SQL Injection was found in the /lms/admin/edit_content.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 title, content parameter in a POST HTTP request.

- **Official Website URL**

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

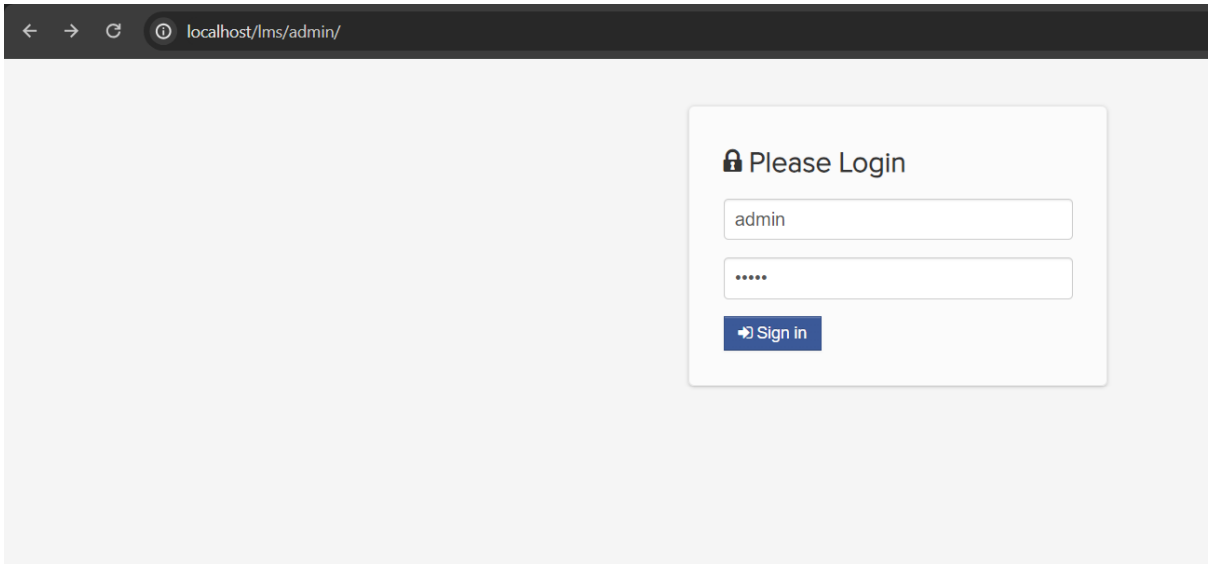
- **Affected Product Name**

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

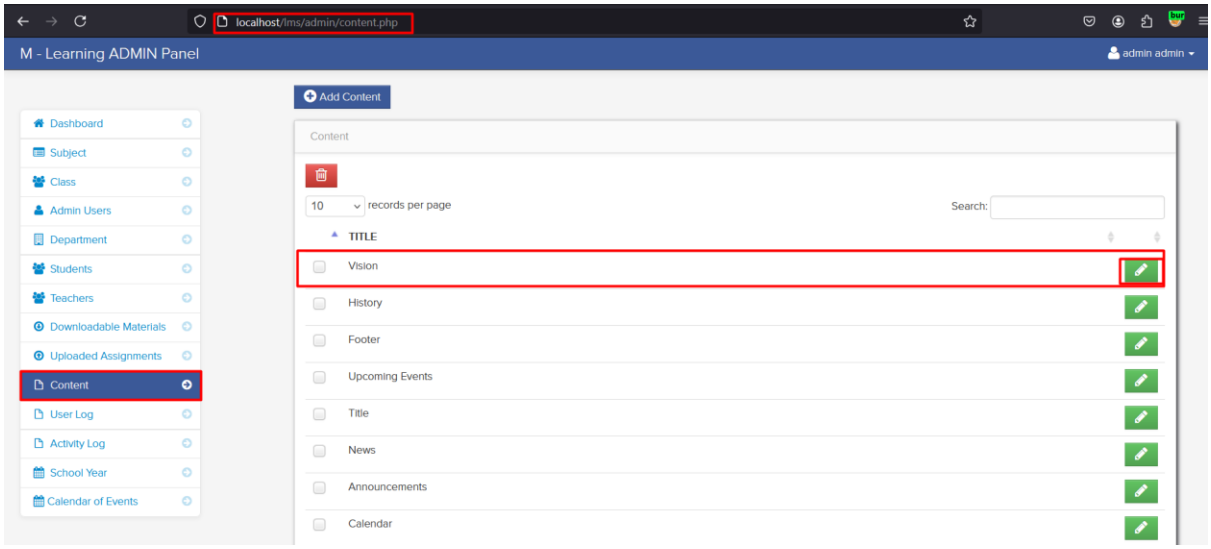
Affected Vendor	kashipara
Affected Code File	/lms/admin/edit_content.php
Affected Parameter	title, content
Method	POST
Type	time-based blind
Version	V1.0

Steps to Reproduce:

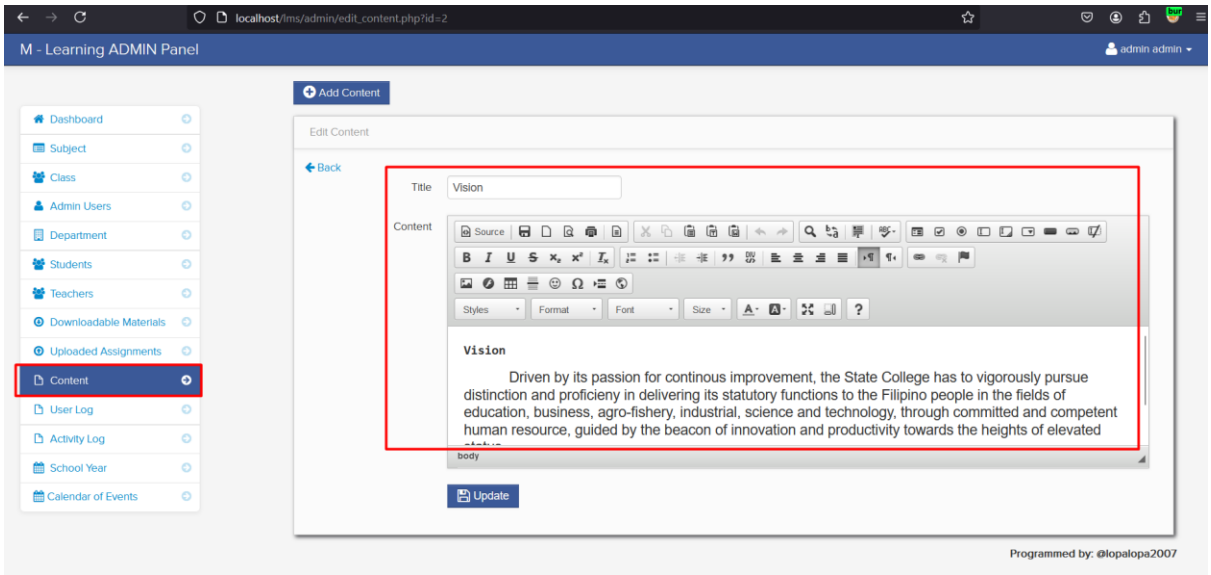
Step 1: Visit to admin login page and login with admin credential.



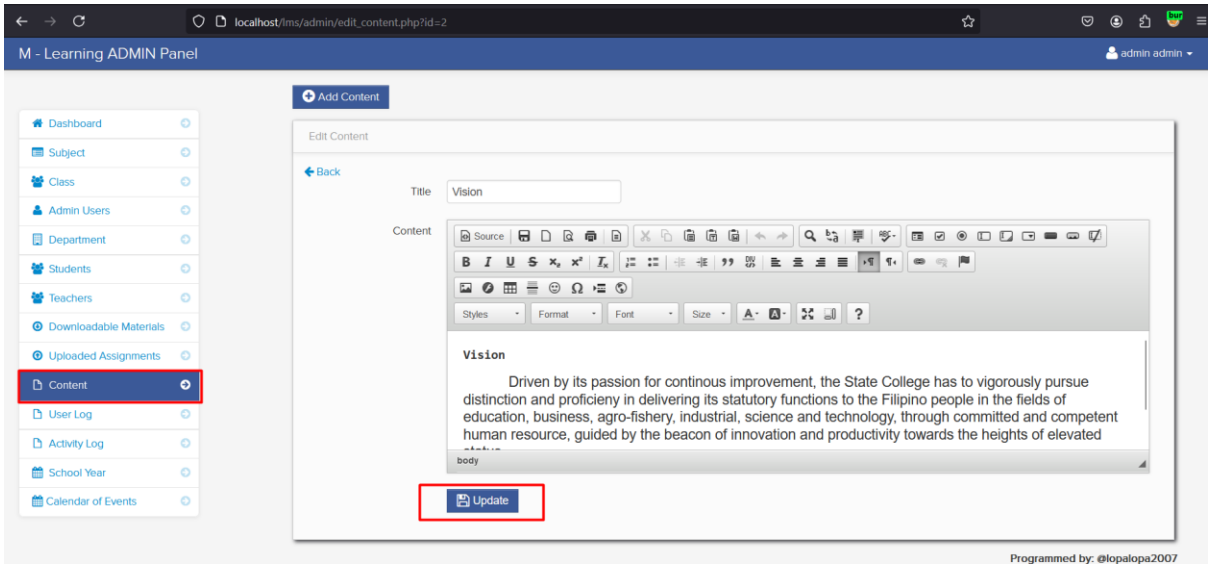
Step 2: Navigate the ‘Content’ page click edit on any content from list.



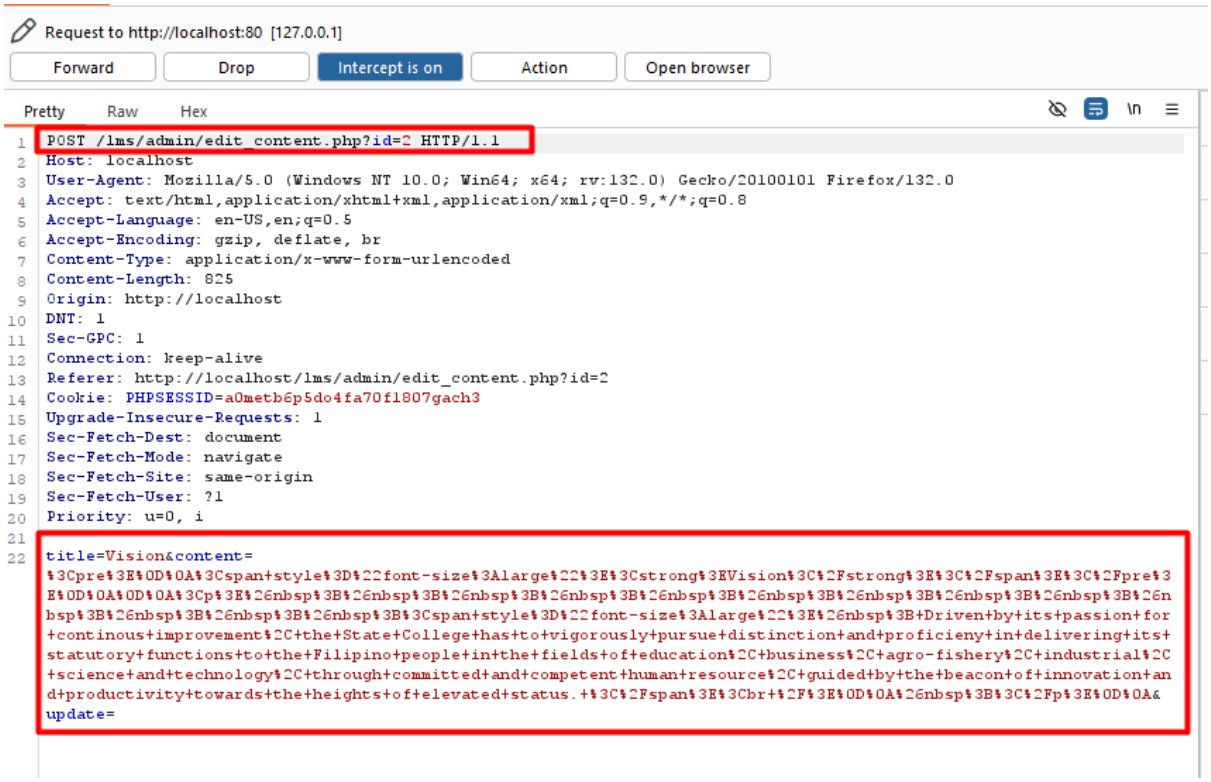
Step 3: Leave the content as it is.



Step 4: Now enable intercept in bupsuite and click ‘Update’ button.

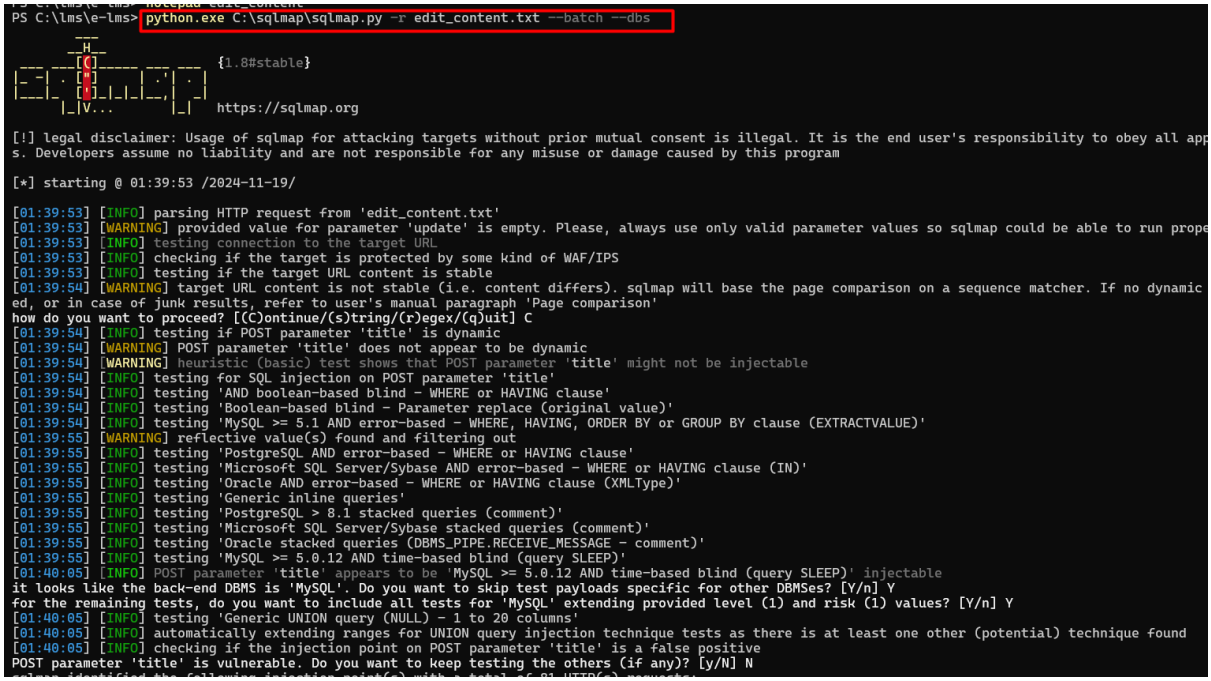


Step 5: Save the burpsuite request in a file.

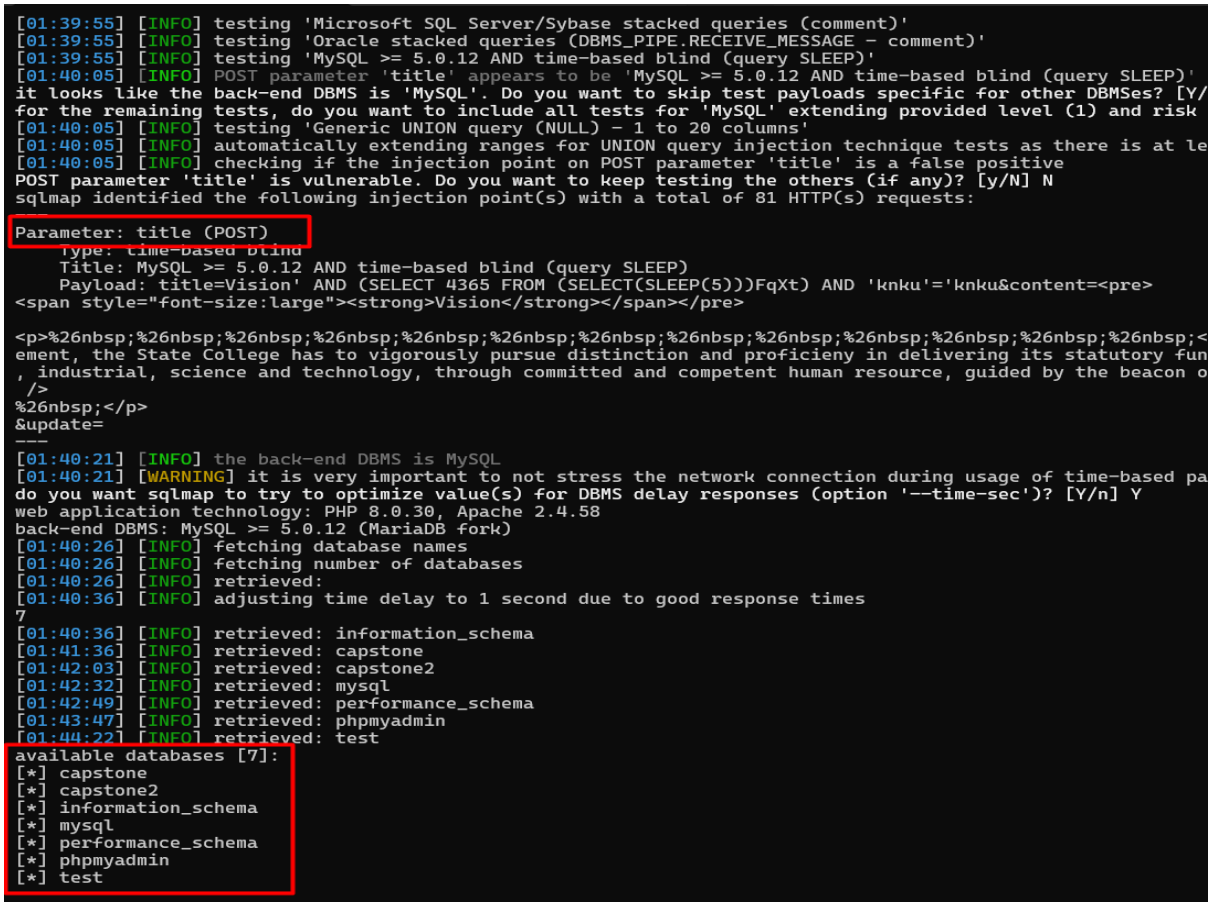


Step 6: Now run the sqlmap command against request saved in file.

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



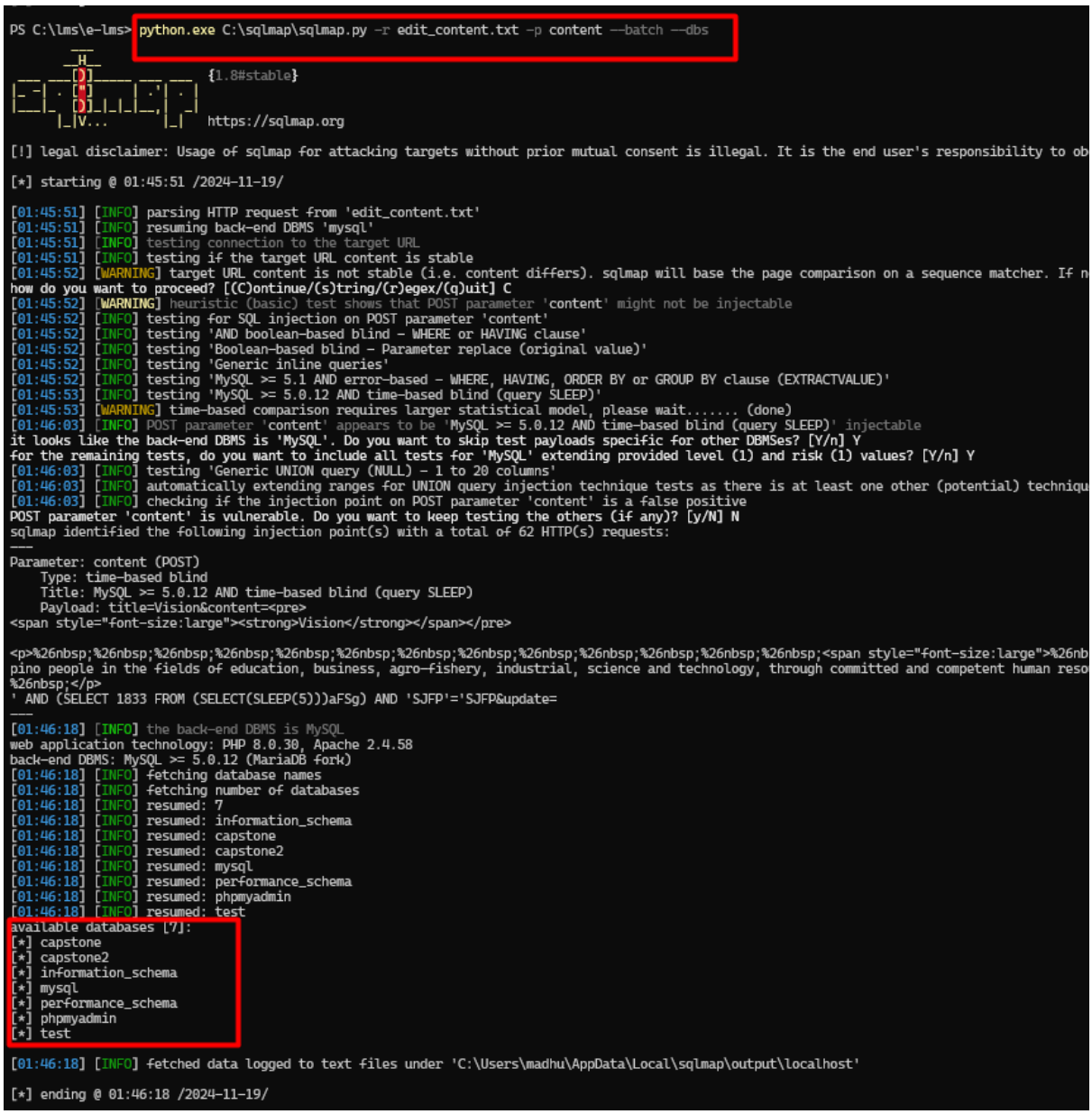
Step 7: Now notice that 'title' parameter is detected vulnerable and all database is successfully retrieved.



Parameter: content

Step 8: Run the sqlmap against 'content' parameter by using switch -p. Notice that 'content' parameter is detected vulnerable and all database is successfully retrieved.

- python.exe C:\sqlmap\sqlmap.py -r edit_content.txt -p content --batch --dbs



Mitigation/recommendations

- https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html
- https://portswigger.net/web-security/sql-injection#how-to-prevent-sql-injection