

SQL Injection Fundamentals Report

Introduction

<https://academy.hackthebox.com/achievement/949661/33>

This report is designed to shed light on the crucial elements of SQL Injection, elaborating on essential concepts associated with databases, MySQL, and SQL Injections, complemented by practical exploitation examples and strategies for mitigation.

Databases

Intro to Databases

Databases are organized collections of data, typically stored and accessed electronically from a computer system. They allow users to store, retrieve, update, and delete data efficiently.

Types of Databases

There are several types of databases, including relational databases, NoSQL databases, in-memory databases, and distributed databases, each serving different use cases and data requirements.

MySQL

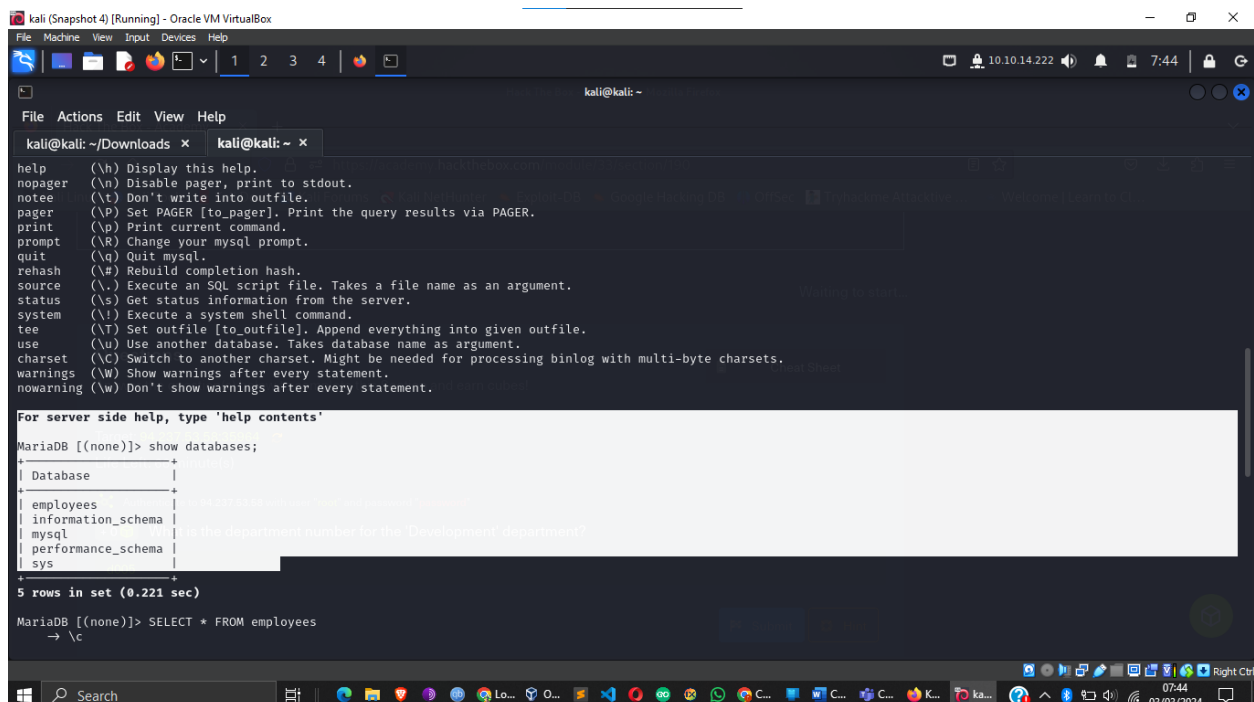
Intro to MySQL

MySQL is a popular open-source relational database management system that uses Structured Query Language (SQL) for accessing and managing the data it stores.

- **Connect to the database using the MySQL client from the command line.**

Question: Use the 'show databases;' command to list databases in the DBMS. What is the name of the first database?

Answer: employees



```
kali (Snapshot 4) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~/Downloads x kali@kali: ~ x
help      (\h) Display this help.
nopager   (\n) Disable pager, print to stdout.
notee     (\t) Don't write into outfile.
pager     (\P) Set PAGER [to_pager]. Print the query results via PAGER.
print     (\p) Print current command.
prompt    (\R) Change your mysql prompt.
quit      (\q) Quit mysql.
rehash    (\#) Rebuild completion hash.
source     (\.) Execute an SQL script file. Takes a file name as argument.
status    (\s) Get status information from the server.
system     (\!) Execute a system shell command.
tee        (\T) Set outfile [to_outfile]. Append everything into given outfile.
use        (\u) Use another database. Takes database name as argument.
charset   (\C) Switch to another charset. Might be needed for processing binlog with multi-byte charsets.
warnings   (\W) Show warnings after every statement.
nowarning  (\w) Don't show warnings after every statement.

For server side help, type 'help contents'

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| employees |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.221 sec)

MariaDB [(none)]> SELECT * FROM employees
→ \c
```

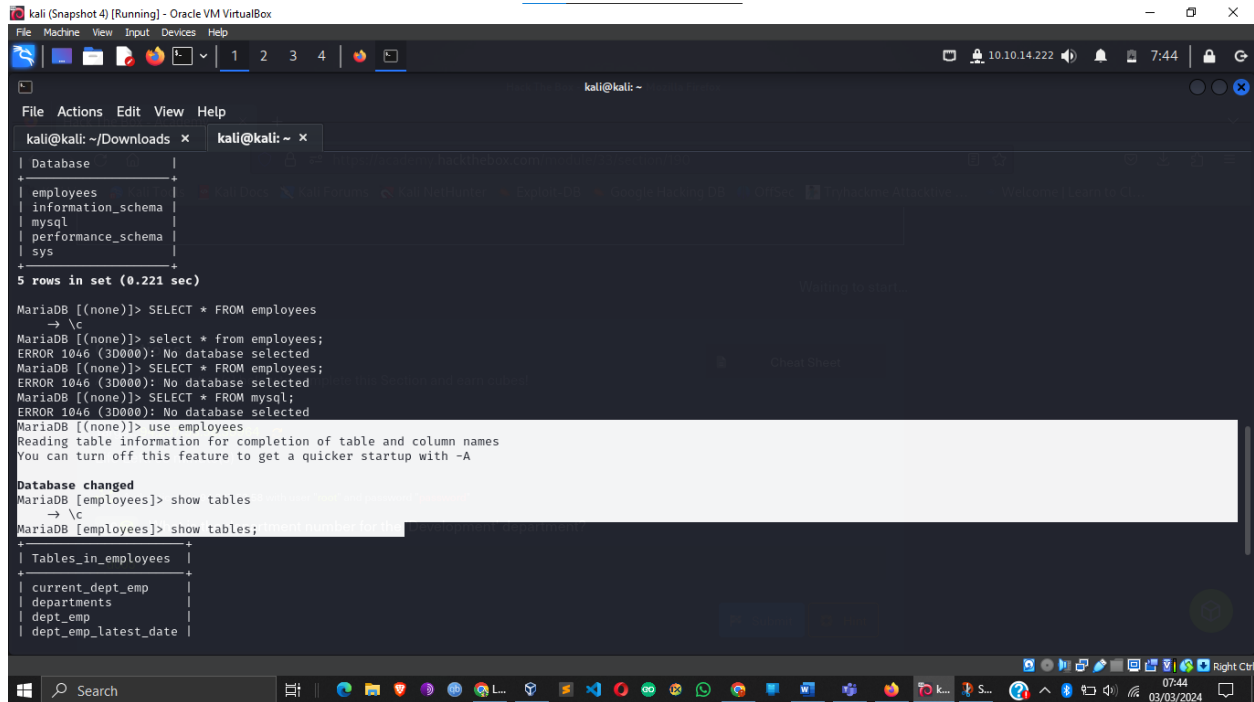
SQL Statements

SQL statements are used to perform tasks such as updating data on a database or retrieving data from a database.

- **SQL Statements Example**

Question: What is the department number for the 'Development' department?

Answer: d005

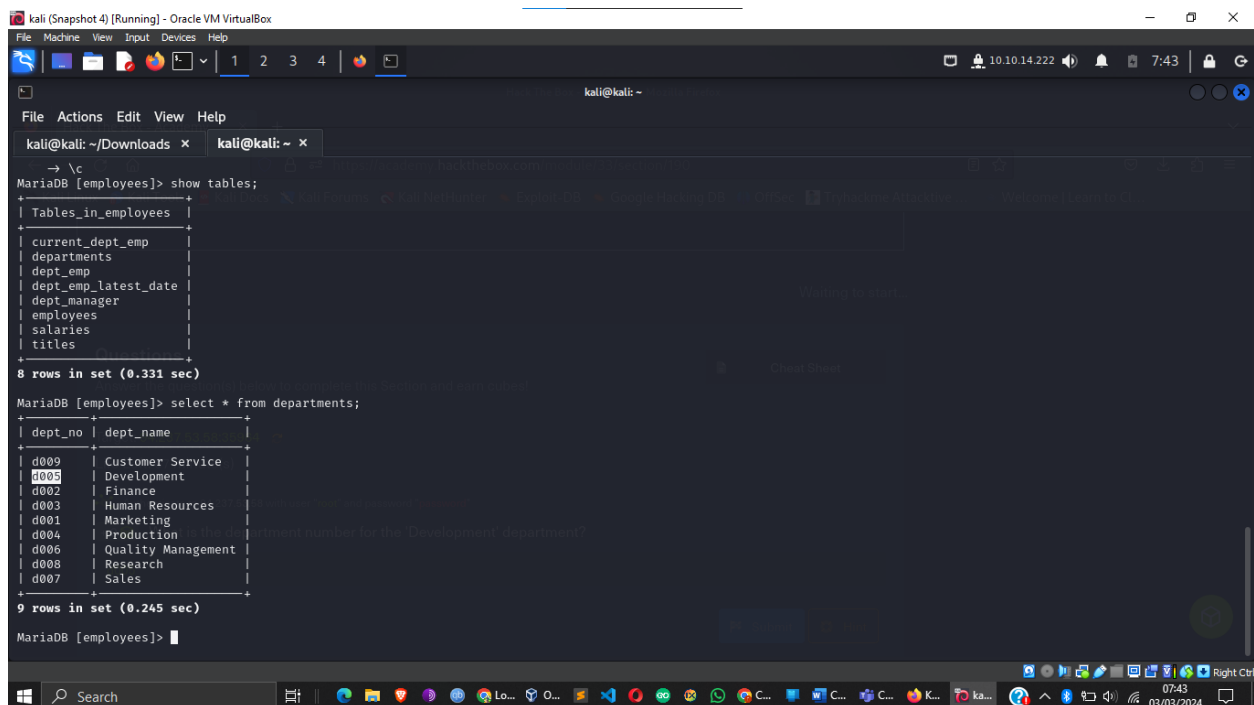


The screenshot shows a terminal window with a MySQL prompt. The user has successfully connected to the 'employees' database. The terminal output shows the following commands and results:

```
MariaDB [(none)]> SELECT * FROM employees
→ \c
MariaDB [(none)]> select * from employees;
ERROR 1046 (3D000): No database selected
MariaDB [(none)]> SELECT * FROM employees;
ERROR 1046 (3D000): No database selected
MariaDB [(none)]> SELECT * FROM mysql;
ERROR 1046 (3D000): No database selected
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
→ \c
MariaDB [employees]> show tables;
```

Tables_in_employees
current_dept_emp
departments
dept_emp
dept_emp_latest_date



The screenshot shows the same terminal window with the following commands and results:

```
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 (0.331 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 (0.245 sec)

MariaDB [employees]>
```

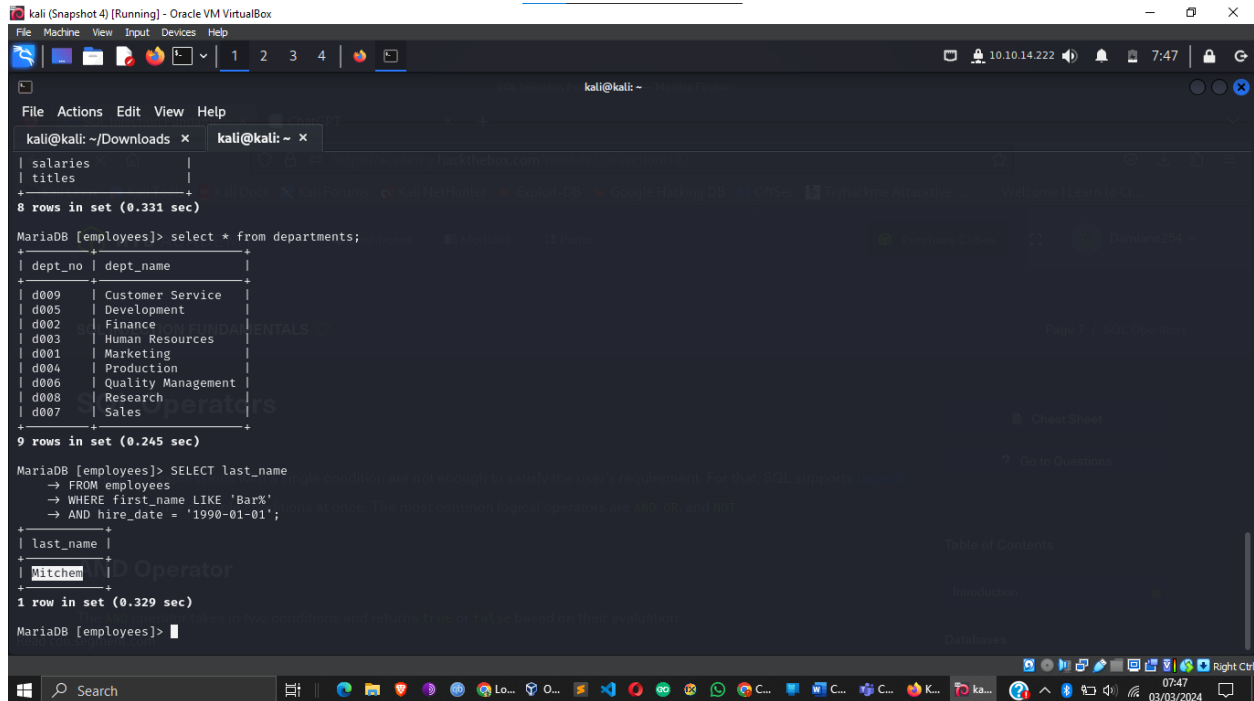
Query Results

Query results are the outputs that SQL statements return, often presented in a tabular format.

- **Query Results Example**

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

Answer: Mitchem



```
kali (Snapshot 4) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~/Downloads x kali@kali: ~ x
| salaries |
| titles |
+-----+
8 rows in set (0.331 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 (0.245 sec)

MariaDB [employees]> SELECT last_name
  FROM employees
  WHERE first_name LIKE 'Bar%'
  AND hire_date = '1990-01-01';
+-----+
| last_name |
+-----+
| Mitchem |
+-----+
1 row in set (0.329 sec)

MariaDB [employees]>
```

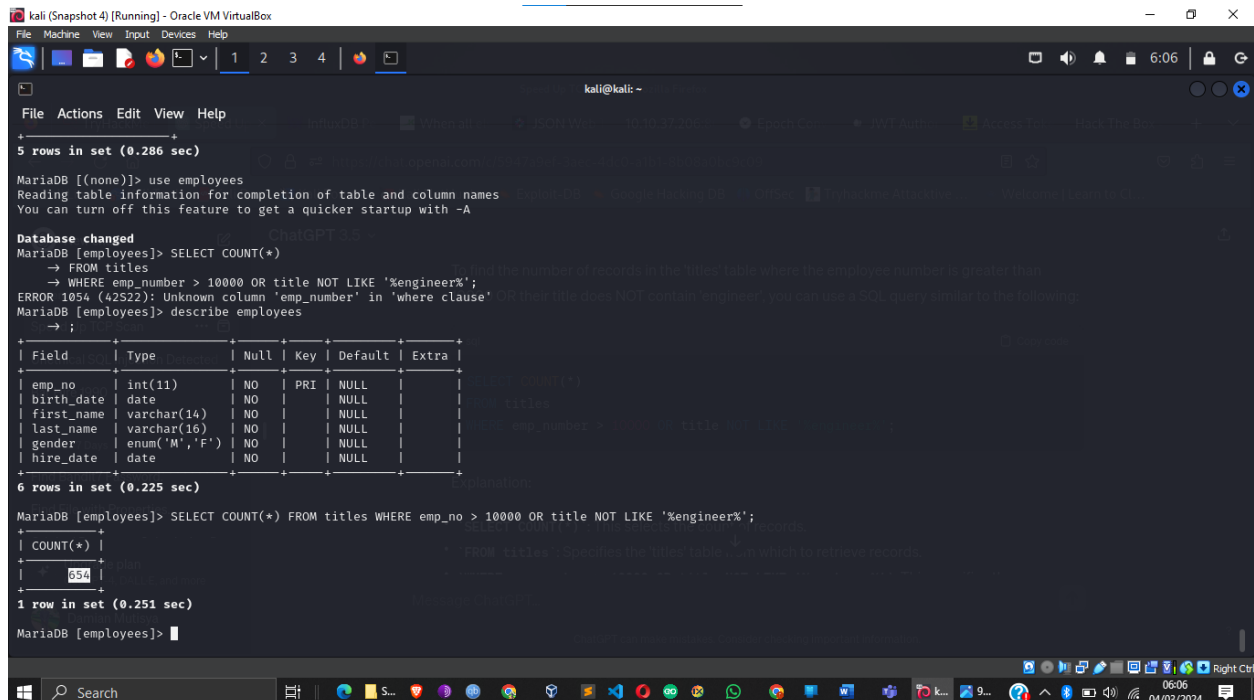
SQL Operators

SQL operators are used to perform operations on data values and to manipulate the data based on conditions.

- SQL Operators Example

Question: 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'?

Answer: 654



```
kali (Snapshot 4) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~ x
+-----+
5 rows in set (0.286 sec)

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]> SELECT COUNT(*)
  FROM titles
  WHERE emp_number > 10000 OR title NOT LIKE '%engineer%';
ERROR 1054 (42S22): Unknown column 'emp_number' in 'where clause'
MariaDB [employees]> describe employees;
+-----+
1 row in set (0.225 sec)

+-----+-----+-----+-----+-----+-----+
| 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 | |
+-----+-----+-----+-----+-----+-----+

MariaDB [employees]> SELECT COUNT(*) FROM titles WHERE emp_no > 10000 OR title NOT LIKE '%engineer%';
+-----+
| COUNT(*) |
+-----+
| 654 |
+-----+
1 row in set (0.251 sec)

MariaDB [employees]>
```

SQL Injections

Intro to SQL Injections

SQL Injection is a code injection technique that exploits a security vulnerability occurring in the database layer of an application.

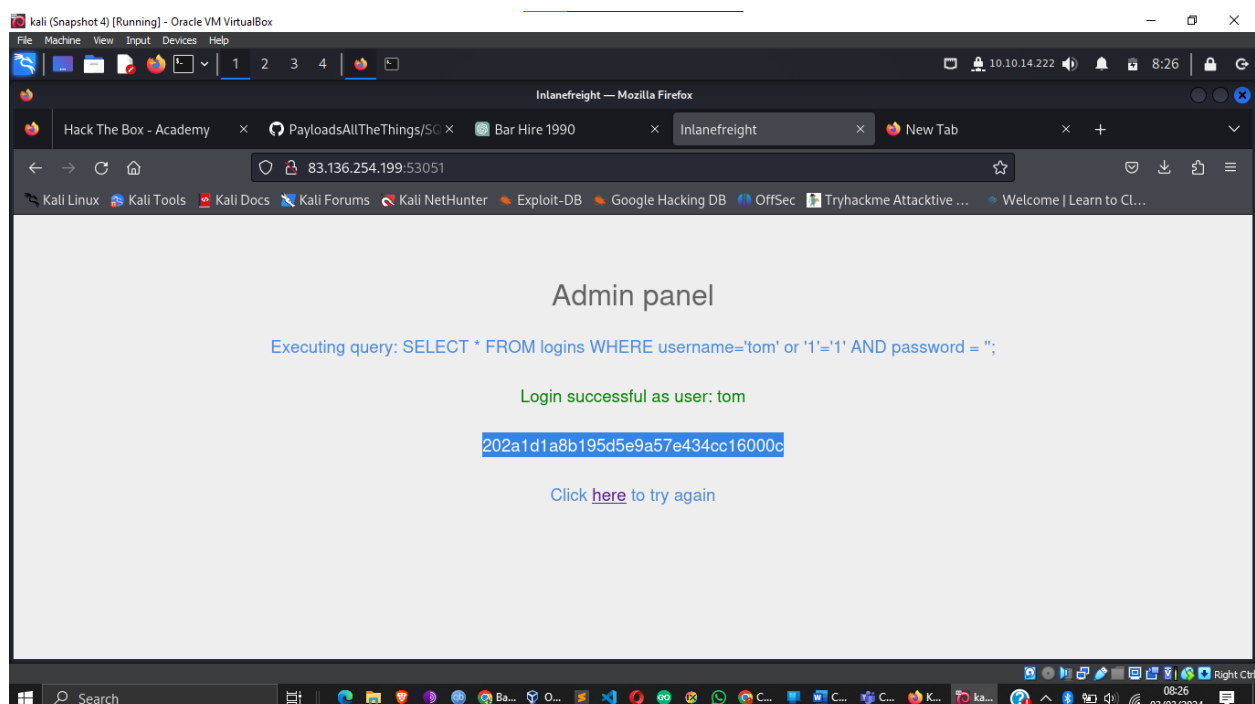
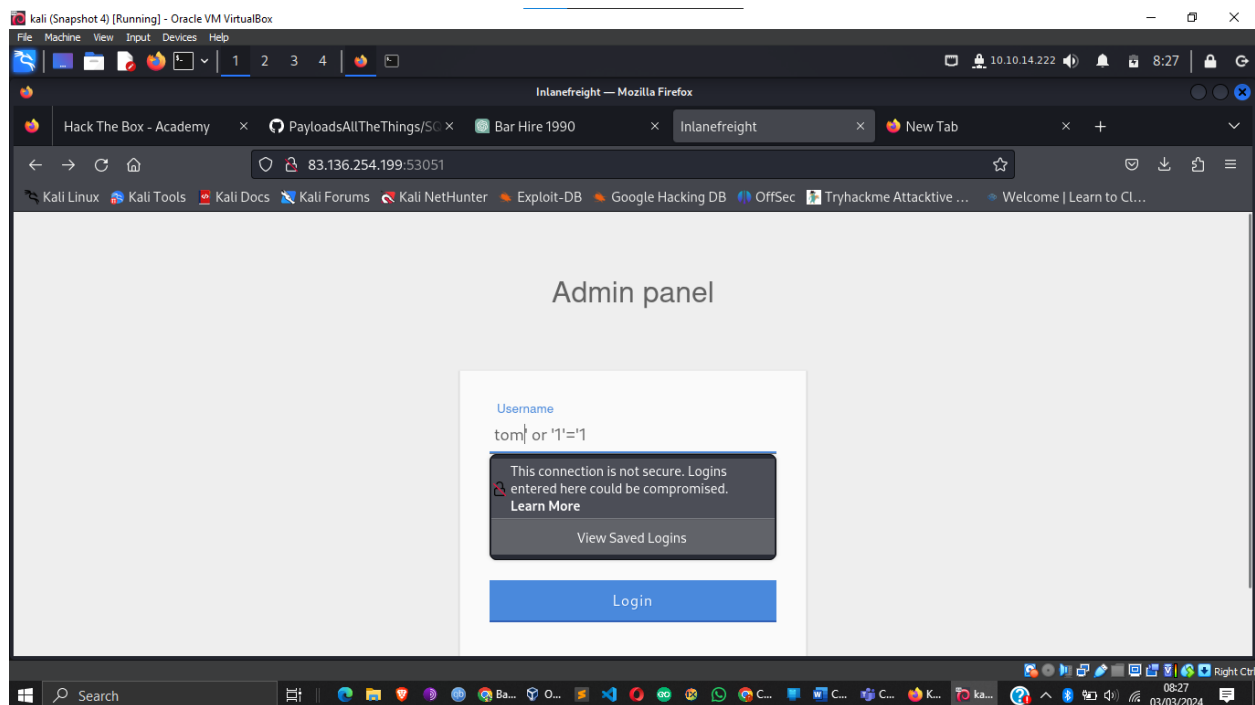
Subverting Query Logic

Manipulating SQL queries can lead to unauthorized access or data retrieval.

- **Subverting Query Logic Example**

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

Answer: 202a1d1a8b195d5e9a57e434cc16000c



Using Comments

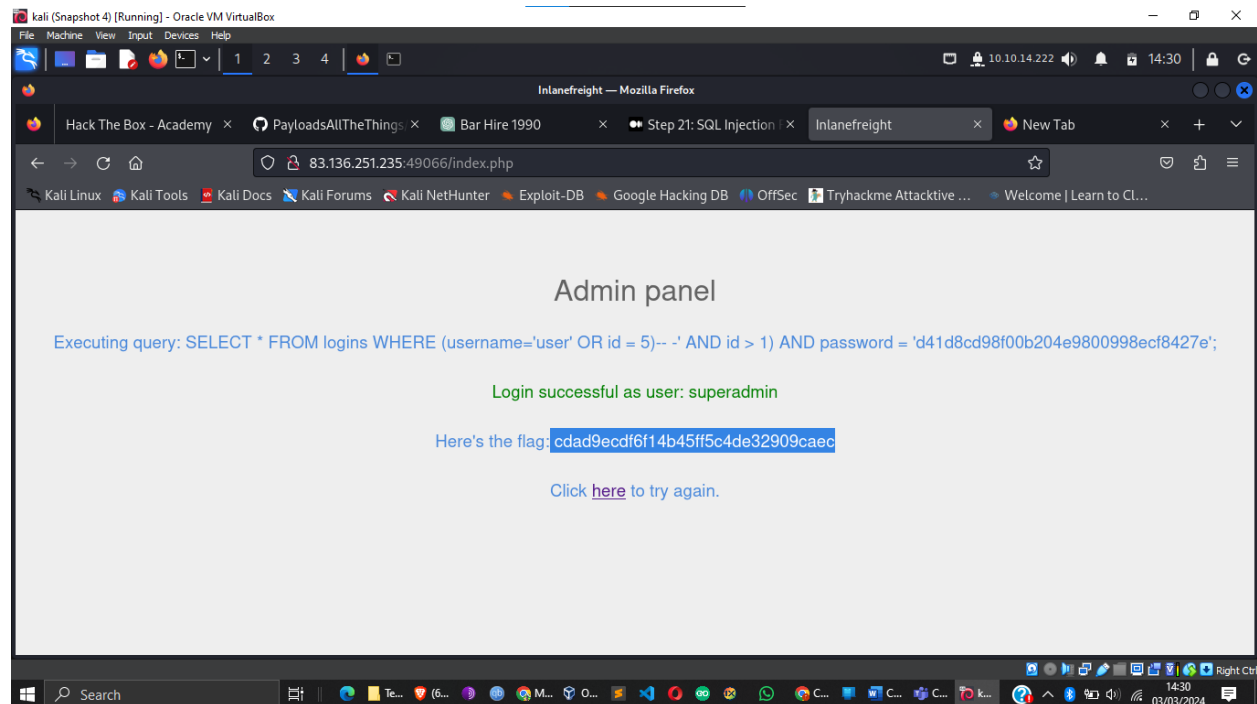
SQL comments can be utilized to manipulate SQL queries and bypass security mechanisms.

- **Using Comments Example**

Question: Login as the user with the id 5 to get the flag.

Answer: cdad9ecdf6f14b45ff5c4de32909caec

User' OR id =5)-- -



Union Clause and Injection

The UNION SQL operator is used to combine the results of two or more SELECT statements into a single result set.

- **Union Clause and Injection Example**

Question: 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.

Answer: 663

```
kali (Snapshot 4) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~/Downloads x kali@kali: ~ x
+-----+
| employees |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.245 sec)

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

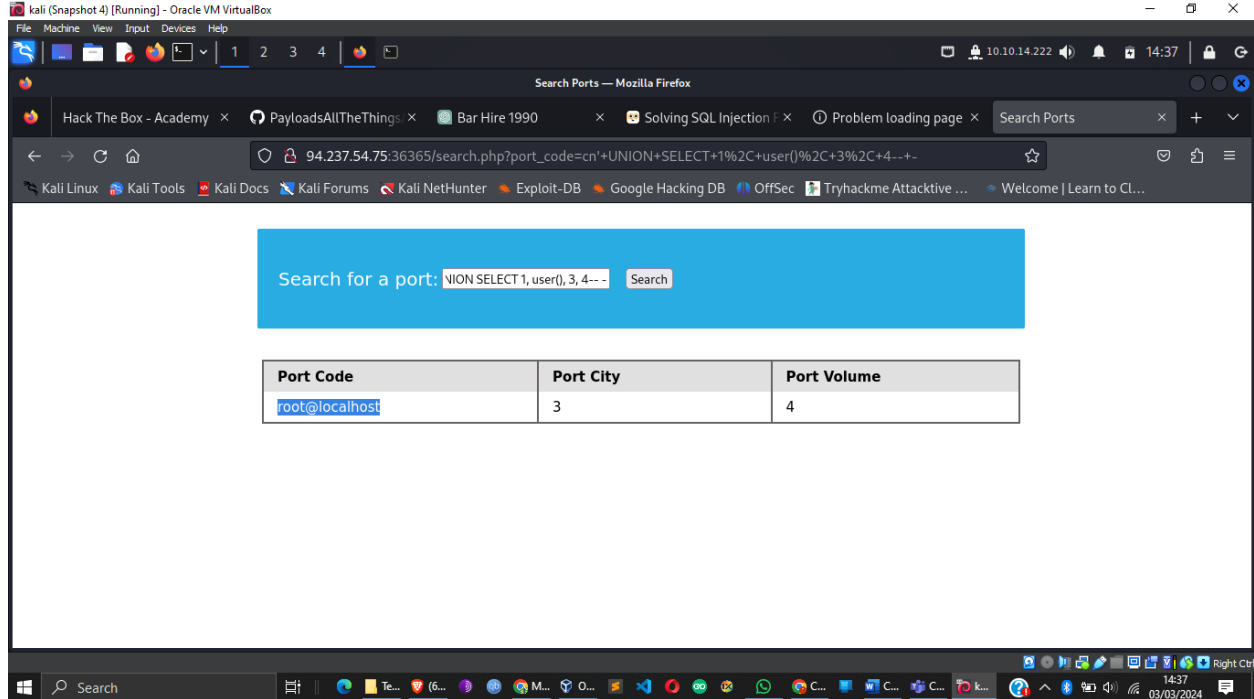
Database changed
MariaDB [employees]> SELECT COUNT(*) AS total_records
  FROM
  (
    SELECT * FROM employees
    UNION
    SELECT * FROM departments
  ) AS combined_tables;
ERROR 1222 (21000): The used SELECT statements have a different number of columns
MariaDB [employees]> select * from employees UNION select dept_no, dept_name, 3, 4, 5, 6 from departments;
+-----+
| emp_no | birth_date | first_name | last_name | gender | hire_date |
+-----+
| 10001 | 1953-09-02 | Georgi | Facello | M | 1986-06-26 |
| 10002 | 1952-12-03 | Vivian | Billawala | F | 1986-12-11 |
| 10003 | 1959-06-16 | Temple | Lukaszewicz | M | 1992-07-04 |
| 10004 | 1956-11-06 | Masanao | Rahimi | M | 1986-12-16 |
| 10005 | 1962-12-11 | Sanjay | Danlos | M | 1985-08-01 |
| 10006 | 1963-12-30 | Marie | Stafford | M | 1988-10-10 |
| 10007 | 1959-06-28 | Huai | Motley | M | 1991-04-04 |
+-----+
```

```
kali (Snapshot 4) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~/Downloads x kali@kali: ~ x
+-----+
| 10637 | 1954-08-25 | Mohammed | Pleszkun | M | 1986-02-21 |
| 10638 | 1955-08-29 | Uri | Juneja | F | 1989-08-28 |
| 10639 | 1959-08-31 | Kaijung | Rodham | M | 1985-09-11 |
| 10640 | 1964-12-26 | Gila | Lukaszewicz | M | 1997-02-11 |
| 10641 | 1952-07-22 | Nathan | Ranta | F | 1985-08-11 |
| 10642 | 1961-09-05 | Rimli | Dusink | F | 1998-09-20 |
| 10643 | 1962-09-28 | Bangqing | Kleiser | F | 1986-06-06 |
| 10644 | 1954-05-26 | Keiichiro | Lindqvist | M | 1993-10-28 |
| 10645 | 1963-11-03 | Khaled | Kohling | M | 1985-10-10 |
| 10646 | 1962-02-26 | Pohua | Siehman | F | 1989-01-12 |
| 10647 | 1960-10-12 | Siamak | Salverda | F | 1987-05-10 |
| 10648 | 1963-06-04 | DeForest | Mullainathan | M | 1997-04-07 |
| 10649 | 1952-02-26 | Navin | Argence | F | 1990-04-24 |
| 10650 | 1958-09-24 | Dekang | Lichtner | F | 1993-01-12 |
| 10651 | 1953-03-07 | Zito | Baaz | M | 1990-09-27 |
| 10652 | 1961-08-03 | Berhard | Lenart | M | 1986-04-21 |
| 10653 | 1956-09-05 | Patricia | Breugel | M | 1993-10-13 |
| 10654 | 1958-05-01 | Sachin | Tsukuda | M | 1997-11-30 |
| d009 | Customer Service | 3 | 4 | 5 | 6 |
| d005 | Development | 3 | 4 | 5 | 6 |
| d002 | Finance | 3 | 4 | 5 | 6 |
| d003 | Human Resources | 3 | 4 | 5 | 6 |
| d001 | Marketing | 3 | 4 | 5 | 6 |
| d004 | Production | 3 | 4 | 5 | 6 |
| d006 | Quality Management | 3 | 4 | 5 | 6 |
| d008 | Research | 3 | 4 | 5 | 6 |
| d007 | Sales | 3 | 4 | 5 | 6 |
+-----+
663 rows in set (0.577 sec)

MariaDB [employees]> select COUNT(*) from employees UNION select dept_no, dept_name, 3, 4, 5, 6 from departments;
ERROR 1222 (21000): The used SELECT statements have a different number of columns
MariaDB [employees]>
```

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

Answer: root@localhost



Exploitation

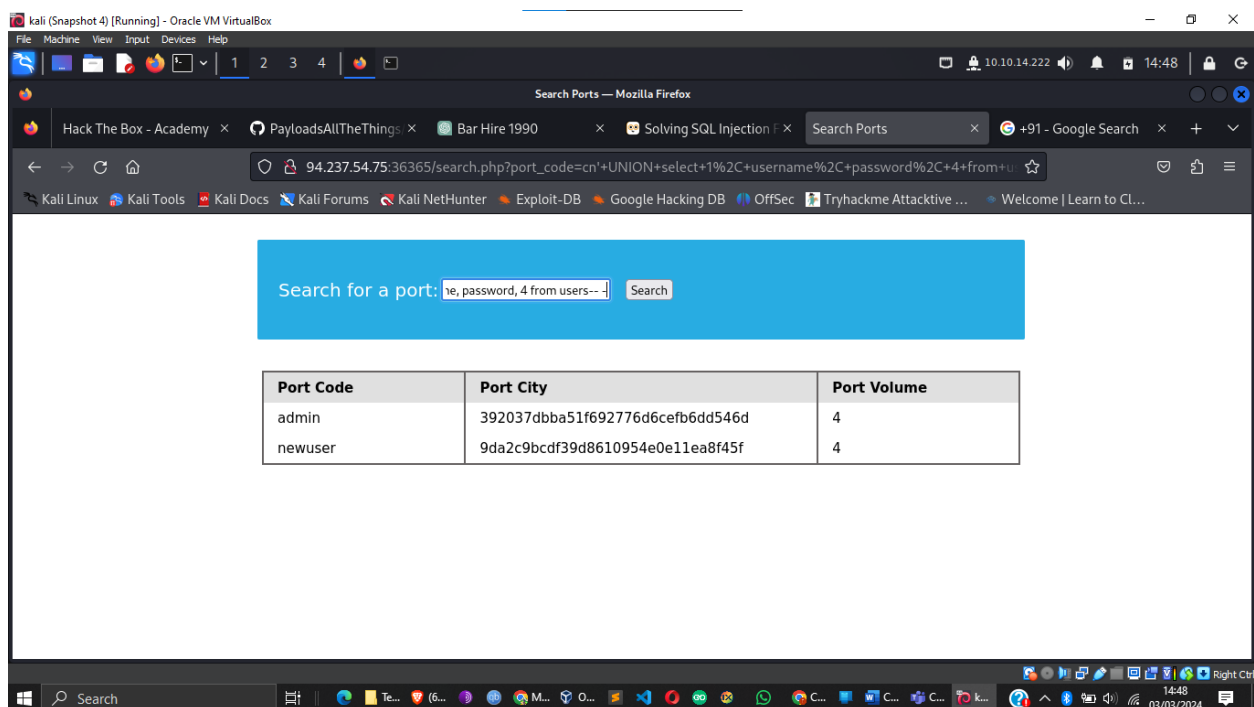
Database Enumeration

Identifying and extracting information from databases can reveal sensitive data or system details.

- **Database Enumeration Example**

Question: What is the password hash for 'newuser' stored in the 'users' table in the 'ilfreight' database?

Answer: 9da2c9bcd39d8610954e0e11ea8f45f



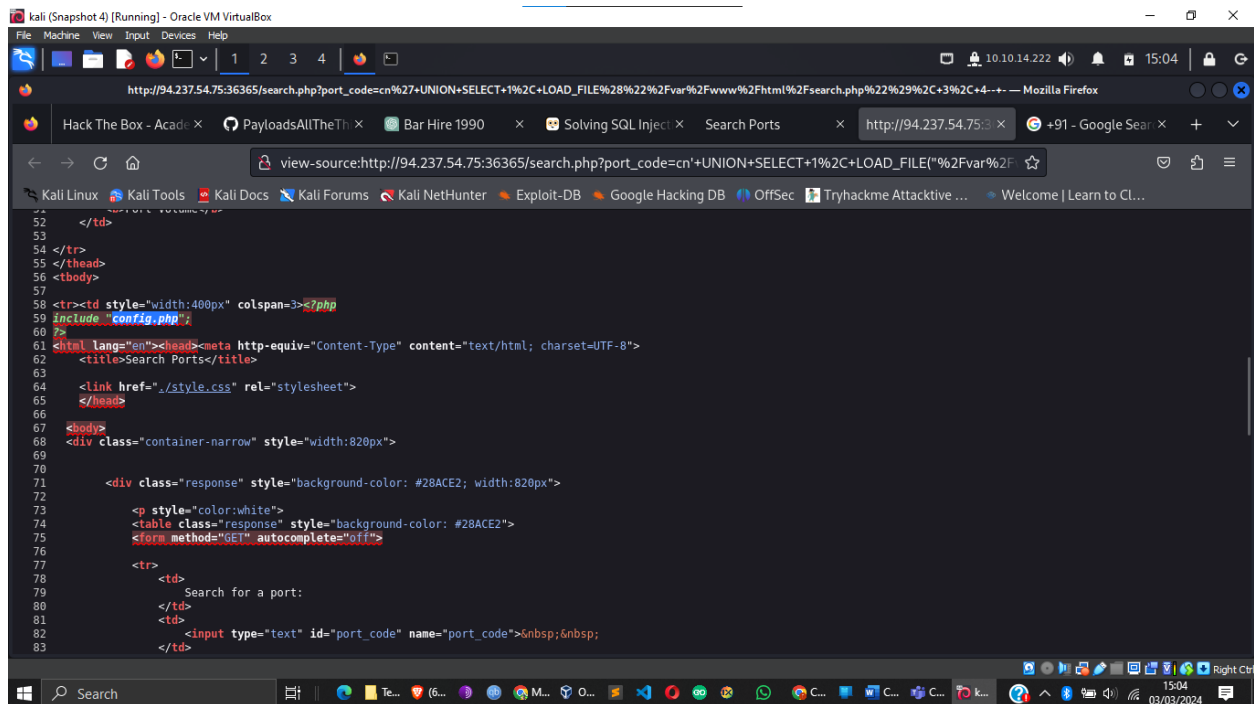
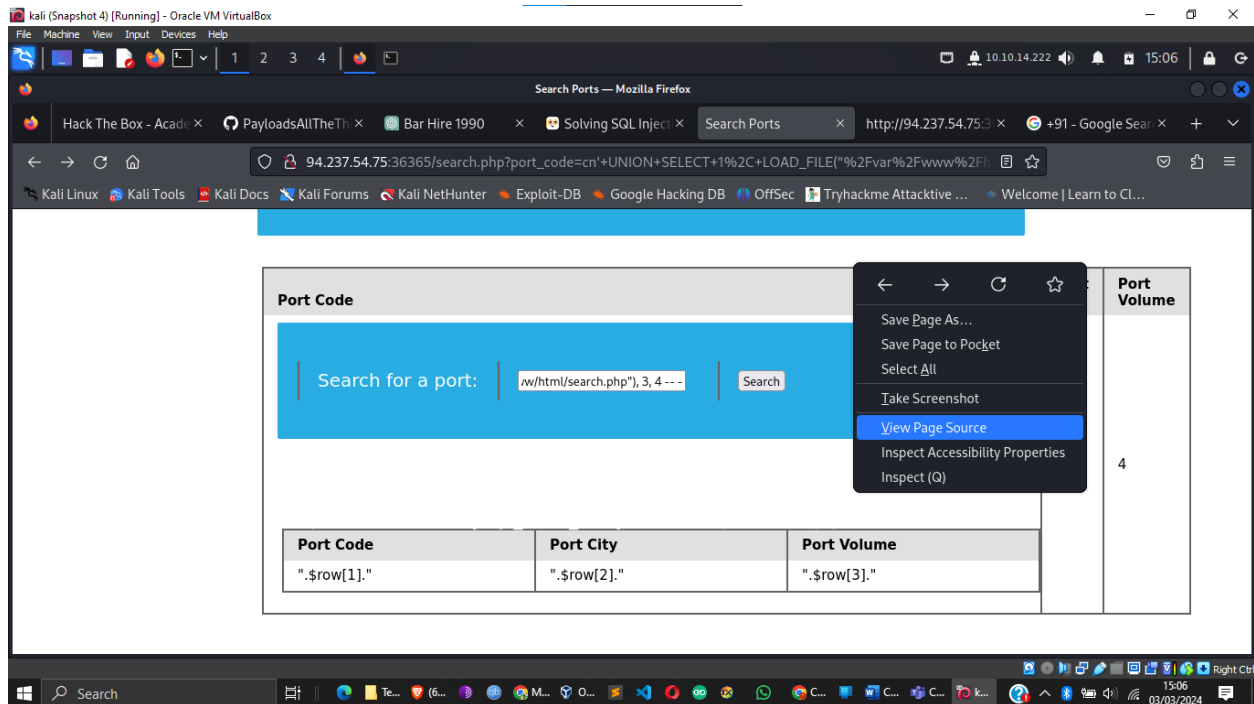
Reading Files

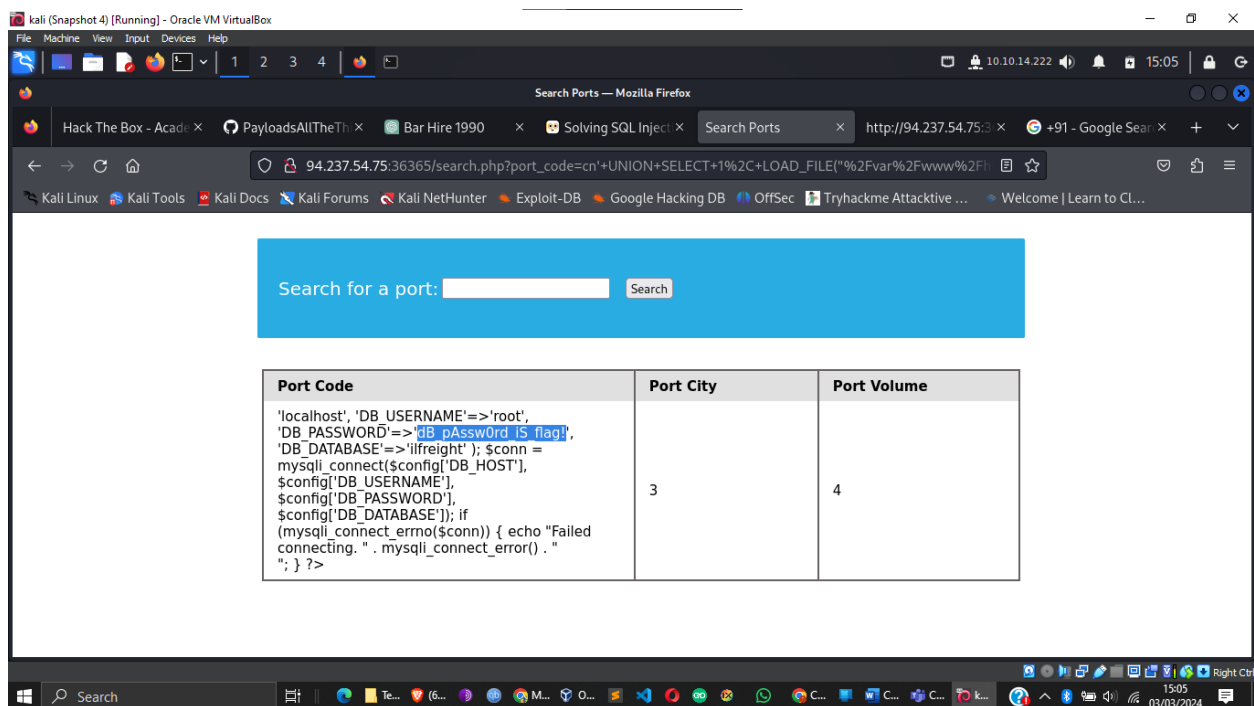
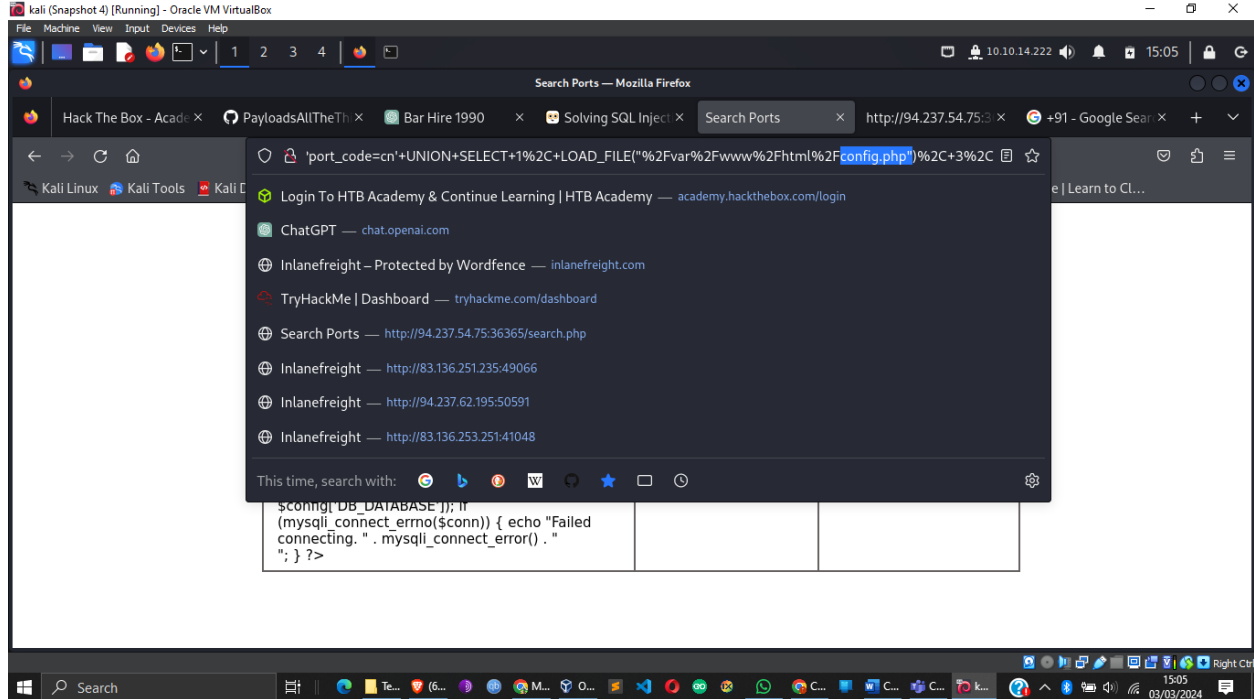
Accessing files through SQL Injection can expose critical information or system vulnerabilities.

- **Reading Files Example**

Question: Check the imported page to obtain the database password.

Answer: dB_pAssw0rd_iS_flag!





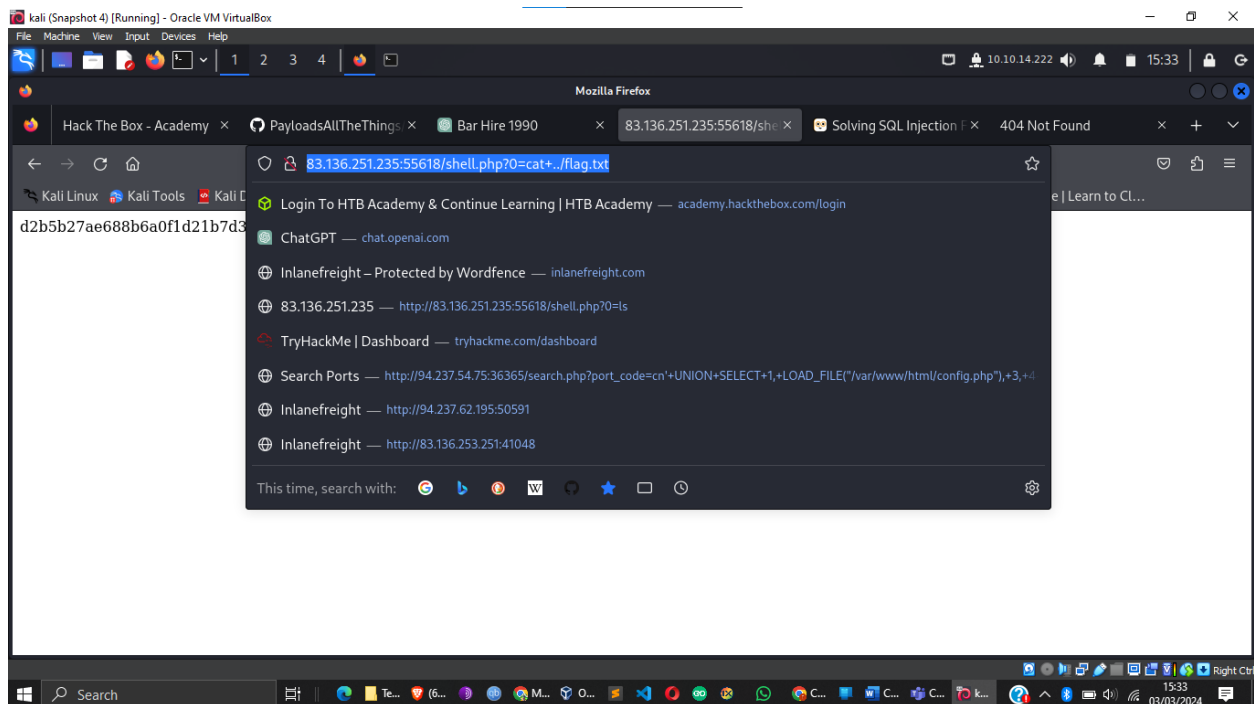
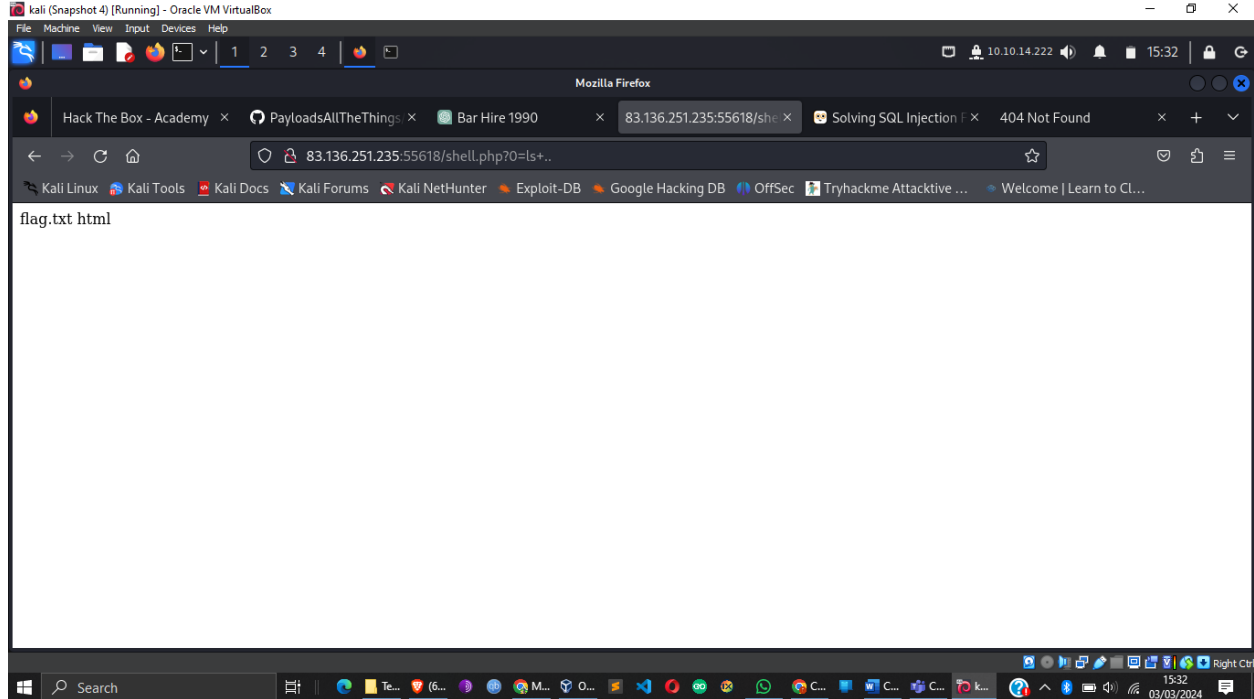
Writing Files

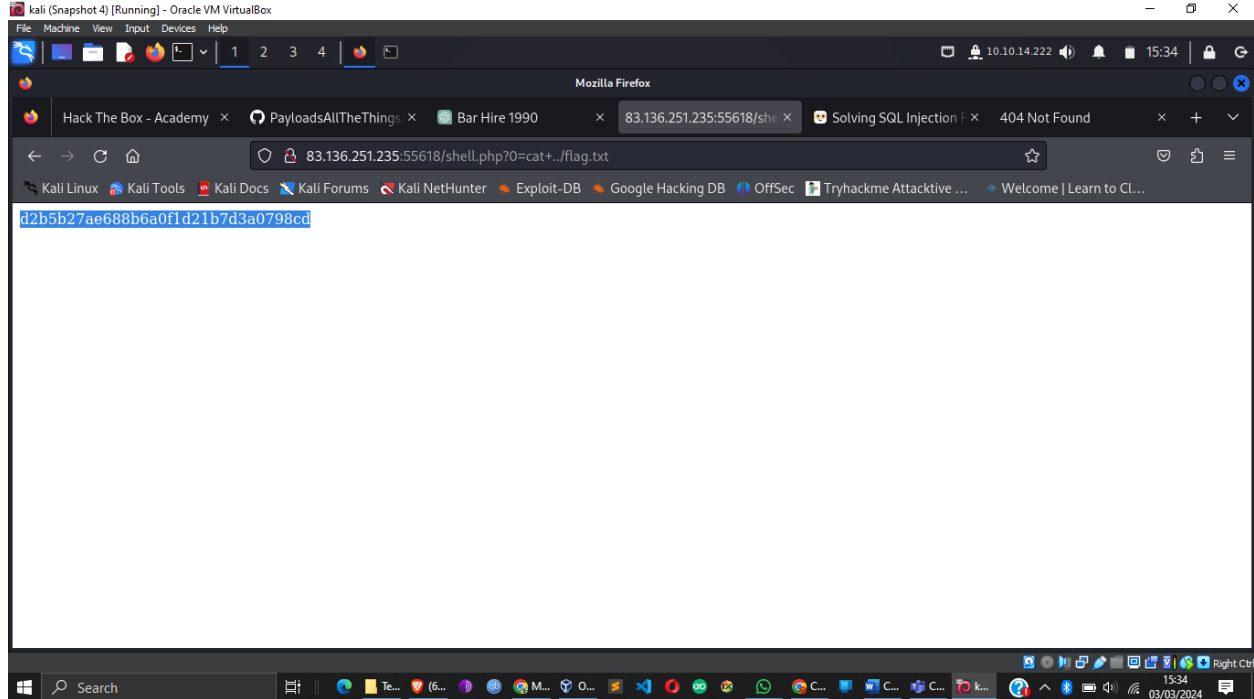
SQL Injection can be used to write files, potentially leading to persistent access or data tampering.

- **Writing Files Example**

Question: Find the flag by using a webshell.

Answer: d2b5b27ae688b6a0f1d21b7d3a0798cd





Mitigations

Mitigating SQL Injections involves validating input, using prepared statements, implementing stored procedures, and regularly updating systems to patch vulnerabilities.

Closing it Out

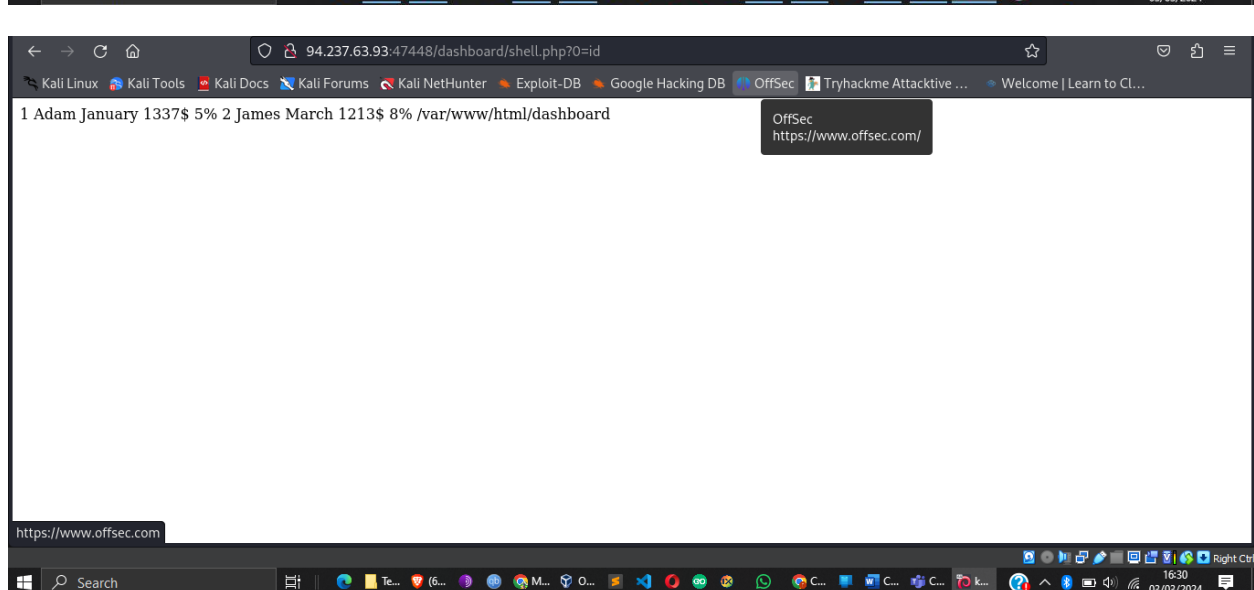
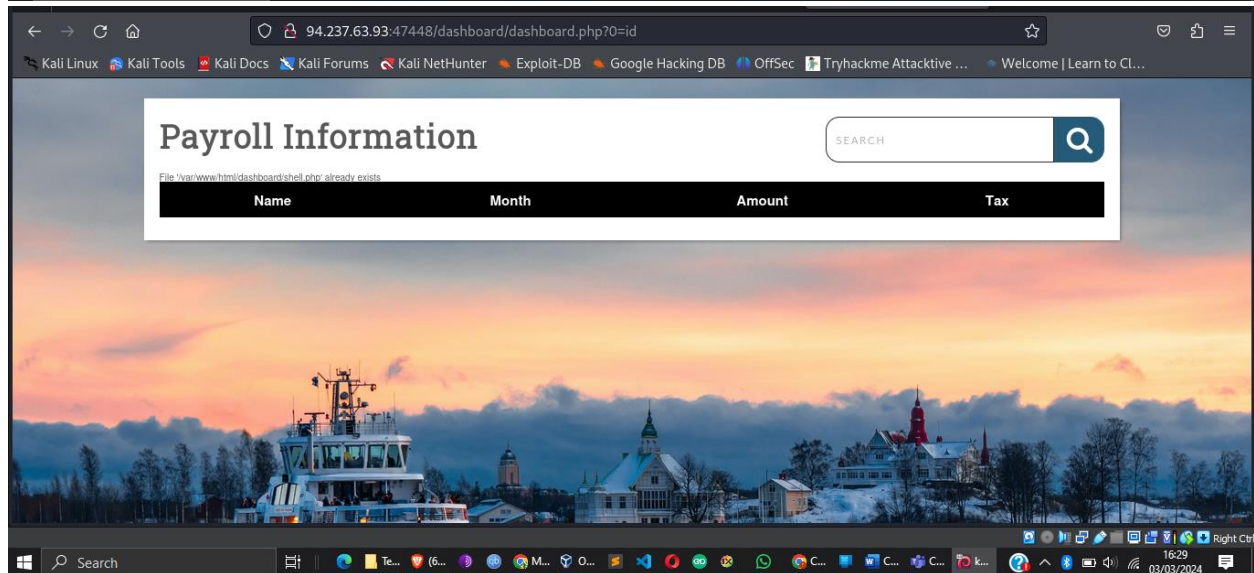
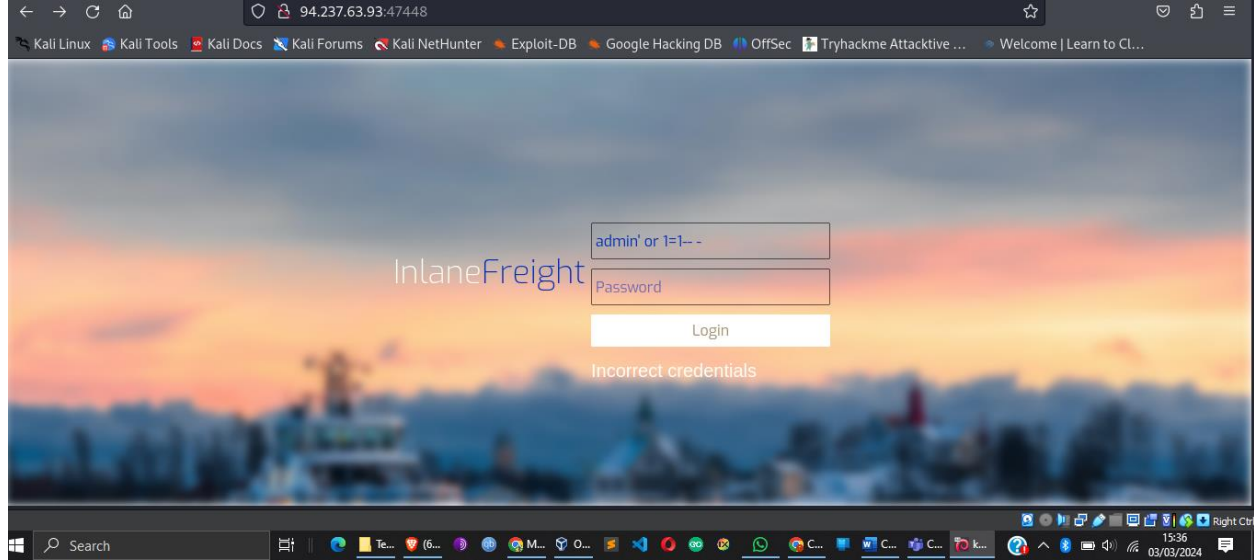
Skills Assessment - SQL Injection Fundamentals

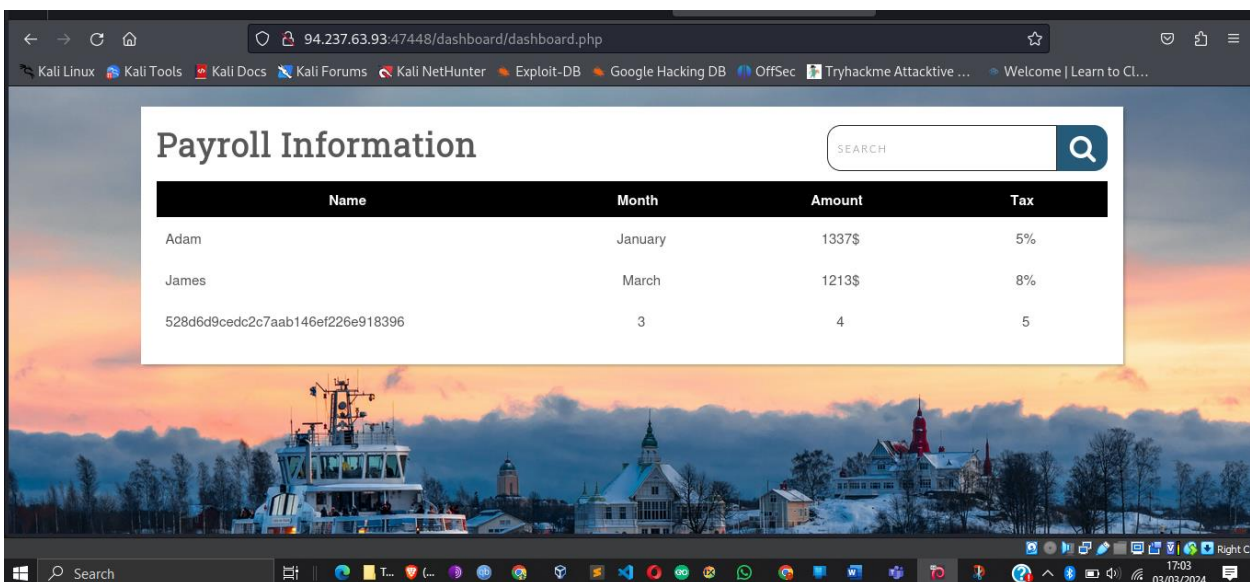
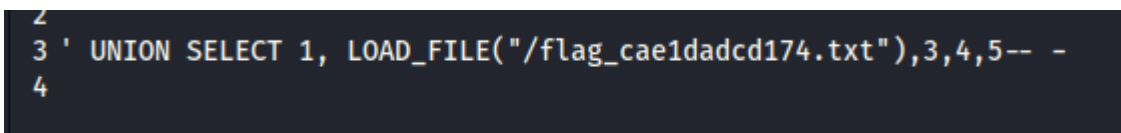
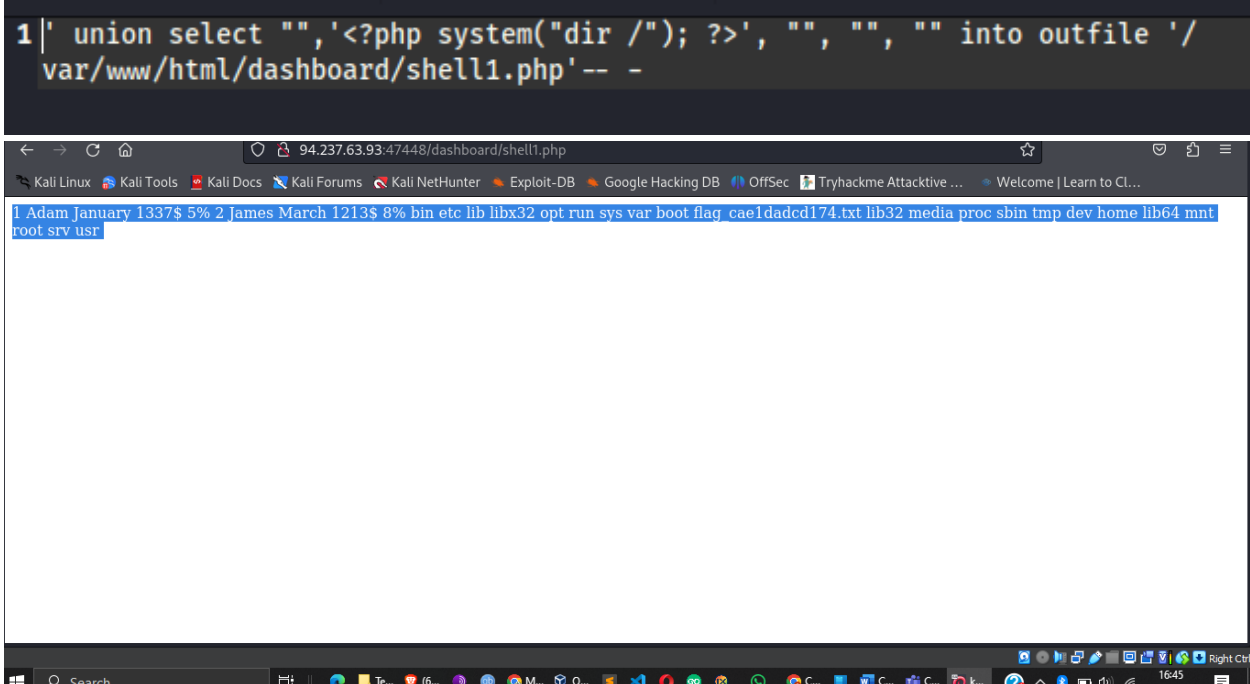
A practical assessment to validate the understanding and application of SQL Injection knowledge in real-world scenarios.

- **Skills Assessment Example**

Question: 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.

Answer: 528d6d9cedc2c7aab146ef226e918396





Conclusion

This report provides a comprehensive overview of SQL Injection, detailing fundamental concepts, practical examples, and mitigation strategies to foster a deeper understanding of this critical security vulnerability.



SQL Injection Fundamentals



Congratulations **Damiano254**, you have completed this module!

Module: **SQL Injection Fundamentals**

Difficulty: **Medium**

Exercises Completed: **12 /12**

Completed at: 03 Mar 2024