report2

report2

Cross-Site Scripting (XSS) Vulnerability Report for Image Manipulation on Deluxe Membership Page

1. Executive Summary

- **Vulnerability Title**: Cross-Site Scripting (XSS) via Image Parameter Manipulation
- **Affected Component**: Deluxe Membership Page (`http://localhost:3000/#/deluxe-membership`)
- **Severity**: High
- **Impact**: The vulnerability allows attackers to manipulate image sources via a query parameter and potentially redirect users to malicious sites or inject malicious content into the application.

Summary:

A Cross-Site Scripting (XSS) vulnerability has been discovered on the Deluxe Membership page at `http://localhost:3000/#/deluxe-membership`. The vulnerability is present in the image logo manipulation functionality, where the `testDecal` parameter allows arbitrary input to change the logo image. An attacker can manipulate this parameter to load external resources, potentially leading to malicious redirects or further exploitation. This issue arises due to a lack of input validation and improper sanitization of the `testDecal` parameter.

---

2. Vulnerability Details

2.1 Description:

The page allows users to customize the logo on the deluxe membership page by manipulating the `testDecal` parameter in the URL. However, this parameter can be tampered with to load arbitrary images or resources, including external URLs. In the worst case, an attacker can redirect users to malicious websites or even load scripts that can result in stored XSS attacks. The vulnerable code is found in `main.js`, which lacks proper validation for the `testDecal` input.

2.2 Steps to Reproduce:

1. Go to the Deluxe Membership page: `http://localhost:3000/#/deluxe-membership`.
2. Modify the URL to include the `testDecal` parameter with a malicious payload:

`http://localhost:3000/#/deluxe-membership?testDecal=../../../../../redirect?to=https://dummyimage.com/600x400/000/fff?x=https://github.com/juice-shop/juice-shop`

- Submit the modified URL.
- Observe that the logo image is replaced with the external image from `dummyimage.com`, or the link redirects to the malicious destination.

2.3 **Impact**:

By manipulating the `testDecal` parameter, an attacker can replace the logo with an external image or redirect the user to a malicious site. This opens the door for potential phishing attacks, as users may be tricked into visiting malicious sites under the guise of the original application. Additionally, the attacker can load external scripts or further escalate the attack through stored XSS, resulting in full compromise of the user's session.

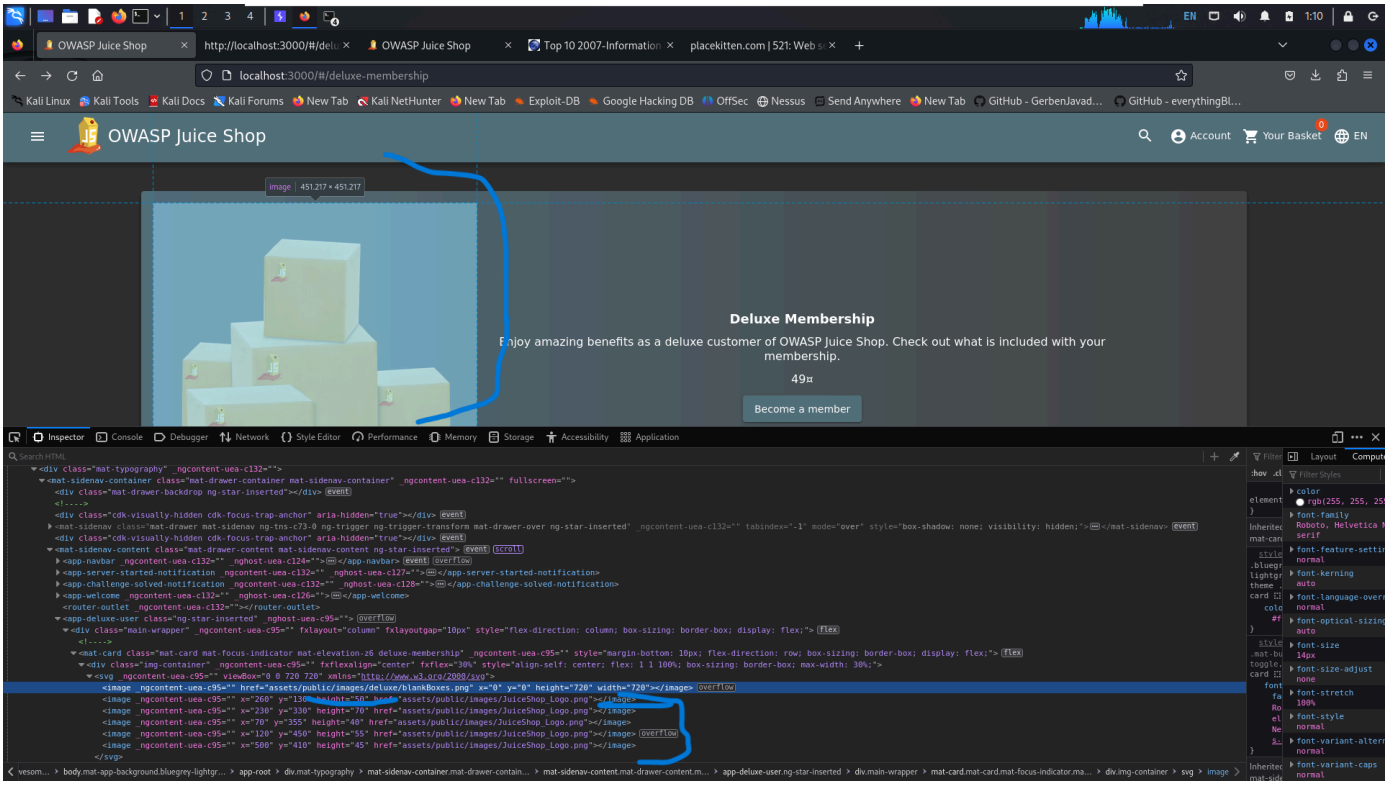
---

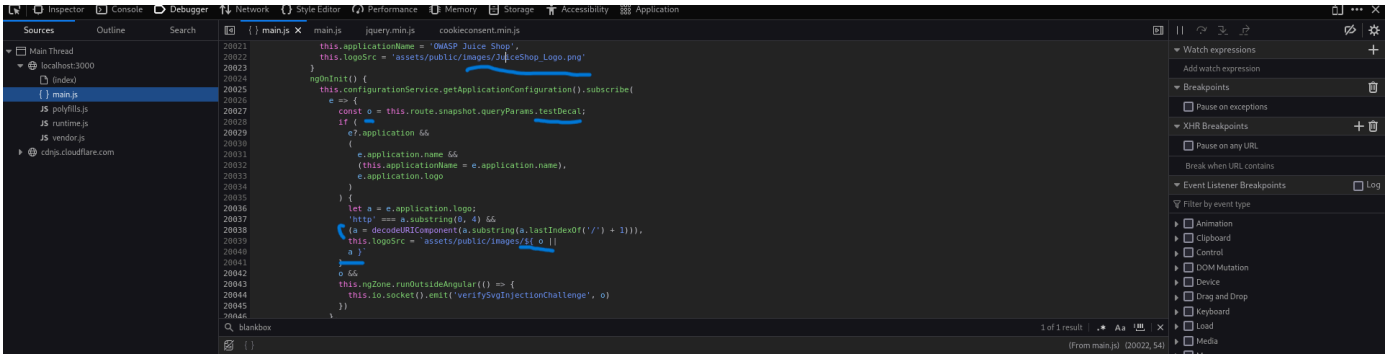
3. Technical Details

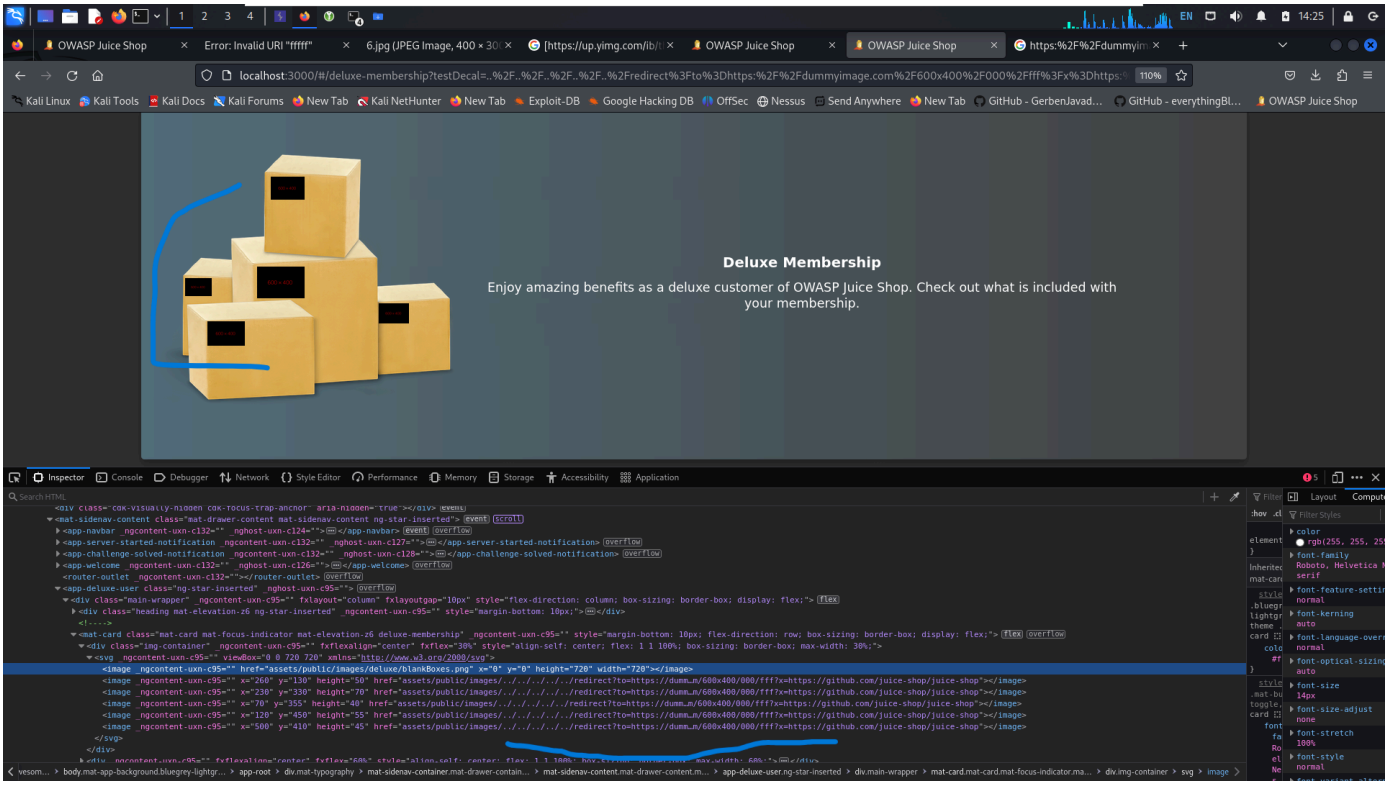
3.1 **Exploitability**:

An attacker can easily exploit this vulnerability by crafting a malicious URL with a manipulated `testDecal` parameter. Since there is no input validation or sanitization on the value of the `testDecal` parameter, external resources such as images or scripts can be injected into the application.

3.2 **Proof of Concept (PoC)**:







3.3 **Root Cause Analysis:**

The vulnerability arises due to improper validation and sanitization of the `testDecal` parameter in the `main.js` file of the application. Instead of validating whether the provided image source is internal and secure, the application directly uses the value of the parameter, allowing attackers to inject arbitrary content.

---

4. **Mitigation Recommendations**

4.1 **Short-Term Fix:**

- Implement strict input validation on the `testDecal` parameter to ensure that only trusted and secure URLs (such as those from the application's own domain) can be used for image replacement.

Sanitize all user inputs before using them in the DOM to prevent XSS attacks.

Example validation:

javascript

- `const allowedImages = ['logo1.png', 'logo2.png'];`
- `if (allowedImages.includes(testDecal)) {`
- `// Proceed with setting the logo`
- `}`
- `else {`
- `// Set a default or error logo`
- `}`

4.2 **Long-Term Fix**:

- Refactor the image customization logic to use a server-side whitelist of allowed images. Only pre-approved images should be used as logos, and users should not be allowed to provide their own URLs.
- Implement a content security policy (CSP) to block the execution of any external scripts or images not hosted on trusted domains.
- Perform security audits on JavaScript files, particularly those handling user inputs, to ensure they do not introduce vulnerabilities like XSS.
- Update the error handling mechanism to gracefully handle invalid input without allowing arbitrary image loading or redirects.

---

5. **Risk Assessment**

5.1 **Likelihood**:

- **Likelihood**: High
  The vulnerability is easily exploitable by anyone who has access to the application's URL and understands the `testDecal` parameter. Since this can be exploited via a URL, the risk of exploitation is high, especially through phishing or social engineering attacks.

5.2 **Impact**:

- **Impact**: High
  Attackers could use this vulnerability to manipulate the appearance of the application or redirect users to malicious websites. In cases of more advanced exploitation, stored XSS could lead to session hijacking, data theft, or full account takeover.

---

6. **Conclusion**

The XSS vulnerability on the Deluxe Membership page, through the `testDecal` parameter, poses a significant risk to the application. An attacker can manipulate the URL to change the displayed logo image or redirect users to external sites. Immediate action is required to validate and sanitize the `testDecal` parameter and ensure that only trusted images are loaded. Furthermore, applying a content security policy (CSP) will help mitigate similar issues in the future.