Lab Task 6: Securing Apache Web Server - 2

Task-1 (5 Marks):

If you want to fully secure your website, you must ensure that nobody can access it via HTTP. For this, we will use a specific module of Apache called mod_rewrite.

Step 1: You can use the a2enmod command to enable a module and the a2dismod command to disable a module. Enable the mod_rewrite module using a2enmod command.

sudo a2enmod rewrite

Step 2: Restart the Apache server

sudo systemctl restart apache2

Task-2 (7 Marks):

In this task, we will utilise a rudimentary authentication mechanism of Apache. The premise is that not all users can access your site. It can be accessed only by properly authenticated users.

Step 1: Add users to your Apache web server using the following command

sudo htpasswd -c /etc/apache2/.htpasswd username (change username with the username that you want for your first user)

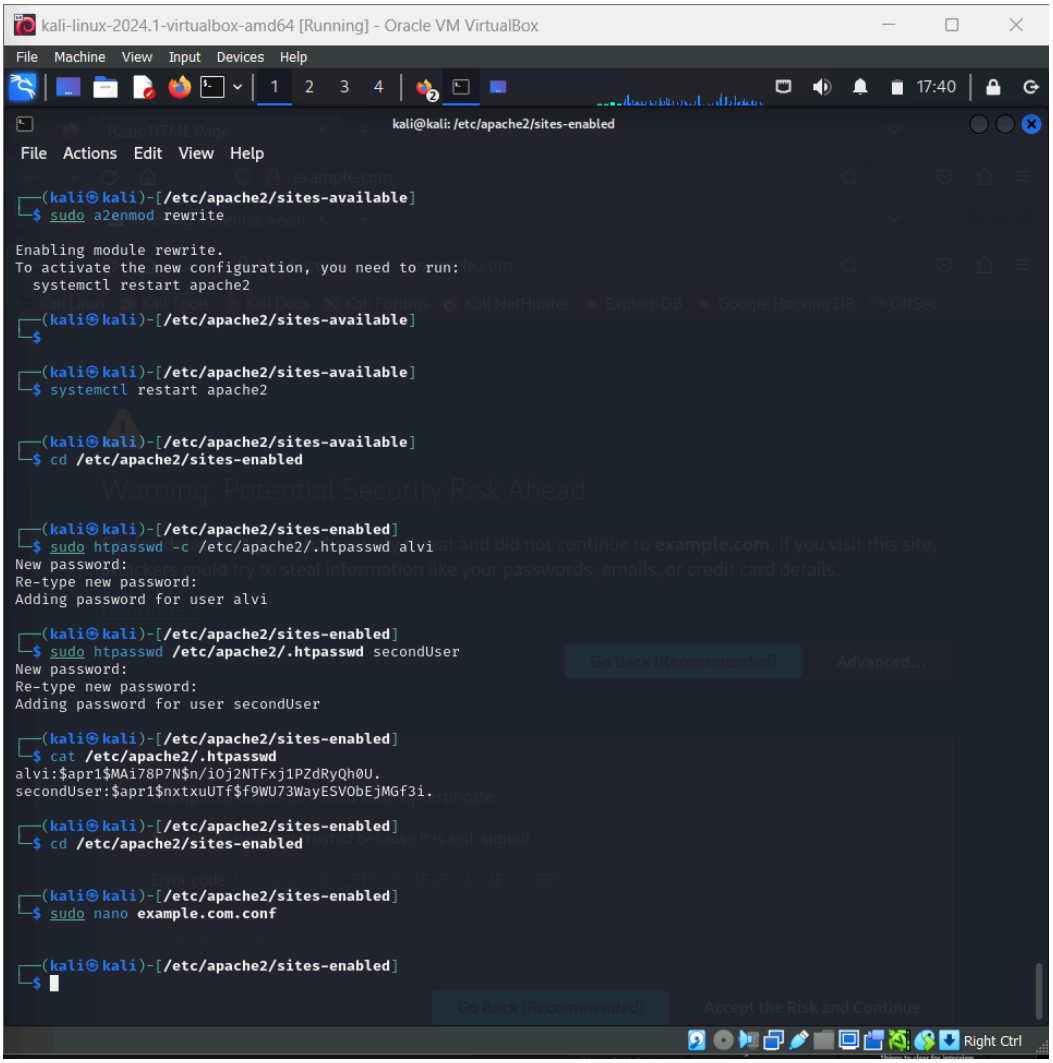
• Add a second user to your Apache server using the htpasswd command.

Step 2: Use the following command to cat the contents of .htpasswd file:
cat /etc/apache2/.htpasswd

You will something like the following, containing the usernames and their corresponding hashed passwords.

Step 3: Next, add the following into your https configuration file for example.com.

Step 4: Restart the apache server.



Task-3 (8 Marks):

Modern web servers often rely on a database to authenticate a user. In this task, you will need an authentication mechanism that relies on MySQL database.

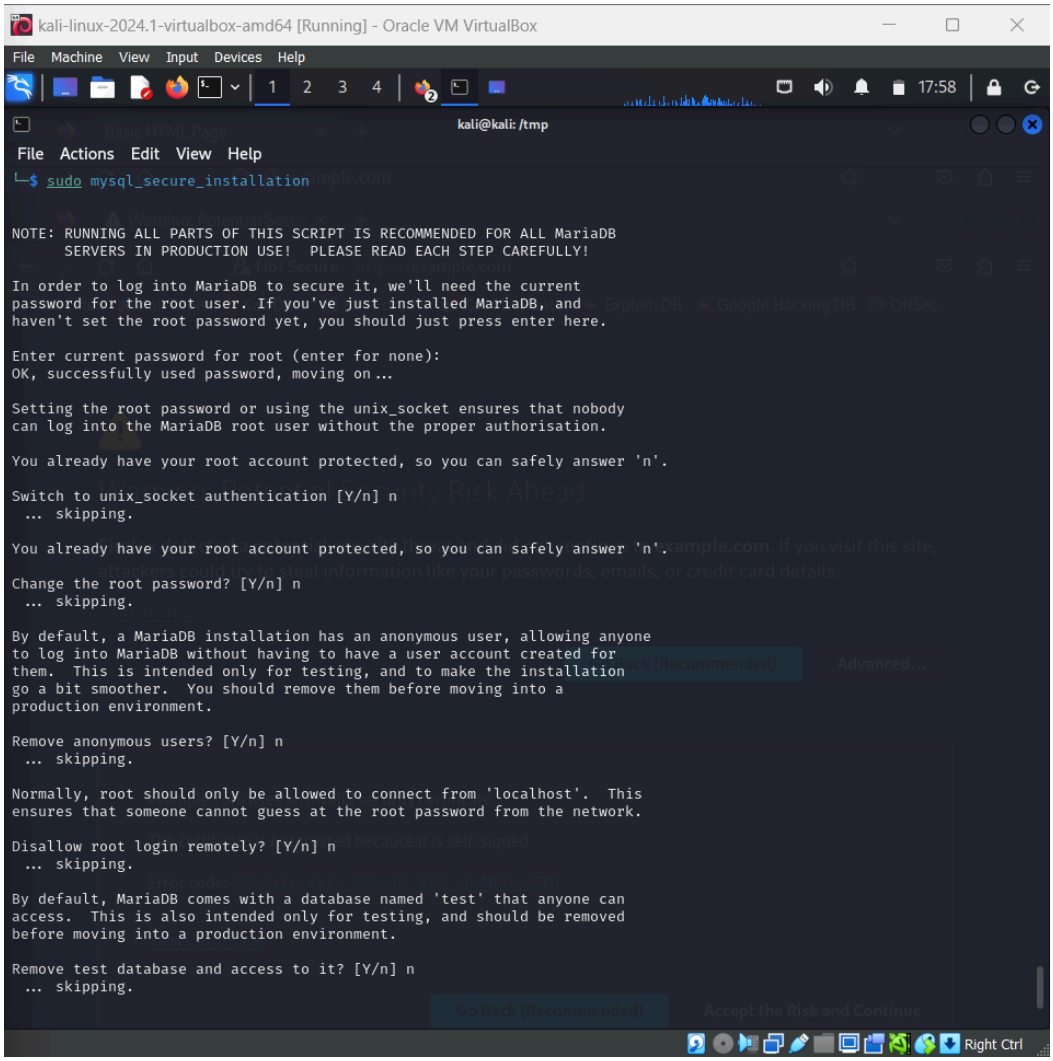
Step 1: Install MySQL server on your ubuntu using the following commands:

sudo apt-get update
sudo apt-get install mysql-server
sudo apt-get installlibaprutil1-dbd-mysql

Step 2: Configuring MySQL using the following command:

Mysql_secure_installation

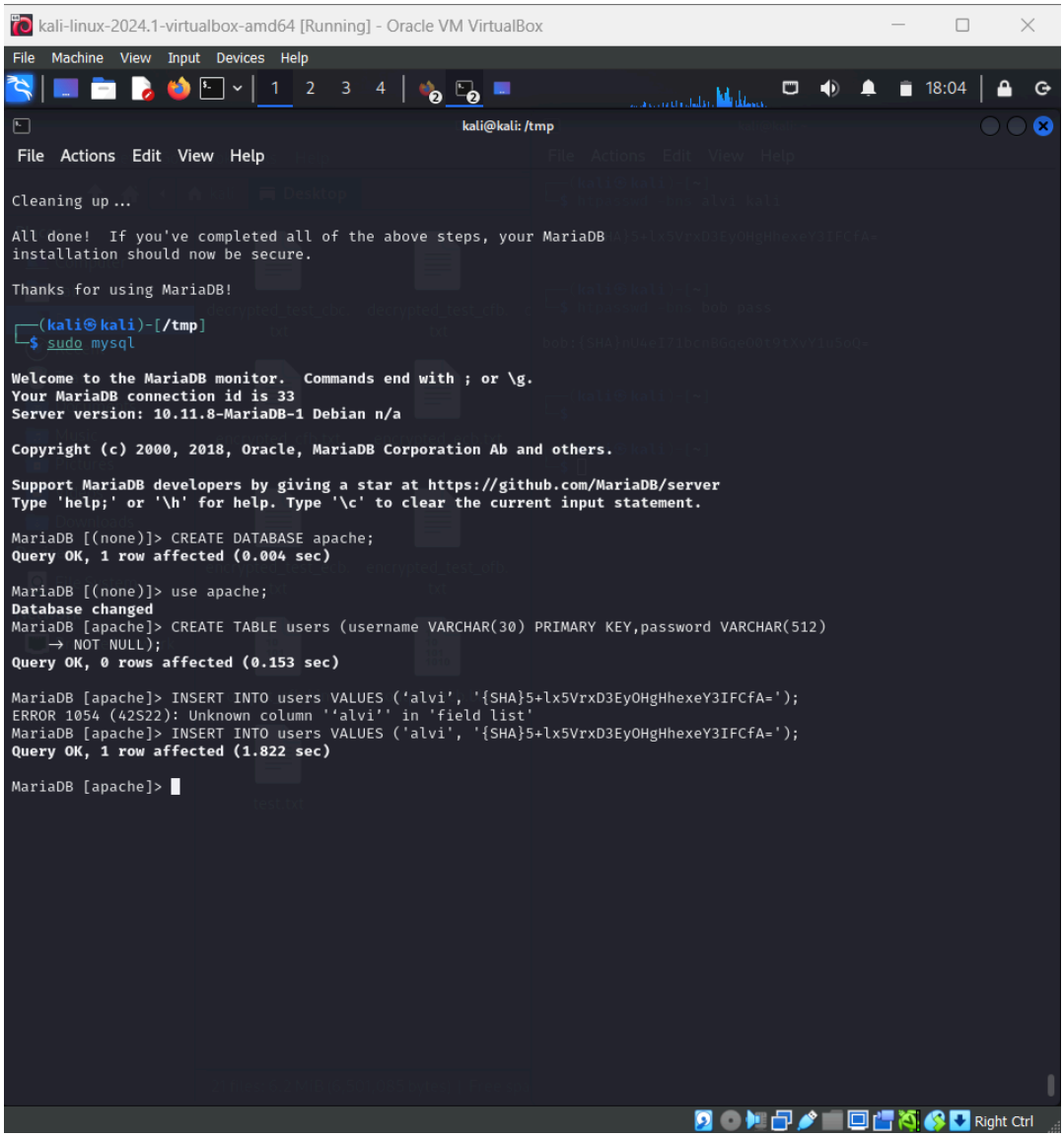
Select no for every option except the last when you are prompted to reload the privilege table.



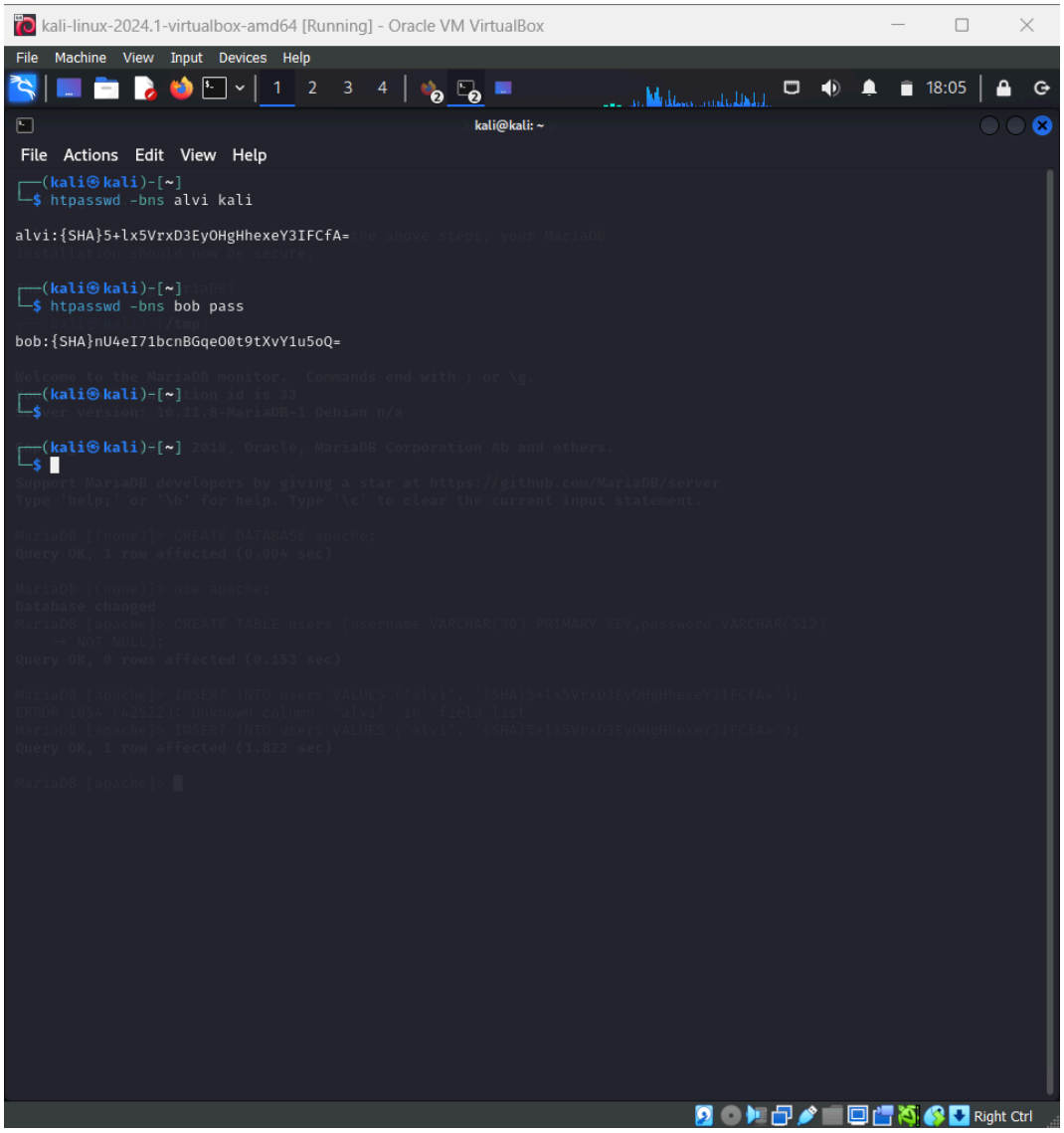
Step 5: Create a database called apache using the following command in the MySQL console:

CREATE DATABASE apache;

Then create a database that will hold the user data and passwords.



Step 6: Now, we will add users to our table. But before that we need to create a hashed password which will be stored in the database.

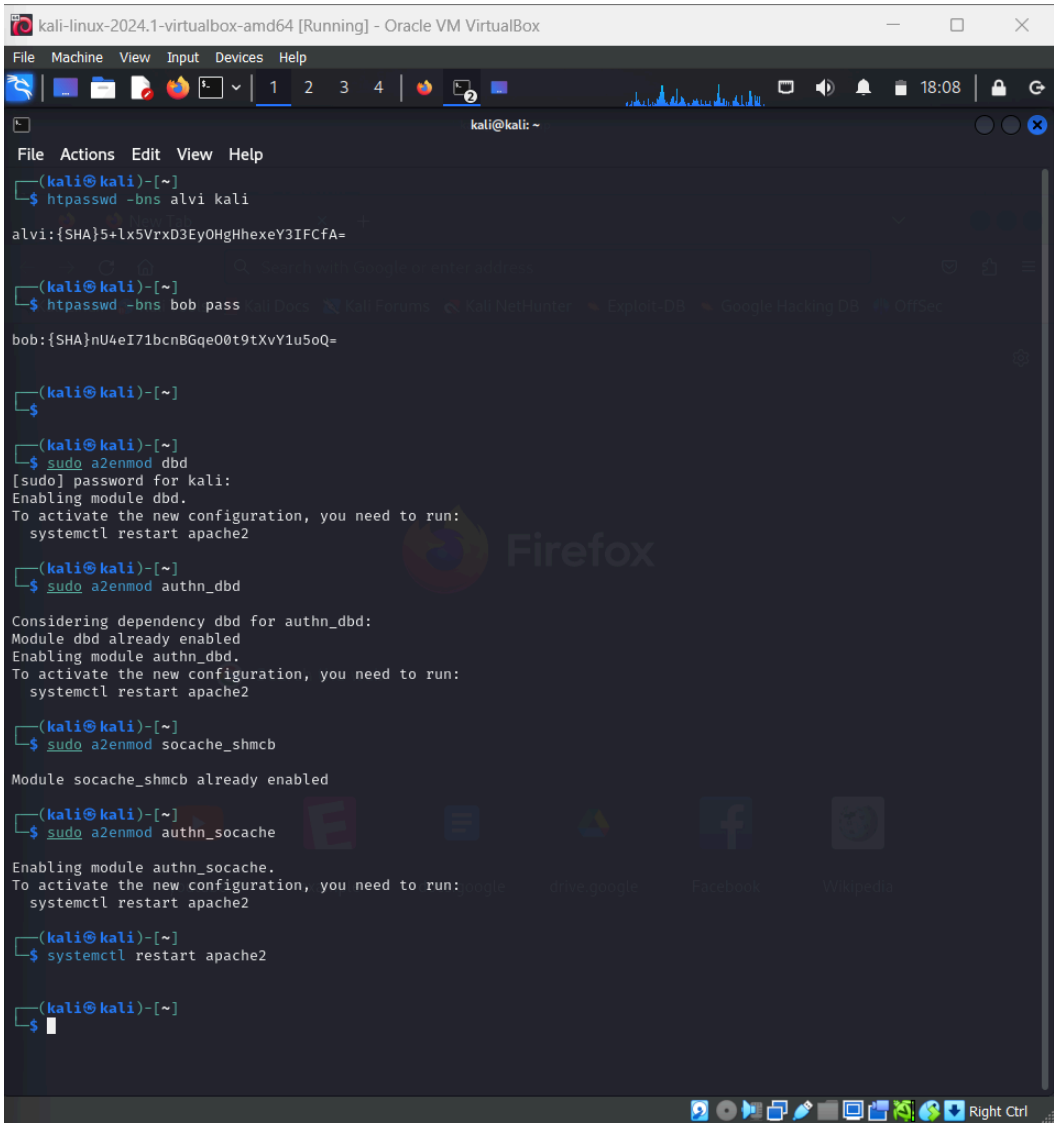


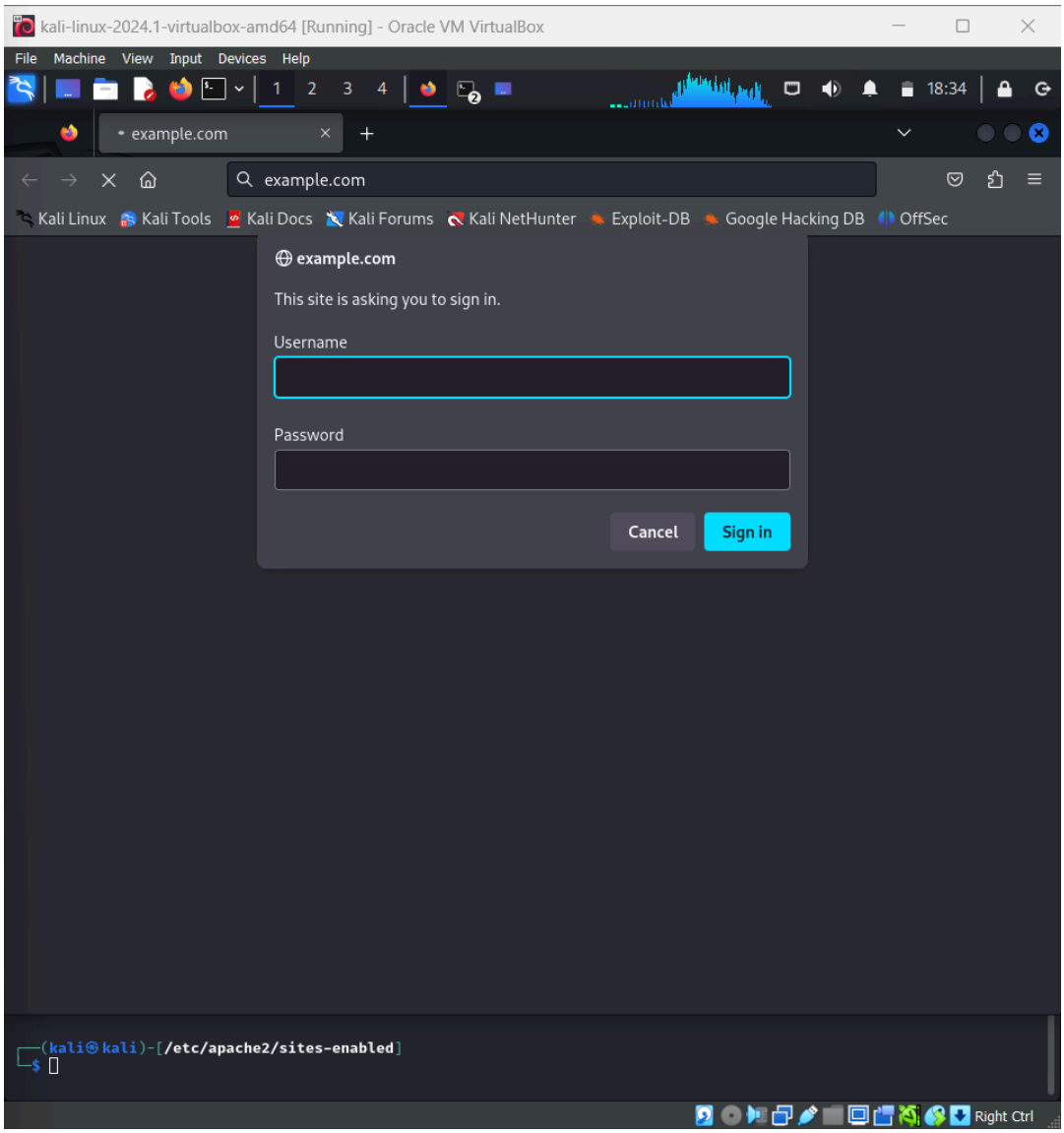
The copy the username and password in the user table using **INSERT** command.

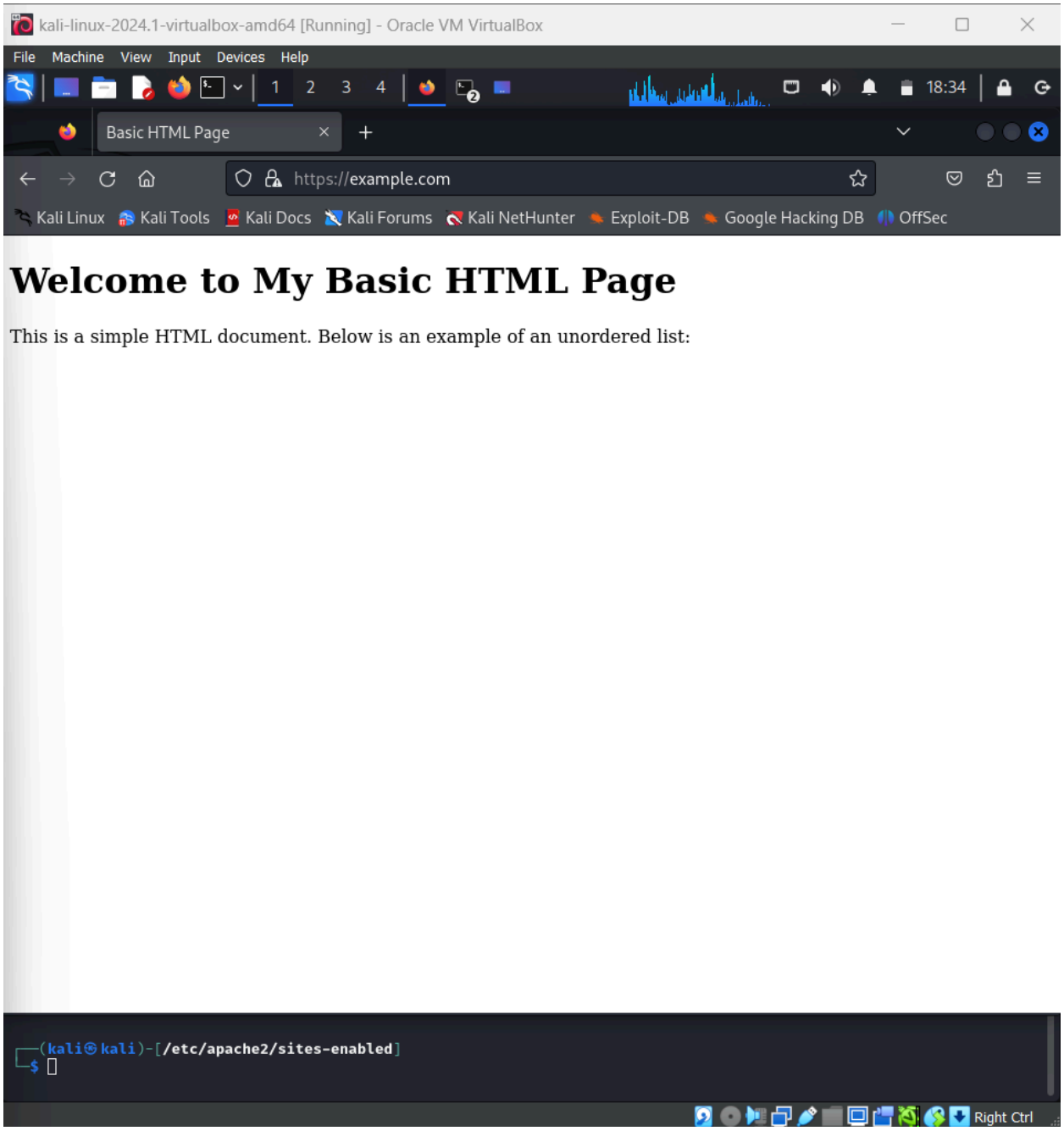
Step 9: Next, issue the following commands to enable the mod_authn_dbd module of Apache. This can be done by enabling the following modules:

sudo a2enmod dbd
sudo a2enmod authn_dbd
sudo a2enmod socache_shmcb
sudo a2enmod authn_socache

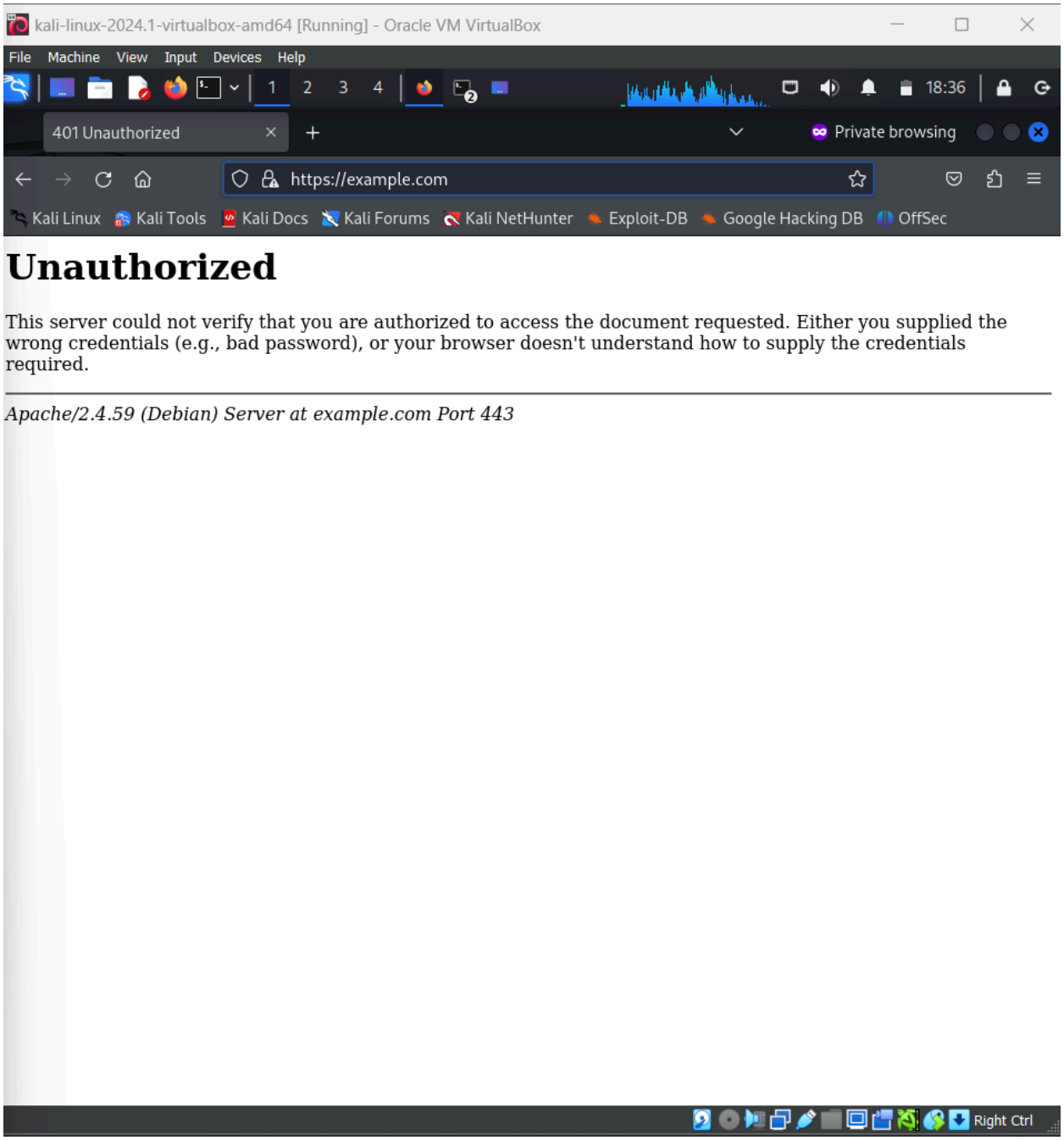
Now restart the apache server and try to access the website. There will be a username / password prompt, and upon providing correct data, everything works good.




kali-linux-2024.1-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
18:34
example.com
example.com
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
example.com
This site is asking you to sign in.
Username
Password
Cancel
Sign in
(kali㉿kali)-[/etc/apache2/sites-enabled]
$
Right Ctrl



If wrong information is provided, the website can not be accessed and the user is identified as “Unauthorized”.


kali-linux-2024.1-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
18:36
401 Unauthorized
Private browsing
https://example.com
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Unauthorized
This server could not verify that you are authorized to access the document requested. Either you supplied the wrong credentials (e.g., bad password), or your browser doesn't understand how to supply the credentials required.
Apache/2.4.59 (Debian) Server at example.com Port 443
Right Ctrl