Account Deletion Module Full Persistent Unauthorized Access for All-Roles via Improper Session Invalidation (Logic Flaw)

High **bertugfahriozer** published **GHSA-4vxv-4xq4-p84h** last week

Package

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

Affected versions	Patched versions
<= 0.28.6.0	0.31.0.0

Description

Summary

Vulnerability: Improper Session Invalidation on Account Deletion (Broken Access Control / Logic Flaw)

- This vulnerability is caused by a backend logic flaw that maintains a false trust assumption that already-authenticated users remain trustworthy, even after their accounts are explicitly deleted. As a result, administrative security actions do not behave as intended, allowing persistent unauthorized access.

Description

The application fails to immediately revoke active user sessions when an account is **deleted**. Due to a logic flaw in the backend design, account state changes are enforced only during authentication (login), not for already-established sessions.

The system implicitly assumes that authenticated users remain trusted for the lifetime of their session. There is no session expiration or account expiration mechanism in place, causing deleted accounts to retain indefinite access until the user manually logs out. This behavior breaks the intended access control policy and results in persistent unauthorized access, representing a critical security flaw.

Affected Functionality

- User session management and authentication logic
- Account **deletion** mechanism
- All authenticated endpoints, including administrative and content interfaces

Attack Scenario

- A user logs into the application.
- An administrator **deletes** the user account.
- The user remains fully logged in and can continue performing all actions allowed by their role indefinitely, as there is no session expiration.
- The user can continue invoking backend methods, triggering application actions, accessing sensitive interfaces (including user management if permitted), and interacting with the system as if the account were still active.
- Access is only lost if the user manually logs out, which may never occur.

Impact

- **Unauthorized Continued Access:** Deleted users retain full access indefinitely, violating intended access control and expected security behavior.
- **Bypass of Administrative Controls:** Administrative actions (**deletion**) fail to immediately restrict active sessions.
- **Logic Flaw Resulting in Broken Behavior:** Backend authorization logic relies on a flawed trust assumption that authenticated users remain valid, enforcing account state only at login.
- **Full Functional Access Retained:** Deleted users can continue invoking application methods, executing actions, interacting with protected endpoints, and using the system exactly as before deletion.
- **Privilege Abuse:** Users with elevated roles (moderator, editor, administrator) can continue performing privileged actions after account deletion, including accessing user management interfaces and modifying application state.
- **Service Disruption Potential:** Persistent access allows attackers to disrupt services, manipulate content, or interfere with normal application operations.
- **Attack Persistence:** Attackers can maintain access indefinitely, increasing the risk of data exfiltration, unauthorized modifications, or further privilege escalation.
- **False Sense of Remediation:** Administrators may believe a threat has been mitigated while the deleted user remains active within the system.

Endpoint Example: Any endpoint accessible to authenticated users, including dashboards, administrative interfaces, user management pages, and API endpoints.

Steps To Reproduce (PoC)

1. Create or use an existing user account.

2. Log into the application using this account.
3. From an administrative account, **delete** the logged-in user account.
4. Observe that the target user remains authenticated.
5. Verify that the user can still access protected functionality, invoke actions, and interact with the application as before.
6. Confirm that the user only loses access after manually logging out (if they choose to do so).

Remediation

- Immediately invalidate all active sessions when an account is **deleted**.
- Enforce account status checks on every authenticated request, not only during login.
- Introduce proper session expiration or account expiration mechanisms to prevent indefinite access.
- Correct the backend logic flaw to ensure access control behavior aligns with intended security design and does not rely on unsafe trust assumptions.

Ready Video POC:

https://mega.nz/file/7dlUTQAB#0oXOapF5XYN4DRRG1xYj6DajmuP72MpMdsHqbVBMmWw

Severity

High 8.8 / 10

CVSS v3 base metrics

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

Learn more about base metrics

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

CVE ID

CVE-2026-34570

Weaknesses

- CWE-284
- CWE-613
- CWE-1254

Credits

bugmithlegend Reporter