# SQL injection vulnerability in “/music/index.php?page=view_playlist” in Kashipara Music Management System v1.0 allows attacker to execute arbitrary SQL commands via the "id" parameter.

**Affected Project:** Kashipara (https://www.kashipara.com/)

**Product Official Website URL**: Music Management System v1.0 (https://www.kashipara.com/project/php/12978/music-management-system-in-php-php-project-source-code)

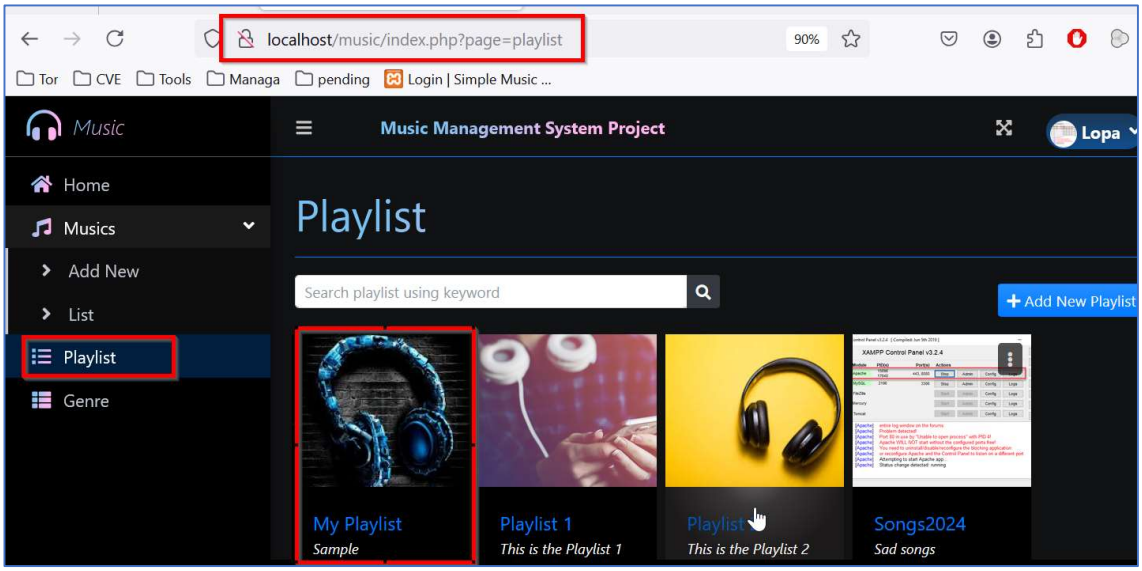
**Version:** 1.0

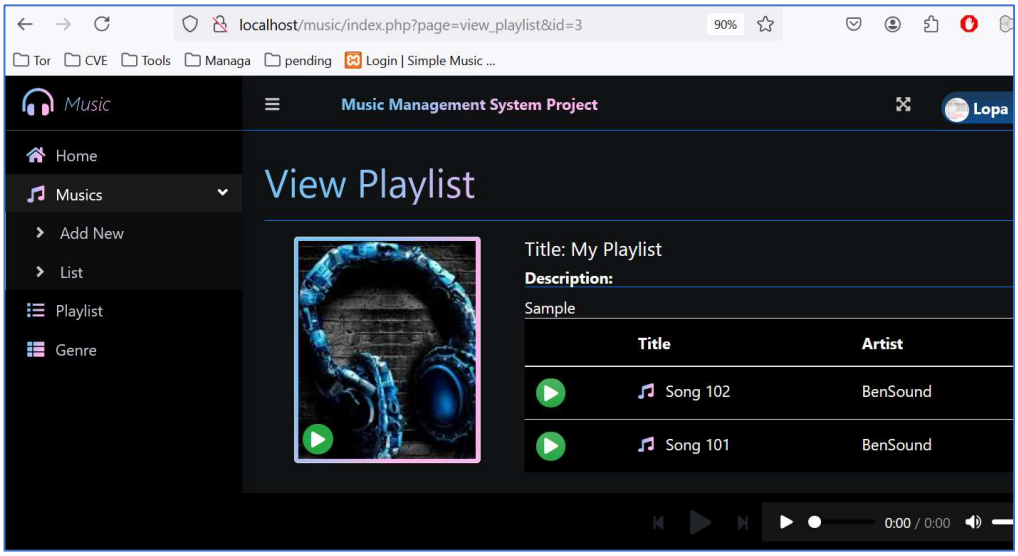
**Affected Components:**

- **Affected Code File:** /music/index.php?page=view_playlist
- **Affected Parameter:** "id" HTTP POST request parameter

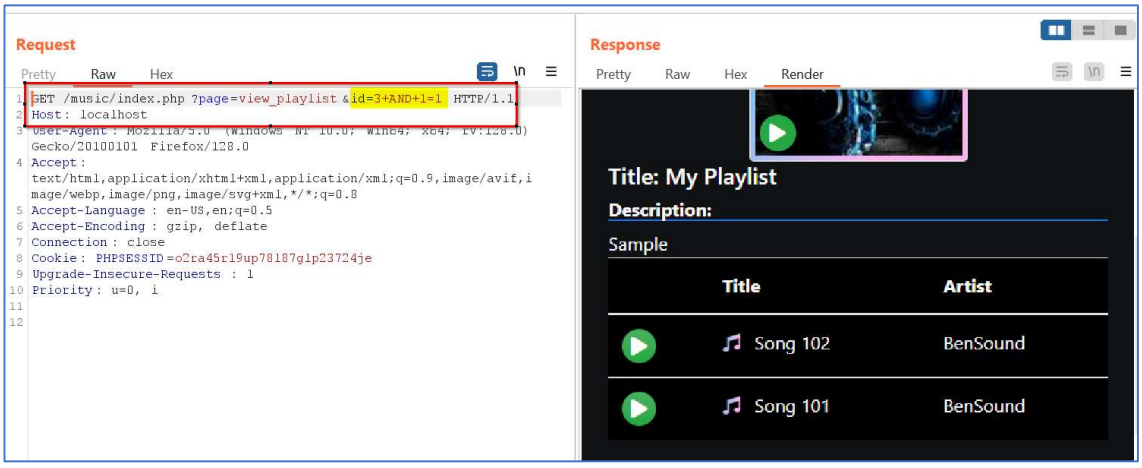
**Steps:**

1. Login into the Music Management System v1.0 application. (http://localhost/music/login.php).
2. Navigate to “Playlist” menu. Click on any one of the Playlist names.

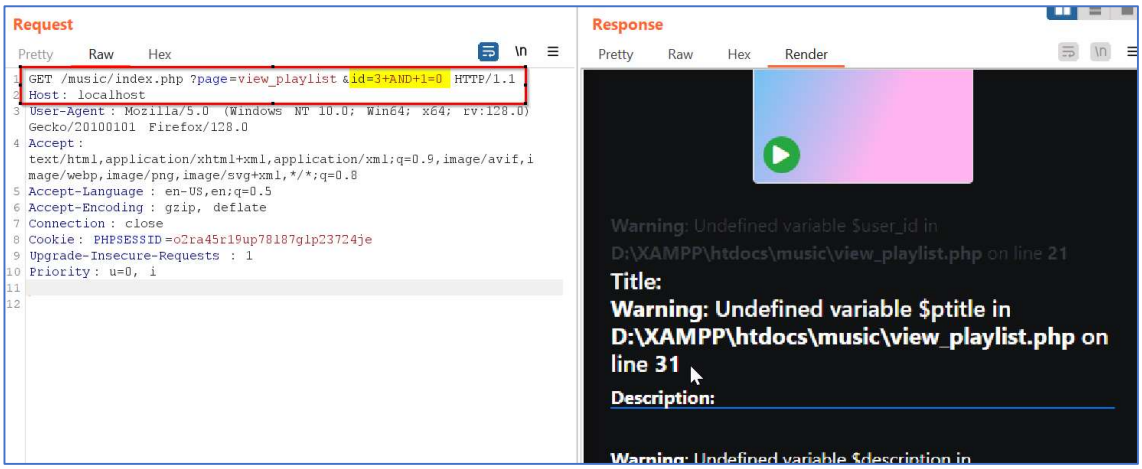




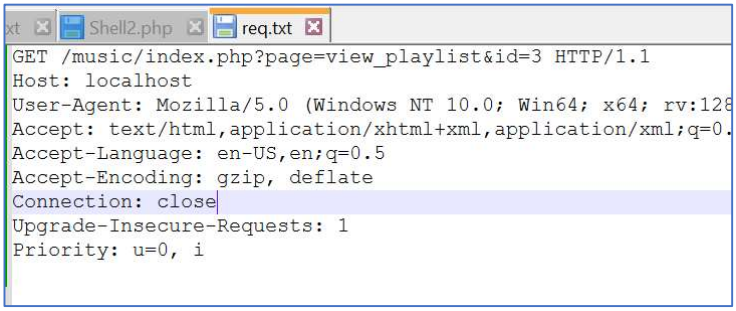

3. In the “id” parameter, insert SQL command ***+AND+1=1***. The response remains same.

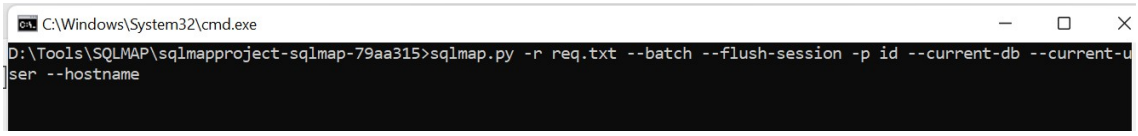



4. In the “id” parameter, insert SQL command ***+AND+1=0***. The response is different. The response does not have any song entry.

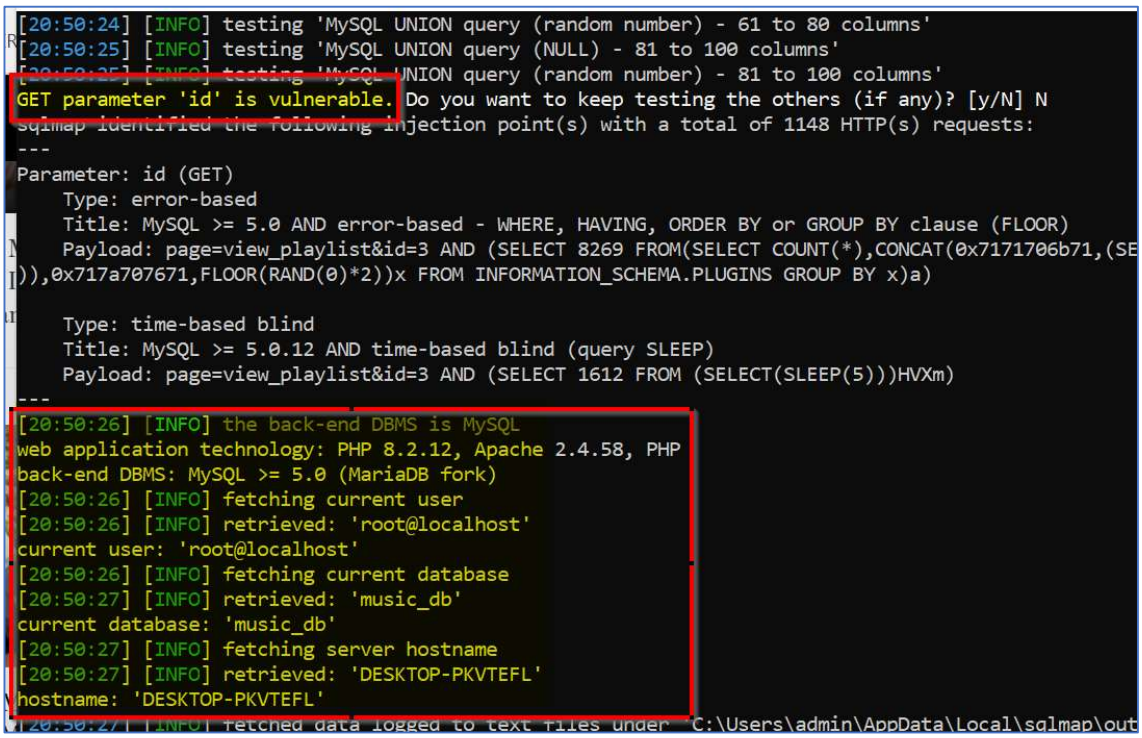


5. This indicates the "**id**" request parameter is vulnerable to SQL injection.
6. We will run SQLMAP against the Login request. Command: ***sqlmap.py -r req.txt --batch --flush-session -p id --current-db --current-user --hostname***




7. SQLMAP identifies parameter "**id**" as vulnerable. Also, SQLMAP successfully lists out the database, current user and hostname.




**Solution/Good Reads:**

User parameterized SQL queries instead of the dynamic SQL queries.

- https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html