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

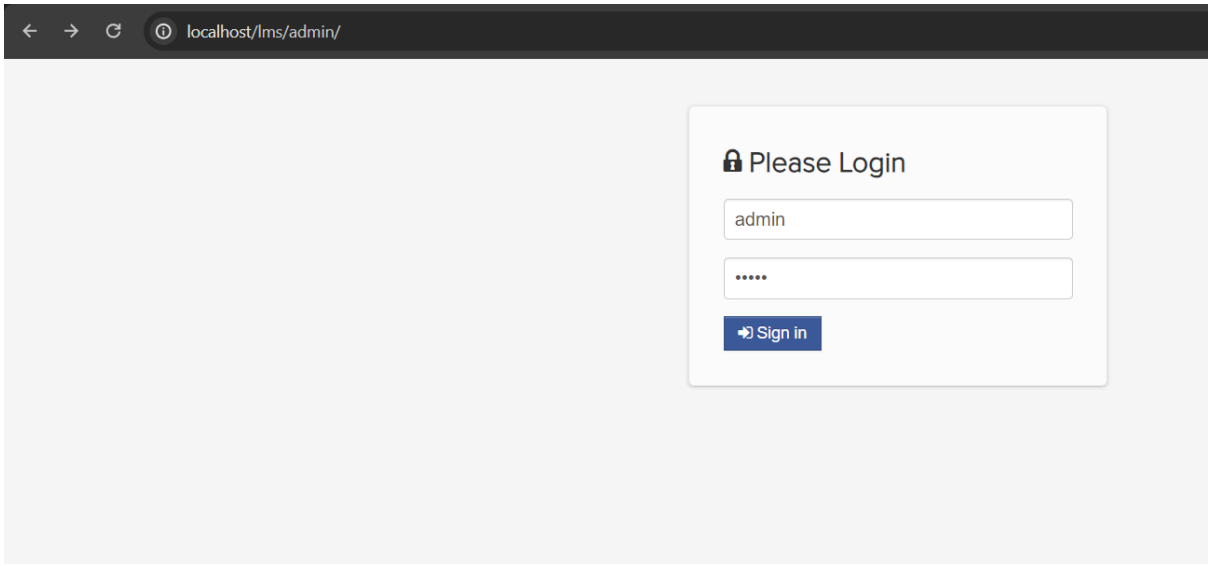
- **Affected Product Name**

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

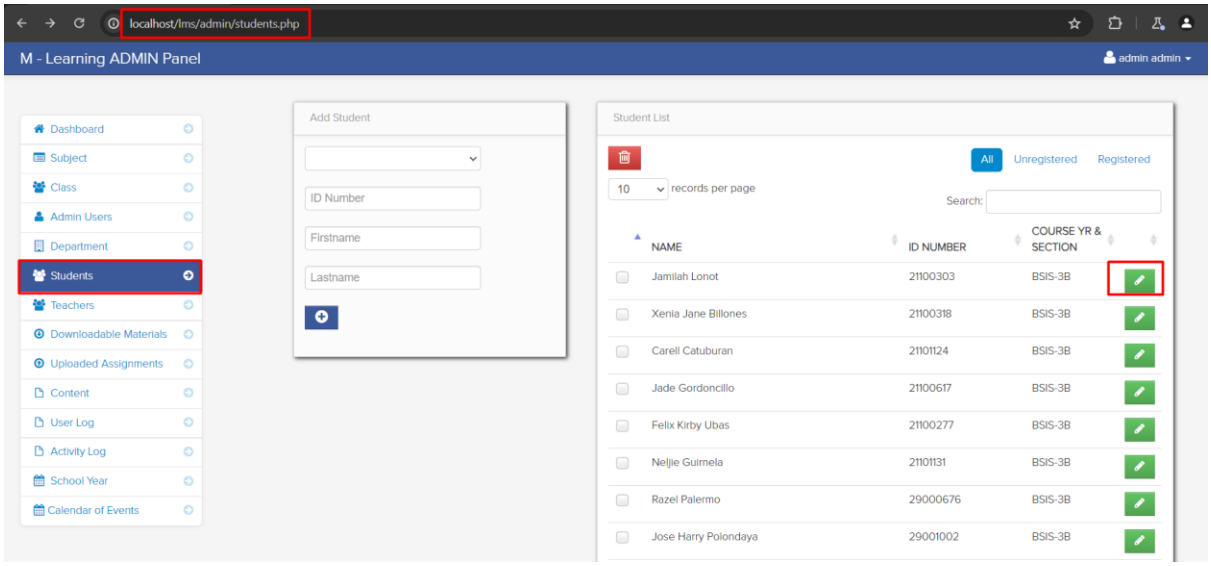
| | |
|---|---|
| **Affected Vendor** | kashipara |
| **Affected Code File** | /lms/admin/edit_student.php |
| **Affected Parameter** | cys, un, ln, fn |
| **Method** | POST |
| **Type** | time-based blind |
| **Version** | V1.0 |

## Steps to Reproduce:

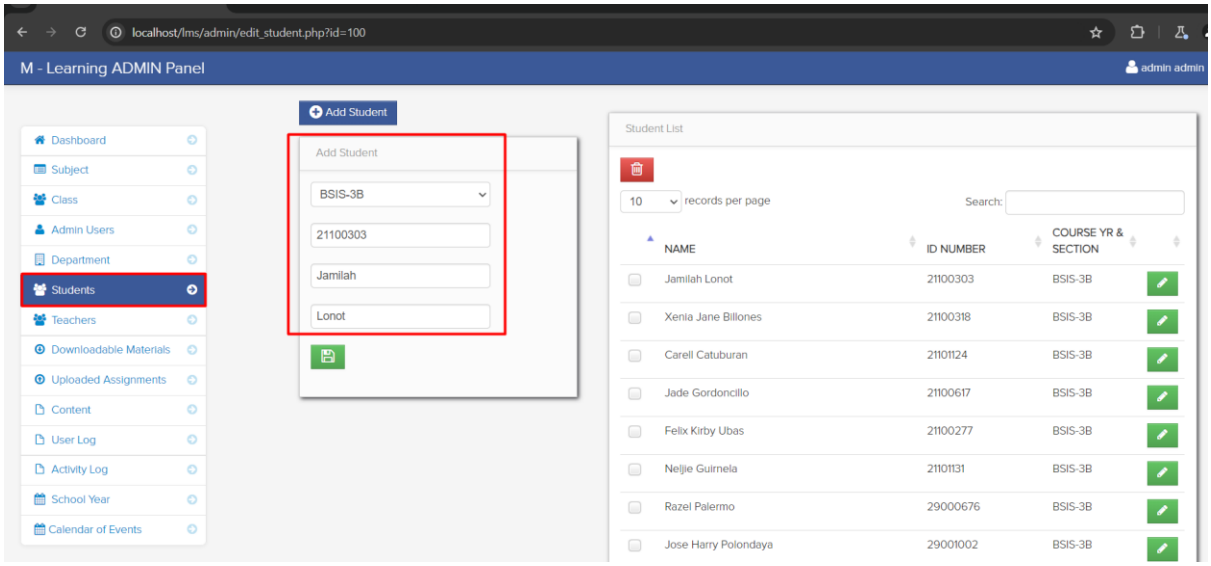
**Step 1**: Visit to admin login page and login with admin credential.



**Step 2:** Navigate the ‘Student’ page click edit on any users from list.



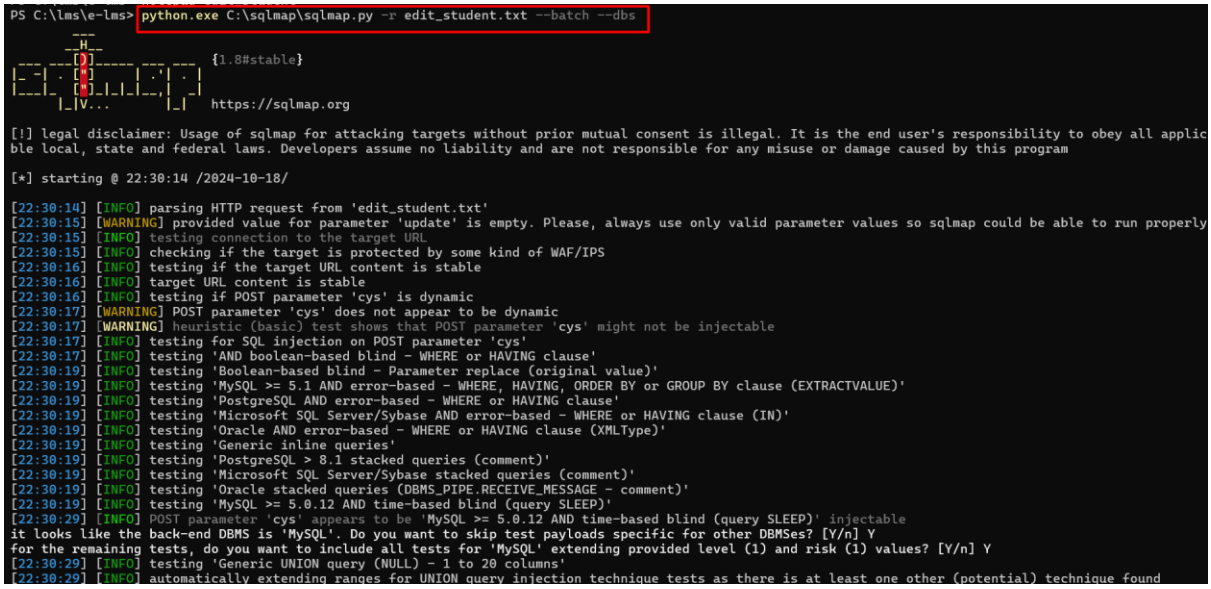

**Step 3**: Now enable intercept in bupsuite and click on save button.



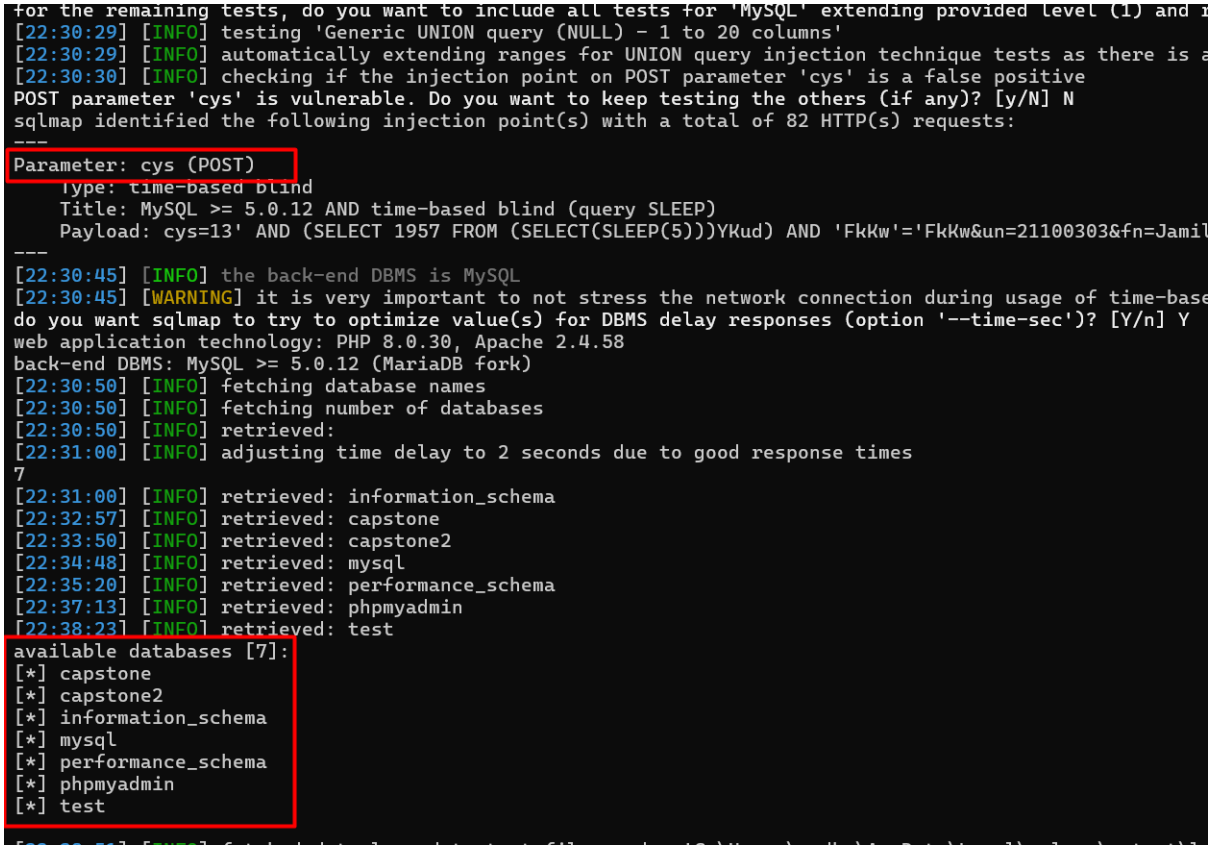

**Step 4:** Save the burpsuite request in a file.


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

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



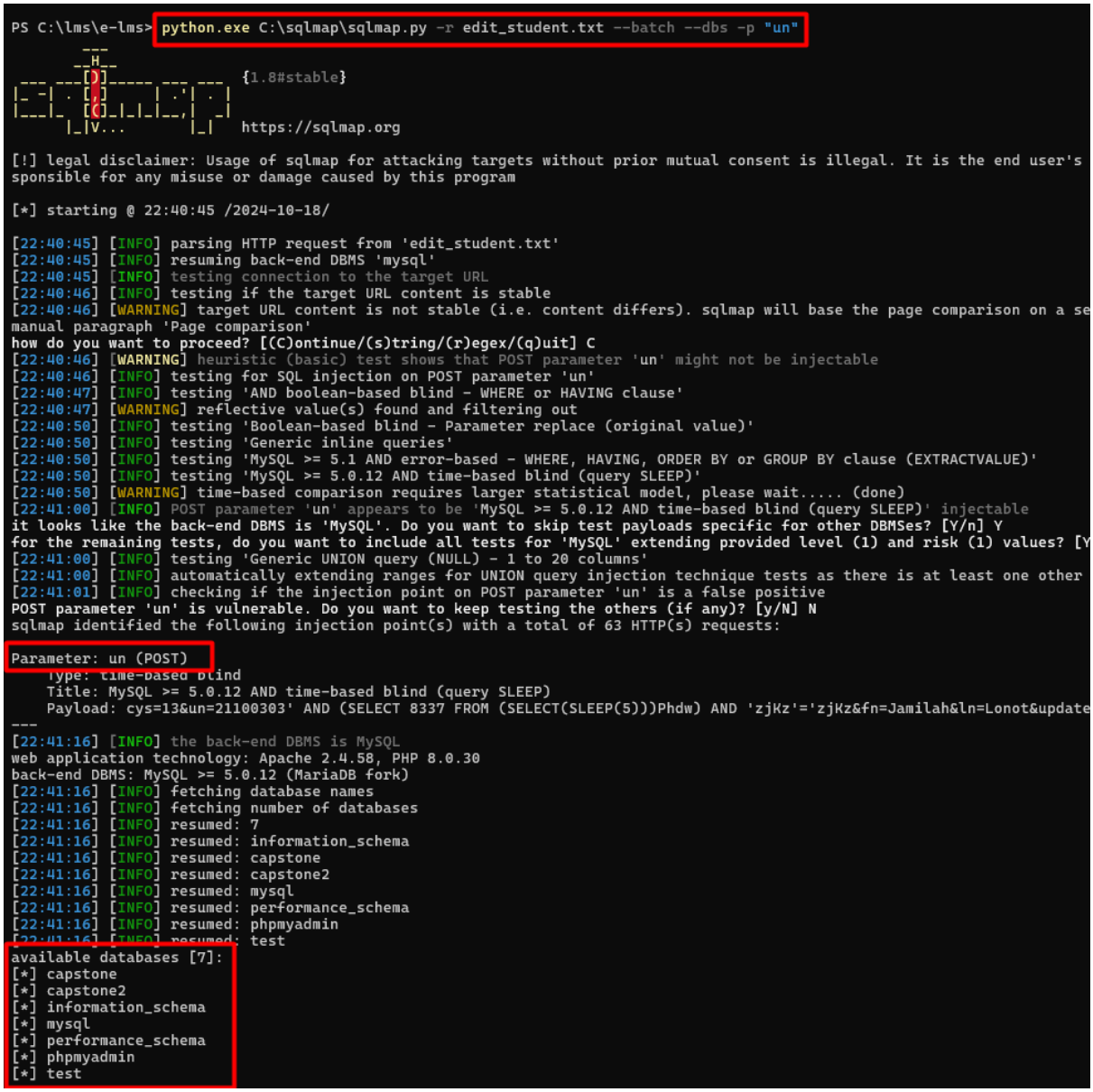

**Step 6:** Now notice that ‘cys’ parameter is detected vulnerable and all database is successfully retrieved.




## 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.

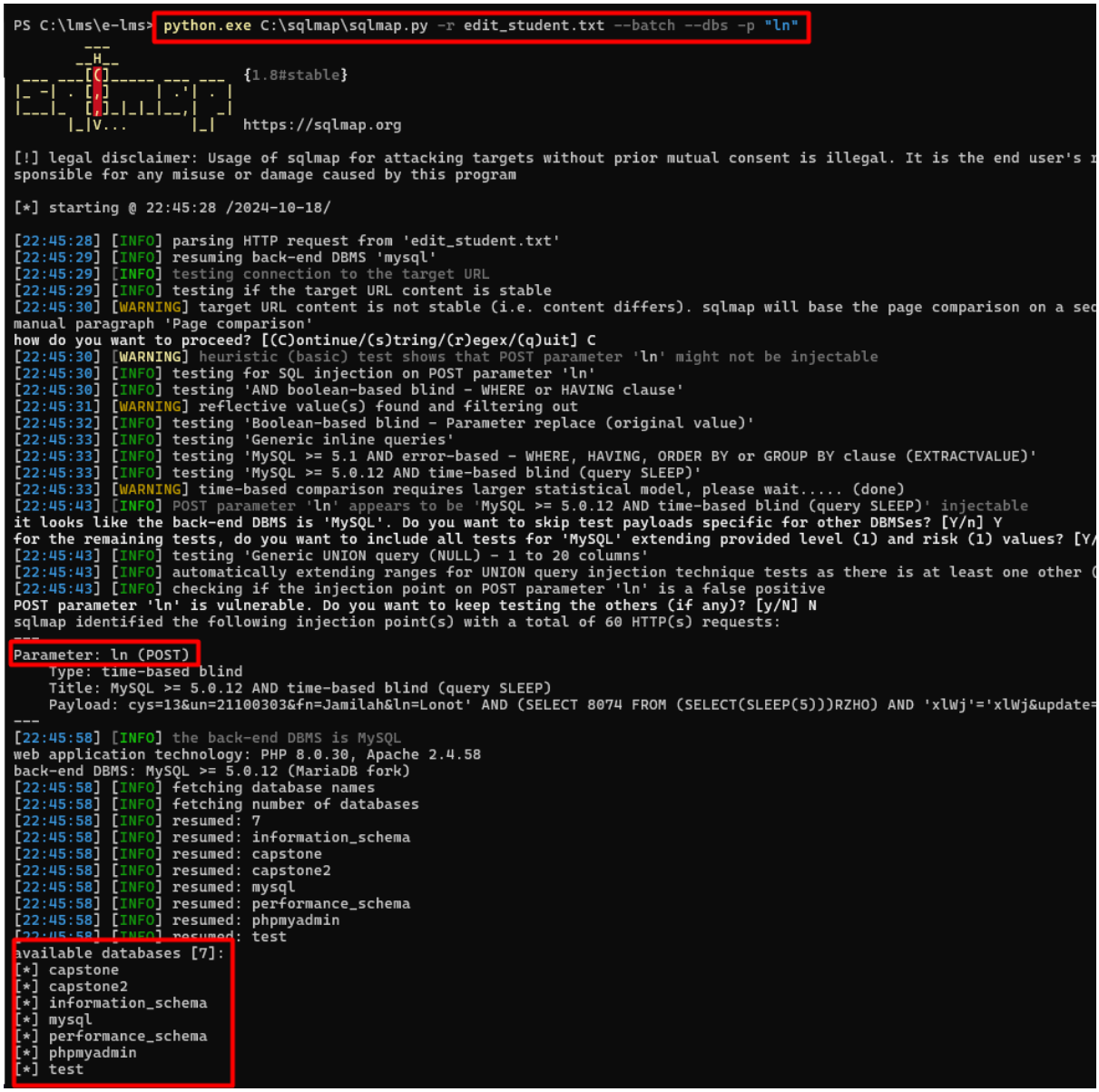
- python.exe C:\sqlmap\sqlmap.py -r edit_student.txt --batch --dbs -p "un"




## Parameter: ln

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

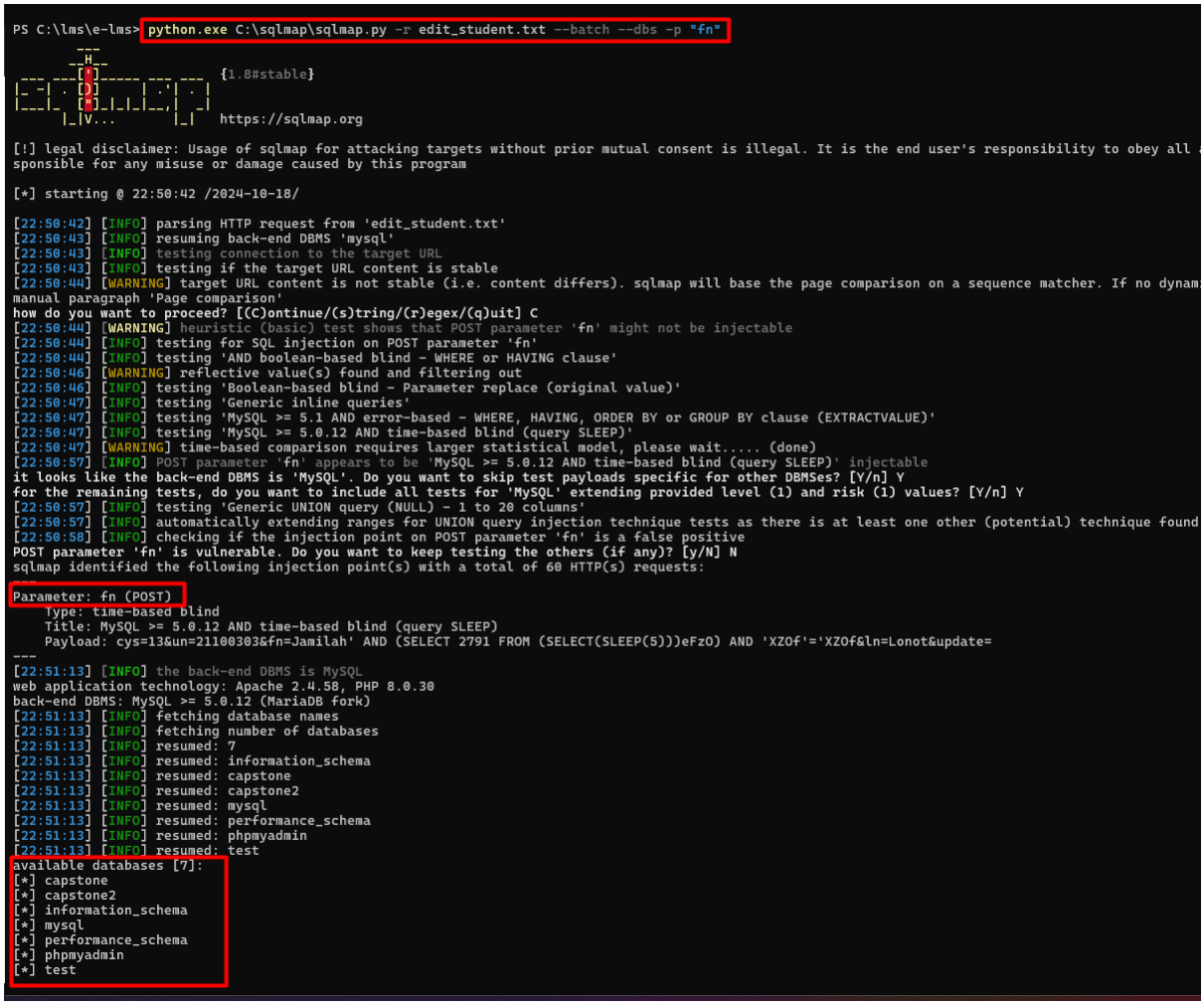
- python.exe C:\sqlmap\sqlmap.py -r edit_student.txt --batch --dbs -p "ln"




## Parameter: fn

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

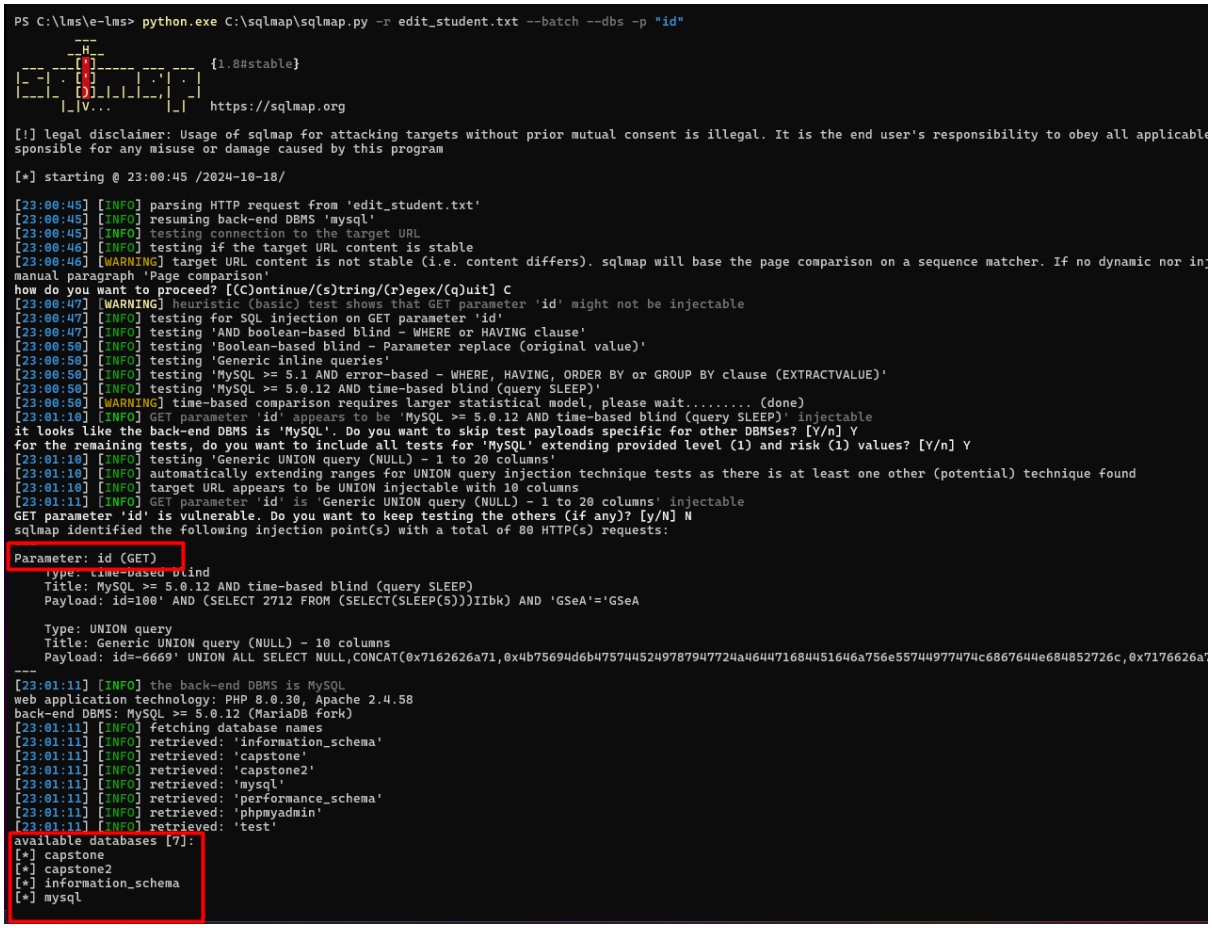
- python.exe C:\sqlmap\sqlmap.py -r edit_student.txt --batch --dbs -p "fn"




## Parameter: id

**Step 9:** Run the sqlmap against ‘id’ parameter by using switch -p. Notice that ‘id’ parameter is detected vulnerable and all database is successfully retrieved.

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




## 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