VULNUS

OFFENSIVE SECURITY — ATTACK SIMULATION

Security Assessment

Target	app.[redacted].io
Client	[redacted] B.V. (sample — client name redacted)
Date	2026-09-01
Tier	Professional — €2,999
Findings	5 total
Classification	CONFIDENTIAL

vulnus.io — contact@vulnus.io — SAMPLE REPORT, FOR ILLUSTRATION ONLY

01 — EXECUTIVE SUMMARY

Overview

This assessment identified 5 findings across the client's web application and supporting infrastructure, including one high-severity issue involving an unauthenticated internal API endpoint that exposes customer records. The remaining findings relate to missing security headers, weak cookie configuration, and email authentication gaps — all of which are straightforward to remediate but collectively increase the application's exposure to common attack techniques. Overall security posture is moderate: no critical, actively-exploited vulnerabilities were found, but the high-severity finding should be addressed before the next external security review or insurance renewal.

5	1	3	1
Total	Critical/High	Medium	Info

02 — FINDINGS

Detailed Findings

F-01 Unauthenticated Internal API Endpoint Exposing Customer Records — HIGH

An internal API endpoint (/api/v1/users/export) is reachable without authentication and returns customer names, email addresses, and account status in plain JSON. This means anyone who discovers the URL — including automated scanners run by attackers — can pull customer data without needing a login.

RECOMMENDED FIX

Require authentication (session token or API key) on this endpoint, and add it to your route-level auth middleware so future endpoints inherit the same protection by default.

F-02 Missing HTTP Security Headers — MEDIUM

The application does not set several standard security headers (Content-Security-Policy, X-Frame-Options, Strict-Transport-Security). Without these, the site is more vulnerable to clickjacking, mixed-content attacks, and certain cross-site scripting scenarios.

> RECOMMENDED FIX
> Add the missing headers at the reverse proxy or framework level — most frameworks (Next.js, Express) support this in a few lines of middleware config.

F-03 Session Cookie Missing Secure and SameSite Flags

MEDIUM

The session cookie (JSESSIONID) is set without the Secure or SameSite=Strict flags. This makes it easier for the session to be intercepted over an unencrypted connection or reused in a cross-site request forgery attack.

> RECOMMENDED FIX
> Set Secure and SameSite=Strict (or Lax, if cross-site flows are required) on all session cookies.

F-04 SPF and DMARC Records Misconfigured

MEDIUM

The domain's SPF record is overly permissive and no DMARC policy is published. This makes it easier for attackers to send convincing phishing emails that appear to come from your domain, targeting employees or customers.

> RECOMMENDED FIX
> Tighten the SPF record to only include mail servers you actually use, and publish a DMARC policy starting at p=quarantine before moving to p=reject.

F-05 Web Server Version Disclosed in Response Headers

INFO

The server's Server header discloses the exact software version in use. While not directly exploitable, this makes it easier for an attacker to identify and target known vulnerabilities for that specific version.

> RECOMMENDED FIX
> Suppress or generalize the Server header at the web server or reverse proxy configuration level.

SAMPLE REPORT NOTICE

This is a sample report for illustration purposes, using representative findings. Your actual report will reflect the real findings from your engagement. Client and target details shown here have been redacted/fictionalized.