Pages Management Full Account Takeover for All-Roles & Privilege-Escalation via Stored DOM XSS

Critical **bertugfahriozer** published **GHSA-458r-h248-29c5** 2 days ago

Package

php **ci4-cms-erp/ci4ms** (Composer)

Affected versions	Patched versions
<= 0.28.6.0	0.31.0.0

Description

Summary

Vulnerability: Stored DOM XSS via Page Management Fields (Persistent Payload Injection)

- Stored Cross-Site Scripting via Unsanitized Page Creation and Editing Inputs

Description

The application fails to properly sanitize user-controlled input within the **Page Management** functionality when creating or editing pages. Multiple input fields accept attacker-controlled JavaScript payloads that are stored server-side.

These stored values are later rendered without proper output encoding across administrative page lists and public-facing page views, leading to stored DOM-based cross-site scripting (XSS).

Affected Functionality

- Page creation functionality
- Page editing functionality
- Page list and management views

- Public-facing page rendering
- Storage and retrieval of page-related data

Affected Fields

- Title
- URL
- Content
- Cover Image
- Image URL
- Image Width
- Image Height
- SEO Description
- SEO Keywords

Attack Scenario

- An attacker creates or edits a page and injects a malicious XSS payload into one or more page-related input fields.
- The application stores these values without sanitization or encoding.
- The payload is rendered in administrative page lists and public-facing page views.
- The payload executes automatically in the browser context of administrators, authenticated users, and unauthenticated visitors.

Impact

- Persistent Stored XSS
- Execution of arbitrary JavaScript in victims' browsers
- Privilege escalation when viewed by administrators or privileged users
- Full administrator account takeover
- Full account takeover across all roles
- Full compromise of the entire application

Endpoints:

- `/backend/pages/create`
- Page list management view
- Public-facing page views

Steps To Reproduce (POC)

1. Navigate to the Page Management -> Add Page interface

2. Insert an XSS payload into any page-related field such as:
   `<img src=x onerror=alert(document.domain)>`
3. Save or publish the page
4. View the page via the administrative page list or public-facing page
5. Observe the XSS payload executing automatically

Remediation

- Never use .html() again or any innerHTML-style like JS in your PHP, or any other sink, even if user inputs that flow into them are not clear, they still represent real world danger as an attacker can make use of this to exploit the application via XSS. And do HTML Encoding as much as possible and always do Sanitization, theres no sanitization there unfortunately. Also apply CSP, HttpOnly, SameSite, and Secure upon all application, they reduce severity of XSS & escalated-CSRF via XSS and do great jobs

Ready Video POC:

https://mega.nz/file/iAkWAKQY#hCUv4DlMPFykPvb4gO94ZVGj64tpUk99gLxE6u1kASk

Severity

Critical 9.1 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	Low
User interaction	None
Scope	Changed
Confidentiality	High
Integrity	Low
Availability	Low

Learn more about base metrics

CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:L/A:L

CVE ID

CVE-2026-34566

Weaknesses

▶ CWE-79

Credits

bugmithlegend Reporter

peeefour Reporter