PlainGuard

PENETRATION TEST REPORT

Acme Corporation — Development Environment

dev.acme-app.example | 203.0.113.42 (Cloud Region US-Central)

Engagement ID: PT-2026-SAMPLE
Date: February 4, 2026
Tester(s): PlainGuard Security Team
Methodology: OWASP Top 10 + PTES
Classification: CONFIDENTIAL

SAMPLE REPORT — Data Anonymized for Portfolio Use

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY

A penetration test was conducted against the client's development environment. The assessment identified **12 findings**: 1 High, 5 Medium, 5 Low, and 1 Informational (reclassified). The most critical finding is a potential SSRF vulnerability in a webhook endpoint that accepts arbitrary URL payloads without authentication.

The application demonstrates strong security fundamentals — authentication, CSRF protection, rate limiting, and route protection all function correctly. However, the development environment lacks the WAF and security header protections that should be present in production, and several API endpoints expose unnecessary information or accept overly permissive input.

Severity	Count
Critical	0
High	1
Medium	5
Low	5
Informational	1
Total	**12**

2. ENGAGEMENT DETAILS

Engagement ID	PT-2026-SAMPLE
Date	2026-02-04
Tester(s)	PlainGuard Security Team
Target Env.	Development (dev.acme-app.example)
Target IP	203.0.113.42 (Cloud Region US-Central)
Methodology	OWASP Top 10 + PTES
Tools Used	curl, testssl.sh, Nuclei, dig, custom toolkit
Report Status	Sample

3. FINDINGS SUMMARY

ID	Severity	CVSS	Title	Status
F-001	High	7.5	Webhook Endpoint — Potential SSRF	Open
F-002	Medium	6.1	CORS Wildcard on API Endpoints	Open
F-003	Medium	5.3	No WAF on Dev Environment	Open
F-004	Medium	5.3	Missing Security Headers (All 6)	Open
F-005	Medium	5.3	Debug Endpoints Accept All Methods	Open
F-006	Medium	5.3	API Endpoint Returns 500 Errors	Open
F-007	Medium	4.3	Session Cookies Missing Secure Flag	Open
F-008	Low	3.7	Server Version Disclosure	Open
F-009	Low	3.7	DMARC Policy Set to None	Open
F-010	Low	3.7	Health Endpoint Info Disclosure	Open
F-011	Low	3.1	Config File Publicly Accessible	Open
F-012	Low	3.1	API Exposes Internal Catalog	Open

4. DETAILED FINDINGS

F-001: Webhook Endpoint Accepts All Payloads (Potential SSRF)

Severity	High | CVSS 7.5
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:N/A:N
Endpoint	https://dev.acme-app.example/api/webhooks/integration
CWE	CWE-918 (Server-Side Request Forgery)
OWASP	A10:2021 Server-Side Request Forgery

Description

The webhook endpoint accepts arbitrary JSON payloads without authentication and returns {"success":true} for all of them, including payloads containing SSRF vectors. All tested payloads — including URLs targeting the cloud instance metadata service at 169.254.169.254, localhost services, file:// protocol, and IPv6 loopback — returned HTTP 200 with success. While it is unknown whether the server actually follows these URLs, the lack of input validation and authentication represents a significant risk.

Evidence

```
$ curl -X POST .../api/webhooks/integration \
-H "Content-Type: application/json" \
-d '{"url": "http://169.254.169.254/latest/meta-data/"}'
{"success":true}

$ curl -X POST ... -d '{"url": "file:///etc/passwd"}'
{"success":true}

All 8 SSRF payloads tested returned HTTP 200 {"success":true}.
```

Impact

If the server processes URL fields from the payload: an attacker could access the cloud IMDS to retrieve instance metadata, managed identity tokens, or subscription information. Internal services on localhost or the private network could be scanned. Local files could potentially be read via file:// protocol. Even without URL processing, the lack of authentication means anyone can send fake webhook data.

Remediation

• Add webhook authentication — implement HMAC signature verification or shared secret validation.

• Validate input — only accept expected fields from the integration API.

• Block internal URLs — if any URL processing occurs, block private IP ranges (10.x, 172.16-31.x, 192.168.x, 169.254.x, 127.x, ::1).

• Return proper errors — reject malformed payloads with 400/422 instead of accepting everything.

F-002: CORS Wildcard on API Endpoints

Severity	Medium | CVSS 6.1
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N
Endpoint	https://dev.acme-app.example/api/*
CWE	CWE-942 (Permissive Cross-domain Policy)
OWASP	A05:2021 Security Misconfiguration

Description

The API endpoints return Access-Control-Allow-Origin: * for all origins, including malicious domains and null. While Access-Control-Allow-Credentials is not set (meaning cookies won't be sent cross-origin), this allows any website to read API responses via JavaScript.

Evidence

```
$ curl -I -H "Origin: https://evil.example" \
https://dev.acme-app.example/api/public/initialize
Access-Control-Allow-Origin: *

$ curl -I -H "Origin: null" \
https://dev.acme-app.example/api/public/initialize
Access-Control-Allow-Origin: *
```

Impact

Any website can make cross-origin requests to the API and read the responses. Combined with other information disclosure findings, an attacker's website could silently enumerate internal data and capabilities.

Remediation

• Configure the application's CORS settings to restrict allowed_origins to specific domains.

F-003: No WAF on Dev Environment

Severity	Medium | CVSS 5.3
Component	Infrastructure
CWE	CWE-16 (Configuration)
OWASP	A05:2021 Security Misconfiguration

Description

The development environment resolves directly to the application server, bypassing the CDN/WAF entirely. The production environment uses a CDN with WAF and DDoS protection, but dev is directly exposed to the internet.

Evidence

```
dev.acme-app.example. CNAME app-dev.cloudprovider.example.
Resolved IP: 203.0.113.42 (Cloud Region US-Central)
```

Impact

The dev environment is directly exposed to the internet without WAF protection, making it susceptible to automated scanning, DDoS, and exploitation attempts.

Remediation

• Route dev through the CDN/WAF, or add access restrictions to limit source IPs.

F-004: Missing Security Headers (All 6)

Severity	Medium | CVSS 5.3
Component	Web Application
CWE	CWE-693 (Protection Mechanism Failure)
OWASP	A05:2021 Security Misconfiguration

Description

All six recommended security headers are absent from responses: Strict-Transport-Security (HSTS), X-Content-Type-Options, X-Frame-Options, Content-Security-Policy, Referrer-Policy, and Permissions-Policy.

Evidence

```
Headers checked via curl -I. None of the 6 security headers are present in the response.
```

Impact

Without these headers, the application is more vulnerable to clickjacking, MIME-type confusion, cross-site scripting escalation, and information leakage through referrer headers.

Remediation

• Add middleware or server configuration to set all six headers.

• Minimum: HSTS max-age=31536000; X-Content-Type-Options: nosniff; X-Frame-Options: DENY; CSP: default-src 'self'; Referrer-Policy: strict-origin-when-cross-origin; Permissions-Policy: camera=(), microphone=(), geolocation=().

F-005: Debug Endpoints Accept All HTTP Methods

Severity	Medium | CVSS 5.3
Endpoints	/api/webhooks/*/test
CWE	CWE-749 (Exposed Dangerous Method)

OWASP	A05:2021 Security Misconfiguration

Description

Test endpoints for various webhooks accept all HTTP methods (GET, POST, PUT, DELETE, PATCH, OPTIONS) and return HTTP 200 with detailed debug information including expected payload schemas, internal webhook URLs, request metadata, and test IDs with timestamps.

Evidence

```
All HTTP methods return 200 with full payload schema and internal URL disclosure.
```

Impact

Information disclosure of internal API schemas and URLs. Test endpoints should not be accessible in deployed environments.

Remediation

• Remove test endpoints entirely from non-local environments, or restrict to authenticated access.

• If kept, restrict to POST method only and remove expected payload schema from responses.

F-006: API Endpoint Returns 500 on All Requests

Severity	Medium | CVSS 5.3
Endpoint	https://dev.acme-app.example/api/webhooks/partner
CWE	CWE-755 (Improper Handling of Exceptional Conditions)
OWASP	A05:2021 Security Misconfiguration

Description

A partner webhook endpoint returns HTTP 500 with an HTML error page for all request content types. This indicates an unhandled exception in the webhook controller.

Evidence

```
JSON, XML, form-urlencoded, and plain text content types all produce HTTP 500 HTML error
pages.
```

Impact

Partner webhooks are not being processed (broken integration). 500 errors may leak stack traces if debug mode is enabled.

Remediation

• Fix the root cause of the 500 error in the webhook handler.

• Add try/catch blocks to return proper JSON error responses.

• Ensure debug mode is disabled in all non-local environments.

F-007: Session Cookies Missing Secure Flag

Severity	Medium | CVSS 4.3
Component	Session Management
CWE	CWE-614 (Sensitive Cookie Without Secure Attribute)
OWASP	A05:2021 Security Misconfiguration

Description

Session cookies are set without the Secure flag. If a user is tricked into visiting the HTTP version of the site, cookies would be transmitted in plaintext.

Evidence

```
Set-Cookie headers for session tokens lack the Secure attribute.
```

Impact

Session cookies could be intercepted via man-in-the-middle if HTTP is used.

Remediation

• Enable secure cookies in the application configuration.

F-008: Server Version Disclosure

Severity	Low | CVSS 3.7
Component	Web Server
CWE	CWE-200 (Exposure of Sensitive Information)

Description

Response headers disclose specific web server and runtime version numbers.

Evidence

```
curl -I response shows server and runtime version headers.
```

Impact

Attackers can use version information to identify known vulnerabilities for these specific versions.

Remediation

• Configure the web server to suppress version information in response headers.

F-009: DMARC Policy Set to None

Severity	Low | CVSS 3.7
Component	Email Security
CWE	CWE-290 (Authentication Bypass by Spoofing)

Description

DMARC is configured with p=none, which monitors but does not reject spoofed emails from the domain.

Evidence

```
DNS TXT record shows DMARC p=none policy.
```

Impact

Attackers could spoof emails from the domain. DMARC reports are collected but no enforcement occurs.

Remediation

• Gradually tighten: p=none → p=quarantine → p=reject after reviewing DMARC reports.

F-010: Health Endpoint Exposes Infrastructure Details

Severity	Low | CVSS 3.7
Component	Web Application
CWE	CWE-200 (Exposure of Sensitive Information)

Description

The unauthenticated /health endpoint returns JSON confirming database and cache connectivity with response times, disclosing infrastructure details.

Evidence

```
/health returns detailed status of backend services including response times.
```

Impact

Reveals infrastructure components and their response characteristics.

Remediation

• Return only {"healthy":true} publicly. Move detailed checks behind authentication or restrict to internal networks.

F-011: Config File Publicly Accessible

Severity	Low | CVSS 3.1
Component	Web Server
CWE	CWE-538 (Sensitive Info in Externally-Accessible File)

Description

A server configuration file is publicly accessible (HTTP 200), exposing rewrite rules. This is likely a leftover from the application's public directory.

Evidence

```
curl returns HTTP 200 with file contents.
```

Impact

Discloses rewrite rules and internal routing logic.

Remediation

• Block access to configuration files in the web server configuration.

F-012: API Exposes Internal Catalog

Severity	Low | CVSS 3.1
Component	Public API
CWE	CWE-200 (Exposure of Sensitive Information)

Description

An unauthenticated API initialization endpoint returns the complete internal catalog (50+ items across 15 categories), service configuration, and capability flags.

Evidence

```
GET request to initialization endpoint returns full catalog and internal configuration flags.
```

Impact

Competitors could enumerate full offerings. Configuration flags provide internal implementation details.

Remediation

• Remove internal configuration flags from the response (information disclosure).

• Rate limit the endpoint.

• Review whether the full catalog should be exposed or just categories.

5. POSITIVE FINDINGS

The following security controls were tested and confirmed working correctly:

Control	Test Performed	Result
Admin route protection	Accessed /admin/* unauthenticated	302 redirect (PASS)
User route protection	Accessed /user/* unauthenticated	302 redirect (PASS)
CSRF protection	POST without CSRF token	419 (PASS)
Login rate limiting	10 rapid login attempts	Rate limited at 6 (PASS)
API authentication	No/invalid/empty tokens	401 (PASS)
Secondary API auth	No auth + wrong token	401 (PASS)
TLS configuration	Full TLS scan	Grade A+ (PASS)
TLS protocols	Protocol version check	TLS 1.2 + 1.3 only (PASS)
XSS reflection	Injected payloads in login + URL params	No reflection (PASS)
Open redirect	7 redirect parameter patterns	None found (PASS)
HTTP methods	TRACE/PUT/DELETE/PATCH on root	405 (PASS)
.env protection	Accessed /.env	404 (PASS)
.git protection	Accessed /.git/config	403 (PASS)
Debug endpoints	Debug/profiler endpoints	All 404 (PASS)
Registration disabled	Accessed /register	404 (PASS)
API rate limiting	Rapid POST requests	429 after ~10 (PASS)
SPF record	DNS check	SPF hard fail (PASS)

6. SCOPE LIMITATIONS

• No authenticated testing performed — All tests were unauthenticated. RBAC matrix validation, data isolation, and IDOR testing require valid credentials.

• Cloud configuration not reviewed — Requires cloud provider console access with appropriate permissions.

• Identity and access review not performed — Requires identity provider access.

• API prompt injection incomplete — Rate limiting prevented full testing. Initial tests with wrong payload format consumed the rate limit.

• SSRF unconfirmed — The endpoint accepts SSRF payloads and returns success, but it could not be confirmed if the server actually follows the URLs. Code review is needed.

• Automated scan — Running in background; results to be appended when available.

7. RECOMMENDED PRIORITY ACTIONS

Immediate (High Priority)

Investigate the webhook endpoint — review code to determine if URL fields are processed. Add authentication to all webhook endpoints.

Short-term (Medium Priority)

Add security headers across all responses. Fix CORS configuration to restrict allowed origins. Set the Secure cookie flag. Fix the broken webhook endpoint.

Medium-term (Low Priority)

Remove debug/test endpoints from non-local environments. Restrict health endpoint details. Configure DMARC enforcement. Suppress server version headers.

Next Assessment

Perform authenticated testing with valid credentials for RBAC/IDOR validation. Conduct cloud configuration review. Complete prompt injection testing on the API.

This is a sample report with anonymized data, provided for demonstration purposes. The findings and recommendations herein represent typical security assessment results and reflect common vulnerabilities found in web application penetration tests.

Sample report generated by PlainGuard.