**BSQL INJECTION TUTORIAL**

1. **INSTALLATION OF DVWA**

Open Terminal

A.Type cd ..

You will be redirected to home

B. Again type cd ..

You will be redirected to start of file system

Type **ls** for var directory

Type cd var

Type cd www

Type cd html

Then you will be redirected to html directory.

C. Open browser type download DVWA

You will get digininja repo (first link)

Clone the URL

**Now switch to root user in kali**

sudo su

Password:

Then type

git clone (link that you copied from git repository)

**1.ls**

**2.mv DVWA dvwa**

**3.ls**

**4. chmod -R 777 dvwa** (for updating the permission of folder)

**5. cd dvwa** (for moving to config folder)

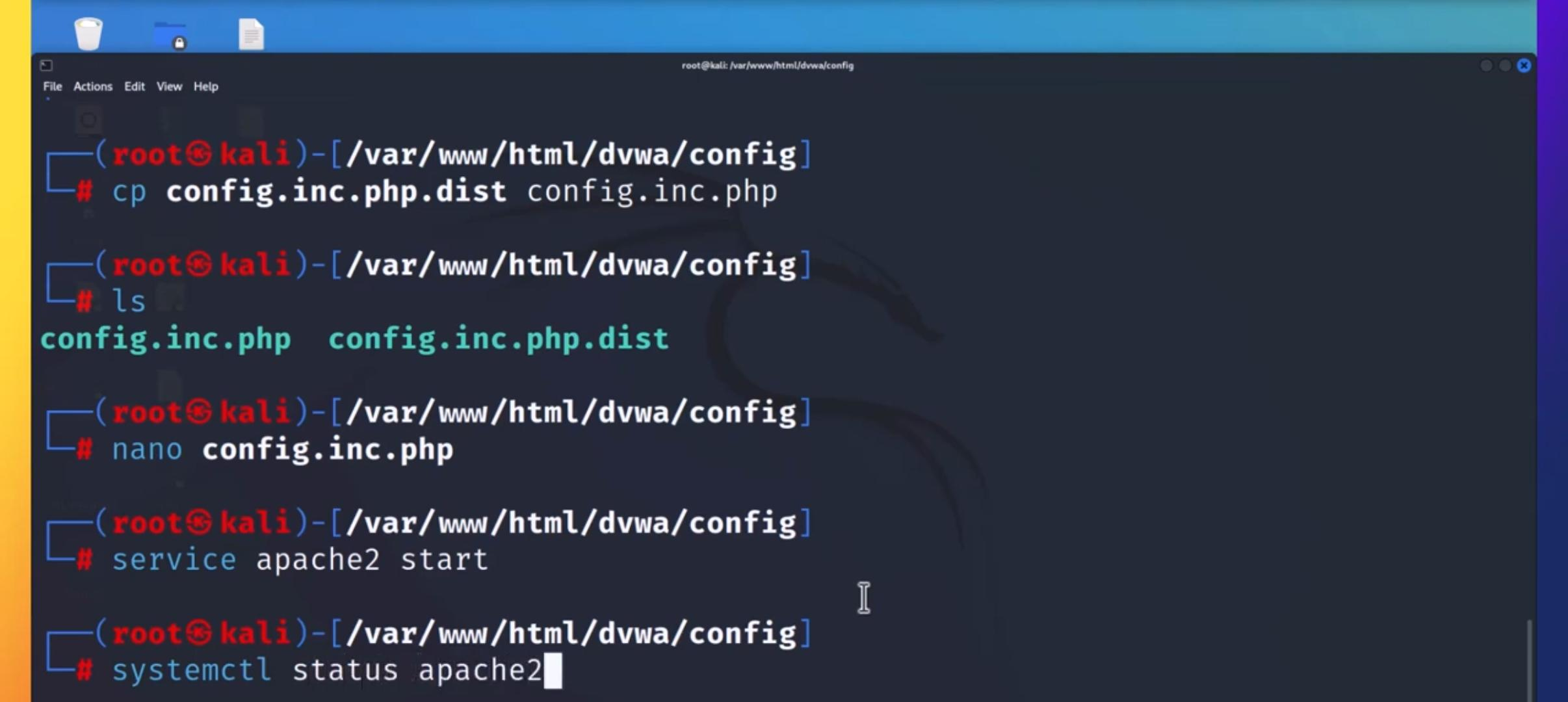
**6.ls**

**7.cd config**

**8. cp config.inc.php.dist config.inc.php**

9.ls

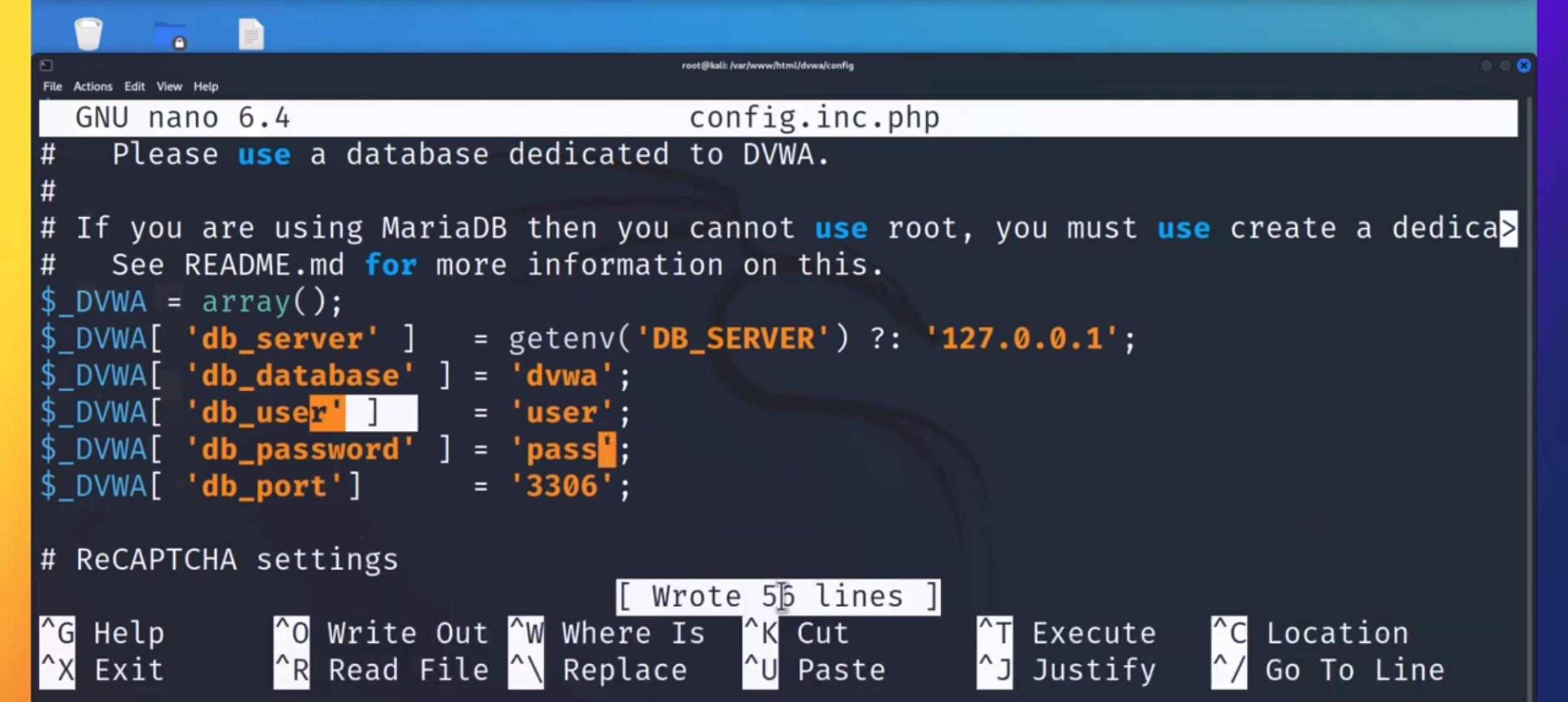
**10. nano config.inc.php** (for editing the file)



For maintaining the database we have to change the credentials

Make [‘db\_user’] = ‘user’;

Make [‘db\_password’] = ‘pass’;

Q

Press ctrl+O it will be write out then press enter

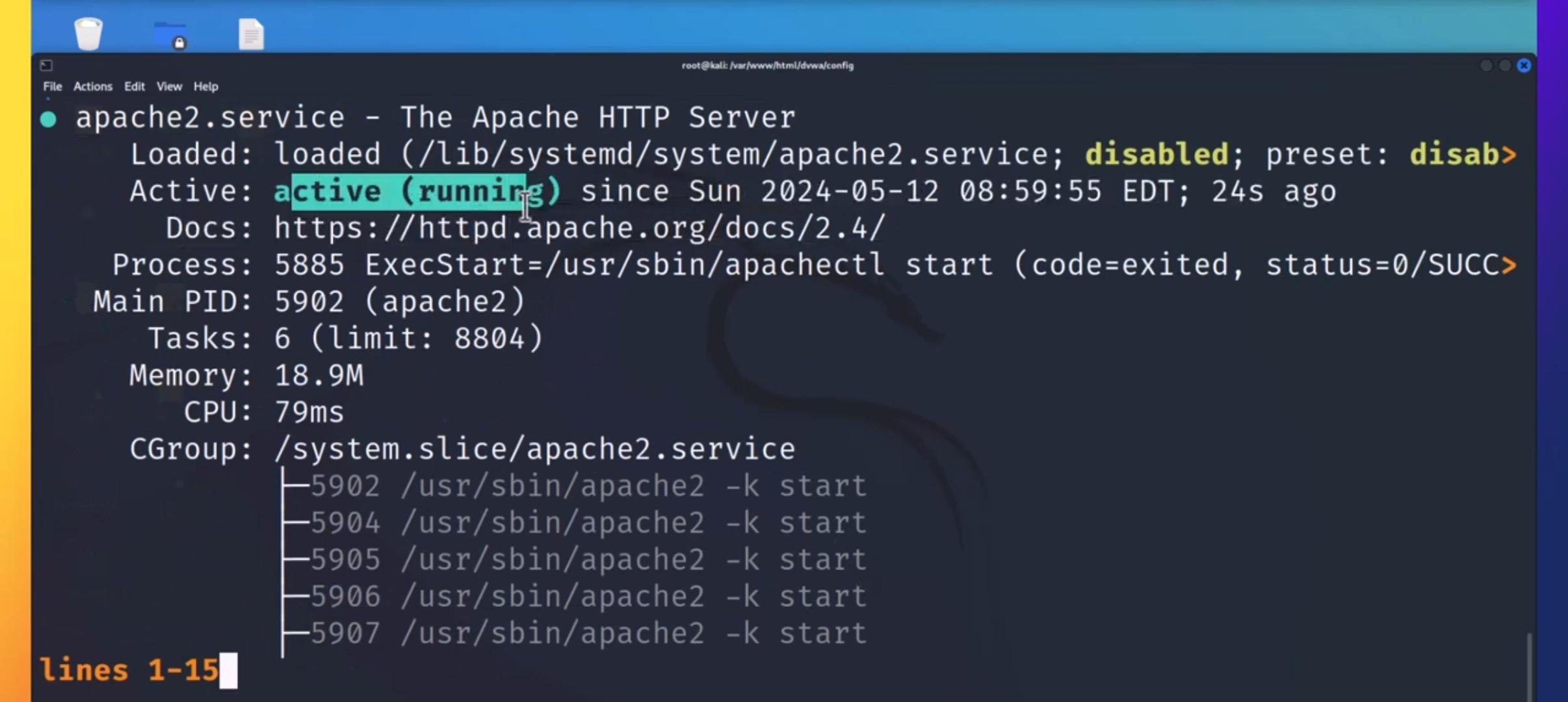
Ctrl x then it will be exit

Now we have to start 2 services

**For web application -Apache 2**

Type : service apache2 start

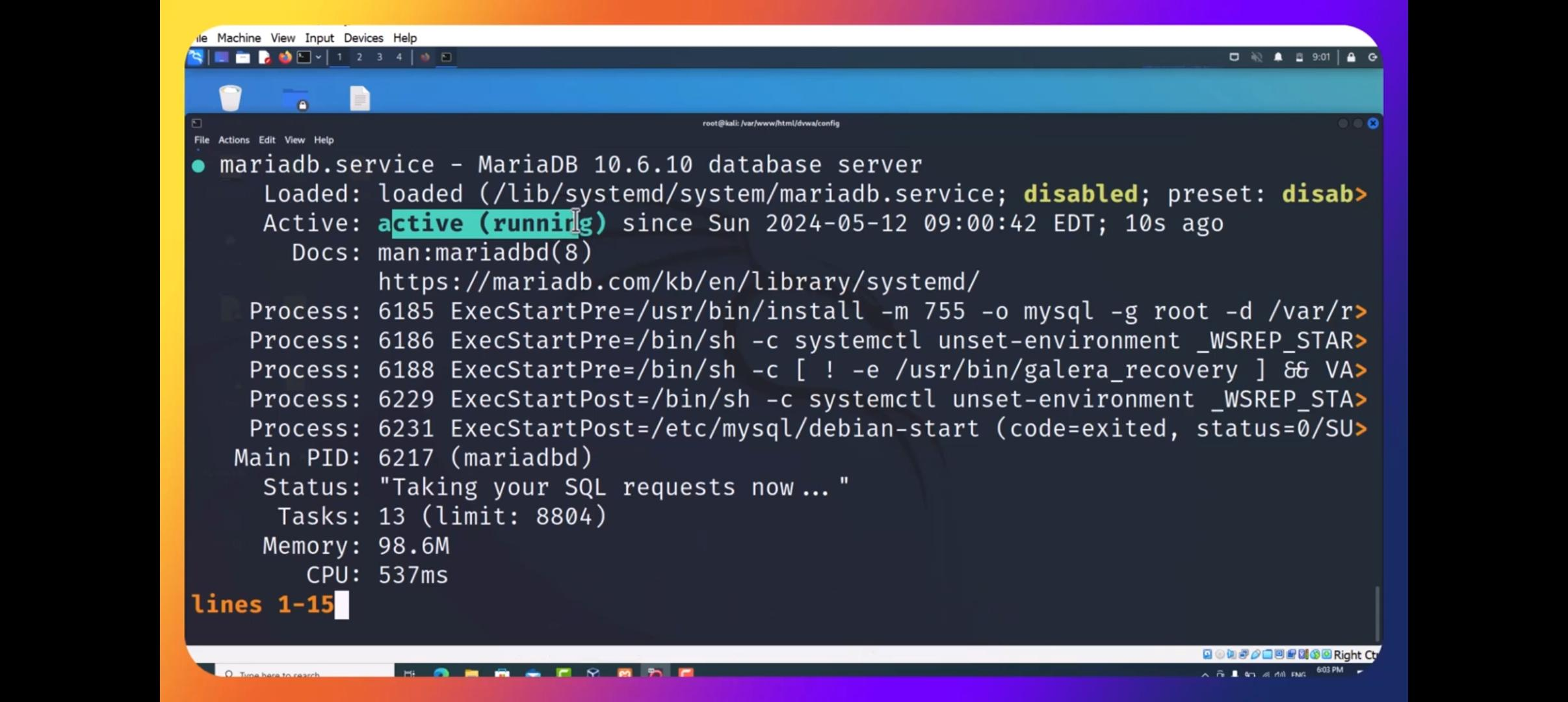
Type : systemctl status apache2



**For Database Server**

Type: service mysql start

Type: systemctl status mysql



For going to database

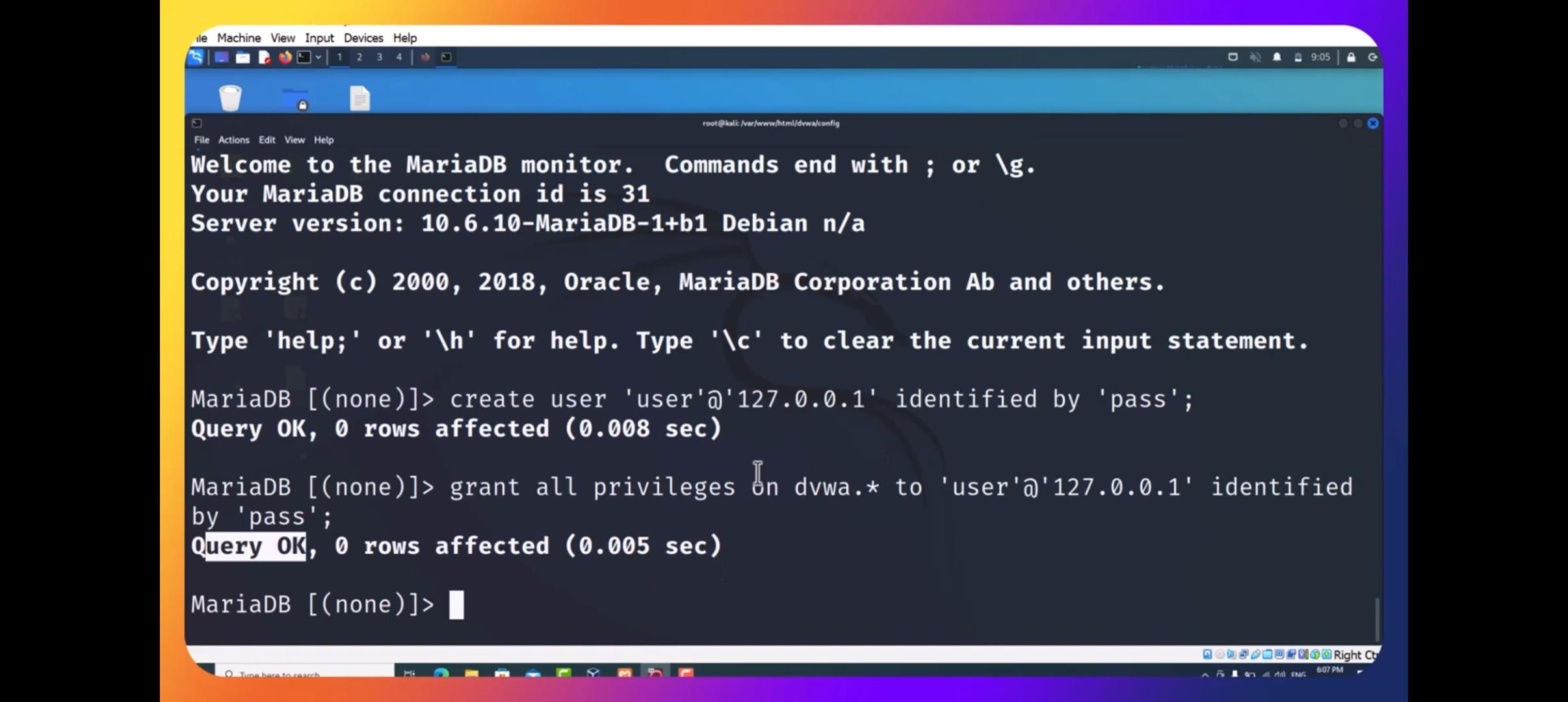
Type : mysql -u root -p

Enter password:

create user ‘user’@’127.0.0.1’ identified by ‘pass’;

Query will be ‘OK’

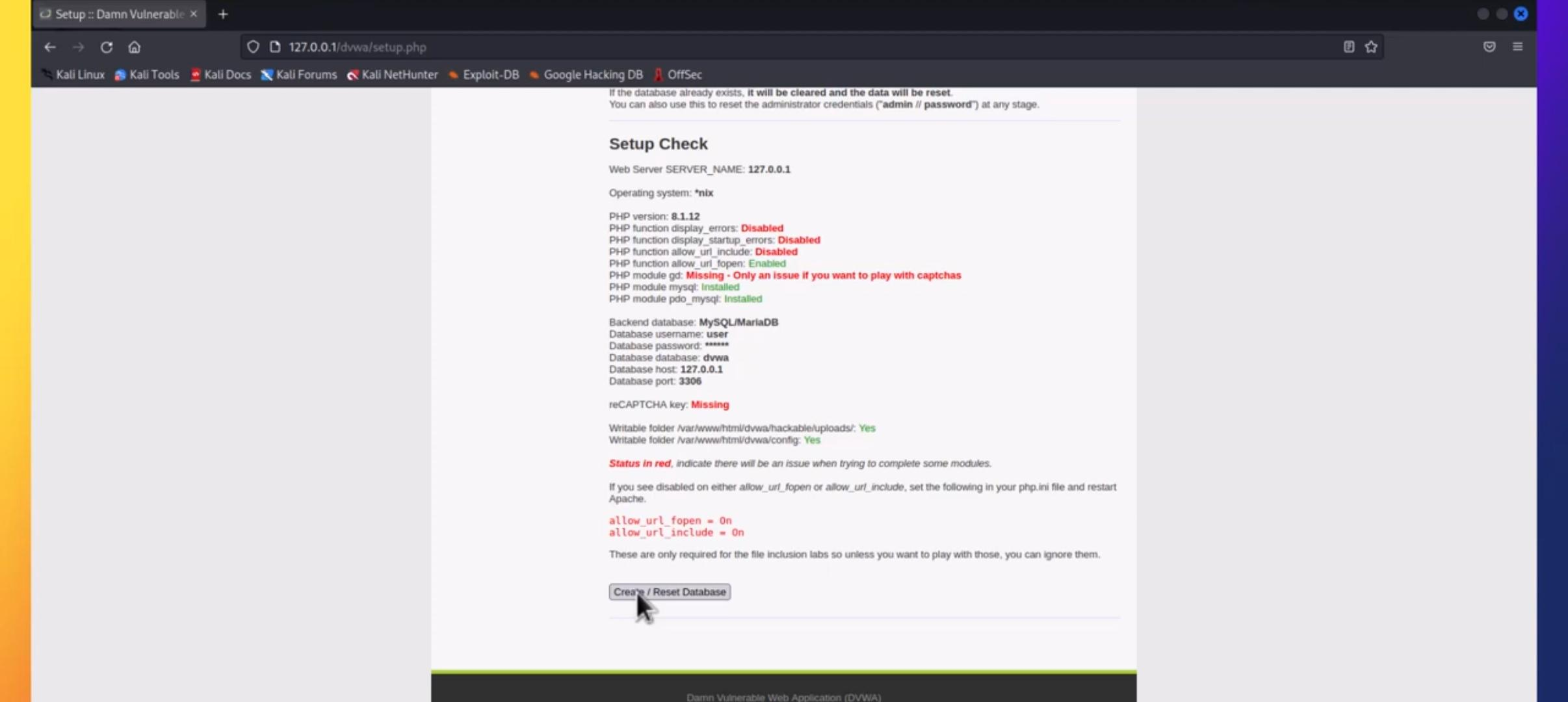
Type : grant all privileges on dvwa.\* to ‘user’@’127.0.0.1’ identified by ‘pass’;



Go to browser

Type 127.0.0.1/dvwa/setup.php

Go to below of the page and click on create/reset password.



You will be redirected to DVWA

Blind SQL Injection Vulnerability Solution

1.Login to DVWA application and go to Blind SQL Injection challenge

2.Its asking to provide User ID:

3.Lets provide 1 as User ID

4.We received response as User ID exists in the database. for User ID 1

5.Now provide 9 as User ID

6.For 9, we received response as User ID is MISSING from the database.

7. Now provide special character '

8.For ', response is - User ID is MISSING from the database.

So for all invalid data we are getting response as User ID is MISSING from the database

9 Lets prepare a payload which always return True value l.e. %' or '1'='1

10. We received a response as User ID exists in the database., for %' or '1'='1

11.now modify the payload, which will return False response %' or '1'='2

12. Now response is - User ID is MISSING from the database.

13. Application considered %' or '1'='1, as a valid User ID.

14. We can conclude an application is vulnerable to SQL Injection, though it's not disclosing any sensitive data such as SQL error, table name or database name.

15. We need to plan our attack further to get sensitive data.