COMMUNITY CONTENT

Submit #782170: Mario Zechner pi-mono 0.58.4 SVG Artifact Stored XSS Leading to Credential Theft

Title	Mario Zechner pi-mono 0.58.4 SVG Artifact Stored XSS Leading to Credential Theft
Description	A stored Cross-Site Scripting (XSS) vulnerability exists in the SVG artifact rendering component of @mariozechner/pi-web-ui. When the LLM generates an SVG artifact, the content is rendered directly into the parent page DOM using the unsafeHTML() Lit directive without any sanitization (no DOMPurify, no allowlist filtering, no iframe sandboxing). Unlike HTML artifacts, which are isolated within sandboxed <iframe> elements (sandbox="allow-scripts allow-modals"), SVG artifacts are rendered inline in the main application context using light DOM (createRenderRoot() { return this; }). This allows embedded JavaScript in SVG event handlers (e.g., onload, onerror, onclick) to execute with full access to the parent page's origin context, including document.cookie, localStorage, and IndexedDB. This vulnerability is chained with a second vulnerability: LLM provider API keys (Anthropic, OpenAI, Google, etc.) are stored as plaintext strings in the browser's IndexedDB without any encryption. When the XSS payload executes, it can read all stored API keys and exfiltrate them to an attacker-controlled server. The combined effect is a full credential theft of all configured LLM provider API keys, authentication tokens, and chat session history, triggered by a single malicious SVG artifact that the LLM is manipulated into generating via prompt injection.
Source	https://github.com/August829/CVEP/issues/20
User	Yu Bao (UID 88956)
Submission	03/18/2026 08:22 AM (18 days ago)
Moderation	04/04/2026 08:35 AM (17 days later)
Status	Verified
VulDB entry	[badlogic pi-mono 0.58.4 SVG Artifact SvgArtifact.ts cross site scripting]
Points	20

Community Content

Submissions are made by VulDB community users. VulDB is *not responsible* for their content nor the links to external sources.

Please use the raw information shown and the links listed *with caution*. They might contain malicious and harmful actions, code or data.

The corresponding VulDB entries contain the moderated, verified, and normalized information provided within the raw submission.

Documentation

- Submission Policy
- Data Processing
- CVE Handling