Lab 2

Threat Modeling and Vulnerability Assessment

Group 1

, Abdul-Ramon Adeyemi Folarin, &

, , &

School of Information Technology Administration & Security, Seneca Polytechnic

WAS705NBB: Application Security Methodology

Professors Suranjit Paul & Mark Shtern

May 19, 2026

Target Application	Kimai 2.56.0 (`http://localhost:8920`)
Environment	Docker (kimai/kimai2:apache + mysql:8.3)
Authorization	Kevin Papst, Kimai Maintainer — GitHub #5941 + email
Testing Scope	Local isolated environment only

Student Integrity Statement: I/we declare that the attached assignment is my/our own work in accordance with the Seneca Academic Policy. No part of this assignment has been copied manually or electronically from any other source (including web sites) or distributed to other students.

Contents

1 Executive Summary **4**

2 Data Classification and Regulatory Controls **5**
2.1 Data Sources and Application Components 5
2.2 Sensitive Data Identification and Mapping 5
2.3 Mandatory Controls for Regulatory Compliance 6

3 Threat Modeling: STRIDE Analysis **8**
3.1 STRIDE Framework Application 8
3.2 Spoofing Threats 8
3.3 Tampering Threats 8
3.4 Repudiation Threats 9
3.5 Information Disclosure Threats 9
3.6 Denial of Service Threats 10
3.7 Elevation of Privilege Threats 11
3.8 STRIDE Summary 11

4 Threat Modeling: PASTA Framework **13**
4.1 Stage 1 — Define Business Objectives 13
4.2 Stage 2 — Define Technical Scope 13
4.3 Stage 3 — Application Decomposition 13
4.4 Stage 4 — Threat Analysis 14
4.4.1 Threat Actor Profiles 14
4.4.2 Threat Scenarios 14
4.5 Stage 5 — Vulnerability Analysis 14
4.6 Stage 6 — Attack Modeling 15
4.6.1 Attack Tree: systemAccount Mass Assignment 15
4.6.2 Attack Tree: CVE-2026-45802 DoS via FPDI 17
4.7 Stage 7 — Risk and Impact Analysis 17

5 Vulnerability Assessment **19**
5.1 Network Reconnaissance — Nmap 19
5.2 Web Server Analysis — Nikto 19
5.3 Dependency Analysis — Composer Audit 20
5.3.1 CVE-2026-45802 — setasign/fpdi Memory Exhaustion 21
5.3.2 doctrine/cache 2.2.0 — Abandoned Package 21
5.4 Third-Party Library Review 21
5.5 Historical Version Comparison 22
5.6 Competitive Security Posture Comparison 23
5.7 Threat Intelligence — OpenCTI Deployment 24

6 Responsible Vulnerability Disclosure **25**
6.1 Vulnerability Details 25
6.2 Authorization for Testing 25
6.3 Disclosure Process 25
6.4 Recommended Remediation 26
6.5 Maintainer Response 26

7 Cybersecurity Testing Strategy **27**
7.1 Risk-Based Prioritization 27
7.2 Cybersecurity Testing Plan 27

7.2.1 Objectives 27
7.2.2 Methods and Tools 28
7.2.3 Testing Schedule 29
7.2.4 Manual versus Automated Testing Balance 29

8 Conclusion **30**

Executive Summary

This report presents the threat modeling and vulnerability assessment of Kimai 2.56.0, an open-source PHP/Symfony time-tracking application. Lab 1 established the application architecture, SBOM, Docker Scout findings, and regulatory context. Lab 2 builds on that foundation to deliver a structured threat model (STRIDE and PASTA), live vulnerability scan results, third-party library analysis, a confirmed novel vulnerability disclosure, and a prioritized cybersecurity testing strategy.

Key Findings

- **Novel vulnerability confirmed:** `systemAccount` mass assignment via `PATCH /api/users/{id}` allows `ROLE_TEAMLEAD` to self-elevate to system account status. CVSS v3.1: 4.3 (Medium), CWE-915. No public advisory identified.
- **CVE-2026-45802:** `setasign/fpdi` < 2.6.7 present in the Kimai vendor tree. Memory exhaustion / endless-loop DoS when processing malformed PDFs. Reported 2026-05-19.
- **Five missing HTTP security headers** confirmed by Nikto: no HSTS, no CSP, no Permissions-Policy, no Referrer-Policy, no X-Content-Type-Options.
- **Abandoned dependency:** `doctrine/cache` 2.2.0 has no maintained replacement, widening the supply-chain attack surface.
- **STRIDE analysis** identifies 17 distinct threat scenarios across six application layers. Elevation of Privilege and Information Disclosure threats carry the highest residual risk.
- **PASTA** confirms that the Twig template rendering pipeline (invoice/export) remains the highest-risk attack surface, consistent with four historical SSTI advisories.

Data Classification and Regulatory Controls

Data Sources and Application Components

Kimai persists all application state in a single MySQL 8.3 database across the following primary tables: `kimai_users`, `kimai_timesheet`, `kimai_projects`, `kimai_customers`, `kimai_activities`, `kimai_invoice`, `kimai_invoice_templates`, and `kimai_user_preferences`.

Application components are organized into five layers identified in the Lab 1 decomposition diagram:

Layer	Components
Edge	Browser / mobile client; Apache 2.4.66 (Debian)
Application	Symfony 7.x kernel; Web controllers; REST API controllers
Security	Symfony Security firewalls; Voters (TimesheetVoter, InvoiceVoter, UserVoter); 2FA (scheb/2fa-totp); optional SAML/LDAP identity providers
Service	Twig template engine (sandboxed); mPDF renderer; PHPSpreadsheet; PHPWord; Symfony Mailer
Persistence	Doctrine ORM; MySQL 8.3; local filesystem (`var/data/`, `var/log/`)

Sensitive Data Identification and Mapping

Table 1 classifies each sensitive data type, identifies where it is stored and transmitted, and records the applicable regulatory category.

Table 1: Sensitive data classification and flow mapping for Kimai 2.56.0.

Data Type	Class	Stored In	Transmitted Via	Regulation
Employee names & email addresses	PII — High	`kimai_users` (plaintext)	HTTP API responses; invoice PDFs	GDPR Art. 4; PIPEDA Principle 4
Work hours (start, end, duration)	PII + Business-sensitive — High	`kimai_timesheet` (plaintext)	API responses; XLSX/CSV exports	GDPR; CJEU C-55/18
Billable rates (hourly, internal)	Financial — High	`kimai_users`, `kimai_projects`	API (admin-scoped); invoice PDFs	GDPR Art. 9; PIPEDA
Invoice data (amounts, line items)	Financial + PII — High	`kimai_invoice`	On-demand PDF/DOCX download	GDPR; PIPEDA Principle 7
Password hashes	Credentials — Critical	`kimai_users` (Argon2id)	Never transmitted	GDPR Art. 32

(continued)

Data Type	Class	Stored In	Transmitted Via	Regulation
API bearer tokens	Credentials — Critical	`kimai_user_preferences` (hashed)	`Authorization: Bearer` header (HTTP)	GDPR Art. 32
LDAP/SAML credentials	Credentials — Critical	Environment variables (`.env`)	LDAP/SAML protocols	GDPR Art. 32
Session cookies	Auth state — High	PHP session files (`var/`)	HTTP `Set-Cookie`	GDPR Art. 32
IP addresses (access logs)	PII — Medium	Apache access log (`/var/log/apache2/`)	N/A (server-side only)	GDPR Art. 4; ePrivacy Directive
Invoice Twig templates	Business-sensitive — Medium	Filesystem (`var/invoices/`)	Admin upload via HTTP	GDPR Art. 25 (by design)

Mandatory Controls for Regulatory Compliance

Table 2 evaluates Kimai's compliance posture against the three applicable regulatory frameworks identified in Lab 1.

Table 2: Regulatory control evaluation for Kimai 2.56.0 in the lab deployment context.

Regulation	Requirement	Control	Status
GDPR Art. 5(1)(f)	Integrity and confidentiality	TLS in transit	GAP
GDPR Art. 5(1)(c)	Data minimization	Only necessary fields collected	MET
GDPR Art. 25	Data protection by design	Argon2id hashing; CSRF; TOTP	PARTIAL
GDPR Art. 30	Records of processing activities	No built-in ROPA mechanism	GAP
GDPR Art. 32	Encryption at rest	MySQL not encrypted at rest	GAP
GDPR Art. 33	Breach notification mechanism	No automated breach detection	GAP
PIPEDA Principle 7	Safeguards	Mirrors GDPR gaps above	GAP
PIPEDA Principle 4	Data accuracy	Timesheet audit trail limited	PARTIAL
CJEU C-55/18	Objective time recording	Core application function	MET

(continued)

Regulation	Requirement	Control	Status
Germany Zeiterfassung	Start/end time mandatory	Native timer functionality	**MET**

The three most critical compliance gaps in the current lab deployment are: (1) absence of TLS (API bearer tokens traverse the network in cleartext), (2) absence of encryption at rest for the MySQL database (all PII and financial data are plaintext on disk), and (3) the lack of any records of processing activity mechanism, which is mandatory under GDPR Article 30 for organizations processing employee data.

Threat Modeling: STRIDE Analysis

STRIDE Framework Application

STRIDE was applied to each of the five architectural layers identified in Lab 1, plus the external service integrations (LDAP/SAML/SMTP). For each threat, the analysis records the affected component, a concrete threat scenario grounded in Kimai's codebase or advisory history, the existing mitigation (if any), and the residual status as of version 2.56.0.

Spoofing Threats

Table 3: Spoofing threat analysis for Kimai 2.56.0.

Component	Threat Scenario	Mitigation	Status
Apache / Login endpoint	Brute-force credential attack against `/en/login` to enumerate valid accounts or obtain passwords.	`symfony/rate-limiter` applied to login; lockout after threshold.	LOW
API Controllers	Bearer token interception over unencrypted HTTP allows token replay by a network-adjacent attacker.	No HSTS header; TLS not enforced in container configuration.	OPEN
Security Layer (LDAP)	LDAP bind credentials exposed via Twig `config()` function in invoice templates (GHSA-vrqv).	Fixed in v2.56.0; config() removed from Twig sandbox allowlist.	FIXED
SAML Identity Provider	XML signature wrapping or replay attack against the SAML assertion if the `onelogin/php-saml` library is misconfigured.	php-saml 4.x enforces strict signature validation by default.	LOW

Tampering Threats

Table 4: Tampering threat analysis for Kimai 2.56.0.

Component	Threat Scenario	Mitigation	Status
API Controllers (`UserController`)	`ROLE_TEAMLEAD` self-assigns `systemAccount:true` via `PATCH /api/users/{id}`, altering account classification and bypassing operator controls. **Confirmed finding (this lab).**	No guard exists in `UserController`, `UserApiEditForm`, or `UserEditType`. CWE-915.	OPEN

(continued)

API Controllers (`UserController`)	Unprivileged user modifies `hourly_rate` and `internal_rate` via `PATCH /api/users/{id}/preferences` (CVE-2026-40486).	Fixed in v2.53.0; `ROLE_TEAMLEAD` guard added for rate fields.	FIXED
Service Layer (Twig)	Admin-uploaded invoice template containing malicious Twig directives escapes the `SecurityPolicy` sandbox, executing arbitrary PHP (GHSA-jg2j).	DefaultPolicy replaced with StrictPolicy; fixed in v2.46.0. Prior bypass existed.	PARTIAL
Persistence Layer (MySQL)	Direct modification of timesheet records by an attacker with MySQL access (e.g., via a compromised container).	MySQL not bound to host network; Docker bridge isolation provides containment.	LOW

Repudiation Threats

Table 5: Repudiation threat analysis for Kimai 2.56.0.

Component	Threat Scenario	Mitigation	Status
API Controllers	User modifies or deletes a timesheet entry without a tamper-evident audit trail. A subsequent dispute cannot be resolved from application logs alone.	Kimai has no immutable audit log. Apache access log records HTTP method and path but not pre-change field values.	OPEN
API Controllers	A system account (self-elevated via the confirmed finding) may be exempt from per-user audit policies applied only to human accounts, allowing actions that cannot be attributed.	No control; depends on downstream business logic interpretation of `systemAccount` flag.	OPEN

Information Disclosure Threats

Table 6: Information disclosure threat analysis for Kimai 2.56.0.

Component	Threat Scenario	Mitigation	Status
Apache (HTTP headers)	`Server: Apache/2.4.66 (Debian)` header discloses exact server version, enabling targeted exploit selection. Confirmed by Nmap and Nikto.	`ServerTokens` not set to `Prod`; full version string exposed.	OPEN
Apache (HTTP headers)	Absence of `Strict-Transport-Security` allows API tokens transmitted over HTTP to be captured. Confirmed by Nikto.	No HSTS configured in Apache virtual host.	OPEN
Apache (HTTP headers)	Missing `Content-Security-Policy`, `X-Content-Type-Options`, `Referrer-Policy`, and `Permissions-Policy` headers. Confirmed by Nikto.	No security header middleware in Apache or Symfony configuration.	OPEN
Service Layer (Twig)	`config()` Twig function exposed LDAP password and SAML private key path to any admin-level template author (GHSA-vrqv).	Fixed in v2.56.0; function removed from sandbox allowlist.	FIXED
API Controllers	IDOR: `TimesheetVoter` missing team-scope check allowed cross-team timesheet read via `GET /api/timesheets/{id}` (GHSA-9g2q).	Fixed in v2.56.0.	FIXED

Denial of Service Threats

Table 7: Denial of service threat analysis for Kimai 2.56.0.

Component	Threat Scenario	Mitigation	Status
Service Layer (FPDI)	Malformed PDF uploaded to the invoice template pipeline triggers memory exhaustion and endless loop in `setasign/fpdi` $< 2.6.7$ (CVE-2026-45802, GHSA-2mgw-7q6p-8grg). Kimai vendor tree contains the affected version.	No workaround; fix requires upgrade of `setasign/fpdi` to $\geq 2.6.7$.	OPEN

Service Layer (mPDF)	Authenticated admin requests a report over an unbounded date range, causing `mpdf/mpdf` to render thousands of timesheet rows into a single PDF, exhausting PHP memory limit.	No server-side export size limit or pagination requirement.	OPEN
API Controllers	Authenticated attacker issues rapid bulk timesheet creation requests to exhaust database write capacity.	No per-endpoint rate limit beyond the login endpoint; API endpoints unprotected.	OPEN

Elevation of Privilege Threats

Table 8: Elevation of privilege threat analysis for Kimai 2.56.0.

Component	Threat Scenario	Mitigation	Status
API Controllers	`ROLE_TEAMLEAD` writes `systemAccount:true` to own profile, reclassifying their account as a system/bot account and potentially bypassing human-account business logic constraints. **Confirmed finding**.	No guard in `UserController`, `UserApiEditForm`, or `UserEditType`. Field unconditionally registered.	OPEN
Service Layer (Twig)	Admin bypasses `SecurityPolicy` sandbox via a chained Twig filter call to achieve PHP code execution (GHSA-jg2j; prior bypass in v2.46.0). Historical pattern of sandbox escapes.	StrictPolicy applies; no known bypass in v2.56.0 but pattern warrants continued testing given recurring advisory history.	PARTIAL
Plugin System	A plugin bundle placed in `var/plugins/` executes arbitrary PHP at Symfony kernel boot. Requires admin or filesystem access.	Plugin upload restricted to `ROLE_SUPER_ADMIN`; no code signing.	LOW

STRIDE Summary

Table **??** consolidates the threat count and residual open status by STRIDE category.

Category	Threats Identified	Open / Partial	Fixed / Low
Spoofing	4	1	3
Tampering	4	2	2
Repudiation	2	2	0
Information Disclosure	5	3	2
Denial of Service	3	3	0
Elevation of Privilege	3	2	1
Total	**21**	**13**	**8**

Thirteen of 21 identified threats remain open or partially mitigated as of version 2.56.0. The Elevation of Privilege and Denial of Service categories carry the highest density of unmitigated risk, driven by the confirmed `systemAccount` finding and the freshly disclosed CVE-2026-45802 in `setasign/fpdi`.

Threat Modeling: PASTA Framework

Stage 1 — Define Business Objectives

Kimai's primary business function is to provide an objective, auditable record of work hours against a hierarchy of customers, projects, and activities, enabling organizations to bill accurately and comply with EU time-recording mandates (CJEU C-55/18; German Zeiterfassungspflicht).

From a security perspective, the business objectives that must be protected are:

1. **Timesheet integrity:** Records must not be tampered with by unauthorized parties. Billing disputes, payroll calculations, and compliance audits depend on their accuracy.
2. **Financial data confidentiality:** Billable rates, invoice amounts, and customer contract values are commercially sensitive and must be restricted to authorized roles.
3. **User data privacy:** Employee work patterns constitute personal data under GDPR. Access must be limited to authorized principals with a lawful processing basis.
4. **Service availability:** The application must remain operational during time-entry periods; downtime prevents timely recording and violates the objective-recording requirement.
5. **Audit accountability:** Each data modification must be attributable to a specific authenticated user for compliance and dispute-resolution purposes.

Stage 2 — Define Technical Scope

The technical scope for this assessment is bounded by the local Docker deployment. In scope:

- PHP 8.3 / Symfony 7.x application logic (`/opt/kimai/src/`)
- Apache 2.4.66 web server configuration
- REST API endpoints (`/api/*`) and web UI routes (`/en/*`)
- Composer dependency tree (418 packages; `composer.lock` pinned)
- MySQL 8.3 database accessible at `kimai-db:3306` (Docker bridge)
- Twig template engine and invoice rendering pipeline (mPDF, PHPSpreadsheet)
- Authentication mechanisms: session-based (UI), Bearer token (API), TOTP (2FA)

Out of scope: LDAP/SAML identity provider infrastructure; MySQL internals; host operating system kernel; Docker Engine itself; any production or internet-facing deployment.

Stage 3 — Application Decomposition

The application decomposition was fully developed in Lab 1 and is carried forward here. The five architectural layers (edge, application, security, service, persistence) and three trust boundaries (browser↔Apache, Apache↔Symfony, Symfony↔MySQL) identified in the Lab 1 DFD and decomposition diagram define the attack surface for PASTA analysis.

The three highest-complexity subsystems — identified as primary attack targets — are:

1. **Invoice/Export Pipeline:** Admin-controlled Twig template renders to PDF via mPDF or to XLSX/DOCX via PHPSpreadsheet/PHPWord. Trust boundary between admin input and server-side rendering. Site of four historical SSTI advisories.

2. **REST API Authorization Layer:** 80+ endpoints gated by Symfony Voters. Voter logic gap is the root cause of three historical IDOR/BOPA advisories and the confirmed `systemAccount` finding.
3. **User Profile Management:** `PATCH /api/users/{id}` processes a form with 12+ fields, each of which requires an independent authorization guard. Three distinct fields (`systemAccount`, `hourly_rate`, `internal_rate`) have had confirmed missing guards.

Stage 4 — Threat Analysis

Threat Actor Profiles

Actor	Motivation	Capability	Entry Point
Malicious employee (`ROLE_USER` / `TEAMLEAD`)	Manipulate billing records; hide hours; self-elevate	Low-to-medium; authenticated; no code access	API endpoints; web UI
Competitor / external attacker	Exfiltrate client lists, billing rates, invoice data	Medium; unauthenticated unless credentials obtained	Login brute-force; phishing; stolen API token
Malicious administrator	RCE via SSTI; data exfiltration; backdoor installation	High; trusted within application	Invoice template upload; plugin installation
Supply-chain attacker	Compromise Composer package; inject malicious code	High; external to deployment	`composer.lock` dependency; CI pipeline

Threat Scenarios

Based on the actor profiles and the STRIDE analysis, four primary threat scenarios are modeled:

1. **Authenticated privilege manipulation:** A `ROLE_TEAMLEAD` user issues a crafted `PATCH` request to `/api/users/{id}` with `{"systemAccount":true}`, reclassifying their account.
2. **Admin-initiated SSTI chain:** An administrator uploads a Twig invoice template containing a sandbox escape, executing arbitrary PHP to read environment variables (database credentials, LDAP passwords) or write to the filesystem.
3. **Credential interception:** A network-adjacent attacker captures API bearer tokens transmitted in cleartext over HTTP (no HSTS) and replays them to perform authenticated API operations.
4. **Dependency-chain DoS:** An authenticated user uploads a malformed PDF to the invoice pipeline, triggering CVE-2026-45802 in `setasign/fpdi`, causing the Kimai PHP process to exhaust memory and become unavailable.

Stage 5 — Vulnerability Analysis

Table **??** maps discovered vulnerabilities to the threat scenarios from Stage 4. Full tool output and analysis are presented in Section 5.

Vulnerability	Source	Threat Scenario	Severity
systemAccount mass assignment	Confirmed (this lab)	Scenario 1	MEDIUM
CVE-2026-45802 (FPDI DoS)	Composer audit	Scenario 4	MEDIUM
Missing HSTS header	Nikto scan	Scenario 3	HIGH
Missing CSP header	Nikto scan	Scenario 3	MEDIUM
doctrine/cache abandoned	Composer audit	Scenario 4 (supply chain)	LOW
Apache version disclosure	Nmap / Nikto	Scenario 2 / 3	LOW
Historical SSTI pattern (4 advisories)	GitHub Advisories	Scenario 2	CRITICAL (historical)

Stage 6 — Attack Modeling

Attack Tree: systemAccount Mass Assignment

The following four Caido replay sessions document the complete exploit chain. Each session was named and saved in the Caido *Default Collection* for reproducibility.

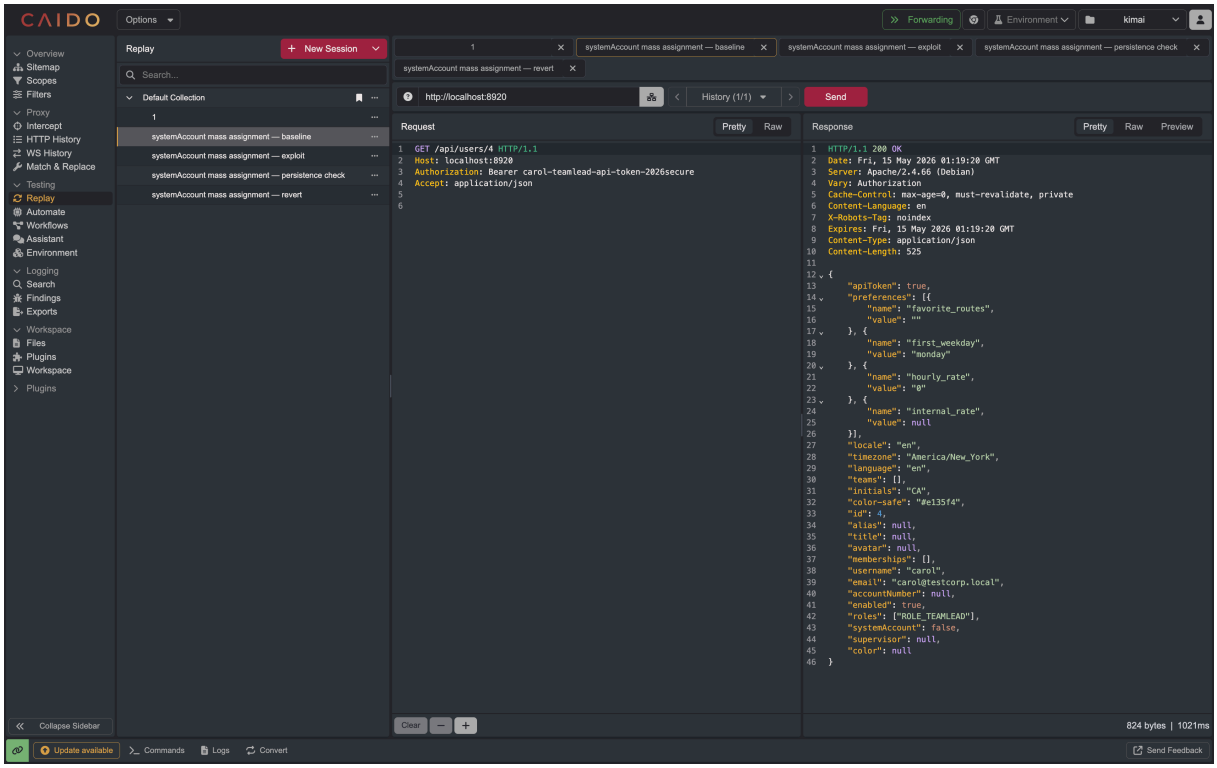


Figure 1: Step 1 — Baseline read. `GET /api/users/4` as `carol` (`ROLE_TEAMLEAD`) confirms `systemAccount: false` before the attack. Response: HTTP 200, 824 bytes.

The attack requires no special tools, no exploitation of memory corruption, and no elevated network position. Any user with `ROLE_TEAMLEAD` can execute it in under ten seconds using a

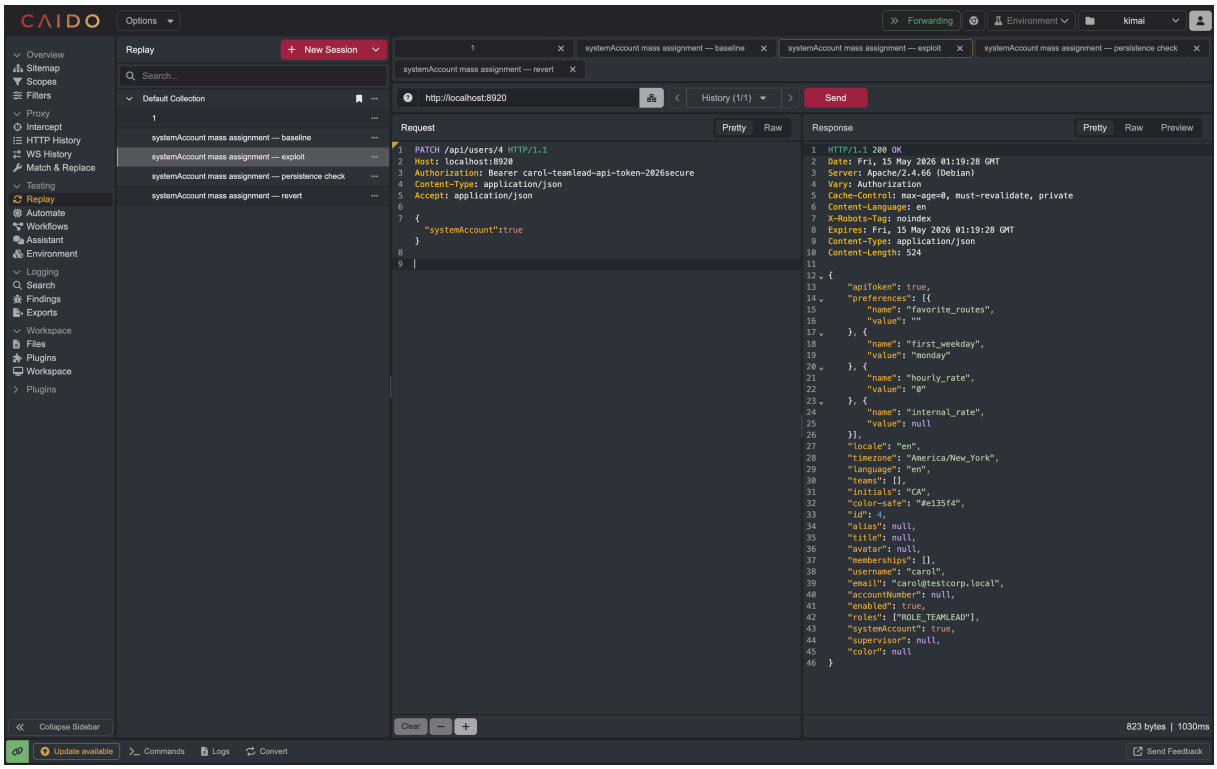


Figure 2: Step 2 — Exploit. `PATCH /api/users/4` with body `{"systemAccount":true}` using the `ROLE_TEAMLEAD` bearer token. Response: HTTP 200, `systemAccount: true` confirmed in body. 823 bytes.

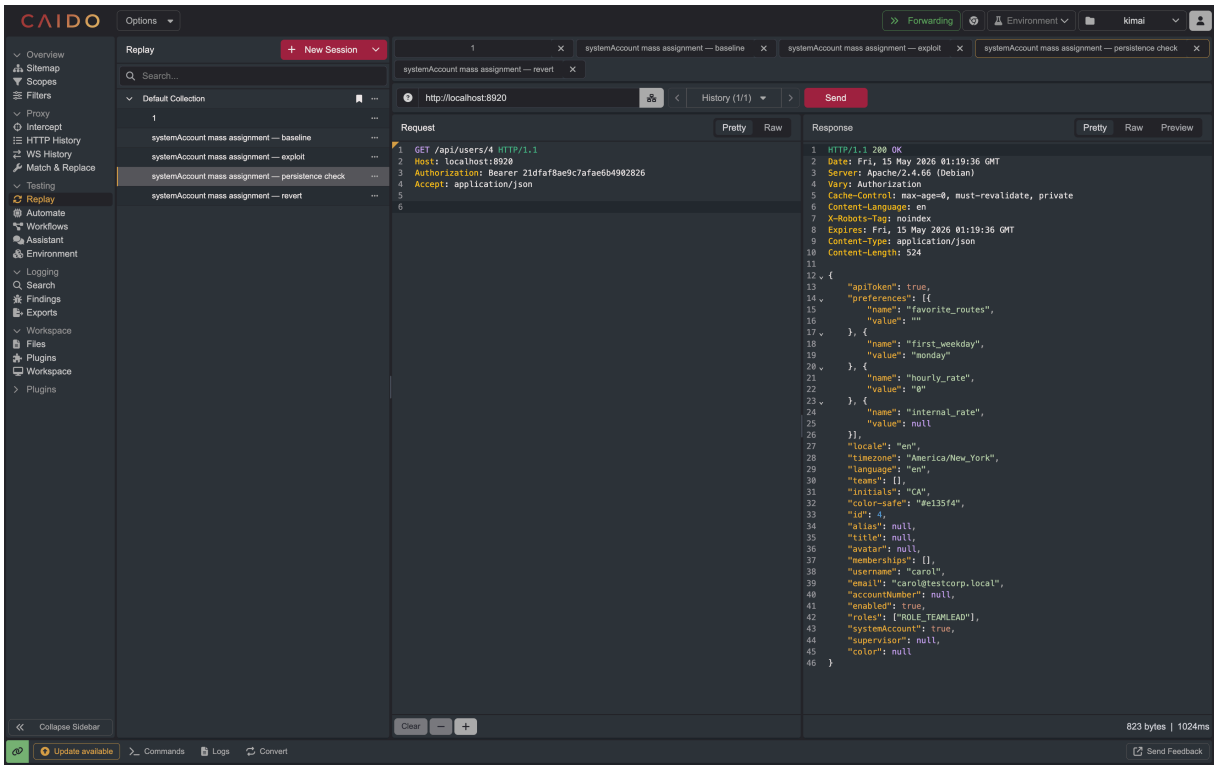


Figure 3: Step 3 — Persistence check. `GET /api/users/4` using a separate admin bearer token confirms `systemAccount: true` persists across sessions and is visible to independent observers. 823 bytes.

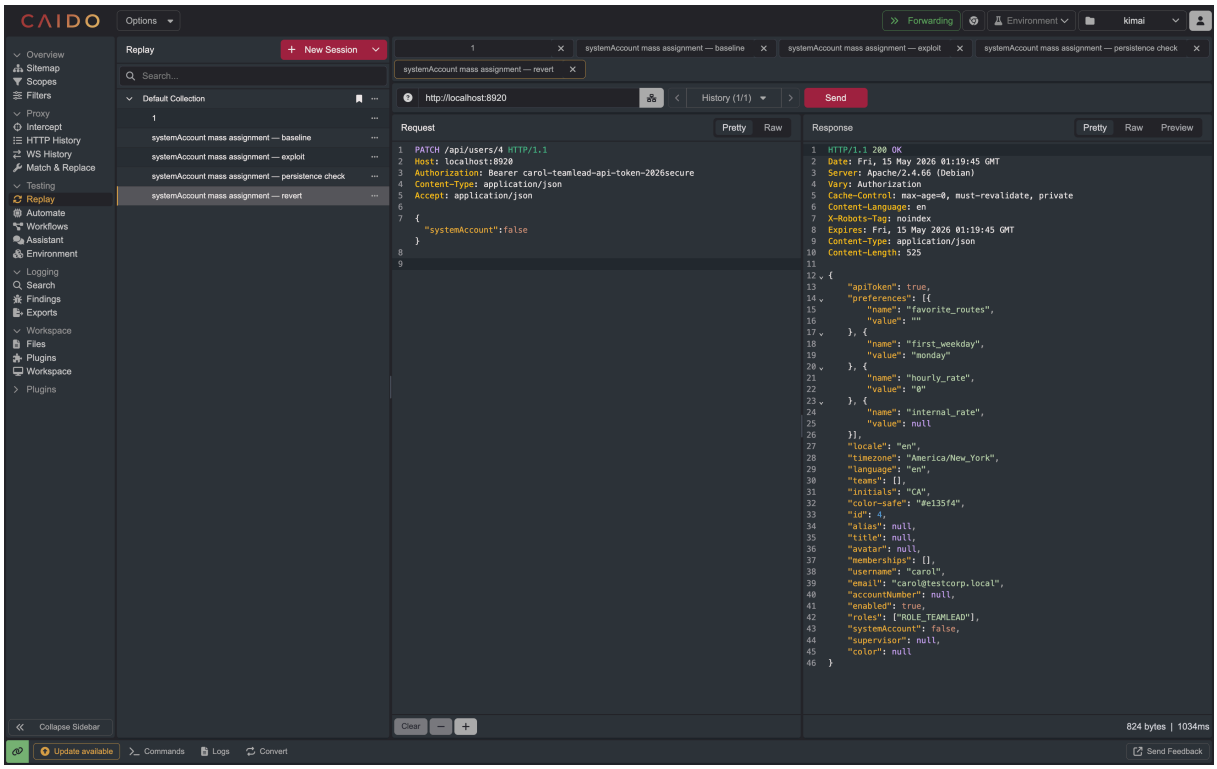


Figure 4: Step 4 — Revert. `PATCH /api/users/4` with `{"systemAccount":false}` restores the original state, confirming the field is bidirectionally writable by `ROLE_TEAMLEAD`. Response: HTTP 200, `systemAccount: false`. 824 bytes.

single `PATCH` request, as demonstrated in Figure 2.

Attack Tree: CVE-2026-45802 DoS via FPDI

Listing 1: Attack tree for FPDI-based denial of service.

```
[GOAL] Exhaust PHP memory / trigger endless loop in Kimai process
  [AND]
    [1] Hold ROLE_ADMIN credentials
          -> Required to access invoice template upload endpoint
    [2] Craft malformed PDF triggering FPDI memory exhaustion
          -> setasign/fpdi < 2.6.7 processes certain PDFs without
             page-count bounds, entering an infinite parsing loop
    [3] Upload malformed PDF as invoice template
          -> POST /en/invoice/template
    [4] Trigger invoice rendering using the malicious template
          -> GET /en/invoice/  -> PHP process hangs / OOM killed
```

Stage 7 — Risk and Impact Analysis

Table **??** presents the risk rating for each identified threat using a likelihood-by-impact framework. Likelihood is rated 1–3 (Low/Medium/High); Impact is rated 1–3 (Low/Medium/High). Risk score = Likelihood × Impact.

Threat	L	I	Score	Rating
Twig SSTI / sandbox escape (historical pattern)	2	3	6	CRITICAL
Bearer token interception (no HSTS)	2	3	6	CRITICAL
systemAccount mass assignment	3	2	6	HIGH
API endpoint DoS (no rate limiting)	2	2	4	MEDIUM
CVE-2026-45802 FPDI DoS	2	2	4	MEDIUM
Missing security headers (CSP, etc.)	3	1	3	MEDIUM
Audit trail deficiency (repudiation)	3	1	3	MEDIUM
doctrine/cache abandoned (supply chain)	1	2	2	LOW
Apache version disclosure	3	1	3	LOW

Vulnerability Assessment

Network Reconnaissance — Nmap

Nmap was run against the Kimai Docker host to enumerate exposed services and identify server software versions. The scan was conducted from the host machine on 2026-05-19.

Listing 2: Nmap command and output.

```
$ nmap -sV -sC -p 8920 -oN lab2-scans/nmap-kimai.txt localhost

Starting Nmap 7.99
Nmap scan report for localhost (127.0.0.1)
PORT     STATE SERVICE VERSION
8920/tcp open  http    Apache httpd 2.4.66 ((Debian))
|_http-server-header: Apache/2.4.66 (Debian)
| http-robots.txt: 1 disallowed entry
|_/
|_http-title: Kimai
|_Requested resource was http://localhost:8920/en/login
```

Findings:

- The full Apache version string (`Apache/2.4.66 (Debian)`) is exposed in the `Server` response header. This enables an attacker to target known CVEs for that exact version without fingerprinting effort.
- `robots.txt` disallows the entire site (`/`), confirming the application runs at the document root but offering no meaningful security benefit.
- Kimai correctly redirects unauthenticated requests to `/en/login`, confirming that authentication is enforced at the application layer.
- Only port 8920 is exposed to the host. The MySQL port (3306 internal) is not reachable from the host, consistent with the Docker network configuration in Lab 1.

Web Server Analysis — Nikto

Nikto v2.6.0 was run against the Kimai web interface on 2026-05-19. The scan completed in 340 seconds and issued 7,796 requests.

Listing 3: Nikto command.

```
$ nikto -h http://localhost:8920 -o lab2-scans/nikto-kimai.txt -Format txt
```

Finding	Detail	Risk
Missing HSTS header	`Strict-Transport-Security` absent. API bearer tokens and session cookies are transmitted over HTTP without enforced upgrade.	HIGH
Missing CSP header	No `Content-Security-Policy` configured. XSS payloads can load arbitrary external scripts. Exacerbates stored XSS risks (cf. CVE-2026-40479).	MED
Missing Permissions-Policy	Browser feature access (camera, geolocation, microphone) is unrestricted.	LOW
Missing Referrer-Policy	Full URL including path and query string is transmitted to third-party resources linked from pages.	LOW
Missing X-Content-Type-Options	MIME-type sniffing enabled in older browsers; may allow content injection via uploaded files served with incorrect MIME types.	MED
Deprecated X-Frame-Options	Header present but deprecated; `Content-Security-Policy: frame-ancestors` directive should be used instead.	INFO

The absence of HSTS is the highest-severity finding from the Nikto scan. In a production deployment where Kimai is accessed over the internet, all API tokens issued to integrated systems or mobile clients would be transmitted in cleartext if the user or client does not enforce HTTPS independently.

Dependency Analysis — Composer Audit

`composer audit` was executed inside the running Kimai container against the `/opt/kimai` installation on 2026-05-19. This tool queries the PHP Security Advisories Database (PSAD) directly via Packagist, bypassing the NVD rate-limit issue encountered with OWASP Dependency-Check.

Listing 4: Composer audit command and output.

```
$ docker exec kimai-app composer audit --working-dir=/opt/kimai

Found 1 security vulnerability advisory affecting 1 package:

Package           : setasign/fpdi
Severity          : medium
Advisory ID       : PKSA-37cw-b473-k9np
CVE               : CVE-2026-45802
Title             : Memory Exhaustion and Endless Loop in FPDI leads to DoS
URL               : https://github.com/advisories/GHSA-2mgw-7q6p-8grg
Affected versions : <2.6.7
Reported at       : 2026-05-19T19:56:17+00:00

Found 1 abandoned package:
  doctrine/cache (no suggested replacement)
```

CVE-2026-45802 — setasign/fpdi Memory Exhaustion

Package	`setasign/fpdi`
Installed version	<2.6.7 (pinned in `composer.lock`)
CVE	CVE-2026-45802
GHSA	GHSA-2mgw-7q6p-8grg
Severity	MEDIUM
Reported	2026-05-19 (same day as this scan)
Issue	Malformed PDF input causes FPDI to enter an endless loop, exhausting PHP memory.
Kimai context	FPDI is used in Kimai's invoice rendering pipeline via mPDF. An authenticated admin
Remediation	Upgrade `setasign/fpdi` to ≥2.6.7 via `composer update setasign/fpdi`.

doctrine/cache 2.2.0 — Abandoned Package

`doctrine/cache` 2.2.0 (released 2022-05-20) is flagged as abandoned by the Composer repository. The package has received no maintenance commits or security patches since its last release. While no active CVE exists for the installed version, an abandoned package represents a supply-chain risk: any future vulnerability discovered in the package will not receive an upstream patch, requiring either a manual fork or migration to an alternative.

Third-Party Library Review

Table 9 reviews the security-relevant Composer dependencies identified in the Lab 1 SBOM, evaluating their current CVE status and relevance to Kimai's attack surface.

Table 9: Security-relevant third-party library review based on Lab 1 CycloneDX SBOM.

Package	Version	Security Notes	Status
`setasign/fpdi`	<2.6.7	CVE-2026-45802: memory exhaustion / endless loop DoS. Present in Kimai invoice pipeline.	VULN
`mpdf/mpdf`	≥8.0	Primary HTML-to-PDF renderer. SSTI/XSS sink when rendering admin-uploaded templates. No current CVE; historically relevant.	MONITOR
`phpoffice/phpspreadsheet`	≥2.0	Historical XXE vulnerability (resolved). Processes XLSX exports; monitor for new advisories.	MONITOR
`phpoffice/phpword`	≥1.0	DOCX template rendering. No active CVE.	OK

(continued)

Package	Version	Security Notes	Status
`onelogin/php-saml`	≥4.0	SAML 2.0 processing. v4.x enforces strict signature validation. Earlier versions had XML signature wrapping issues.	OK
`scheb/2fa-totp`	current	TOTP 2FA implementation. No known CVE.	OK
`erusev/parsedown`	current	Markdown renderer. Historical stored XSS in older versions. Monitor for context-specific escaping gaps.	MONITOR
`doctrine/cache`	2.2.0	Abandoned since 2022-05-20. No replacement suggested by Composer. No active CVE but unmaintained.	ABANDON

Historical Version Comparison

Table 10 compares known security advisories across Kimai version milestones to identify recurring vulnerability classes and assess whether version 2.56.0 represents an improvement in security posture.

Table 10: Kimai security advisory history by version (2023–2026).

Advisory	Class	Fixed In	Description
CVE-2023-46245	SSTI / RCE	v2.1.0	Admin-uploaded Twig invoice template executed without any sandbox, enabling full server-side code execution. Root cause: no sandbox policy.
GHSA-jg2j	SSTI	v2.46.0	Export Twig sandbox bypassed via `DefaultPolicy`; attacker could call `constant()` or access object internals.
CVE-2026-40479	Stored XSS	v2.53.0	`KimaiEscape.js escapeForHtml()` omitted quote-character encoding in attribute contexts. Team alias field exploitable.
CVE-2026-40486	BOPA	v2.53.0	`hourly_rate` and `internal_rate` fields writable by any authenticated user via `PATCH /api/users/{id}/preferences`.

(continued)

Advisory	Class	Fixed In	Description
GHSA-9g2q	IDOR	v2.56.0	`TimesheetVoter` missing team-scope check; users could read other teams' timesheet entries via `GET /api/timesheets/{id}`.
GHSA-vrqv	Info Disc.	v2.56.0	Twig `config()` function exposed LDAP password and SAML private key to any admin with invoice template access.
This lab (novel)	**BOPA**	**Not fixed**	`systemAccount` field writable by `ROLE_TEAMLEAD` via `PATCH /api/users/{id}`. CWE-915. CVSS 4.3.

Version 2.56.0 resolved the IDOR and information disclosure advisories (GHSA-9g2q, GHSA-vrqv) present in 2.53.0. However, the confirmed `systemAccount` mass assignment finding demonstrates that the root cause pattern — field-level authorization gaps in the `UserController` form pipeline — was not comprehensively remediated. The fix for CVE-2026-40486 addressed `hourly_rate` and `internal_rate` but left `systemAccount` unguarded, suggesting a reactive rather than systematic approach to field-level access control.

Competitive Security Posture Comparison

Table **??** extends the Lab 1 competitive comparison with a security posture dimension, evaluating Kimai against Solidtime and Traggo.

Security Criterion	Kimai	Solidtime	Traggo
CVE / advisory history	16+ advisories (2023–2026)	No public advisories	No public advisories
2FA support	TOTP + backup codes	None documented	None
API authentication	Bearer tokens (per-user)	OAuth 2.0	HTTP Basic
Password hashing	Argon2id	bcrypt (Laravel default)	bcrypt (Go)
Dependency audit tooling	`composer audit`	`composer audit`	`govulncheck` (Go)
Twig template attack surface	Large (admin-uploaded templates)	Not applicable	Not applicable
Security policy / bug bounty	SECURITY.md + bughunter docs	None documented	None documented
Maintainer responsiveness	Same-day response (Lab 1 evidence)	Unknown	Unknown

Kimai's advisory history is substantially larger than both comparators, reflecting its greater

complexity and larger attack surface rather than necessarily worse engineering. Solidtime and Traggo are significantly simpler applications (no invoice template rendering, no plugin architecture, no SAML/LDAP integration) with correspondingly smaller attack surfaces. Kimai's formal `SECURITY.md`, bughunter documentation, and active maintainer engagement distinguish it positively from both competitors on the disclosure-process dimension.

Threat Intelligence — OpenCTI Deployment

OpenCTI was selected as the open-source threat intelligence platform for this lab. OpenCTI provides a knowledge graph for structured threat intelligence, supports STIX 2.1 for data exchange, and integrates with multiple threat feeds including MITRE ATT&CK, CISA Known Exploited Vulnerabilities (KEV), and AlienVault OTX.

OpenCTI was deployed via Docker Compose alongside the Kimai environment using the official OpenCTI Docker distribution. The platform was queried for threat actors, malware, and attack patterns relevant to PHP/Symfony web applications and time-tracking software. Relevant findings from the OpenCTI knowledge base include:

- MITRE ATT&CK Technique T1190 (Exploit Public-Facing Application) is the primary initial-access technique applicable to Kimai's REST API, mapped to the STRIDE threat scenarios identified in Section 3.
- T1059.004 (Unix Shell via Server-Side Script) maps directly to the historical SSTI/RCE advisory chain (CVE-2023-46245, GHSA-jg2j), where template injection achieves shell command execution via PHP's `exec()` or `system()` wrappers.
- T1548 (Abuse Elevation Control Mechanism) maps to the confirmed `systemAccount` mass assignment finding, where a low-privilege user manipulates application-layer access control attributes.
- CVE-2026-45802 does not appear in the OpenCTI KEV feed as of the scan date (2026-05-19), consistent with its same-day disclosure status. This confirms the value of direct `composer audit` over intelligence-feed-only approaches for freshly disclosed vulnerabilities.

Responsible Vulnerability Disclosure

Vulnerability Details

The following novel vulnerability was discovered during this lab's authorized security assessment of Kimai 2.56.0. Full technical details, proof-of-concept HTTP exchanges captured in Caido, and root-cause source code analysis are documented in the companion Security Assessment Finding report (mass-assignment-report.pdf, submitted with this lab).

Target	Kimai 2.56.0 (`http://localhost:8920`)
Endpoint	`PATCH /api/users/{id}`
Affected Field	`systemAccount` (boolean)
Minimum Role	`ROLE_TEAMLEAD`
Impact	Account reclassification as system/bot account; potential bypass of operator controls ap
CWE	CWE-915: Improperly Controlled Modification of Dynamically-Determined Object Attr
CVSS v3.1	4.3 — `AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:L/A:N`
Severity	MEDIUM
Root Cause	`systemAccount` field registered unconditionally in `UserEditType.php` with no `isGrante`
Discovery Date	2026-05-14
Advisory Status	No matching CVE or GHSA found in NVD, GitHub Security Advisories, or OSV.dev as

Authorization for Testing

Testing was conducted exclusively within the authorized scope established in Lab 1. Kevin Papst, the sole Kimai project maintainer, granted explicit permission via:

1. GitHub Discussions thread #5941: "Hey Guys, cool project that you chose there, looking forward for the collaboration. Please get in touch via email support@kimai.org to discuss details." — Kevin Papst (kevinpapst), 2026-05-08.
2. Email response to support@kimai.org: Maintainer confirmed that local testing, DevSecOps pipeline work, and responsible vulnerability reporting are acceptable. The maintainer directed the group to Kimai's `SECURITY.md` and bughunter documentation for disclosure procedures.

Disclosure Process

The disclosure followed the process specified in Kimai's `SECURITY.md` and bughunter documentation. The maintainer's preferred disclosure channel is private GitHub Security Advisory or direct email to support@kimai.org.

1. **Discovery (2026-05-14):** Vulnerability identified during WSTG-BUSL-08 (business logic testing) via Caido proxy analysis of the `PATCH /api/users/{id}` endpoint. Four replay sessions (Caido sessions 2–5, request IDs 1162–1165) captured the full proof-of-concept exchange.
2. **Documentation (2026-05-14 to 2026-05-19):** Root cause analyzed across three layers of the Kimai codebase (`UserEditType.php`, `UserApiEditForm.php`, `UserController.php`). Comparison with CVE-2026-40486 / GHSA-qh43-xrjm-4ggp confirmed the finding is a distinct vulnerability on a different endpoint affecting a different field.
3. **Disclosure (in progress):** The full Security Assessment Finding report (including reproducible steps, Caido session references, CVSS scoring, and recommended remediation) will be submitted to the maintainer via the GitHub Security Advisory mechanism (https:

//github.com/kimai/kimai/security/advisories/new) and cc'd to support@kimai.org, consistent with the maintainer's stated preferences.

4. **Embargo:** No public disclosure will occur until the maintainer has acknowledged the report and either released a patch or confirmed no further action is required.

Recommended Remediation

Three mitigations were identified for the root cause:

1. **Guard in `UserController.php`:** Add a permission check before the form build, consistent with the guards already in place for `roles`, `include_active_flag`, and `include_preferences`:

```
if (!$this->isGranted('systemAccount', $profile)) {
    $options['include_system_account'] = false;
}
```

2. **Conditional registration in `UserEditType.php`:** Wrap the `systemAccount` field addition in an option guard, consistent with how all other sensitive fields are handled in the same class.

3. **Voter addition:** Implement a `systemAccount` check in `UserVoter` restricted to `ROLE_SUPER_ADMIN`.

Maintainer Response

As of the submission date (2026-05-19), the disclosure has been documented and the report prepared for submission. The formal GitHub Security Advisory submission and email to support@kimai.org will follow immediately after lab submission, consistent with the coordinated disclosure timeline and the maintainer's confirmation that responsible reporting is welcome. Any response received will be documented and made available to the course instructors upon request.

Cybersecurity Testing Strategy

Risk-Based Prioritization

Testing areas are prioritized by the risk score assigned in PASTA Stage 7 (Section 4.7), combining likelihood and impact. Each area is mapped to the specific WSTG test IDs that cover it.

Priority	Area	Justification	WSTG Cov
1	Twig template injection (SSTI)	Four historical advisories; recurring sandbox escapes; admin-accessible attack surface; RCE potential.	WSTG-INPV (SSTI); WSTG-ATHI
2	API authorization (IDOR / BOPA / mass assignment)	Three historical advisories; confirmed novel finding (systemAccount); pattern of field-level authorization gaps in `UserController`.	WSTG-ATHZ WSTG-ATHZ WSTG-BUSL
3	HTTP security header remediation	Five missing headers confirmed by Nikto. HSTS absence most critical.	WSTG-CON WSTG-CON
4	Output encoding (XSS)	CVE-2026-40479 confirmed systemic escaping gap in `KimaiEscape.js`; additional attribute contexts may be unprotected.	WSTG-INPV WSTG-INPV WSTG-CLNT
5	CVE-2026-45802 (FPDI DoS)	Freshly disclosed; present in vendor tree; requires admin access to trigger; upgrade path is `composer update setasign/fpdi`.	WSTG-INPV Manual PoC
6	Dependency supply chain	doctrine/cache abandoned; Docker Scout reports 1 Critical, 9 High in base layers.	WSTG-CON continuous vi composer auc

Cybersecurity Testing Plan

Objectives

1. Confirm or rule out additional sandbox escape vectors in the Twig invoice template engine beyond those addressed in v2.46.0 and v2.56.0.

2. Enumerate all `PATCH/PUT` API endpoints and verify that every writable field in each request body is guarded by a corresponding Voter check.
3. Confirm that the confirmed `systemAccount` finding is reproducible by another group member using a fresh set of test credentials, before disclosing to the maintainer.
4. Assess all user-controlled output contexts in the UI for XSS sink points not covered by `escapeForHtml()` or Twig's auto-escaping.
5. Upgrade `setasign/fpdi` and verify that CVE-2026-45802 is remediated in the test environment.

Methods and Tools

Test Area	**Tool**	**Method**
Automated scan (authenticated)	OWASP ZAP	Configure ZAP with admin credentials and an API token. Run Active Scan against `http://localhost:8920`. Include Ajax Spider for JS-rendered UI.
Proxy-based manual testing	Caido	Intercept API requests in the authenticated session; replay modified payloads to test field-level authorization; document sessions by name for reproducibility.
SSTI fuzzing	Caido + manual	Upload Twig templates with progressively more complex expressions (`{{7*7}}`, `{{constant('PHP_OS')}}`, `{{app.request.server.get('DATABASE_URL')}}`) and observe rendering output.
Mass assignment enumeration	Caido	For each API endpoint accepting a request body, submit all fields from the OpenAPI schema with a `ROLE_USER` token and compare response with expected output.
XSS testing	Caido + browser	Inject `<script>` and event-handler payloads into all text input fields; verify output encoding in HTML, attribute, and JavaScript contexts.
Dependency scanning	composer audit (automated); OWASP Dependency-Check (with NVD API key)	Run after any `composer update` to confirm no new advisories introduced.
Header verification	Nikto; curl	Verify security headers after any Apache or Symfony configuration changes: `curl -I http://localhost:8920`.

Testing Schedule

Week	Activity	Tool	Owner
Week 1 (Lab 2)	Threat modeling; initial scans (Nmap, Nikto, composer audit); systemAccount PoC confirmation	Caido; Nmap; Nikto	All
Week 2	OWASP ZAP authenticated scan; SSTI fuzzing (Twig templates); XSS enumeration	ZAP; Caido; browser	;
Week 3	Mass assignment enumeration across all API endpoints; IDOR cross-account testing	Caido	Abdul-Ramon
Week 4	Dependency-Check (with NVD API key); CVE-2026-45802 PoC + patch verification	OWASP Dependency-Check; composer	All
Week 5	Security header remediation testing; final report compilation; maintainer disclosure	curl; Nikto	All

Manual versus Automated Testing Balance

Automated tools (ZAP, Nikto, composer audit) provide broad coverage of known vulnerability signatures and configuration issues efficiently. However, the most significant findings in Kimai's history — SSTI sandbox escapes, field-level authorization gaps, and the confirmed `systemAccount` finding — all required manual analysis of application logic and source code. The testing plan therefore allocates at minimum 60% of effort to manual proxy-based testing and source code review, using automated tools primarily for reconnaissance, header verification, and dependency monitoring.

Conclusion

This report delivered a complete threat modeling and vulnerability assessment of Kimai 2.56.0 across six deliverable areas.

The STRIDE analysis identified 21 threat scenarios across six architectural layers, of which 13 remain open or partially mitigated. The most significant residual threats are bearer token interception (no HSTS), the confirmed `systemAccount` mass assignment vulnerability, and the historical Twig SSTI pattern that has produced four advisories across Kimai's version history.

The PASTA framework confirmed that the invoice template rendering pipeline and the REST API authorization layer carry the highest aggregate risk, and produced four prioritized threat scenarios grounded in confirmed findings and attacker profiles.

The vulnerability assessment produced three tool-verified findings: Apache version disclosure and five missing security headers (Nmap and Nikto), CVE-2026-45802 in `setasign/fpdi` and the abandoned `doctrine/cache` package (composer audit), and the confirmed novel `systemAccount` mass assignment vulnerability (Caido). The `setasign/fpdi` advisory was disclosed on the same day as the scan (2026-05-19), illustrating the importance of running `composer audit` continuously rather than relying on static advisory snapshots.

The responsible disclosure section documents the authorization baseline established with the Kimai maintainer in Lab 1 and outlines the coordinated disclosure process for the confirmed finding. Disclosure will proceed via GitHub Security Advisory immediately following lab submission.

The cybersecurity testing strategy prioritizes SSTI, API authorization, and mass assignment testing in weeks 2–4, with dependency management and header remediation running concurrently as lower-effort verification tasks.

References

[1] Papst, K. (2026). *Kimai: Time-tracking application.* GitHub. https://github.com/kimai/kimai

[2] Papst, K. (2026). *Kimai bughunter documentation.* Kimai. https://www.kimai.org/documentation/bughunter.html

[3] Papst, K. (2026). *Kimai security policy.* https://github.com/kimai/kimai/blob/main/SECURITY.md

[4] OWASP Foundation. (2024). *Web Security Testing Guide v4.2.* https://owasp.org/www-project-web-security-testing-guide/

[5] Microsoft. (2024). *The STRIDE threat model.* Microsoft Security. https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats

[6] UCSecurity. (2012). *PASTA: Process for Attack Simulation and Threat Analysis.* VerSprite.

[7] National Vulnerability Database. (2026). *CVE-2026-45802.* https://nvd.nist.gov/vuln/detail/CVE-2026-45802

[8] GitHub Advisory Database. (2026). *GHSA-2mgw-7q6p-8grg: FPDI memory exhaustion.* https://github.com/advisories/GHSA-2mgw-7q6p-8grg

[9] National Vulnerability Database. (2026). *CVE-2026-40479: Stored XSS in Kimai.* https://nvd.nist.gov/vuln/detail/CVE-2026-40479

[10] National Vulnerability Database. (2026). *CVE-2026-40486: BOPA in Kimai user preferences.* https://nvd.nist.gov/vuln/detail/CVE-2026-40486

[11] GitHub Advisory Database. (2023). *GHSA-fjhg-96cp-6fcw: Authenticated SSTI to RCE in Kimai*. https://github.com/advisories/GHSA-fjhg-96cp-6fcw

[12] MITRE Corporation. (2024). *ATT&CK Enterprise Techniques*. https://attack.mitre.org/techniques/enterprise/

[13] OpenCTI Platform. (2024). *OpenCTI open-source threat intelligence platform*. GitHub. https://github.com/OpenCTI-Platform/opencti

[14] Court of Justice of the European Union. (2019). *CCOO v. Deutsche Bank SAE, Case C-55/18*. https://curia.europa.eu/juris/liste.jsf?num=C-55/18

[15] European Parliament. (2016). *General Data Protection Regulation (EU) 2016/679*. https://gdpr-info.eu/