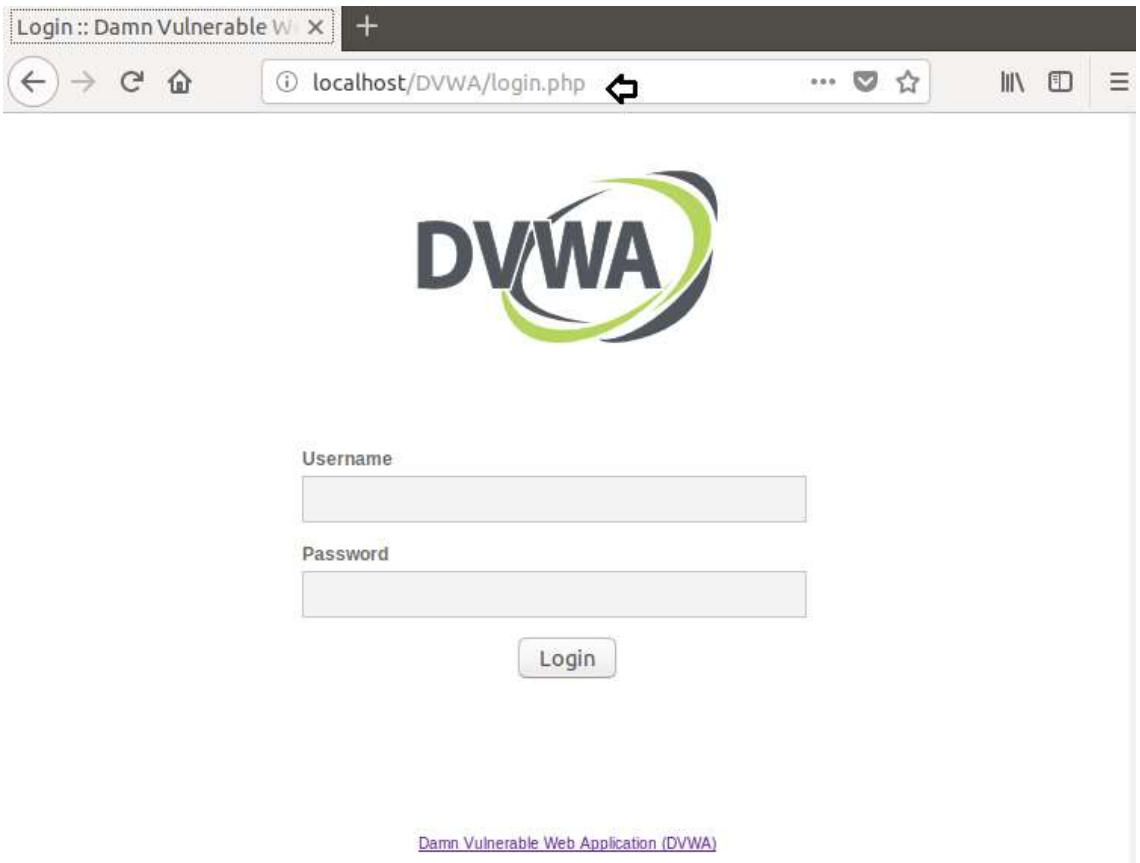
Now we will try to open dvwa lab in the browser by the following URL and click on **Create/Reset Database**



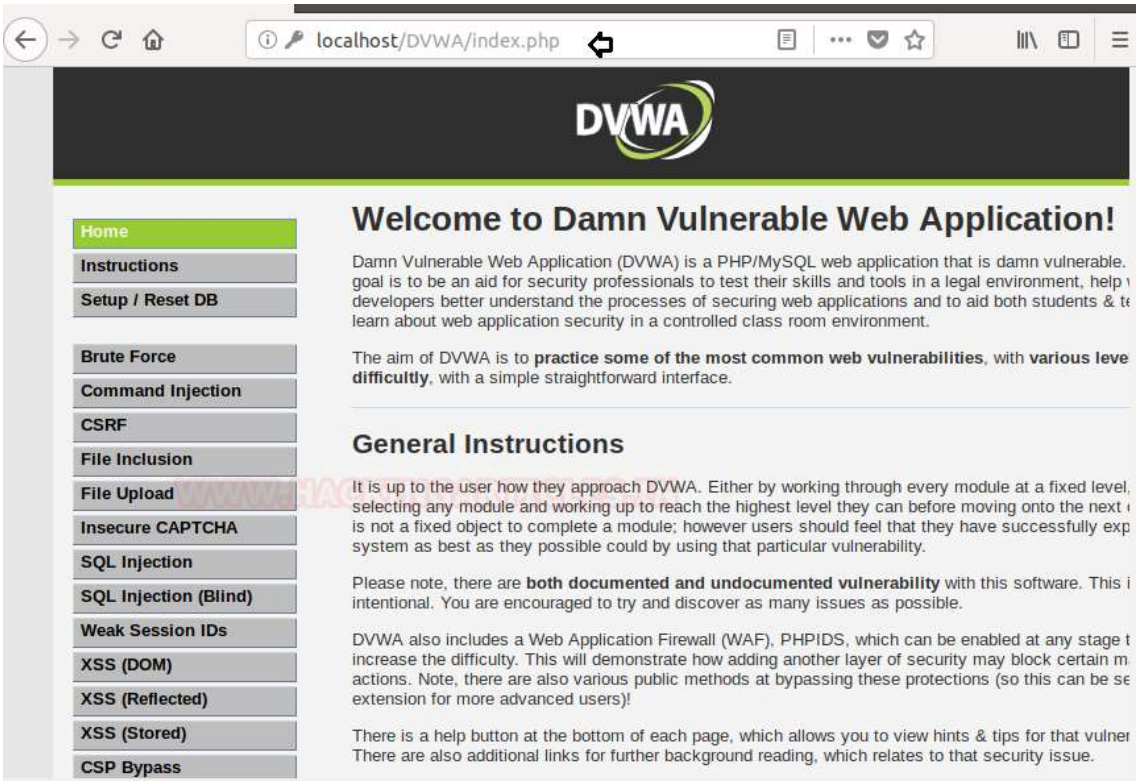
1 http://localhost/dvwa/setup.php



Good! We have successfully configured the dvwa lab in ubuntu 18 as we can see that we are welcomed by the login page.



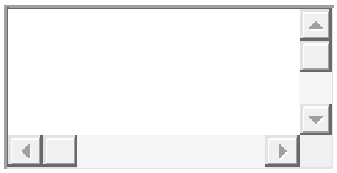
For login, we will use the dvwa username which is **admin** and **password** which is dvwa password by default.



bWAPP

A buggy web application that is purposely unsafe. Enthusiasts of security, system engineers, developers can find out about Web vulnerabilities and prevent them.

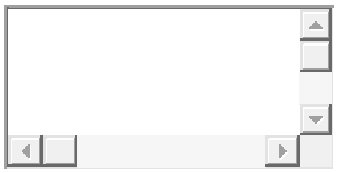
bWAPP prepares you for successful tests and penetration testing. Now we will configure bWAPP lab in Ubuntu 18. First, we will **download bWAPP** and then we will move inside the Downloads folder and then unzip the bWAPP file by the following command-



```
1 unzip bWAPP_latest.zip
```

```
root@ubuntu:~# cd Downloads/
root@ubuntu:~/Downloads# ls
bWAPP_latest.zip
root@ubuntu:~/Downloads# unzip bWAPP_latest.zip
Archive:  bWAPP_latest.zip
  inflating: apache2/default
  inflating: apache2/httpd.conf
  inflating: bWAPP/666
   creating: bWAPP/admin/
  inflating: bWAPP/admin/index.php
  inflating: bWAPP/admin/phpinfo.php
  inflating: bWAPP/admin/settings.php
  inflating: bWAPP/aim.php
   creating: bWAPP/apps/
  inflating: bWAPP/apps/movie_search
  inflating: bWAPP/ba_captcha_bypass.php
  inflating: bWAPP/ba_forgotten.php
  inflating: bWAPP/ba_insecure_login.php
  inflating: bWAPP/ba_insecure_login_1.php
  inflating: bWAPP/ba_insecure_login_2.php
```

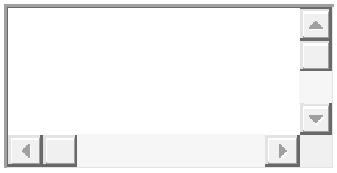
Now we will move bWAPP into var/www/html by the following command-



```
1 mv bWAPP /var/www/html
```

```
root@ubuntu:~/Downloads# ls
apache2  bWAPP  bWAPP_intro.pdf  bWAPP_latest.zip  ClientAccessPolicy.xml
root@ubuntu:~/Downloads# mv bWAPP /var/www/html
root@ubuntu:~/Downloads#
```

Now we will edit the config file; so, move inside the config file by the following command and where you can see that db username is root and db password is bug b default.



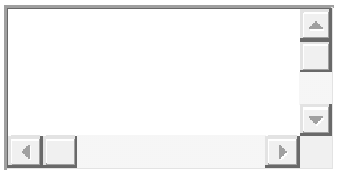
```
1 cd admin
2 ls
3 nano setting.php
```

```
*/

// Database connection settings
$db_server = "localhost";
$db_username = "root";
$db_password = "bug";
$db_name = "bWAPP";

// SQLite database name
$db_sqlite = "db/bwapp.sqlite";

// SMTP settings
$smtp_sender = "bwapp@mailinator.com";
$smtp_recipient = "bwapp@mailinator.com";
$smtp_server = "";
```

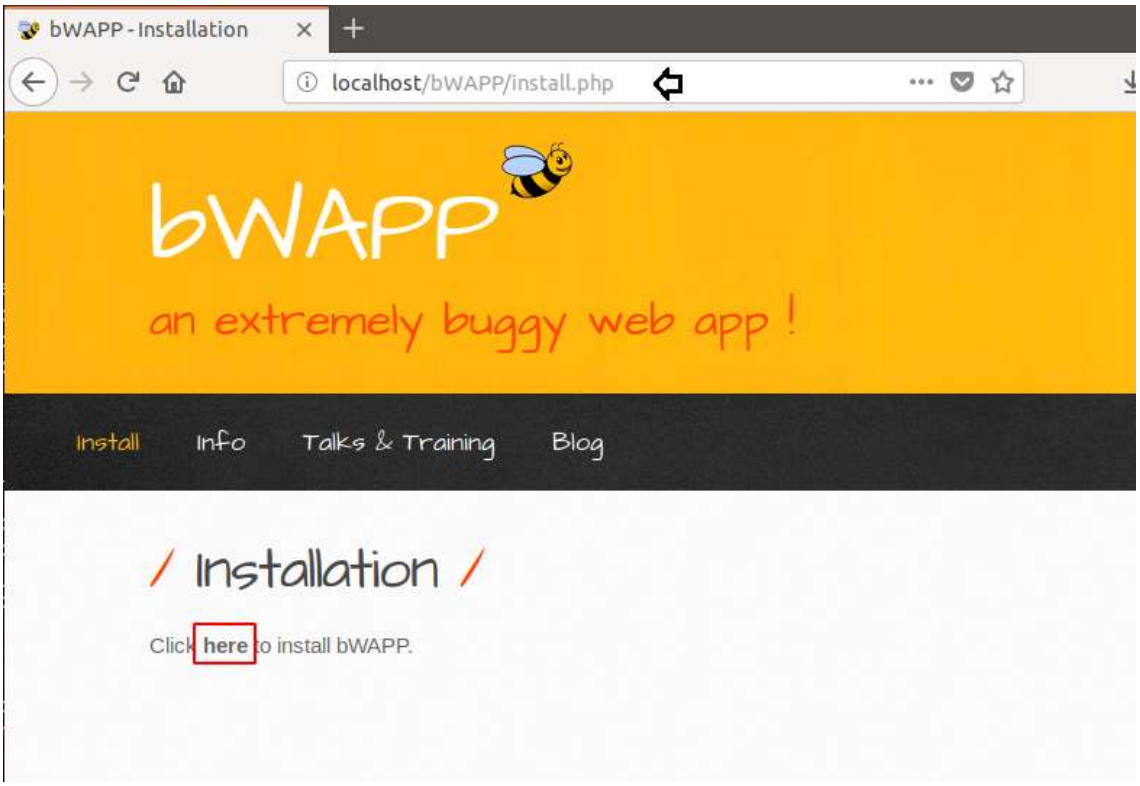
Now we will make some changes and will set our ubuntu user **raj** in place of root and set password **123** in place of bug. Save it and then exit the config file.

```
bWAPP is licensed under a Creative Commons Attribution-NonCommercia

*/

// Database connection settings
$db_server = "localhost";
$db_username = "raj";
$db_password = "123";
$db_name = "bWAPP";

// SQLite database name
$db_sqlite = "db/bwapp.sqlite";

// SMTP settings
$smtp_sender = "bwapp@mailinator.com";
$smtp_recipient = "bwapp@mailinator.com";
$smtp_server = "";
```

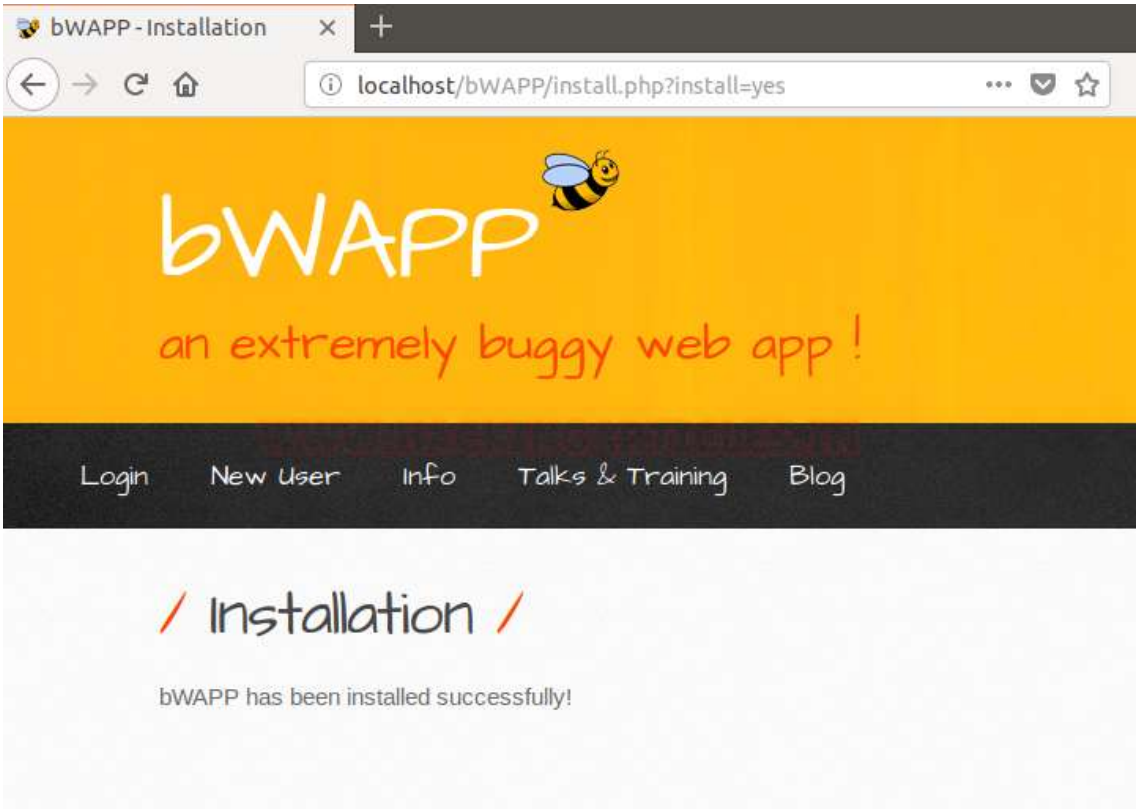
Now go to your browser and open bWAPP installation file by the following command and click on here as shown in the image below



1 http://localhost/bWAPP/install.php

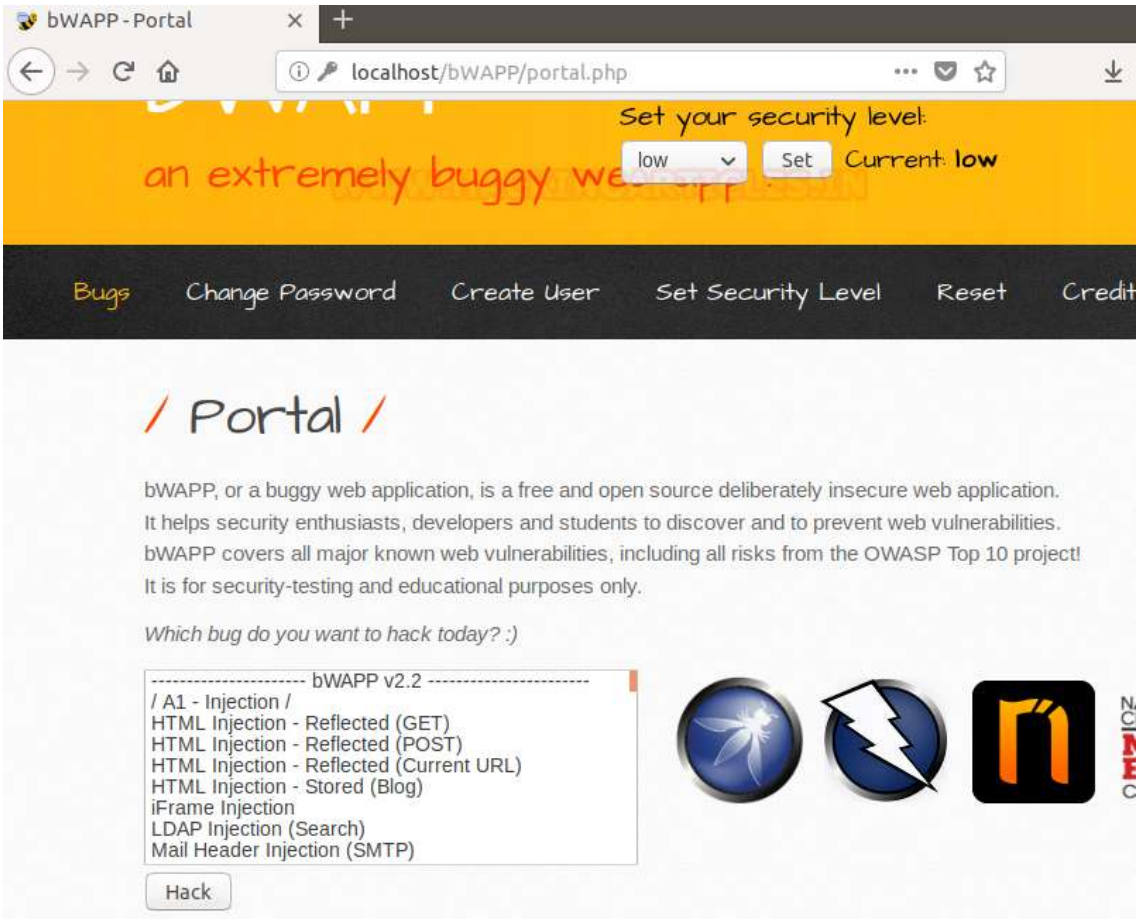


Now you will get a login page of bWAPP where we will use the default username which is **bee** and default password which is **bug** and you are logged in in bWAPP.



Now you can start working on bWAPP.

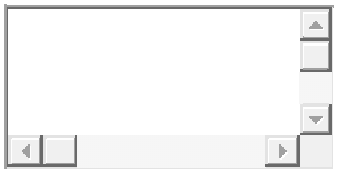
When you will login as bee:bug; you will get the portal to test your penetration testing skill.



XVWA

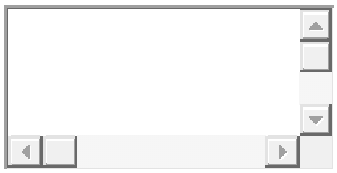
XVWA is poorly coded written in PHP/MYSQL web application that helps security lovers learn security from applications. This application is not advisable online because it is Vulnerable to extremes as the name also suggests. This application should be hosted in a controlled and safe environment where you can improve your skills with the tool of your choice. So, let's start-

First, we will download XVWA from GitHub; so, go to ubuntu terminal and open the following link to download XVWA lab inside html directory by the following link-



```
1 git clone http://github.com/s4n7h0/xvwa.git
```

Once it is downloaded, we will open the config file of xvwa by the following command



```
1 cd xvwa
2 nano config.php
```

Now we can see that the username of xvwa is root and password is left blank.

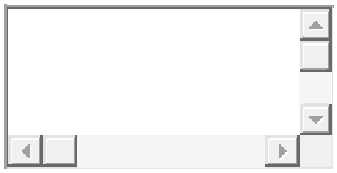
Now we will remove the root user from here and we will be using the ubuntu username and password here which is **raj:123**

Afterwards, we will save the file and exit.

Now browse web application through URL-localhost/xvwa and we can see that we are successfully logged in-

SQLI Labs

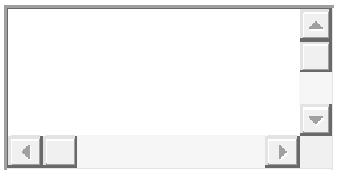
A laboratory that offers a complete test environment for those interested in acquiring or improving SQL injection skills. Let's start. First, we will download SQLI lab inside html directory by the following link-



```
1 git clone http://github.com/Rinkish/Sqli_Edited_Version
```

Once the download is done, we will move sqli labs into the /var/www/html directory and rename it to sqli. Then go inside the sqli directory where we will find **/sqli-connections** directory. Here we will run ls command to check the files and we can see that here is file by the name of db-creds.inc

we need to make some changes in the config file by the following command-



```
1 cd Sqli_Edited_Version/
2 ls
3 mv sqlilabs/ ../sqli
4 cd sqli
5 cd sql-connections/
6 ls
7 nano db-creds.inc
```

```
root@ubuntu:/var/www/html# git clone https://github.com/Rinkish/Sqli_Edited_Version
Cloning into 'Sqli_Edited_Version'...
remote: Enumerating objects: 406, done.
remote: Total 406 (delta 0), reused 0 (delta 0), pack-reused 406
Receiving objects: 100% (406/406), 6.39 MiB | 1.07 MiB/s, done.
Resolving deltas: 100% (81/81), done.
root@ubuntu:/var/www/html# cd Sqli_Edited_Version/
root@ubuntu:/var/www/html/Sqli_Edited_Version# ls
README.md  sqlilabs
root@ubuntu:/var/www/html/Sqli_Edited_Version# mv sqlilabs/ ../sqli
root@ubuntu:/var/www/html/Sqli_Edited_Version# cd ..
root@ubuntu:/var/www/html# ls
bWAPP  DVWA  index.html  master.zip  sqli  Sqli_Edited_Version  xvwa
root@ubuntu:/var/www/html# cd sqli
root@ubuntu:/var/www/html/sqli# cd sql-connections/
root@ubuntu:/var/www/html/sqli/sql-connections# ls
db-creds.inc  functions.php  setup-db-challenge.php  setup-db.php  sql-connect-1.php  sql-connect.php
root@ubuntu:/var/www/html/sqli/sql-connections# nano db-creds.inc
```

As we can see that username is given root and password is left blank which we need to modify.

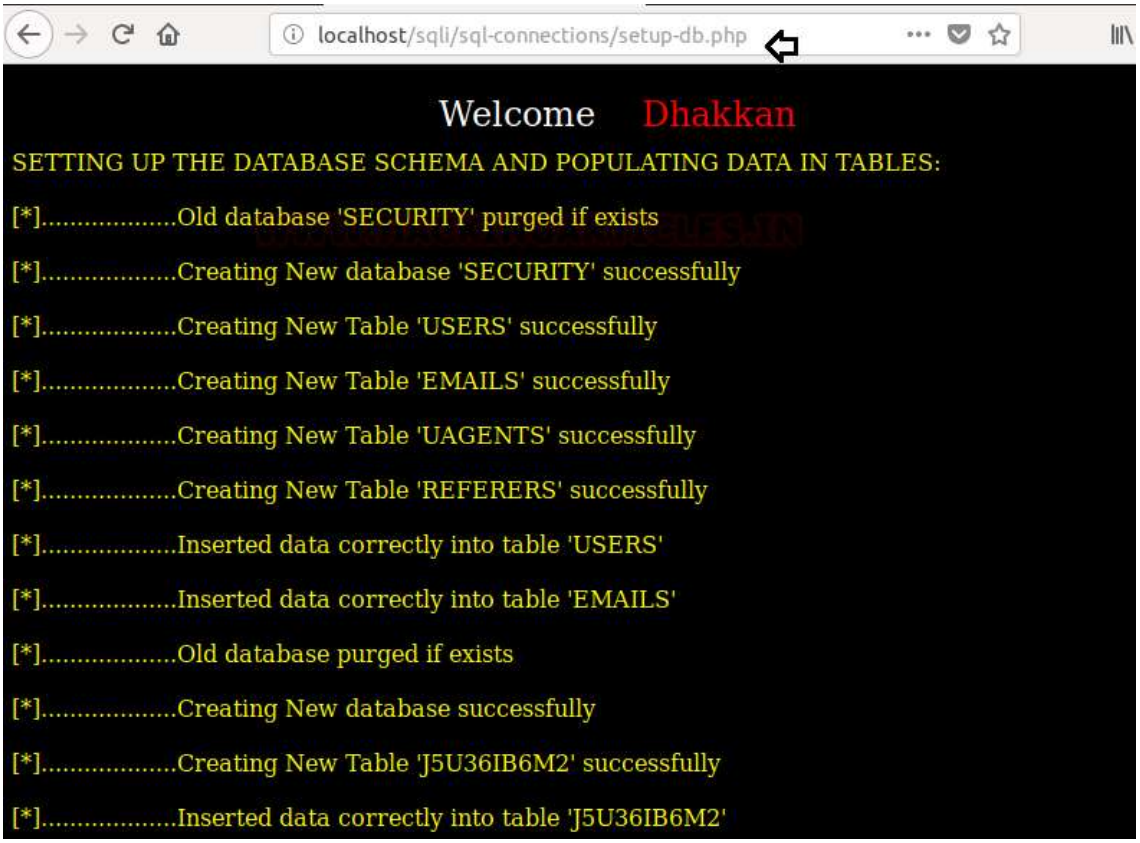
```
<?php

//give your mysql connection username n password
$dbuser ='root';
$dbpass ='';
$dbname ="security";
$host = 'localhost';
$dbname1 = "challenges";



?>
```

Now here we will set the username and password as **raj:123** Now save the file and exit.

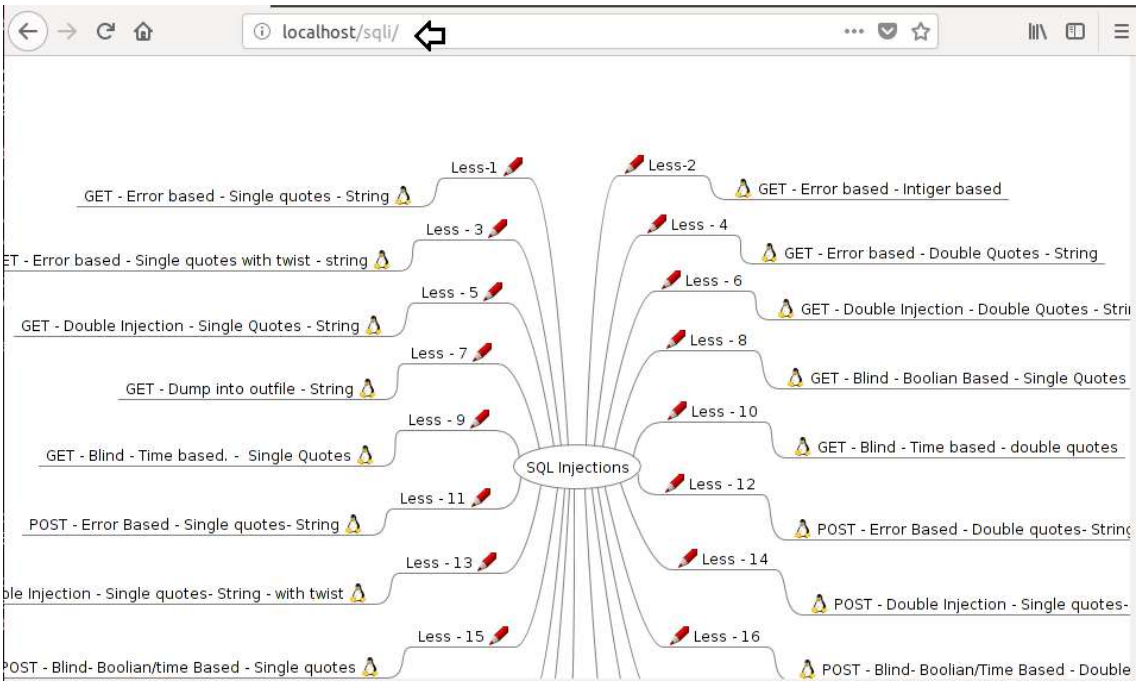
```
<?php

//give your mysql connection username n password
$dbuser ='raj';
$dbpass ='123';
$dbname ="security";
$host = 'localhost';
$dbname1 = "challenges";



?>
```

Now browse this web application from through this URL*:* **localhost/sqli** and click on **Setup/reset** Databases for labs.

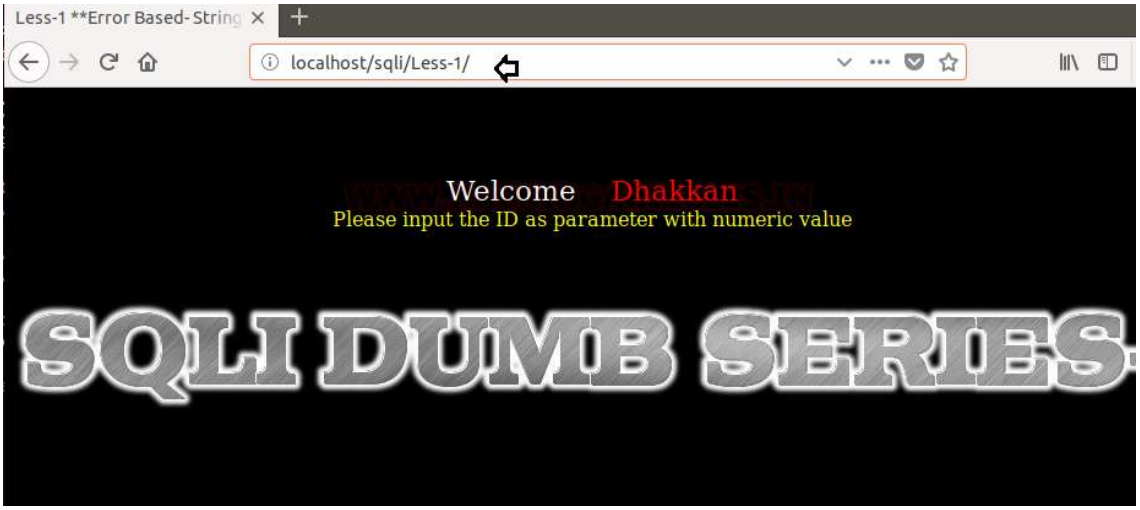
Now the sqli lab is ready to use.



Now a page will open up in your browser which is an indication that we can access different kinds of Sqli challenges



Click on lesson 1 and start the Sqli challenge.

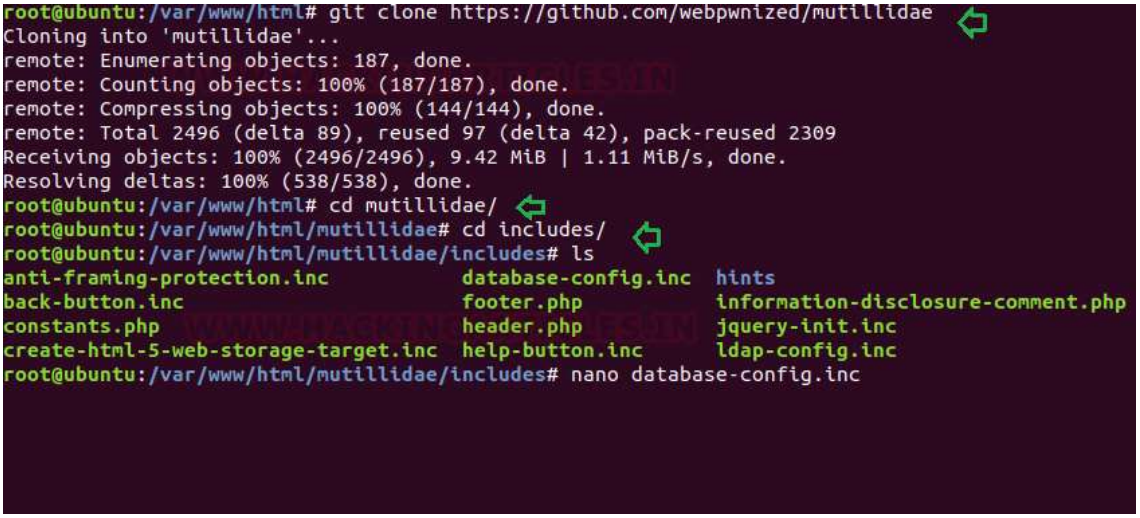


Mutillidae

OWASP Mutillidae is a free open source purposely vulnerable web application providing an enthusiastic goal for web security. It's a laboratory which provides a complete test environment for those who are interested in SQL injection acquisition or improvement. This is an easy-to-
use Web hacking environment designed for laboratories, security lovers, classrooms, CT Fs, and vulnerability assessment targets, and has dozens of vulnerabilities and tips to he lp the user.

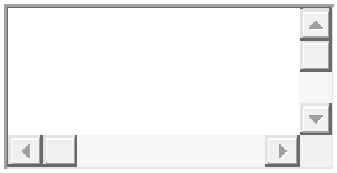
So, let's start by downloading by the clicking on the following link given below-

```
1 git clone https://github.com/webpwnized/mutillidae
```



After the downloading, go inside the Mutillidae directory and where you will find a directory /includes, go inside this directory.

Inside this directory, we will find database-config.inc file which we need to open by nano command as shown in the image below.



```
1 cd mutillidae
2 cd includes
3 ls
4 nano database-config.inc
```

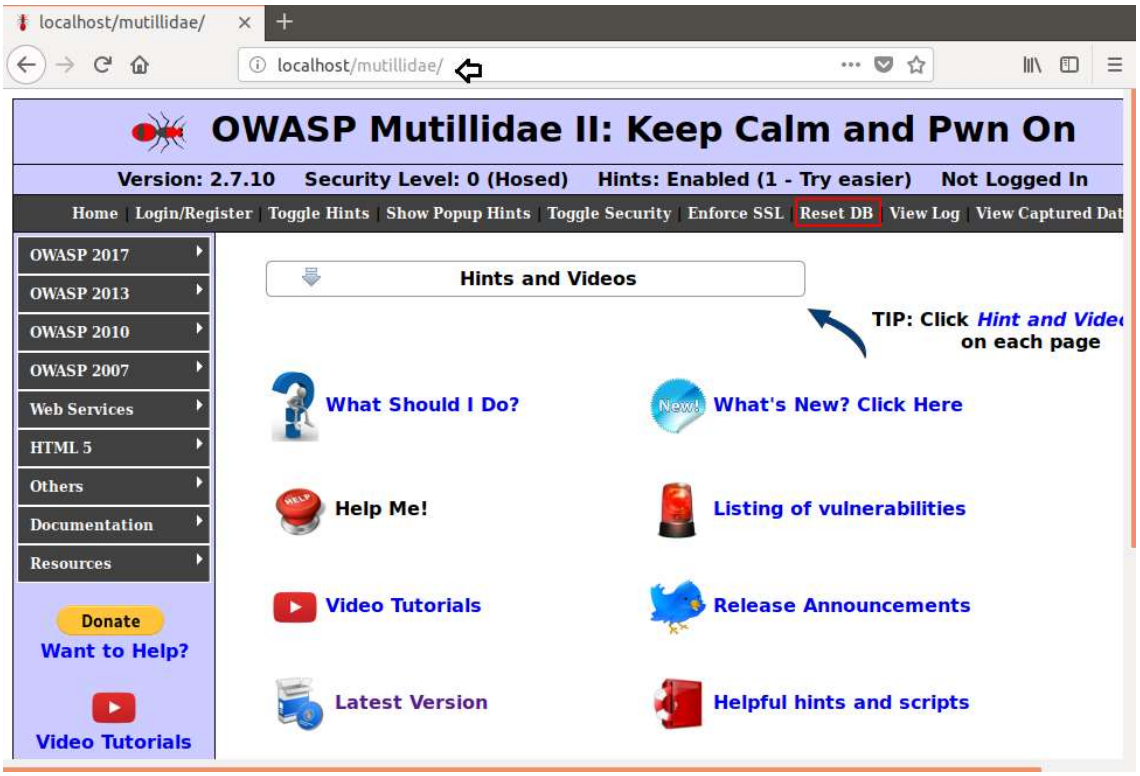
Now here you will find that username is root and password is Mutillidae, by default and which we need to change.

```
<?php
define('DB_HOST', '127.0.0.1');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', 'mutillidae');
define('DB_NAME', 'mutillidae');
?>
```

Now we will use our ubuntu username and password which is raj:123. Save the changes and then exit

```
<?php
define('DB_HOST', '127.0.0.1');
define('DB_USERNAME', 'raj');
define('DB_PASSWORD', '123');
define('DB_NAME', 'mutillidae');
?>
```

Now we will open this our local browser by the following
URL: **localhost/mutillidae** where we will find an option of reset database. Just click on it to reset the database.



Now you will be redirected to a page which will ask you to click ok to proceed. Here you need to click on ok and you are done with the configuration of the Mutillidae lab.

So, In this way, we can setup our vulnerable web application lab for penetration testing.