SQL Injection was found in the /lms/admin/add_content.php page 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 and 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

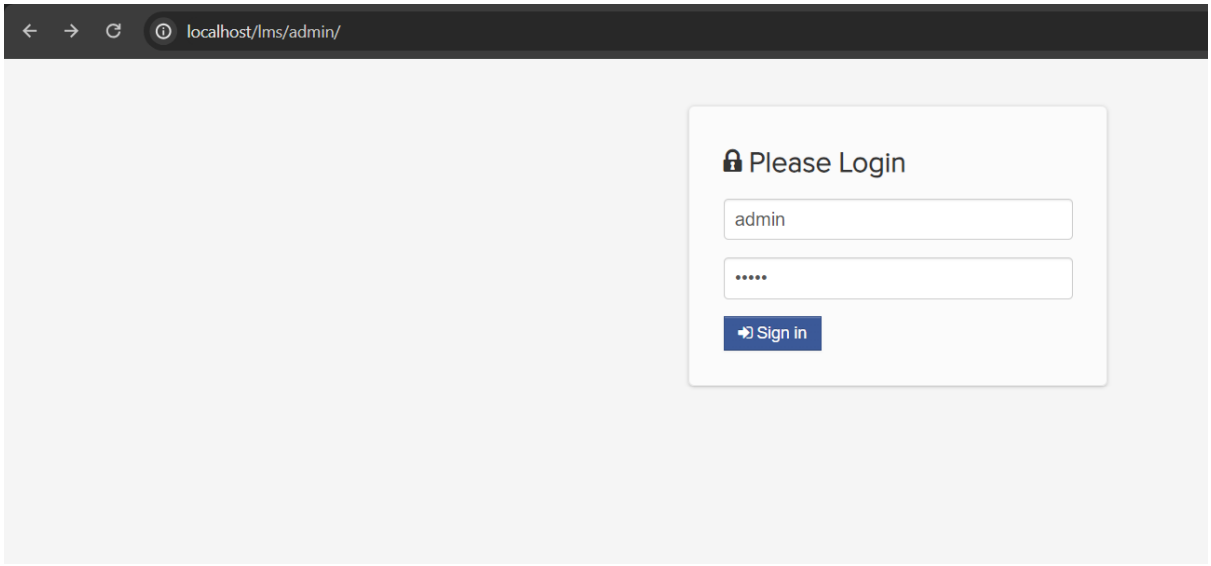
- **Affected Product Name**

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

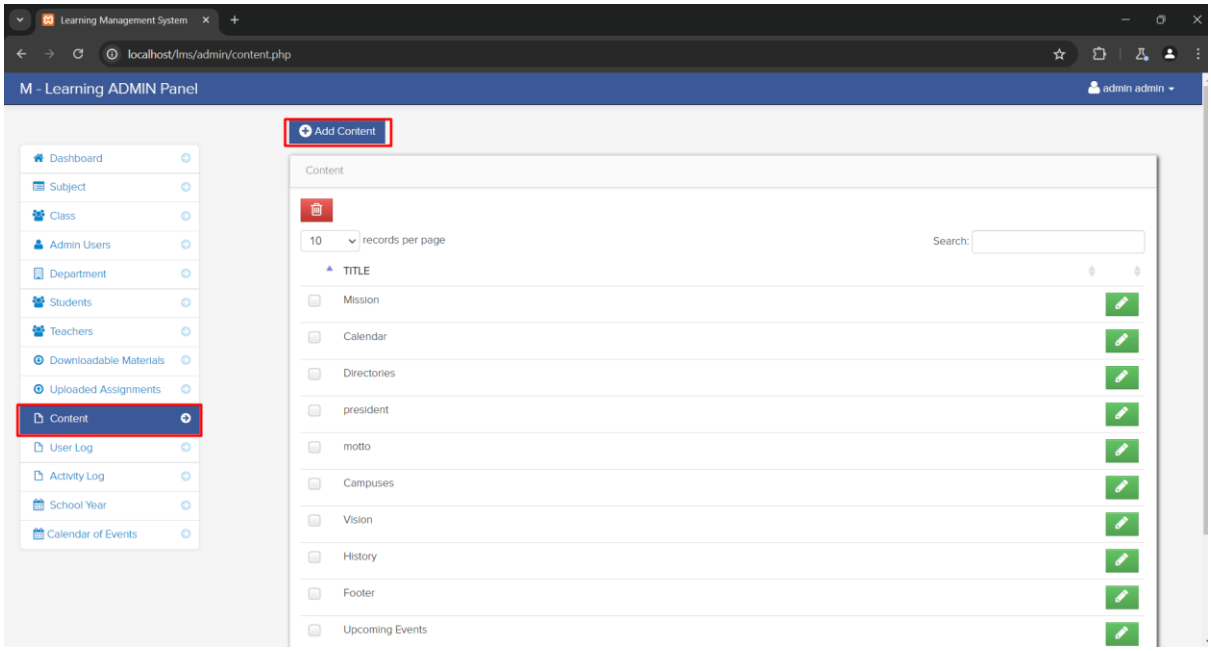
Affected Vendor	kashipara
Affected Code File	/lms/admin/add_content.php
Affected Parameter	title, content
Method	POST
Type	time-based blind
Version	V1.0

Steps to Reproduce:

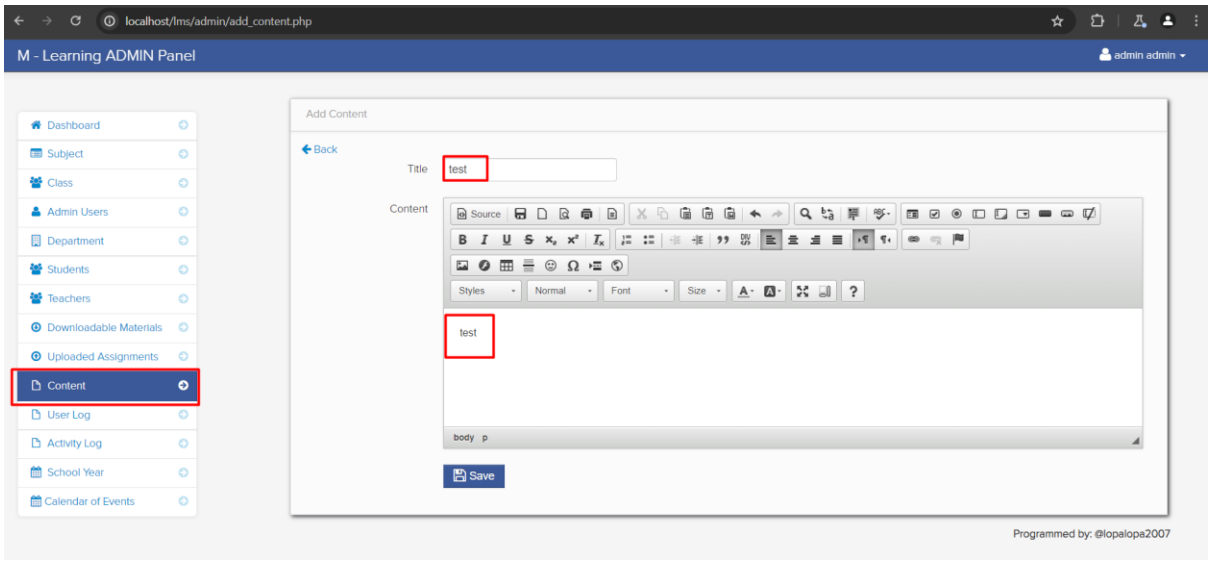
Step 1: Visit to admin login page and login with admin credential.

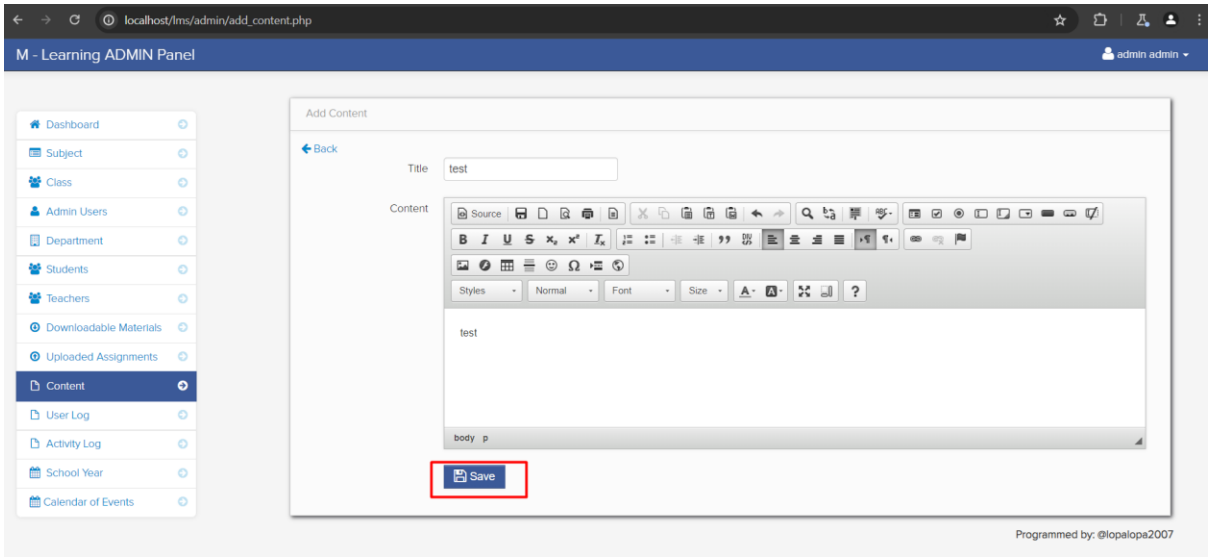


Step 2: Navigate the ‘content’ page and click on ‘Add Content’.

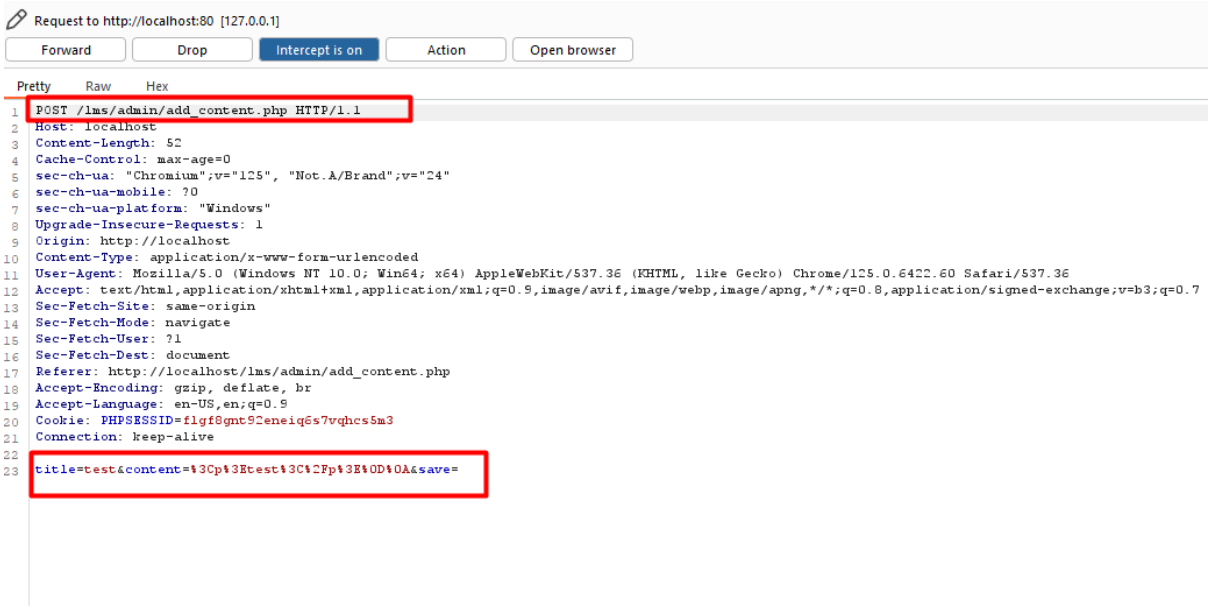


Step 3: Fill the Title and Content values and enable intercept in burpsuite.



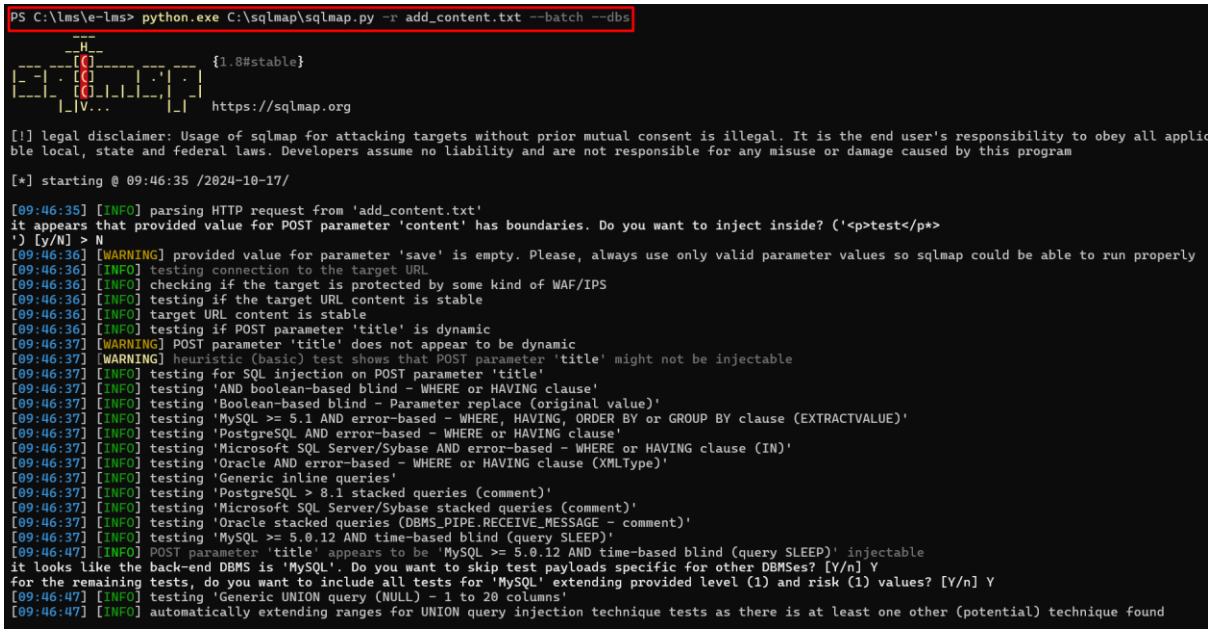


Step 4: Save the burpsuite request in a file.

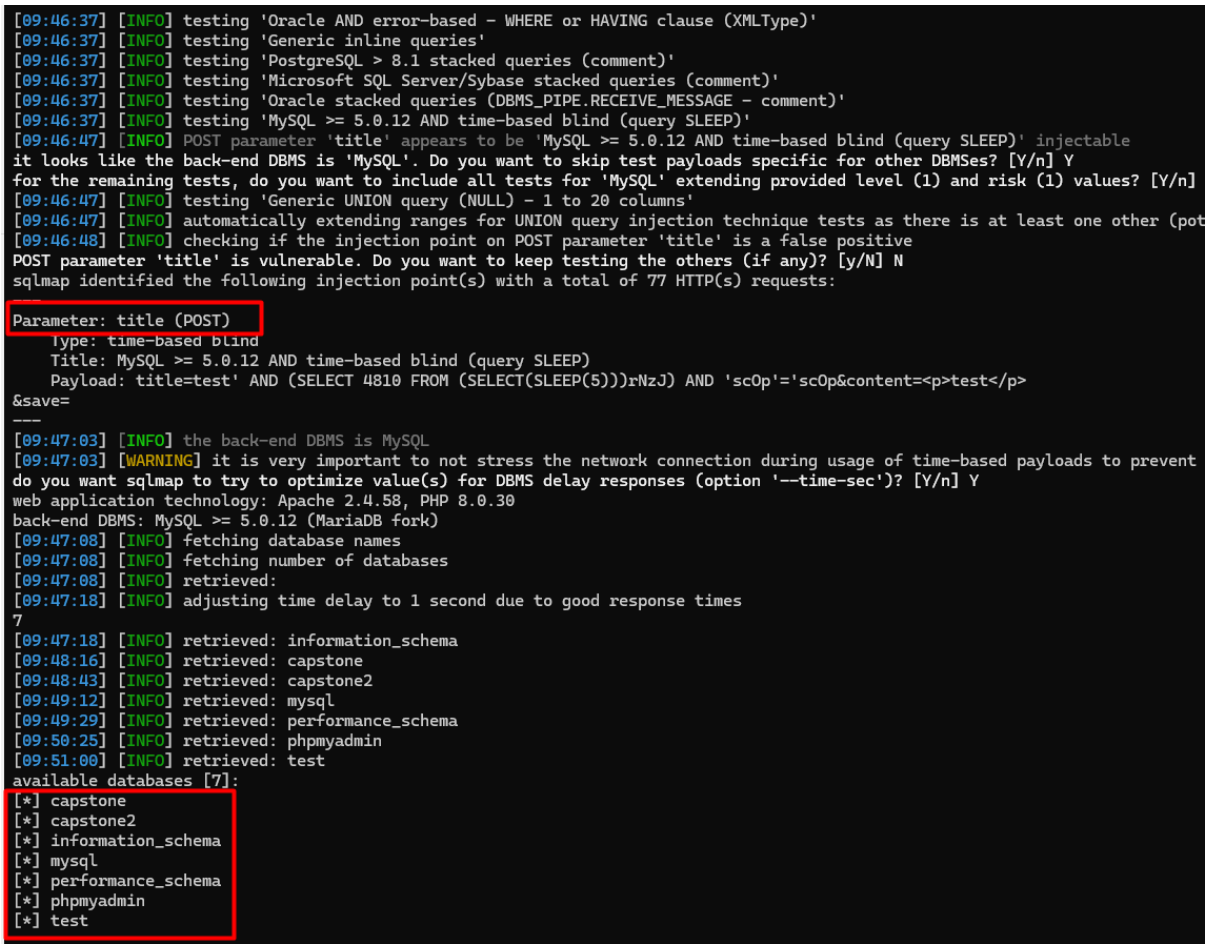


Step 5: Now run the sqlmap command against burpsuite request saved in file.

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



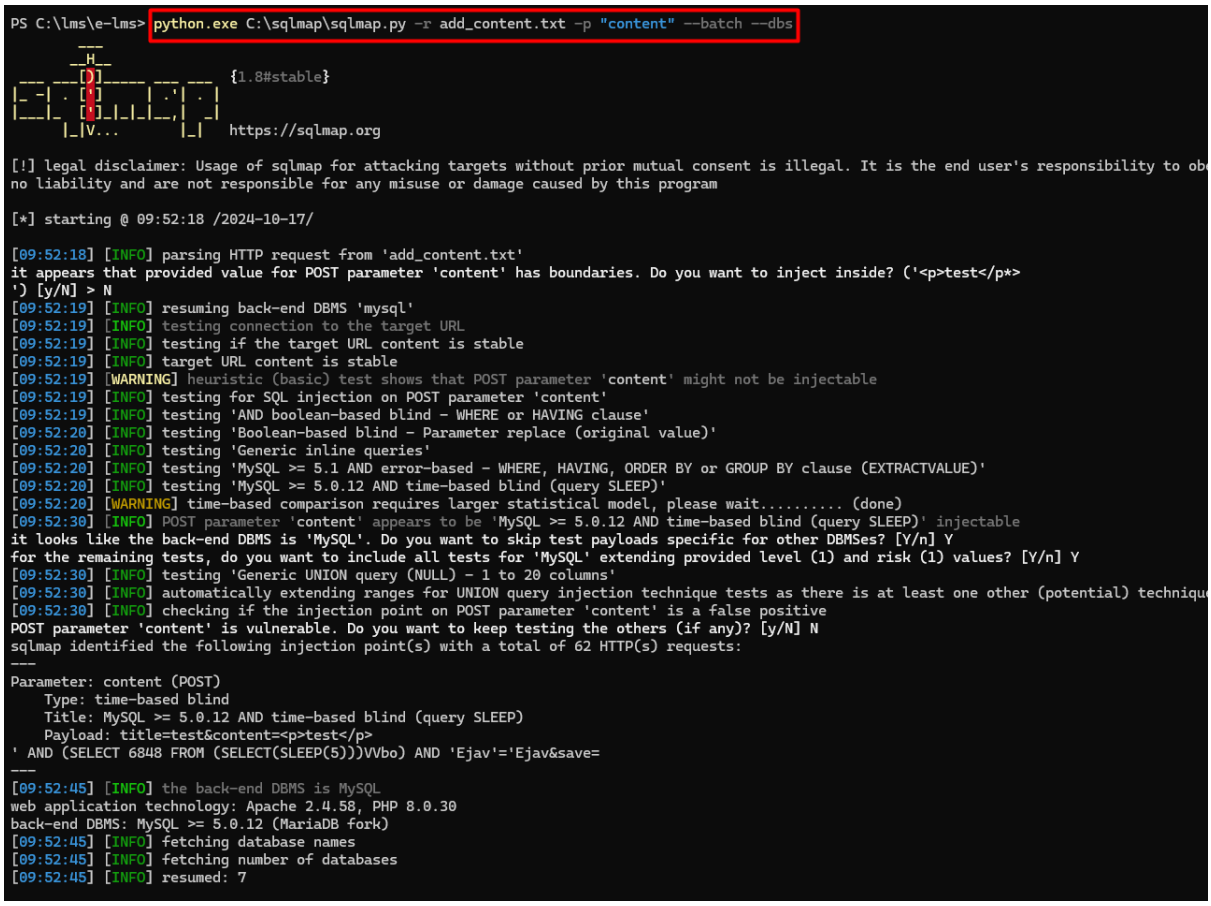
Step 6: Now notice that ‘title’ parameter is detected vulnerable and all database is successfully retrieved.



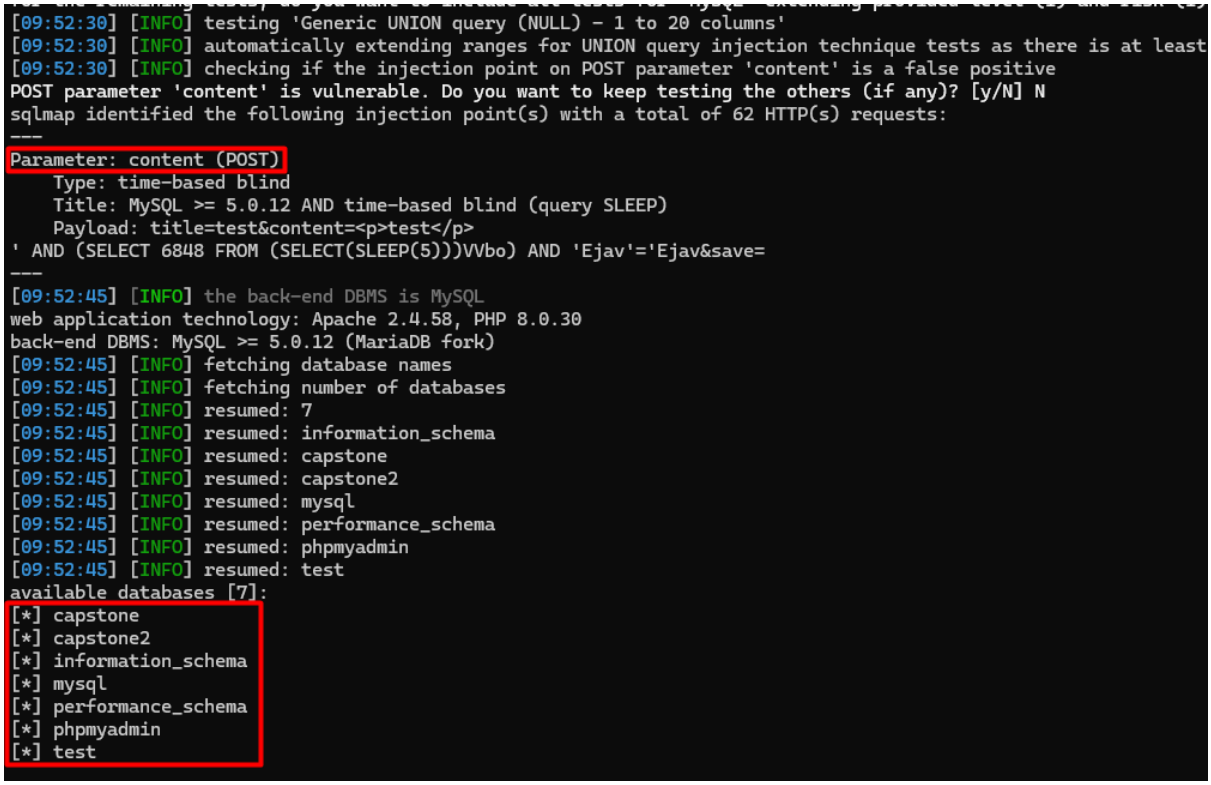
Parameter: content

Step 7: Now try to run sqlmap against `content' parameter with switch '-p'

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



Step 8: Notice that ‘content’ parameter is vulnerable and all database is successfully retrieved.



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