Stored Cross-Site Scripting (Stored XSS) in Backend User Management Allows Session Hijacking and Full Administrative Account Compromise

Critical **bertugfahriozer** published **GHSA-fc4p-p49v-r948** 2 days ago

Package

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

Affected versions	Patched versions
<= 0.28.6.0	0.31.0.0

Description

Summary

A critical Stored Cross-Site Scripting (Stored XSS) vulnerability exists in the backend user management functionality. The application fails to properly sanitize user-controlled input before rendering it in the administrative interface, allowing attackers to inject persistent JavaScript code. This results in automatic execution whenever backend users access the affected page, enabling session hijacking, privilege escalation, and full administrative account compromise.

Details

The vulnerability resides in the backend user creation feature accessible via:

```
/backend/users
```

User-supplied input in the **name** and **surname** fields is stored without proper validation or sanitization. When this data is later rendered in the backend users listing page, it is injected directly into the HTML without output encoding.

Because of this, attackers can embed malicious JavaScript payloads that execute in the context of authenticated backend users.

This indicates missing contextual output escaping (e.g., HTML encoding) and insufficient input sanitization, leading to persistent script execution.

The vulnerability is particularly severe because:

- The payload is stored in the database (persistent XSS).
- The script executes automatically on page load.
- The affected page appears to be an administrative/backend interface, increasing the risk of privilege escalation.

PoC

Steps to reproduce:

1. Navigate to:

```
http://localhost:8080/backend/users
```

2. Click **Add New User**.

3. Create a new user.

4. In the **name** and **surname** fields, insert the following payload:

```
adnan"><img src=1 onerror=alert(document.cookie)><<e>img src=1
onerror=alert(document.cookie)>
```

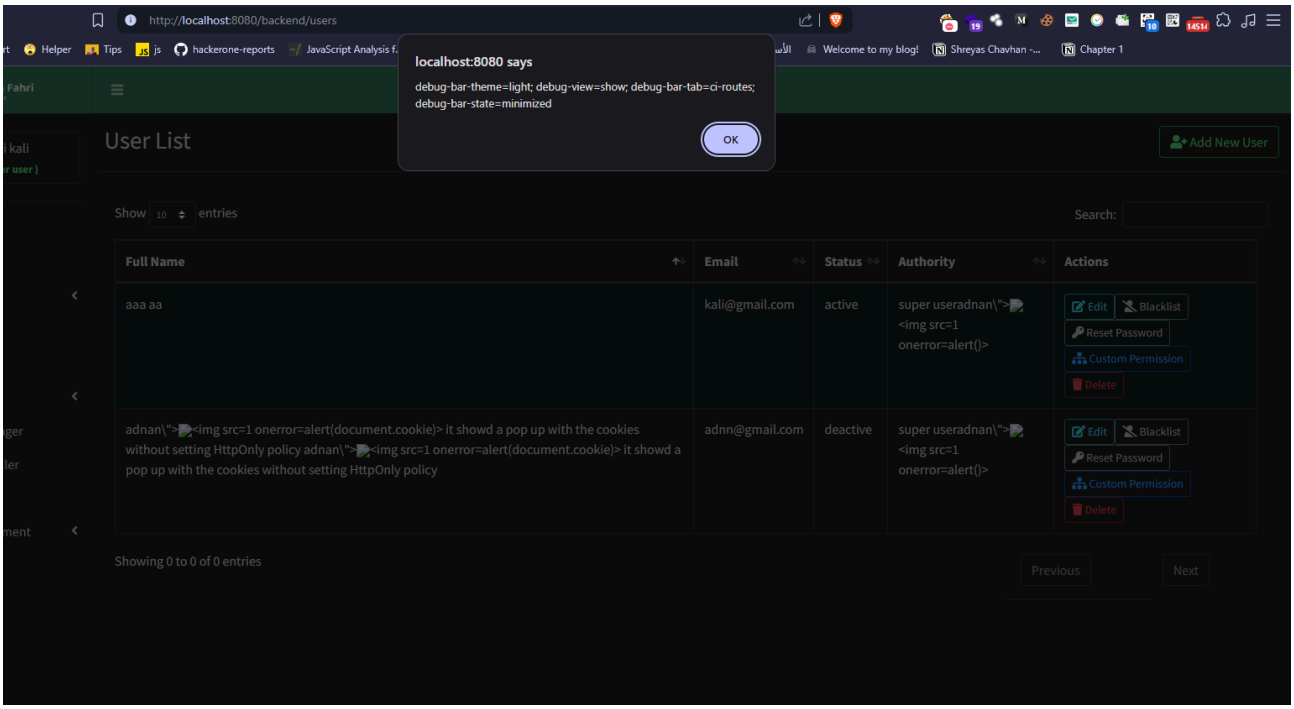
5. Save the user.

6. After saving, a popup displaying cookies will appear, demonstrating JavaScript execution.

7. Revisit:

```
http://localhost:8080/backend/users
```

8. The popup automatically triggers again, confirming that the malicious script is stored and executed persistently.



Impact

Severity: **Critical**

This vulnerability enables:

- Persistent execution of attacker-controlled JavaScript in privileged backend contexts.
- Theft of session cookies, potentially leading to full account takeover.
- Unauthorized actions performed on behalf of administrators (CSRF-like behavior via XSS).
- Privilege escalation if a high-privilege user views the page.
- Injection of keyloggers, credential harvesting scripts, or malicious redirects.
- Full compromise of backend administrative functionality depending on role permissions.

Since the payload executes automatically without user interaction once stored, exploitation requires minimal effort and can impact all backend users.

Severity

Critical 10.0 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	Low

User interaction	None
Scope	Changed
Confidentiality	High
Integrity	High
Availability	High

Learn more about base metrics

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

CVE ID

CVE-2026-34571

Weaknesses

▶ CWE-79

Credits

LAW6ZX7 Reporter

bugmithlegend Reporter