Proof of Concept and Mitigation Report for Reflected XSS Vulnerability in LinkAce

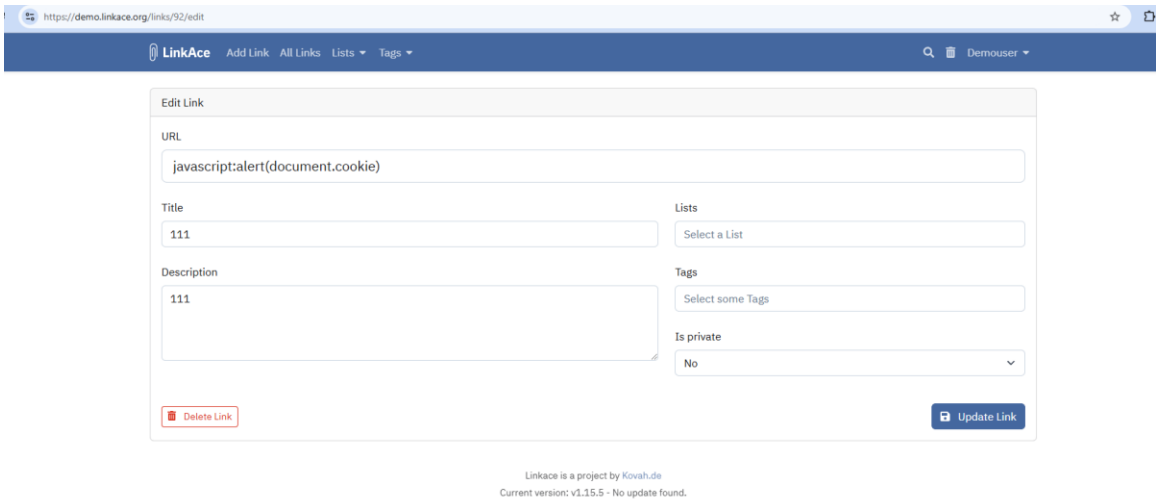
Title
Reflected Cross-Site Scripting (XSS) Vulnerability in LinkAce v1.15.5

Summary
A reflected cross-site scripting (XSS) vulnerability was identified in the "URL" field of the LinkAce web application (v1.15.5). This vulnerability allows attackers to inject and execute arbitrary JavaScript in the context of the victim's browser. This poses a severe risk, including potential session hijacking, cookie theft, and unauthorized actions on behalf of the victim.

Affected Component
Module: Edit Link
Field Name: URL

Vulnerability Details
Description:
The application fails to sanitize or encode user-supplied input in the "URL" field. When malicious input is provided, it is reflected in the application's response and executed by the browser. This enables an attacker to execute arbitrary JavaScript in the victim's session.

Proof of Concept (PoC)
Steps to Reproduce:
1. Navigate to the Edit Link page of LinkAce.



2. In the "URL" field, input the following payload:
   javascript:alert(document.cookie)

3. Click Update Link.

4. Observe the reflected JavaScript executing in the browser context, displaying the session cookie of the user.

Evidence:
See the attached screenshot demonstrating the payload execution in the "URL" field.

Mitigation

1. Input Validation:
   - Enforce strict validation on the "URL" field to ensure only valid URLs are allowed.
   - Reject input that includes JavaScript protocols (javascript:).

2. Output Encoding:
   - Apply context-aware encoding to user-supplied input before rendering it in the HTML response.
   - For attributes, use HTML attribute encoding.
   - For inline JavaScript, use JavaScript encoding.

3. Content Security Policy (CSP):
   - Implement a strict CSP to disallow inline JavaScript execution.
   - Example CSP:
     Content-Security-Policy: default-src 'self'; script-src 'self';

4. Additional Hardening:
   - Remove support for dangerous protocols like javascript:.
   - Use server-side sanitization libraries (e.g., OWASP ESAPI).

Contact Information

Researcher Name: Kwangyun Keum
Email: kwangyunkeum@gmail.com