AUDITCORE

Security & SEO Audit Report

Automated penetration testing & compliance assessment. 13 findings across the ENTERPRISE-tier scan pipeline.

TARGET

`https://demo-shop.example.com`

TIER	ENTERPRISE	GENERATED	2026-05-10 12:15 UTC
APP TYPE	Ecommerce	ACCOUNT	Authenticated
SCAN ID	a7f3e291-4b8c-4d9e-bf12-c8a7d3e5f690		

AUDIT-CORE.TECH · 5-PHASE PIPELINE · 50+ SCANNERS

01 · OVERVIEW

Executive Summary

This automated audit identified **13 findings** across `https://demo-shop.example.com`. The site scored **87/100** on SEO and **72/100** on AI-readiness.

2	3	4	2	2
CRITICAL	HIGH	MEDIUM	LOW	INFO

> **Action required.** 5 critical/high severity findings require immediate attention. Review the corresponding sections below.

Detected stack: `Next.js` `Stripe` `Cloudflare` `Shopify Hydrogen` `PostgreSQL`

Compliance Impact

Findings that may put the target out of compliance with each framework. Detailed control references appear under each finding below.

8	5	4	2
OWASP TOP 10	PCI DSS 4.0	ISO 27001	GDPR ART. 32

AI-GENERATED SUMMARY

The target shows the typical risk profile of an early-stage e-commerce platform: critical SQL injection on the product search and a missing rate limit on the login endpoint dominate exposure. Headers and SSL configuration are adequate. SEO is strong (87/100) but AI-readiness trails (72/100) — primarily because GPTBot is blocked at the CDN layer, making products invisible to ChatGPT shopping recommendations.

Vulnerability Findings

CRITICAL — 2 findings

CRITICAL CVSS 9.1 CWE-639 Authorization

Broken Object Level Authorization (BOLA) on User Profile API

Affected: `GET /api/v1/users/42/profile`

User A (role: customer) can access User B's profile data by changing the user ID in the URL. The API does not verify that the authenticated user owns the requested resource.

EVIDENCE

```
Request: GET /api/v1/users/42/profile (Auth: user_a_token) Response: 200 OK {"id": 42, "email": "userb@example.com", "phone": "+1-555-0142", "address": "123 Main St"}
```

> **Recommendation:** Implement object-level authorization checks. Verify that the authenticated user has permission to access the specific resource before returning data.

CRITICAL CVSS 9.8 CVE-2024-XXXXX CWE-89 Injection

SQL Injection in Product Search (Time-Based Blind)

Affected: `GET /api/v1/products/search?q=` (parameter: q)

The search parameter is vulnerable to time-based blind SQL injection. An attacker can extract database contents including user credentials and payment information.

EVIDENCE

```
Payload: q=1' AND (SELECT SLEEP(5))-- Response time: 5.02s (baseline: 0.12s) Database: MySQL 8.0.35
```

> **Recommendation:** Use parameterized queries or an ORM for all database interactions. Never concatenate user input into SQL strings.

HIGH — 3 findings

HIGH CVSS 8.2 CWE-347 Authentication

JWT Algorithm Confusion (alg: none accepted)

Affected: `POST /api/v1/auth/login`

The API accepts JWT tokens with algorithm set to 'none', allowing token forgery without knowing the secret key.

EVIDENCE

```
Modified token header: {"alg": "none", "typ": "JWT"} Server accepted forged token with admin role.
```

> **Recommendation:** Explicitly validate the JWT algorithm on the server side. Reject tokens with alg='none' or unexpected algorithms. Use a well-maintained JWT library.

HIGH CVSS 7.1 CWE-79 XSS

Cross-Site Scripting (Reflected XSS) in Error Page

Affected: `GET /search?error=` (parameter: `error`)

The error parameter is reflected in the response without sanitization, allowing execution of arbitrary JavaScript in the user's browser.

EVIDENCE

```
Response body contains:
```

> **Recommendation:** Sanitize and encode all user input before rendering in HTML. Implement Content-Security-Policy headers.

HIGH CVSS 8.6 CWE-918 SSRF

Server-Side Request Forgery (SSRF) via Image URL

Affected: `POST /api/v1/profile/avatar` (parameter: `image_url`)

The avatar URL parameter allows fetching internal resources. An attacker can access cloud metadata endpoints and internal services.

EVIDENCE

```
Payload: image_url=http://169.254.169.254/latest/meta-data/iam/security-credentials/
Response: 200 OK with AWS credentials
```

> **Recommendation:** Validate and whitelist allowed URL schemes and hosts. Block requests to private IP ranges (10.x, 172.16-31.x, 192.168.x, 169.254.x).

MEDIUM — 4 findings

MEDIUM CVSS 5.3 CWE-693 Headers

Missing Content-Security-Policy Header

Affected: `https://demo-shop.example.com`

The application does not set a Content-Security-Policy header, making it easier to exploit XSS vulnerabilities.

> **Recommendation:** Add a Content-Security-Policy header. Start with: default-src 'self'; script-src 'self'; style-src 'self' 'unsafe-inline'

MEDIUM CVSS 4.3 CWE-1275 Cookies

Cookies Without SameSite Attribute

Affected: `https://demo-shop.example.com`

Session cookies are set without the SameSite attribute, potentially enabling CSRF attacks in older browsers.

EVIDENCE

```
Set-Cookie: session=abc123; Path=/; HttpOnly (missing: SameSite)
```

> **Recommendation:** Set SameSite=Lax (or Strict) on all cookies, especially session cookies.

MEDIUM CVSS 5.4 CWE-942 CORS

CORS Allows Wildcard with Credentials

Affected: https://demo-shop.example.com/api/

The API returns Access-Control-Allow-Origin: * with Access-Control-Allow-Credentials: true, allowing any website to make authenticated requests.

EVIDENCE

```
Access-Control-Allow-Origin: * Access-Control-Allow-Credentials: true
```

> **Recommendation:** Replace wildcard origin with an explicit whitelist of trusted domains. Never combine wildcard origin with credentials.

MEDIUM CVSS 5.3 CWE-548 Information Disclosure

Directory Listing Enabled on /assets/

Affected: https://demo-shop.example.com/assets/

The web server has directory listing enabled, exposing file structure and potentially sensitive files.

> **Recommendation:** Disable directory listing in your web server configuration. Add 'autoindex off;' in Nginx or 'Options -Indexes' in Apache.

LOW — 2 findings

LOW CVSS 2.1 Headers

Server Version Disclosed in Headers

Affected: https://demo-shop.example.com

The Server header reveals the software version (nginx/1.24.0), helping attackers identify known vulnerabilities.

EVIDENCE

```
Server: nginx/1.24.0
```

> **Recommendation:** Remove or obscure the Server header. In Nginx: server_tokens off;

LOW CVSS 2.1 Headers

X-Powered-By Header Present

Affected: https://demo-shop.example.com

The X-Powered-By header reveals the backend technology (Express), aiding reconnaissance.

EVIDENCE

```
X-Powered-By: Express
```

Recommendation: Remove the X-Powered-By header. In Express: app.disable('x-powered-by')

INFO — 2 findings

INFO SEO

Missing robots.txt File

Affected: https://demo-shop.example.com/robots.txt

No robots.txt file found. Search engine crawlers may index sensitive pages.

Recommendation: Create a robots.txt file to guide search engine crawlers and prevent indexing of sensitive paths.

INFO SSL

Mixed Content Warning

Affected: https://demo-shop.example.com/checkout

The checkout page loads one image over HTTP instead of HTTPS.

EVIDENCE

Recommendation: Ensure all resources are loaded over HTTPS. Use protocol-relative URLs or enforce HTTPS.

03 · PROCESS

Methodology & Limitations

Pipeline. AuditCore runs a 5-phase pipeline: **(1) Reconnaissance** (DMARC/SPF, subdomain enumeration, port mapping), **(2) Crawling** (Playwright through ZAP proxy capturing HTTP traffic), **(3) Auth Testing** (JWT, OAuth, session, headers, cookies, multi-role replay for BOLA/BFLA when credentials are provided), **(4) Injection & Fuzzing** (sqlmap, Nuclei, ZAP active, custom analyzers covering SQLi/XSS/SSRF/SSTI/CMD/CRLF/path-traversal/etc.), and **(5) Static Analysis** (Semgrep, gitleaks, Trivy when source code is provided; MobSF + 6 native modules for APK/IPA).

AI triage. Findings from noise-prone scanners (Nuclei templates, banner-grab CVE matches, header probes) are passed through a Claude-based triage layer that flags likely false positives. Only HIGH-confidence FP flags are auto-applied to keep the trust signal honest; you can revert any decision in the dashboard.

Limitations. Automated scanning has fundamental boundaries. We cannot guarantee discovery of all vulnerabilities — particularly: complex business-logic flaws beyond our heuristics, vulnerabilities requiring specific application state, dynamic vulnerabilities only present under load, or zero-day issues without published signatures. This report supplements but does not replace manual penetration testing for high-stakes systems. A WAF in front of the target may further reduce coverage; see banners above.

Authorization. Intrusive scanners (sqlmap, ZAP active, Nuclei active templates, ffuf bruteforce) only run after the target domain has been verified as owned by the requester via DNS TXT or HTTP file challenge. Scans of unverified domains are limited to passive reconnaissance.

Confidentiality. This report contains sensitive security findings about `https://demo-shop.example.com`. It is intended only for the recipient and their authorized representatives. Do not publish, share publicly, or transmit over insecure channels.

Generated by AuditCore (audit-core.tech) on `2026-05-10 12:15 UTC`.
Operated by Valer Krystian Szozda, NIP 6912545436, Poland.

Questions, refunds, support: `contact@audit-core.tech`