LORIKEET SECURITY

Penetration Testing Report

Web Application Security Assessment

Prepared for **Acme Corporation**

April 7, 2026

Prepared For	**Acme Corporation**
Project	Web Application Security Assessment
Date	April 7, 2026
Testing Period	March 28 – April 4, 2026
Lead Tester	Jane Doe
SOW Reference	SOW-2026-042
Scope	4 targets (see Section 2)
Prepared By	Lorikeet Security
Contact	security@lorikeetsecurity.com
Classification	CONFIDENTIAL

VERSION	STATUS	AUTHOR	DATE
1.0	**Final**	Jane Doe	April 7, 2026

lorikeetsecurity.com | security@lorikeetsecurity.com

Table of Contents

1. Executive Summary

Lorikeet Security conducted a **Web Application** penetration testing engagement for **Acme Corporation** on the **Web Application Security Assessment** project. This report presents all findings identified during the assessment, along with risk ratings, evidence, and actionable remediation guidance.

A total of **7** unique vulnerabilities were identified, of which **3** are rated Critical or High severity and require immediate attention.

Assessment Narrative

The assessment identified several critical vulnerabilities that pose an immediate risk to Acme Corporation's web application infrastructure. Priority should be given to the SQL injection and XSS findings, which could be chained by a skilled adversary to achieve full account takeover.

Risk Scorecard

SEVERITY	COUNT	%	PRIORITY TIMELINE
Critical	1	14%	Immediate (24–48 hours)
High	2	29%	Within 7 days
Medium	2	29%	Within 30 days
Low	1	14%	Within 90 days
Info	1	14%	Best effort

Key Recommendations

1. **SQL Injection in Login Form** [Critical]
2. **Cross-Site Scripting (XSS) in Search** [High]
3. **Insecure Direct Object Reference (IDOR)** [High]
4. **Missing Rate Limiting on Authentication** [Medium]
5. **TLS 1.0/1.1 Enabled** [Medium]

2. Scope & Methodology

Scope

Statement of Work Reference: SOW-2026-042

Testing Period: March 28 – April 4, 2026

Conducted By: Jane Doe

In-Scope Targets

#	Target / Asset
1	https://app.example.com
2	https://api.example.com
3	https://admin.example.com
4	10.0.0.0/24

Points of Contact

Client Technical Lead: Alice Johnson <alice@acme.com>
Client Project Manager: Bob Lee <bob@acme.com>
Lorikeet Security: security@lorikeetsecurity.com

Methodology

The assessment followed industry-standard methodologies including:

- OWASP Testing Guide v4.2 - for web application security assessment
- OWASP Application Security Verification Standard (ASVS) v4.0
- PTES (Penetration Testing Execution Standard) - for engagement structure
- NIST SP 800-115 - Technical Guide to Information Security Testing
- MITRE ATT&CK Framework - for attack technique classification

Testing Phases

PHASE	DESCRIPTION
Reconnaissance	Passive and active information gathering, OSINT, DNS enumeration, and service discovery.
Mapping & Analysis	Application mapping, authentication flow analysis, input vector identification, and business logic review.
Exploitation	Manual and automated exploitation of identified vulnerabilities with proof-of-concept demonstrations.
Post-Exploitation	Impact assessment, privilege escalation testing, lateral movement analysis, and data access

	verification.
Reporting	Documentation of findings with risk ratings, evidence, and actionable remediation guidance.

3. Risk Rating Methodology

Findings are categorised using the Common Vulnerability Scoring System (CVSS v3.1) and mapped to severity levels consistent with industry standards:

SEVERITY	CVSS RANGE	DESCRIPTION
Critical	9.0 – 10.0	Exploitation is trivial and leads to full system compromise, data breach, or complete loss of confidentiality, integrity, or availability. Immediate remediation required.
High	7.0 – 8.9	Exploitation could lead to significant data exposure, privilege escalation, or service disruption. Should be addressed within 7 days.
Medium	4.0 – 6.9	Exploitation requires specific conditions but could lead to partial information disclosure or limited impact. Address within 30 days.
Low	0.1 – 3.9	Minor security weakness with limited exploitability or impact. Represents defence-in-depth opportunities. Address within 90 days.
Info	0.0	Informational observation or best-practice recommendation. No direct security impact but may improve overall security posture.

4. Vulnerability Statistics

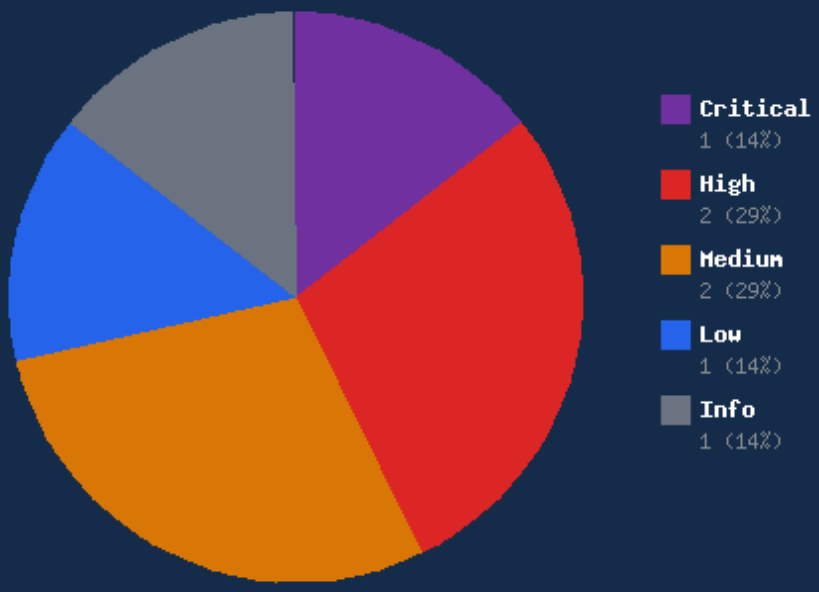
Critical	High	Medium	Low	Info
1	2	2	1	1

Risk Matrix

Findings plotted by likelihood of exploitation vs. impact on the organisation:

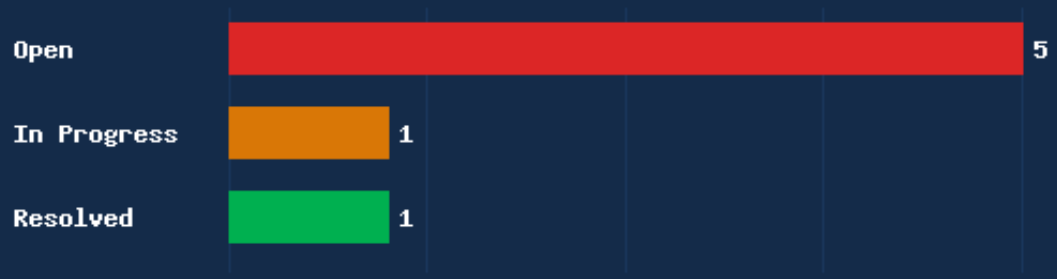
LIKELIHOOD / IMPACT	Critical	High	Medium	Low	Info
High	1	2	-	-	-
Medium	-	-	2	-	-
Low	-	-	-	1	1

5. Severity Distribution



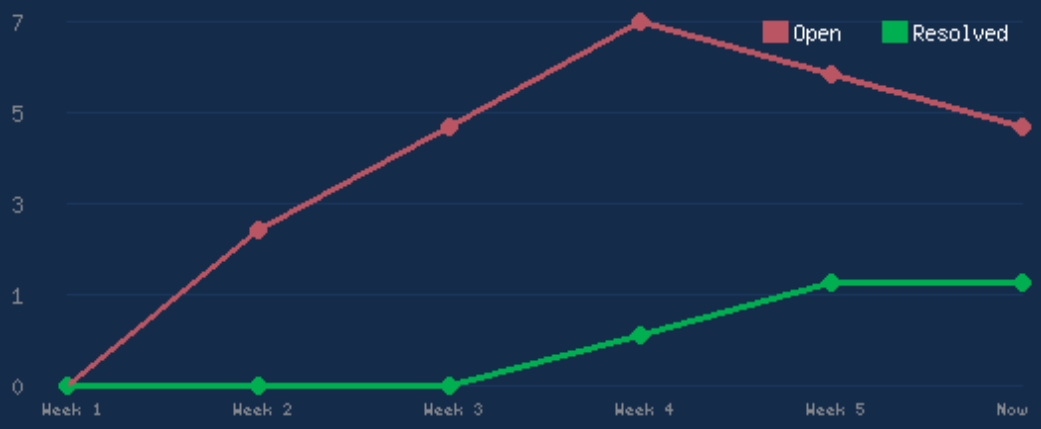
6. Remediation Status

Open	In Progress	Resolved
5	1	1



7. Engagement History

Vulnerability trend over the engagement period:



Engagement Timeline

PHASE	DATE	STATUS
Scoping & Planning	April 7, 2026	Complete
Active Testing	April 7, 2026	Complete
Analysis & Reporting	April 7, 2026	Complete
Report Delivery	April 7, 2026	Delivered
Remediation Review	TBD	Pending

8. Findings Summary

#	TITLE	SEVERITY	STATUS	CATEGORY
1	**SQL Injection in Login Form**	Critical	Open	Injection
2	**Cross-Site Scripting (XSS) in Search**	High	Open	Cross-Site Scripting
3	**Insecure Direct Object Reference (IDOR)**	High	In Progress	Broken Access Control
4	**Missing Rate Limiting on Authentication**	Medium	Open	Authentication
5	**TLS 1.0/1.1 Enabled**	Medium	Resolved	Cryptography
6	**Sensitive Data in API Response**	Low	Open	Information Disclosure
7	**Server Version Disclosed**	Info	Open	Information Disclosure

9. Detailed Findings

FINDING 1 OF 7

SQL Injection in Login Form CVSS 9.8

Severity	Critical
Status	Open
CVSS Score	9.8
Category	Injection
Affected Asset	https://app.example.com/api/login
CWE Reference	CWE-89 — https://cwe.mitre.org/data/definitions/89.html

Description

The login endpoint is vulnerable to SQL injection via the 'username' parameter. An attacker can bypass authentication and extract sensitive data from the database.

The application concatenates user input directly into SQL queries without parameterized statements or input validation.

Impact

Successful exploitation of this vulnerability could result in complete system compromise, full data exfiltration, or total loss of availability. This represents a maximum-severity risk to the organisation.

Evidence

```
Request: POST /api/login
Body: username=admin' OR '1'='1&password=anything
Response: HTTP 200 OK with valid session token, bypassing authentication entirely.
```

Remediation

1. Use parameterized queries (prepared statements) for all database interactions.
2. Implement input validation with whitelist approach.

3. Apply the principle of least privilege to database accounts.
4. Deploy a Web Application Firewall (WAF) as defense-in-depth.

FINDING 2 OF 7

Cross-Site Scripting (XSS) in Search CVSS 7.4

Severity	**High**
Status	Open
CVSS Score	**7.4**
Category	Cross-Site Scripting
Affected Asset	https://app.example.com/search
CWE Reference	CWE-79 — https://cwe.mitre.org/data/definitions/79.html

Description

The search functionality reflects user input without encoding, allowing stored and reflected XSS attacks.

Impact

Exploitation could allow an attacker to gain elevated privileges, access sensitive data, or disrupt critical services. Immediate attention is required.

Evidence

Request: GET /search?q=alert(document.cookie)

The script tag is rendered unescaped in the HTML response.

Remediation

1. Implement context-aware output encoding.
2. Use Content Security Policy (CSP) headers.
3. Enable HttpOnly and Secure flags on session cookies.

FINDING 3 OF 7

Insecure Direct Object Reference (IDOR) CVSS 8.1

Severity	**High**
Status	In Progress
CVSS Score	**8.1**
Category	Broken Access Control
Affected Asset	https://app.example.com/api/users/{id}/documents
CWE Reference	CWE-639 — https://cwe.mitre.org/data/definitions/639.html

Description

The API endpoint for user documents does not verify authorization. Modifying the user ID parameter exposes other users' data.

Impact

An authenticated attacker can iterate over user IDs in the API endpoint to enumerate and download documents belonging to other users. This constitutes a broken access control vulnerability allowing horizontal privilege escalation.

Evidence

Authenticated as user 1001, request to /api/users/1002/documents returned documents belonging to user 1002.

Remediation

1. Implement server-side authorization checks.
2. Use indirect object references.
3. Log unauthorized access attempts.

FINDING 4 OF 7

Missing Rate Limiting on Authentication

Severity	**Medium**
Status	Open
Category	Authentication
Affected Asset	https://app.example.com/api/login
CWE Reference	CWE-307 — https://cwe.mitre.org/data/definitions/307.html

Description

The authentication endpoint has no rate limiting or account lockout. Unlimited brute-force attempts are possible.

Impact

An attacker exploiting this vulnerability could access restricted resources or perform limited unauthorised actions under specific conditions.

Evidence

10,000 login attempts in 60 seconds with no throttling or blocking.

Remediation

1. Implement progressive rate limiting.
2. Add account lockout after 5-10 failed attempts.
3. Implement CAPTCHA.

FINDING 5 OF 7

TLS 1.0/1.1 Enabled

Severity	**Medium**
Status	Resolved
Category	Cryptography
Affected Asset	https://app.example.com
CWE Reference	CWE-326 — https://cwe.mitre.org/data/definitions/326.html

Description

The server supports deprecated TLS 1.0 and 1.1 which have known cryptographic weaknesses.

Impact

An attacker exploiting this vulnerability could access restricted resources or perform limited unauthorised actions under specific conditions.

Evidence

```
TLS 1.0: Enabled (VULNERABLE)
TLS 1.1: Enabled (VULNERABLE)
TLS 1.2: Enabled
TLS 1.3: Enabled
```

Remediation

1. Disable TLS 1.0 and 1.1.
2. Use strong cipher suites with forward secrecy.

FINDING 6 OF 7

Sensitive Data in API Response

Severity	**Low**
Status	Open
Category	Information Disclosure
Affected Asset	https://app.example.com/api/users/profile
CWE Reference	CWE-200 — https://cwe.mitre.org/data/definitions/200.html

Description

The profile API returns excessive information including internal IDs and other users' email addresses.

Impact

This vulnerability presents a limited attack surface but could contribute to a multi-step attack chain or weaken the overall security posture.

Evidence

```
Response includes: internal_id, db_created_at, admin_notes, linked_accounts.
```

Remediation

1. Use a response DTO that explicitly defines returned fields.
2. Remove internal identifiers from API responses.

FINDING 7 OF 7

Server Version Disclosed

Severity	Info
Status	Open
Category	Information Disclosure
Affected Asset	https://app.example.com
CWE Reference	CWE-200 — https://cwe.mitre.org/data/definitions/200.html

Description

HTTP headers reveal server software versions, aiding attackers in identifying known CVEs.

Impact

This is an informational finding with no direct security impact. Addressing it will improve the overall security posture and reduce attack surface.

Evidence

```
Server: Apache/2.4.52 (Ubuntu)
X-Powered-By: PHP/8.1.2
```

Remediation

1. Suppress version information in server config.
2. Remove X-Powered-By header.

---

10. Remediation Priority Guide

Lorikeet Security recommends addressing findings in the following priority order:

PRIORITY	SEVERITY	TIMELINE	COUNT
1	Critical	Immediate (24-48 hours)	1
2	High	Within 7 days	2
3	Medium	Within 30 days	2
4	Low	Within 90 days	1
5	Info	Best effort	1

11. Disclaimer & Limitations

1. This penetration test was conducted as a point-in-time assessment. The security posture of the tested systems may change after the assessment due to updates, configuration changes, or new vulnerabilities being discovered.

2. The scope of testing was limited to the systems and applications agreed upon in the statement of work. No testing was performed on systems outside the defined scope.

3. While Lorikeet Security endeavours to identify all vulnerabilities within the defined scope, no penetration test can guarantee the discovery of every security weakness. The absence of findings does not imply the absence of vulnerabilities.

4. Exploitation of identified vulnerabilities was performed in a controlled manner to minimise disruption. Some exploitation paths may not have been fully explored to avoid impacting production services.

5. This report contains confidential information and is intended solely for the authorised recipient(s). Unauthorised distribution, copying, or disclosure of this report or its contents is strictly prohibited.

6. Remediation recommendations are provided as guidance and should be validated in a staging environment before implementation in production. Lorikeet Security is not responsible for any adverse effects resulting from the implementation of recommended fixes.

7. All findings are classified using the CVSS v3.1 scoring system. Risk ratings may be adjusted based on the specific business context and compensating controls that were not visible during testing.

12. Glossary of Terms

TERM	DEFINITION
APT	Advanced Persistent Threat - A prolonged, targeted cyberattack in which an attacker maintains unauthorized access to a network.
CVSS	Common Vulnerability Scoring System - An open standard for assessing the severity of computer system vulnerabilities (v3.1 used in this report).
CWE	Common Weakness Enumeration - A community-developed list of software and hardware weakness types.
DoS / DDoS	Denial of Service / Distributed Denial of Service - An attack that makes a system unavailable to its intended users.
IDOR	Insecure Direct Object Reference - An access control vulnerability where user-supplied input is used to access objects directly.
OWASP	Open Web Application Security Project - A nonprofit foundation focused on improving the security of software.
Pentest	Penetration Test - An authorized simulated cyberattack performed to evaluate the security of a system.
PoC	Proof of Concept - Demonstration that a vulnerability can be exploited, without causing damage.
RCE	Remote Code Execution - A vulnerability allowing an attacker to run arbitrary code on a target system.
SSRF	Server-Side Request Forgery - A vulnerability where an attacker can make the server perform requests to unintended locations.
SQLi	SQL Injection - An attack that inserts malicious SQL code into application queries.
WAF	Web Application Firewall - A firewall that monitors, filters, and blocks HTTP traffic to and from a web application.
XSS	Cross-Site Scripting - A vulnerability allowing attackers to inject client-side scripts into web pages.
Zero-day	A vulnerability that is unknown to the vendor and has no available patch at the time of discovery.

13. Appendix

A. Tools and Techniques

The following tools and techniques were employed during the engagement:

TOOL	PURPOSE
Burp Suite Professional	Web application security testing, intercepting proxy, scanner
Nmap / Masscan	Network discovery, port scanning, service enumeration
Nuclei	Template-based vulnerability scanning and detection
SQLMap	Automated SQL injection detection and exploitation
Metasploit Framework	Exploitation framework for vulnerability validation
Gobuster / ffuf	Directory and file brute-forcing, content discovery
Nikto	Web server misconfiguration scanning
Custom Scripts	Bespoke scripts for business logic and authorization testing
Manual Testing	Expert-driven testing for logic flaws, race conditions, and access control issues

B. Standards and Frameworks Referenced

STANDARD	DESCRIPTION
OWASP Top 10 (2021)	Top 10 web application security risks
OWASP Testing Guide v4.2	Comprehensive web application testing methodology
OWASP ASVS v4.0	Application Security Verification Standard
PTES	Penetration Testing Execution Standard
NIST SP 800-115	Technical Guide to Information Security Testing and Assessment
MITRE ATT&CK	Knowledge base of adversary tactics and techniques
CVSS v3.1	Common Vulnerability Scoring System
CWE / CAPEC	Common Weakness Enumeration / Common Attack Pattern Enumeration

C. CVSS v3.1 Severity Classification

All findings in this report are scored using the Common Vulnerability Scoring System (CVSS) version 3.1. The base score is calculated from exploitability and impact metrics and mapped to the following severity levels:

SCORE	SEVERITY	SLA	ACTION
9.0 - 10.0	Critical	24-48 hours	Immediate escalation to security leadership. Hotfix or compensating control required.
7.0 - 8.9	High	7 days	Prioritize in current sprint. Remediate before next release.
4.0 - 6.9	Medium	30 days	Schedule for next development cycle. Monitor for exploitation.
0.1 - 3.9	Low	90 days	Address as part of regular maintenance or hardening efforts.
0.0	Info	Best effort	Best-practice recommendation. No direct exploitability.

End of Report