

New issue



Phpgurukul Online DJ Booking Management System V1.0 /admin/booking-search.php SQL injection #2

Open



MoshangChunfeng opened 2 weeks ago



NAME OF AFFECTED PRODUCT(S)

- Online DJ Booking Management System

Vendor Homepage

- <https://phpgurukul.com/online-dj-booking-management-system-using-php-and-mysql/>

AFFECTED AND/OR FIXED VERSION(S)

submitter

- YeyukongXiaodengli

Vulnerable File

- /admin/booking-search.php

VERSION(S)

- V1.0

Software Link

- https://phpgurukul.com/?sdm_process_download=1&download_id=11402

PROBLEM TYPE

Vulnerability Type

- SQL injection

Root Cause

- A SQL injection vulnerability was found in the '/admin/booking-search.php' file of the 'Online DJ Booking Management System' project. The reason for this issue is that attackers inject malicious code from the parameter 'searchdata' 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 "Online DJ Booking Management System", I discovered a critical SQL injection vulnerability in the "/admin/booking-search.php" file. This vulnerability stems from insufficient user input validation of the 'searchdata' 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 Location:

- 'searchdata' parameter

Payload:

Parameter: searchdata (POST)
Type: time-based blind



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.

Sign up for free

 to join this conversation on **GitHub**. Already have an account? [Sign in to comment](#)

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

