Lab 3 — Individual Bug Report

Security Vulnerabilities in Kimai 2.56.0

Course	WAS705NBB — Web Application Security
Institution	Seneca Polytechnic
Student	Abdul-Ramon Adeyemi Folarin
Group	1
Date	2 June 2026
Target	Kimai 2.56.0 (self-hosted, Docker)
Scope	Authorized academic security assessment

#	Title	Severity	Status
1	Mass Assignment: `systemAccount` writable by `ROLE_TEAMLEAD`	MEDIUM	Patched (2.57.0), credited

Contents

1 Bug 1 — Mass Assignment: `systemAccount` Field Writable by `ROLE_TEAMLEAD` 3
1.1 Summary . . . 3
1.2 Root Cause . . . 3
1.3 Proof of Concept . . . 3
1.4 Impact . . . 4
1.5 Remediation . . . 4
1.6 Disclosure and Confirmation . . . 5

2 Disclosure 6
2.1 Bug 1 — systemAccount Mass Assignment . . . 6

3 References 7

Bug 1 — Mass Assignment: `systemAccount` Field Writable by `ROLE_TEAMLEAD`

Severity	MEDIUM
CVSS v3.1	4.3 `AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:L/A:N`
CWE	CWE-915 — Improperly Controlled Modification of Dynamically-Determined Object Attributes
Endpoint	`PATCH /api/users/{id}`
Status	Fixed in Kimai 2.57.0 (PR #5929). Student credited in release notes.

Summary

An authenticated user with `ROLE_TEAMLEAD` can modify the `systemAccount` boolean flag on their own profile by submitting `PATCH /api/users/{id}` with `{"systemAccount":true}` in the request body. The server accepts the value, persists it to the database, and returns `HTTP 200`. No administrator interaction is required. The field has no permission guard in the form, API form, or controller.

Root Cause

The `systemAccount` field is registered unconditionally in `UserEditType.php` with no `isGranted()` check. The API form (`UserApiEditForm`) inherits it, and `UserController.php` applies conditional option flags to every other sensitive field (`roles`, `active_flag`, `preferences`, `supervisor`, `password_reset`) but not `systemAccount`.

Listing 1: UserController.php — conditional guards exist for all fields except systemAccount

```
if (!$this->isGranted('roles', $profile)) { $options['include_roles'] = false; }
if ($profile->getId() === $user->getId()) { $options['include_active_flag'] = false;
    }
if (!$this->isGranted('preferences', $profile)) { $options['include_preferences'] =
    false; }
if (!$this->isGranted('supervisor', $profile)) { $options['include_supervisor'] =
    false; }
if (!$this->isGranted('password', $profile)) { $options['include_password_reset'] =
    false; }
// systemAccount: no guard. Accepted from any authenticated caller.
```

Proof of Concept

The exploit was demonstrated through four named Caido replay sessions:

1. **Baseline** — `GET /api/users/4` as `carol` (`ROLE_TEAMLEAD`) confirms `systemAccount: false`.
2. **Exploit** — `PATCH /api/users/4` with body `{"systemAccount":true}` returns `HTTP 200` with `systemAccount: true`.
3. **Persistence check** — Independent `GET` using the admin bearer token confirms the value is written to the database, not merely echoed.
4. **Revert** — `PATCH` with `{"systemAccount":false}` restores original state; server accepts bidirectional writes.

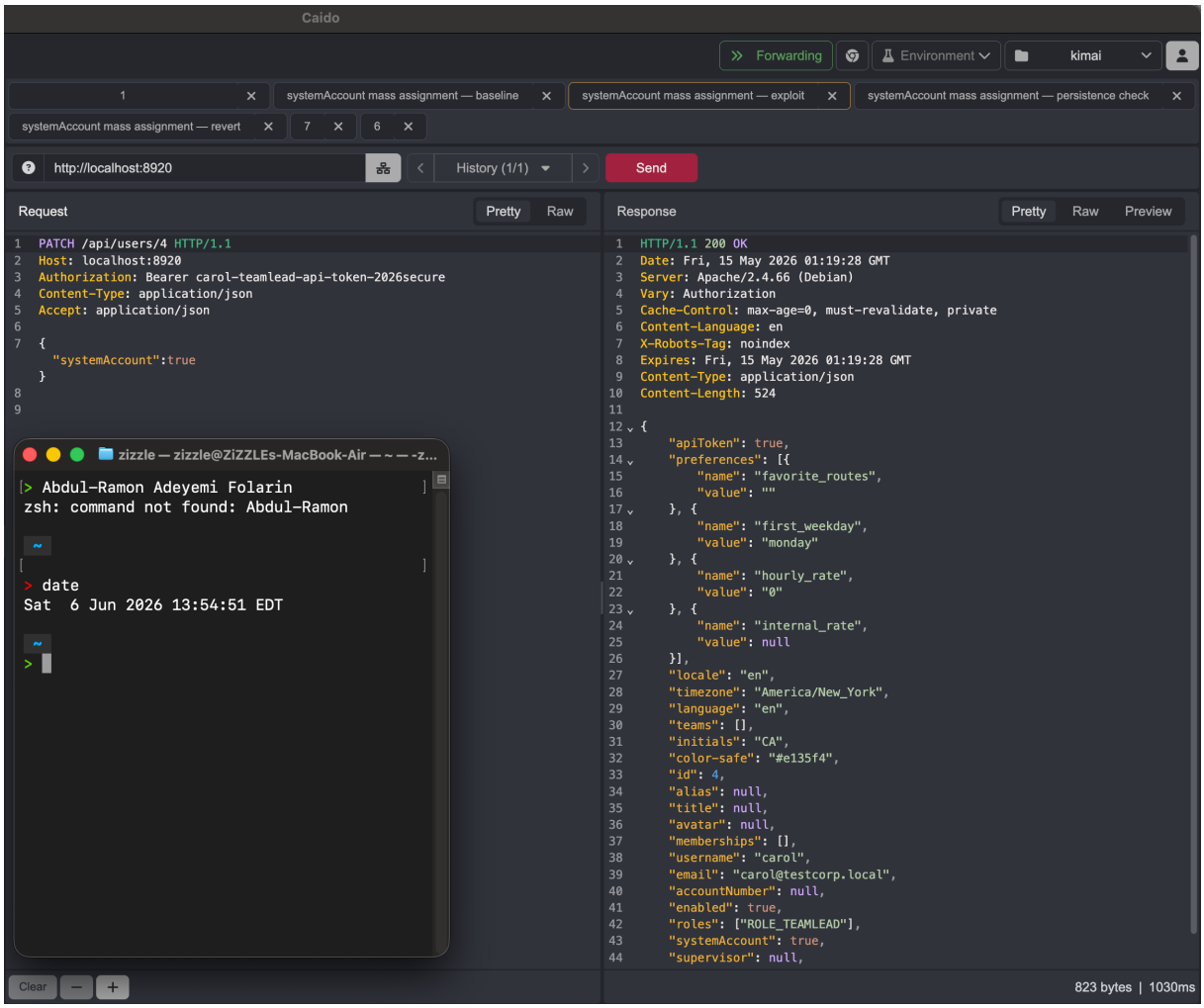


Figure 1: Caido replay — exploit. `PATCH /api/users/4` with 26-byte body accepted. Response shows `systemAccount: true`. HTTP 200.

Impact

A user who flips `systemAccount` to `true` is excluded from all per-user billing and time reports (week, month, year, customer-monthly) because all four report controllers hardcode `$query->setSystemAccount(false)`. Their records remain in the database but do not appear in any management-facing view. Additionally, `UserRepository::getQueryBuilderForFormType()` removes system accounts from all user-picker dropdowns, and `User::canSeeUser()` prevents peer `ROLE_TEAMLEAD` users from accessing the attacker's profile.

Remediation

Add `include_system_account` to the controller option flags, gated on `ROLE_ADMIN`, and condition the `UserEditType` field registration on that option.

Disclosure and Confirmation

- Vulnerability identified and disclosed to `security@kimai.org` on 19 May 2026.
- Fixed in Kimai 2.57.0 (released approximately 19 May 2026) under PR #5929 alongside five related security fixes.
- Student credited by name in the PR #5929 description and the official 2.57.0 release notes. The PR description states: *"Prevent regular users from turning their account into a `systemAccount` (thanks Abdul-Ramon)."* The release notes credit: *"Involved in this release: @cheriimoya and @kevinpapst and @offset and @Mitchell45 and Abdul-Ramon."*

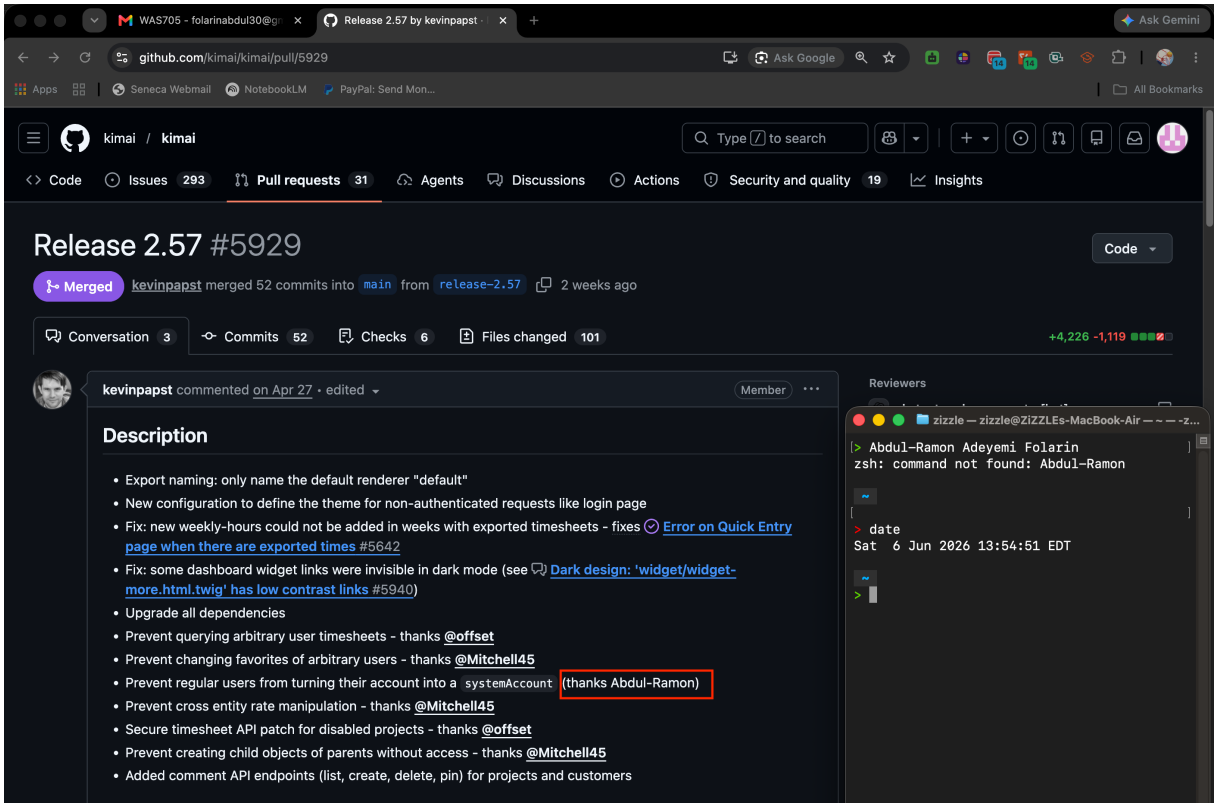


Figure 2: Kimai 2.57.0 release notes on GitHub showing student credit alongside core team members. Source: `github.com/kimai/kimai/releases/tag/2.57.0`.

Disclosure

Bug 1 — systemAccount Mass Assignment

Date	Event
14 May 2026	Vulnerability identified during manual API testing of Kimai 2.56.0
15 May 2026	Confirmed via four Caido replay sessions (baseline, exploit, persistence, revert)
15 May 2026	Source-level root cause verified against `kimai/kimai` main branch
19 May 2026	Responsible disclosure submitted to `security@kimai.org`
∼19 May 2026	Patched in Kimai 2.57.0 (PR #5929); student credited in release notes

References

- Kimai 2.57.0 Release Notes — student credited in release notes and PR #5929
- PR #5929 — Security fixes batch (2.57.0)
- CWE-915 — Improperly Controlled Modification of Dynamically-Determined Object Attributes
- OWASP Web Security Testing Guide v4.2
- Kimai Roles & Permissions Documentation
- `src/Form/UserEditType.php` — kimai/kimai main branch
- `src/Form/API/UserApiEditForm.php` — kimai/kimai main branch
- `src/API/UserController.php` — kimai/kimai main branch

Classification: — Academic Security Assessment
This document is submitted in partial fulfilment of WAS705NBB Lab 3 at Seneca Polytechnic. Distribution is restricted to course instructors and the assessment team. The reported vulnerability details are now public following the Kimai 2.57.0 release.