SAMPLE REPORT — SANITIZED · ALL CLIENT NAMES, DOMAINS, DATA AND IDENTIFIERS ARE FICTIONAL

Web Application Penetration Test: Sample Report

> *Sample report. The client, domain, data and identifiers in this document are fictional. The structure, finding classes and reporting approach come from a real assessment I performed on a SaaS product I own and operate. The client name and every identifier have been replaced.*

Client	Acme Logistics (fictional)
Target	`app.acme-logistics.example` (Next.js SaaS on a serverless platform)
Engagement type	External, unauthenticated black-box assessment
Testing window	Two days, 2026
Prepared by	Ben Hammouda Nour
Classification	Sample, sanitized
Methodology	PTES, OWASP Web Security Testing Guide

CONTENTS

1. Executive Summary
2. Scope, Authorization and Methodology
3. Risk Rating Methodology
4. Summary of Findings
5. Detailed Findings
6. Verified-Clean Test Matrix
7. Remediation Roadmap
Appendix A. Tools and Techniques
Appendix B. Data Reconciliation Note
Disclaimer

1. Executive Summary

An external black-box penetration test was performed against a French-language logistics and fleet-cost SaaS built on Next.js (App Router) and hosted on a serverless platform. Testing was authorized by the site owner and limited to non-destructive techniques.

The application has a better-than-average baseline for a small SaaS: middleware-based route gating, invite-only registration, no exposed source maps or configuration files, no extra open ports, and patches in place for well-known framework bypasses. The assessment still confirmed **one actively exploitable High-severity vulnerability** and **one High-severity control gap**.

- **Most significant finding (High, CVSS 8.2):** the deployed framework version was behind the patched release for a published Server Actions CPU-exhaustion denial-of-service issue. The trigger was reconstructed from the public patch and verified live. One crafted request caused roughly a 6× latency increase, and 32 concurrent requests pushed responses to over 100× baseline. Per-request platform isolation prevented a full outage, but the issue is a billed-CPU amplification vector today and a full unauthenticated DoS on any non-isolated hosting target.
- **Second High finding:** no rate limiting or lockout on the login action. More than 35 rapid attempts all returned the same response with no throttling, which enables credential spraying against the invite-only user base.
- **Secondary findings:** a partially mitigated email/phone enumeration oracle on a public signup form, an unenforced DMARC policy (`p=none`) with SPF softfail and no DKIM, and a permissive Content-Security-Policy.

No SQL/NoSQL injection, IDOR, authentication bypass, SSRF, open redirect or exposed secrets were found from the external, unauthenticated position.

Overall risk rating	**HIGH**, driven by one confirmed-exploited CVE and one unmitigated brute-force exposure. No Critical findings.

2. Scope, Authorization and Methodology

Authorization. Testing was authorized in writing by the site owner against the primary domain and its `www` alias only. Destructive testing and account takeover with real credentials were out of scope. No real user data was accessed. Synthetic test records were created only in the public signup form. A sibling domain on the same hosting project was observed passively and excluded from active testing on the client's instruction.

Methodology.

1. **Reconnaissance:** DNS, TLS and WHOIS enumeration, header and cookie analysis, robots.txt and sitemap review, JavaScript bundle mining for routes and action identifiers, certificate transparency and web-archive review.
2. **Attack-surface mapping:** route and API enumeration, server-action cataloguing, framework and version fingerprinting.
3. **Vulnerability identification:** a structured checklist covering common SaaS weaknesses (exposed env files, client-side-only auth gates, IDOR, injection, open redirect, weak cookies) plus a platform and framework CVE battery.
4. **CVE cross-reference and verification:** the fingerprinted version was cross-referenced against the current security releases. Candidate issues were reconstructed from public patches and verified with instrumented, non-destructive requests.
5. **Tooling pass and gap closure:** standard tooling for independent validation, then a second pass against the OWASP/PTES checklist to close residual gaps.

Constraints. Registration is invite-only, so the authenticated surface (dashboard, billing, role separation) could not be tested. This is a scope gap, with a follow-up recommended in Section 7. A companion mobile app sharing the same backend was not analyzed.

3. Risk Rating Methodology

Findings are rated qualitatively, informed by CVSS v3.1 base scores where a published score exists, and by exploitability and business impact for custom application-layer findings.

RATING	CVSS BAND	DEFINITION
Critical	9.0 – 10.0	Trivially exploitable and unauthenticated, leading to full compromise of data, accounts or infrastructure.
High	7.0 – 8.9	Exploitable with moderate effort, with significant impact, or a confirmed working exploit against a known CVE.
Medium	4.0 – 6.9	Exploitable under specific conditions, or gives meaningful attacker advantage without direct compromise.
Low	0.1 – 3.9	Defense-in-depth weakness that needs another vulnerability to matter.
Info	Not scored	Observation with no direct exploit path.

4. Summary of Findings

ID	TITLE	SEVERITY	STATUS
F-01	Server Action CPU-exhaustion DoS (published framework CVE)	High	Confirmed, exploited
F-02	No rate limiting or lockout on login	High	Confirmed
F-03	Email/phone enumeration oracle on signup form	Medium	Confirmed, well mitigated
F-04	DMARC not enforced, email spoofing possible	Medium	Confirmed
F-05	Permissive Content-Security-Policy	Low	Confirmed
F-06	Missing hardening headers, incomplete HSTS scope	Low	Confirmed
F-07	Deployment and framework fingerprinting	Info	Observed
F-08	Server-action identifiers disclosed in client JS	Info	Observed
F-09	Wildcard DNS and certificate inconsistency	Info	Observed
F-10	Login action instability, conflicting observations	Unverified	Needs re-verification

F-10 is excluded from the severity totals until resolved (see Appendix B).

5. Detailed Findings

F-01. Server Action CPU-exhaustion Denial of Service (High, CVSS 8.2)

Component: login server action (multipart form handling). **Status:** confirmed, live-verified.

Root cause. The deployed framework version was two patch levels behind the fixed release. Affected versions place no bound on the number of action-reference fields accepted in a multipart form body submitted to a server action. Processing those fields triggers a CPU-burning decode loop (CWE-834, Excessive Iteration). The vendor fix caps the number of such fields per form.

Reproduction (conceptual). Submit a multipart POST to the login server action containing a very large number of repeated action-reference fields. No credentials are needed.

Measured impact (rounded, one test window):

SCENARIO	AVG. RESPONSE TIME	VS. BASELINE
Baseline login POST	~0.7 s	1×

SCENARIO	AVG. RESPONSE TIME	VS. BASELINE
1 crafted request	~4 s	~6×
8 concurrent	~12 s	~17×
32 concurrent	~90 s+	~120×

Impact. Response time grows with concurrency. Per-request instance isolation prevented a platform-wide outage (a post-test health check responded normally), but each crafted request burns seconds of billed CPU time, which is a cost-amplification vector. On a non-isolated or self-hosted target, about 30 concurrent requests would be enough for a full unauthenticated denial of service.

Remediation. Upgrade the framework to the fixed release immediately. This is the highest-priority action. Re-test after the upgrade.

F-02. No Rate Limiting or Account Lockout on Login (High)

Status: confirmed.

More than 35 rapid POST requests all returned the same generic error with no delay, lockout, CAPTCHA or IP-based throttling. This enables credential spraying and brute force against the invite-only user base. The application already throttles a different public form (F-03), so the team can add the control. It is missing on the higher-value login path.

Remediation. Add per-IP and per-account throttling with exponential backoff, and a challenge after a small number of consecutive failures. Keep error responses uniform, and alert on spray patterns.

F-03. Enumeration Oracle on the Signup Form (Medium)

Status: confirmed, well mitigated.

The form returns three distinct states (success, duplicate, validation error), which lets an attacker learn whether an email or phone number is already registered. A per-IP throttle engages after roughly 3–5 probes and persists for hours. It was not bypassable with header rotation, phone rotation or fresh email addresses. The residual exposure is the first few probes per source.

Remediation. Return one uniform response regardless of state, and make the throttle block rather than stall.

F-04. DMARC Not Enforced (Medium)

Status: confirmed, live-tested.

SPF uses softfail (`~all`), DMARC is `p=none`, and no DKIM record exists at the default selector. At the SMTP level, a forged envelope sender for the client's domain was accepted by the MX servers of a major consumer mailbox provider. Delivery was blocked only by the testing

IP's reputation, which an attacker controls. A second scenario, a compromised mailbox sending through the legitimate relay with a forged From header, would reach inboxes in full.

Remediation. Move DMARC to `p=quarantine`, then `p=reject`, with aggregate reporting. Tighten SPF to `-all`, and publish DKIM for the sending selector.

F-05. Permissive Content-Security-Policy (Low)

`script-src` includes `'unsafe-inline'` and `'unsafe-eval'`, with a permissive `connect-src`. No XSS was found, but with cookie-session authentication, any future XSS would become full account takeover instead of being contained by CSP.

Remediation. Remove `'unsafe-eval'` and move to nonce- or hash-based inline scripts.

F-06. Missing Hardening Headers (Low)

HSTS is present but lacks `includeSubDomains` and `preload`. `Cross-Origin-Opener-Policy` and `Permissions-Policy` are absent. Other headers (`X-Content-Type-Options`, `frame-ancestors`, `Referrer-Policy`) are correct.

Remediation. Add the missing headers and extend HSTS once all subdomains are HTTPS-only.

F-07 to F-09. Informational

ID	FINDING	DETAIL
F-07	Fingerprinting	Framework, build and deployment identifiers are visible in response headers and static asset URLs. No direct exploit path.
F-08	Action identifiers in client JS	A small set of stable server-action identifiers can be extracted from the client bundle. This is inherent to the framework, but it means unauthenticated actions are callable by anyone who reads the bundle.
F-09	Wildcard DNS and certificates	Wildcard DNS resolves common subdomain names to a generic not-found page (no takeover risk, but it discloses naming conventions). The apex and `www` use certificates from different issuers.

F-10. Login Action Instability, Conflicting Observations (Unverified)

An earlier test pass recorded an unhandled-exception response on every call to the login action, including empty bodies. A later, more extensive pass recorded uniform normal responses with no crashes. These describe mutually exclusive behavior and cannot both be true of the same endpoint at the same time. The conflict could not be resolved by further black-box testing, so it is reported openly (Appendix B) instead of being merged silently.

Recommended action. The client should review platform function logs across the full testing window for the error digests, to check whether a deploy or configuration change happened between the two passes.

6. Verified-Clean Test Matrix

Checks performed with no exploitable result, included for audit traceability.

AREA	RESULT
SQLi / NoSQLi on public actions	Uniform error path, no crash-on-payload correlation
User enumeration via login (message and timing)	Uniform error, flat timing
IDOR on API routes	Not reachable pre-auth, blanket 401
Open redirect on login redirect parameter	Inert or validated server-side
Mass assignment and hidden parameters	All inert
Session fixation, cookie forgery	No pre-auth session, forged tokens ignored
Client-side-only auth gates	Server-side enforcement confirmed
Known middleware auth-bypass CVEs	Patched
Path-normalization bypasses	Blocked at the edge
Image-optimizer SSRF	Blocked, remote patterns properly scoped
CORS wildcard or reflection	No permissive headers
CSRF on server actions	Strong (per-render nonce, action binding)
Host-header poisoning	Not reflected
Exposed `.env`, config files, `.git`	Not reachable
Source maps	Disabled in production
Hardcoded secrets in JS bundles	None found
Open ports	Only 80/443, no origin exposed
Error-page verbosity	Bare errors, no stack traces

7. Remediation Roadmap

PRIORITY	ACTION	RESOLVES
1	Upgrade the framework to the fixed release	F-01

PRIORITY	ACTION	RESOLVES
2	Add per-IP and per-account rate limiting and lockout or CAPTCHA on login	F-02
3	Review function logs to resolve the login-crash discrepancy	F-10
4	Enforce DMARC, tighten SPF, publish DKIM	F-04
5	Make signup responses uniform and convert the throttle to a hard block	F-03
6	Harden the CSP	F-05
7	Add COOP and Permissions-Policy, extend HSTS	F-06
8	Re-test login and signup end to end after items 1 to 3	F-01, F-02, F-10
9	Provision a test account and schedule an authenticated-surface engagement	Scope gap
10	Scope a separate assessment of the sibling domain and the shared hosting project	Out-of-scope observation

Appendix A. Tools and Techniques

CATEGORY	TOOLS / TECHNIQUES
Network	nmap (service and default scripts, top 1000 ports)
Subdomain enumeration	subfinder
Vulnerability scanning	nuclei (framework templates), nikto
Fingerprinting	whatweb, wafw00f, manual header and cookie analysis
Content discovery	ffuf, dirb (rate-limited by the target WAF, so low-and-slow manual probing was used)
Recon / OSINT	Certificate transparency, web-archive review, public secret-dork searches
Custom instrumentation	Python and bash harnesses for CVE verification, login-spray checks, enumeration-oracle checks and SMTP-level DMARC tests
Manual testing	Browser auth-flow walkthrough, static analysis of JavaScript bundles

Appendix B. Data Reconciliation Note

This report consolidates three testing passes into one findings register. Two passes were consistent and were merged directly. The third produced the conflict described in F-10. It is documented here instead of being resolved by assumption.

PASS	OBSERVATION ON THE LOGIN ACTION
Pass 1 (recon and checklist)	Not specifically tested for crash behavior.
Pass 2 (detailed findings)	Unconditional error response on every request, with a stable error digest.
Pass 3 and gap-closure round	Uniform normal responses across 35+ requests, no crashes. Used as the basis for F-02.

Until the client's log review is done, F-02 is treated as confirmed on the Pass 3 evidence, and the earlier behavior is tracked separately as F-10 and not discarded.

Disclaimer

This is a sanitized sample. All names, domains, identifiers and figures are fictional or generalized. It reflects the reporting format and level of analysis of a real black-box assessment, not the security posture of any real organization. Reports for third-party clients are confidential and are shared only under an agreement with the client.