Blogs Posts (Categories) Full Account Takeover for All-Roles & Privilege-Escalation via Stored DOM XSS

Critical **bertugfahriozer** published **GHSA-r33w-c82v-x5v7** 5 days ago

Package

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

Affected versions	Patched versions
<= 0.28.6.0	0.31.0.0

Description

Summary

Vulnerability: Blogs Posts (Categories) Full Account Takeover for All-Roles & Privilege-Escalation via Stored DOM XSS

- Stored Cross-Site Scripting via Unsanitized Blog Post Content in Blog Management (Categories)

Description

The application fails to properly sanitize user-controlled input when creating or editing blog posts within the **Categories** section. An attacker can inject a malicious JavaScript payload into the **Categories** content, which is then stored server-side.

This stored payload is later rendered unsafely when the **Categories** are viewed via blog posts, without proper output encoding, leading to stored cross-site scripting (XSS).

Affected Functionality

- Blog post **Categories** creation functionality
- Blog post **Categories** editing functionality
- Blog post **Categories** storage and retrieval logic

Attack Scenario

- An attacker creates or edits a blog post **Category** to include a malicious XSS payload in the category description or name.
- The application stores this content without sanitization or encoding.
- The payload persists and executes whenever the category is viewed within the blog posts section, leading to the execution of arbitrary JavaScript in the victim's browser.

Impact

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

Endpoints:

- `/backend/blogs/create` (Categories specific)
- `/backend/blogs/` (Categories view)
- `/blog/{id}` (Rendered blog post under Categories)

Steps To Reproduce (POC)

1. Go to the **Categories** section of the blog management panel.
2. Create a new category or edit an existing category.
3. Insert an XSS payload into the category content, such as:
   `<img src=x onerror=alert(document.domain)>`
4. Save or publish the Categories.
5. View the category via the blog posts in the administrative panel or public blog page under the Categories section.
6. Notice the XSS payload executing automatically when the Category is viewed in the Blog Posts.

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/SAdVxK7b#kFW_sFOim_d_1AnVcpwvzOEV4MHv33LLooL4Xa_Ymgg

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-2026-34567

Weaknesses

▶ CWE-79

Credits

bugmithlegend	Reporter
peeefour	Reporter