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.

ttps://demo.linkace.org/links/92/edit					
	🕅 LinkAce Add Link All Links Lists - Tags -		Q 📋 Demouser 🕶		
	Edit Link URL				
	javascript:alert(document.cookie)				
	Title	Lists			
	111	Select a List			
	Description	Tags			
	111	Select some Tags			
		Is private			
		No	~		
	Delete Link		Update Link		
Linkace is a project by Kovahude					

Linkace is a project by Kovah.de Current version: v1.15.5 - No update found.

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.

https://demo.linkace.org/links							
D LinkAce Add	.ink All Links Lists ▼ Tags ▼	demo.linkace.org says	Q 🛅 Demouser 🕶				
Links		XSR- TOREA-ey/pdilialBOTENJMI9UKJ9PdH84W/hIRkINdlE9P5(sinZhbHVI JjojGCTFWW92blFyW ERhcTNiGnF00TFZbVILckVQxQrVyFQyWDFIcXVFdRvQ3ZwRm1WcG dx0ncSnR2QfilUB00WUUWW6xt Tlckunp5SGY5fX2TBCXQFIVUM3UB1jzm1R2210/3Ju0D2LT1B0T0 S5BW4bhnc3UNeX9HHESXTNUL CIPWH/OIIz2TZI/NmVjZTc4NGRmZDMQ2Vg2Yg5ZGI4VTZK/jQwZ mUSMTC2VUUZmMIMDFINCV/zJUSY	+ Add Order by •				
	X 111 javascript:alert(document.coc 111		3				
No Tags		ОК	Added 33 minutes ago 🕚 Show Link 🖉 Edit Link 🖀 Delete Link				

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