

Stored Cross Site Scripting (XSS) vulnerability was found in `"/view_type.php"` of the Kashipara Live Membership System v1.0 allows remote attackers to execute arbitrary code via `"membershipType"` POST parameter fields.

**Affected Vendor:** KASHIPARA (<https://www.kashipara.com/>)

**Product Official Website URL:** Live Membership System v1.0  
(<https://www.kashipara.com/project/php/12997/live-membership-system-in-php-php-project-source-code>)

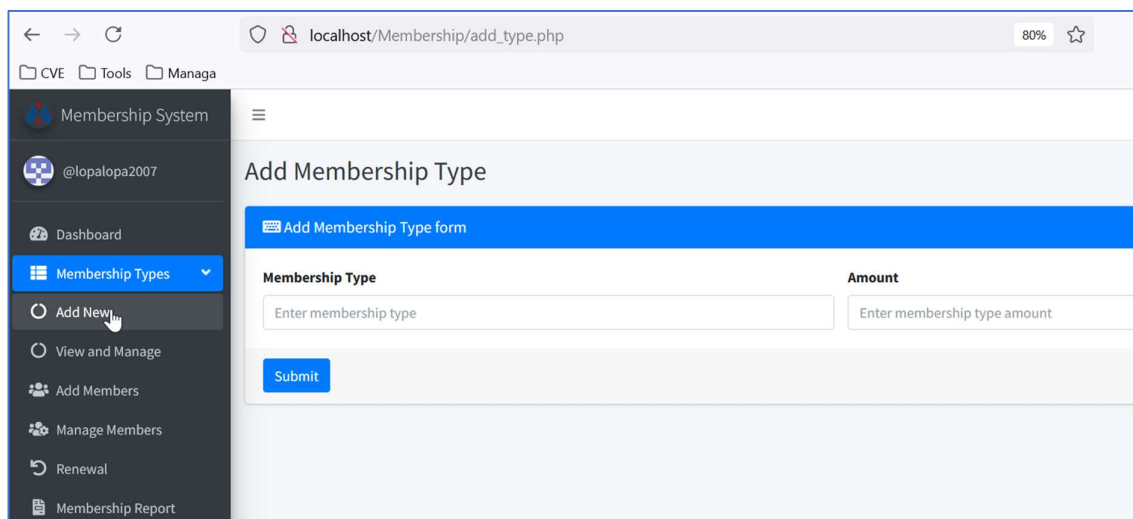
**Version:** 1.0

**Affected Components:**

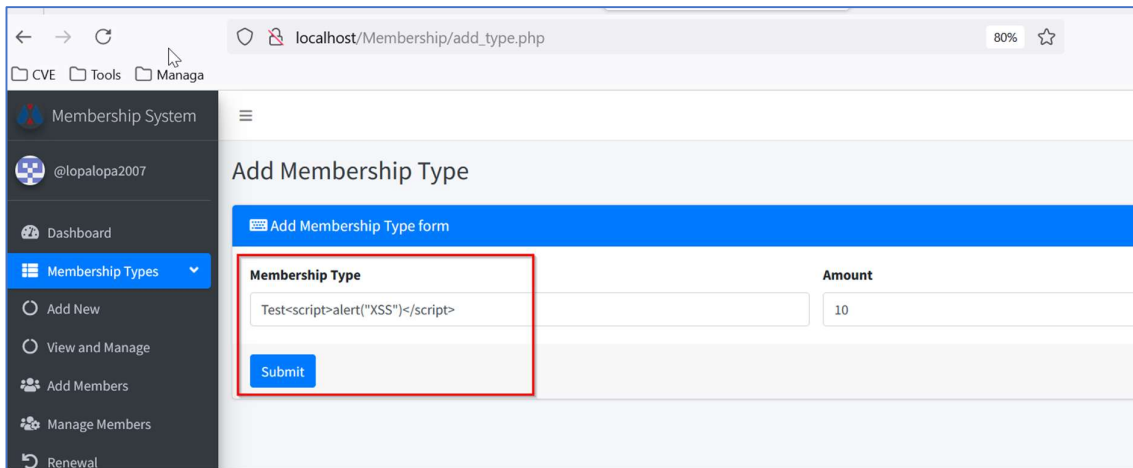
- **Affected Code File:** `/add_type.php` & `/view_type.php`
- **Affected Parameter:** `"membershipType"` POST parameter input
- **Application URL:** [http://localhost/Membership/view\\_type.php](http://localhost/Membership/view_type.php)

**Steps:**

1. Login as administrator in the Live Membership System v1.0 application (URL: <http://localhost/Membership/index.php>).
2. Navigate to "Membership Types" -> "Add New" menu.

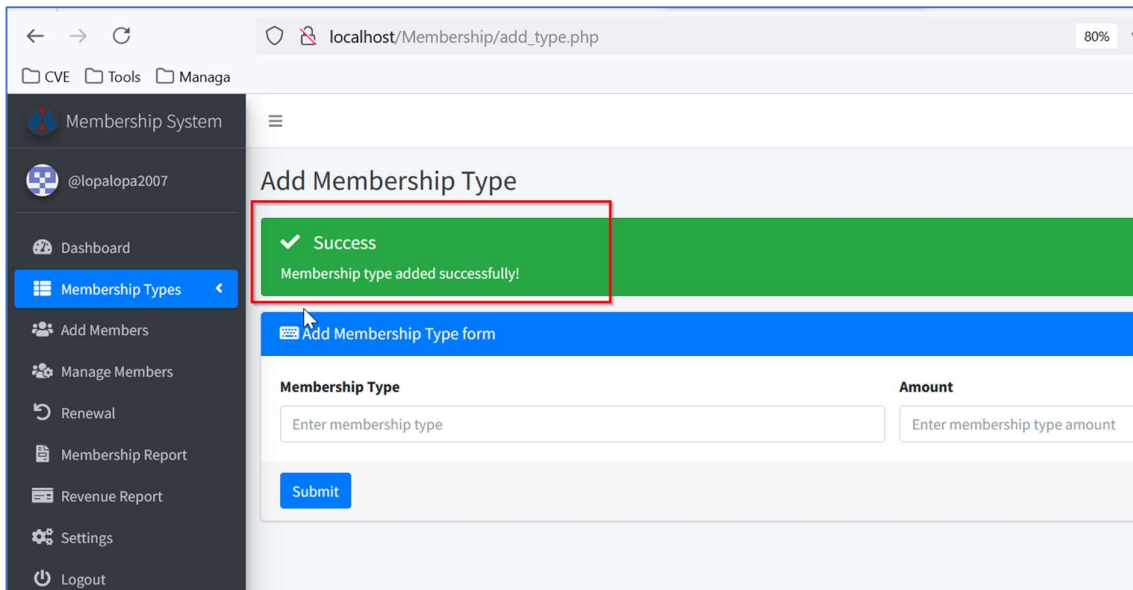


3. Enter the XSS script payload: `Test<script>alert("XSS")</script>` in "Membership Type" input text box and click "Submit" button.



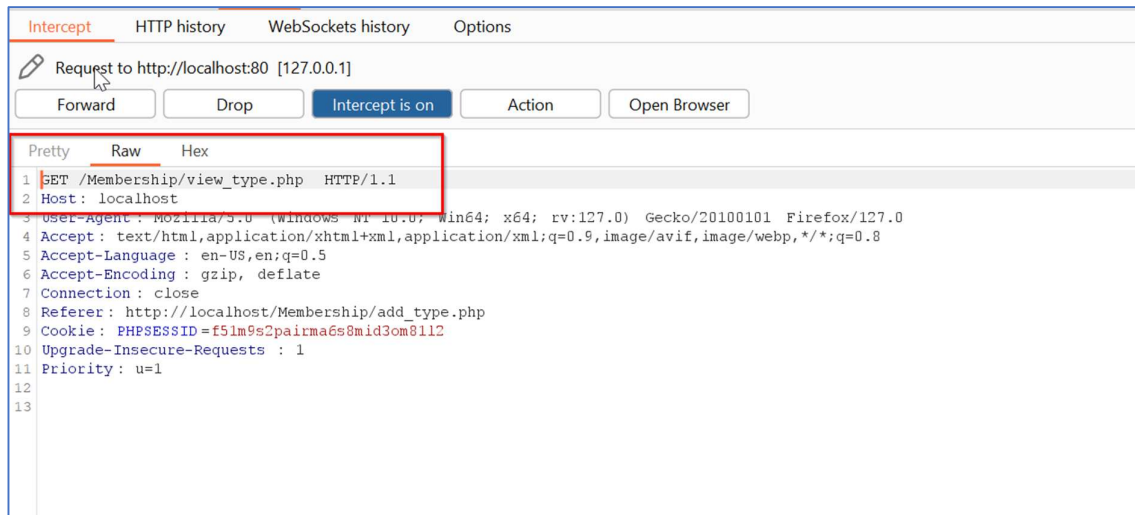
The screenshot shows a web browser at `localhost/Membership/add_type.php`. The page title is "Add Membership Type". On the left is a sidebar menu with "Membership Types" selected. The main form has a blue header "Add Membership Type form". Below it, there are two input fields: "Membership Type" and "Amount". The "Membership Type" field contains the text `Test<script>alert("XSS")</script>`. The "Amount" field contains the number "10". A blue "Submit" button is located below the "Membership Type" field. A red rectangle highlights the "Membership Type" input field and the "Submit" button.

4. The XSS script gets submitted successfully and it gets stored in database.

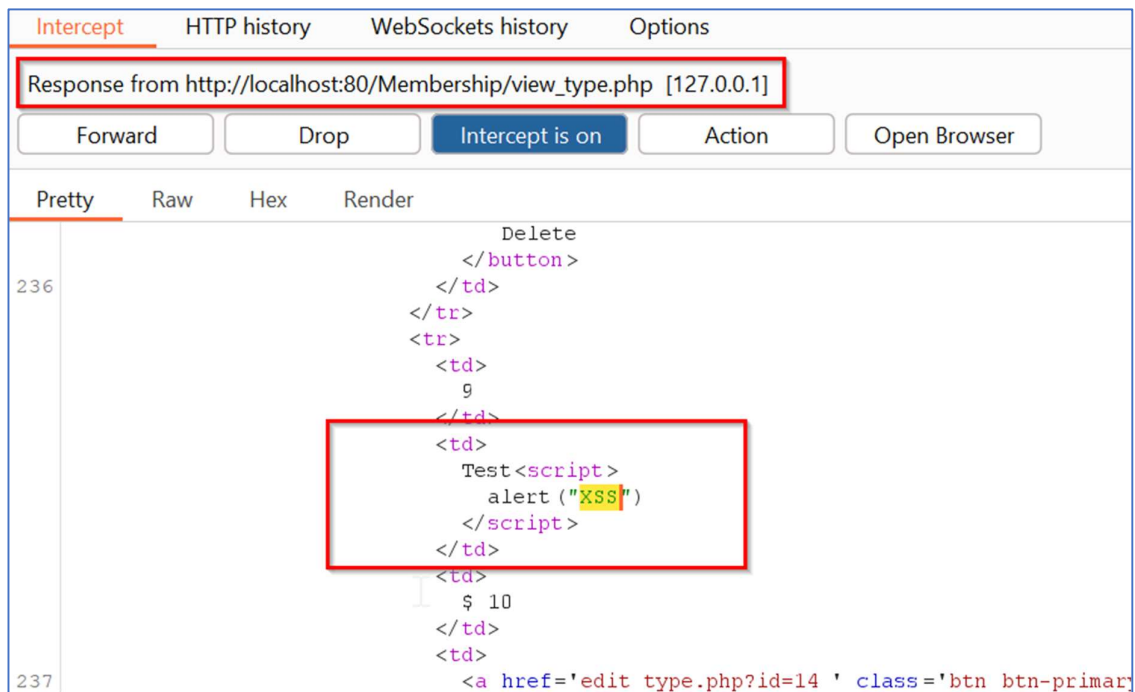


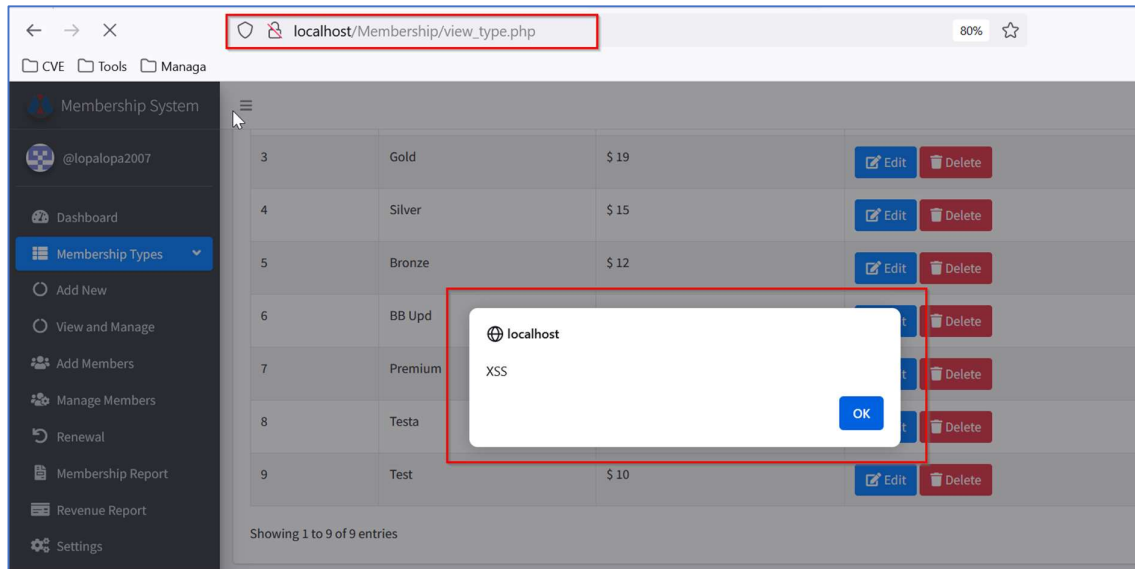
The screenshot shows the same web browser at `localhost/Membership/add_type.php`. The page title is "Add Membership Type". A green success message is displayed at the top of the form area: "✓ Success Membership type added successfully!". Below this, the form is titled "Add Membership Type form". It contains two input fields: "Membership Type" and "Amount". The "Membership Type" field has the placeholder text "Enter membership type". The "Amount" field has the placeholder text "Enter membership type amount". A blue "Submit" button is located below the "Membership Type" field. A red rectangle highlights the green success message.

5. Navigate to "Membership Types" -> "View and Manage" menu (URL: [http://localhost/Membership/view\\_type.php](http://localhost/Membership/view_type.php))

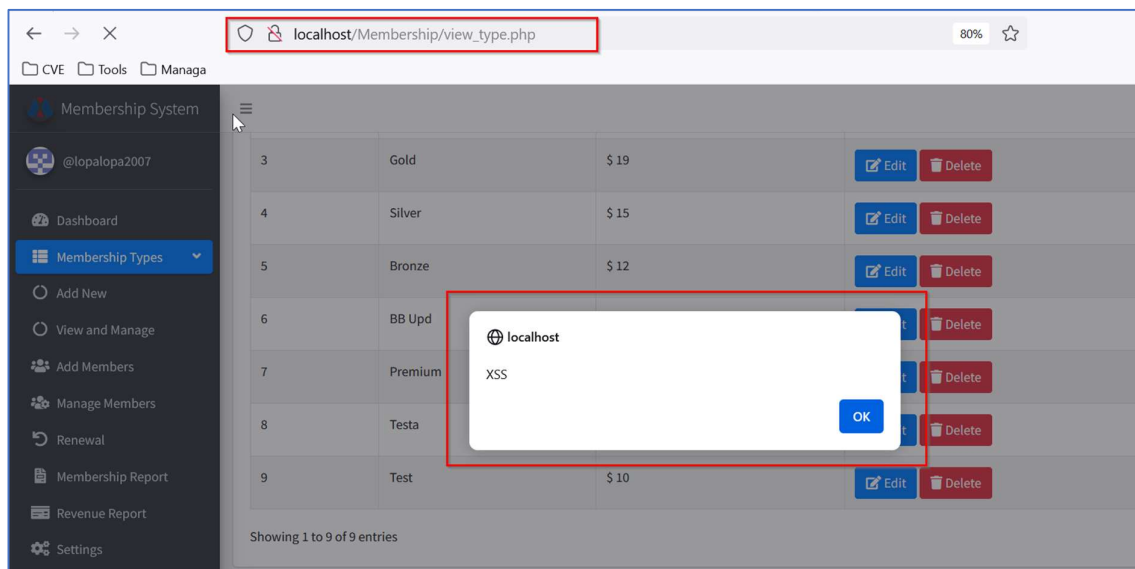


6. The XSS script we submitted in Step 3&4 is reflected back as it is in the response. This script gets executed on the browser.





7. This XSS script is stored in the database. So, every time we navigate to the “Membership Types” -> “View and Manage” menu (URL: [http://localhost/Membership/view\\_type.php](http://localhost/Membership/view_type.php)), this XSS script will get executed.



### **Solution/Good Reads:**

Output Encoding -> When you need to safely display data exactly as a user types it in, output encoding is recommended.

- <https://portswigger.net/web-security/cross-site-scripting>
- [https://cheatsheetseries.owasp.org/cheatsheets/Cross Site Scripting Prevention Cheat Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html)