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

Critical **bertugfahriozer** published **GHSA-v77r-xg3p-75g7** 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 Methods Management Fields (Global Persistent Payload Execution)

- Stored Cross-Site Scripting via Unsanitized Method Creation and Management Inputs
- Automatic Execution Across All Pages Where Method Is Rendered in Navigation

Description

The application fails to properly sanitize user-controlled input within the **Methods Management** functionality when creating or managing application methods/pages. Multiple input fields accept attacker-controlled JavaScript payloads that are stored server-side without sanitization or output encoding.

These stored values are later rendered directly into administrative interfaces and global navigation components without proper encoding, resulting in **Stored DOM-Based Cross-Site Scripting (XSS)**.

Critically, because created methods are automatically rendered inside the system's navigation/menu structure, the injected payload executes globally — meaning **every page visited where the malicious method appears in the menu triggers the XSS payload automatically**.

This significantly increases severity, as exploitation is not limited to a single view — it becomes a platform-wide persistent execution point.

Affected Functionality

- Methods creation functionality
- Methods management and listing functionality
- Administrative navigation rendering
- Permission-related UI rendering
- Global sidebar / menu rendering
- Storage and retrieval of method-related data

Affected Fields

The following fields accept unsanitized input and allow persistent JavaScript injection:

- Page Name
- Description
- Controller
- Method Name
- Seflink
- Page Order
- Symbol (FontAwesome 5)
- Permissions
- Parent Page
- Module

Attack Scenario

1. An attacker creates or edits a method.
2. The attacker injects a malicious XSS payload into any vulnerable field (e.g., Page Name).
3. The application stores the payload without sanitization or encoding.
4. The method is automatically rendered inside the application’s navigation/menu.
5. Every time any user visits any page where the menu is displayed, the malicious JavaScript executes automatically.

Because the navigation is globally rendered across backend pages, the XSS triggers on nearly every administrative page visit.

Impact

- Persistent Stored DOM XSS

- Automatic execution across multiple application pages
- Execution of arbitrary JavaScript in victims' browsers
- Privilege escalation when viewed by administrators
- Full administrator account takeover
- Full account takeover across all roles
- Session hijacking
- CSRF token theft
- Complete compromise of the entire application

This vulnerability is highly severe due to:

- Persistent storage
- Global rendering surface
- Automatic execution without user interaction
- High likelihood of administrator exposure

Endpoints:

- `/backend/methods/`
- `/backend/methods/create`

Steps To Reproduce (POC)

1. Navigate to Methods Management → Create Method
2. Insert the following payload into Page Name (or any vulnerable field):
   `<img src=x onerror=alert(document.domain)>`
3. Save the method
4. Navigate to any backend page
5. Observe the payload executing automatically wherever the malicious method appears in the menu
6. The XSS triggers across all pages where the navigation is rendered.

Remediation

- Never use `.html()`, `innerHTML`, or equivalent unsafe DOM sinks with untrusted data
- Implement strict output encoding (HTML entity encoding) before rendering user input
- Apply server-side input validation and sanitization
- Use contextual escaping depending on rendering context (HTML, attribute, JS, URL)
- Implement a strong Content Security Policy (CSP)
- Set cookies with HttpOnly, Secure, and SameSite flags
- Perform security review of all navigation rendering logic

Failure to properly encode and sanitize user-controlled method fields results in full application compromise through persistent global XSS.

Ready Video POC:

https://mega.nz/file/CFsiQAJS#cBSF2lCMD7YNZEKYEjw3T8YturY92oBvrdRQ08gmw2A

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-2026-34558

Weaknesses

▶ CWE-79

Credits

 **bugmithlegend** Reporter

 **peeefour** Reporter