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

Critical **bertugfahriozer** published **GHSA-rpjr-985c-qhvm** 18 hours 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 Group / Role Management Fields (Administrative Context Execution)

- Stored Cross-Site Scripting via Unsanitized Group / Role Management Inputs

Description

The application fails to properly sanitize user-controlled input within group and role management functionality. Multiple input fields (three distinct group-related fields) can be injected with malicious JavaScript payloads, which are then stored server-side.

These stored payloads are later rendered unsafely within privileged administrative views without proper output encoding, leading to stored cross-site scripting (XSS) within the role and permission management context.

Affected Functionality

- Group creation and editing functionality
- Role and permission assignment interfaces

- Storage and retrieval of group-related data

Attack Scenario

- An attacker injects a malicious XSS payload into one or more group-related input fields.
- The application stores these values without sanitization or encoding.
- An administrator views the group or role management interface.
- The payload executes automatically in the administrator's browser.

Impact

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

Endpoints:

- `/backend/users/groupList/`

Steps To Reproduce (POC)

1. Navigate to the Group / Role Management page
2. Insert an XSS payload into any of the three group-related input fields such as:
   `<img src=x onerror=alert(document.domain)>`
3. Save the group or role changes
4. View the group/role management page as an administrator
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/6QUEXDbR#JXzYXg9bef_NeSUVFB4R03UeXLtAVtYwTRsdrHLlokU

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-2026-34557

Weaknesses

▶ CWE-79

Credits

 **bugmithlegend** Reporter

 **peeefour** Reporter