Security Scan Report

Automated Penetration Test Results

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

Tier: ENTERPRISE | **Generated:** 2026-04-11 23:04 UTC

Scan ID: a7f3e291-4b8c-4d9e-bf12-c8a7d3e5f690

Executive Summary

This automated security scan identified **13 findings** across the target application.

2	3	4	2	2
CRITICAL	HIGH	MEDIUM	LOW	INFO

⚠ **5 critical/high severity issues require immediate attention.**

Findings

CRITICAL (2)

CRITICAL Broken Object Level Authorization (BOLA) on User Profile API

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

CVSS: 9.1

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.

```
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.

CWE-639 Authorization

CRITICAL SQL Injection in Product Search (Time-Based Blind)

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

CVSS: 9.8

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

```
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.

CVE-2024-XXXXX CWE-89 Injection

HIGH (3)

HIGH JWT Algorithm Confusion (alg: none accepted)

Affected: POST /api/v1/auth/login

CVSS: 8.2

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

```
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.

CWE-347 Authentication

HIGH Cross-Site Scripting (Reflected XSS) in Error Page

Affected: GET /search?error= (param: error)

CVSS: 7.1

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

```
Response body contains:
```

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

CWE-79 XSS

HIGH Server-Side Request Forgery (SSRF) via Image URL

Affected: POST /api/v1/profile/avatar (param: image_url)

CVSS: 8.6

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

```
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).

CWE-918 SSRF

MEDIUM (4)

MEDIUM Missing Content-Security-Policy Header

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

CVSS: 5.3

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'

CWE-693 Headers

MEDIUM Cookies Without SameSite Attribute

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

CVSS: 4.3

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

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

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

CWE-1275 Cookies

MEDIUM CORS Allows Wildcard with Credentials

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

CVSS: 5.4

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

```
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.

CWE-942 CORS

MEDIUM Directory Listing Enabled on /assets/

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

CVSS: 5.3

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.

CWE-548 Information Disclosure

LOW (2)

LOW Server Version Disclosed in Headers

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

CVSS: 2.1

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

```
Server: nginx/1.24.0
```

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

Headers

LOW X-Powered-By Header Present

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

CVSS: 2.1

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

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

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

Headers

INFO (2)

INFO 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.

SEO

INFO **Mixed Content Warning**

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

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

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

SSL