PENETRATION TEST REPORT

External API Security Assessment

CONFIDENTIAL

Document Type:	External Penetration Test Report
Target:	REST API (Mobile Application Backend)
Test Type:	Black-box Assessment
Performed by:	Benbo Security Consulting
Contact:	reda@benbo.se

This assessment was performed with explicit written authorization from the system owner.

Table of Contents

1. Executive Summary

Benbo was engaged to perform an external penetration test against the client's REST API, which serves as the backend for a mobile application. The assessment was conducted using black-box methodology without access to source code.

Overall Security Posture

Rating: 5/10 — Foundational security controls are in place, but several high-severity vulnerabilities require immediate remediation before the application can be considered production-ready.

Severity	Count	CVSS Range
Critical	**1**	9.0+
High	**2**	7.0–8.9
Medium	**4**	4.0–6.9
Passed Controls	**6**	N/A

Key Business Risks:

- Account compromise through brute-force attacks (no rate limiting)
- Potential for session hijacking via stored XSS vulnerability
- User enumeration enabling targeted attacks
- Information disclosure through exposed API documentation and debug endpoints

2. Scope & Authorization

2.1 Authorization

This penetration test was performed with explicit written consent from the system owner. Authorization was obtained via direct communication prior to testing commencement.

2.2 Scope

Target System:	Production REST API
Test Type:	External black-box assessment
Authentication:	User-level access (self-registered accounts)
Exclusions:	Denial-of-service testing, social engineering

2.3 Limitations

- No source code access (black-box only)
- No admin-level access provided
- Testing window limited to minimize production impact
- Mobile application binary analysis not in scope

3. Methodology

Testing was conducted in accordance with industry-standard frameworks:

- **OWASP API Security Top 10 (2023)** — Primary reference for API-specific vulnerabilities
- **OWASP Web Security Testing Guide (WSTG)** — General web application testing methodology
- **PTES (Penetration Testing Execution Standard)** — Engagement structure and reporting

Testing Phases:

1. Reconnaissance and API discovery
2. Authentication and session management testing
3. Authorization and access control testing
4. Input validation and injection testing
5. Business logic testing
6. Configuration and deployment review

4. Summary of Findings

ID	Finding	Severity	CVSS
V-01	Missing Rate Limiting on Authentication	**Critical**	9.1
V-02	Stored Cross-Site Scripting (XSS)	**High**	8.1
V-03	Missing Email Verification	**High**	7.2
V-04	Account Enumeration via Registration	**Medium**	5.3
V-05	Exposed API Documentation	**Medium**	5.3
V-06	Active Debug Endpoints	**Medium**	5.3
V-07	Unauthenticated Access to User Ratings	**Medium**	4.3

5. Critical & High Findings

V-01: Missing Rate Limiting on Authentication Endpoint [Critical — CVSS 9.1]

CVSS:3.1 Vector: `AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N`
CWE: CWE-307 (Improper Restriction of Excessive Authentication Attempts)
OWASP API: API4:2023 — Unrestricted Resource Consumption

Description

The authentication endpoint allows unlimited login attempts without any rate limiting, account lockout, or CAPTCHA mechanism. An attacker can perform automated credential stuffing or brute-force attacks.

Evidence

Ten (10) consecutive failed login attempts were submitted without triggering any defensive mechanism. All requests received identical 401 responses with no progressive delays or blocks.

Business Impact

- Account takeover of users with weak passwords
- Credential stuffing attacks using leaked password databases
- Reputational damage from compromised user accounts

Remediation

- Implement rate limiting (e.g., 5 attempts per minute per IP/account)
- Add exponential backoff after failed attempts
- Implement CAPTCHA after 3 failed attempts
- Consider account lockout with secure unlock mechanism

V-02: Stored Cross-Site Scripting (XSS) in Username Field [High — CVSS 8.1]

CVSS:3.1 Vector: `AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:L/A:N`
CWE: CWE-79 (Improper Neutralization of Input During Web Page Generation)
OWASP API: API8:2023 - Security Misconfiguration

Description

The registration endpoint accepts JavaScript code in the username field without sanitization. The payload is stored in the database and may execute when rendered in client applications.

Evidence

Successfully registered a user account with the following username:
```
<script>alert(document.cookie)</script>
```
The payload was stored without modification.

Business Impact

- Session hijacking through cookie theft
- Account takeover without credentials
- Malware distribution to other users

Remediation

- Implement server-side input validation and sanitization
- HTML-encode all user-supplied data on output
- Implement Content Security Policy (CSP) headers
- Use parameterized templates that auto-escape by default

`V-03:` Missing Email Verification on Registration [High — CVSS 7.2]

CVSS:3.1 Vector: `AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:L`
CWE: CWE-287 (Improper Authentication)
OWASP API: API2:2023 — Broken Authentication

Description

New user accounts are activated immediately without email verification. This allows registration with arbitrary or non-existent email addresses.

Business Impact

- Mass creation of fake/automated accounts
- Platform abuse and spam
- Inability to verify user identity for password recovery
- Reduced trust in user data integrity

Remediation

- Implement email verification with time-limited activation tokens
- Restrict account functionality until verification is complete
- Add CAPTCHA to registration to prevent automated signups

6. Medium Findings

V-04: Account Enumeration via Registration Endpoint [Medium — CVSS 5.3]

CWE: CWE-204 (Observable Response Discrepancy)

Description

The registration endpoint returns different error messages for existing vs. non-existing email addresses, allowing attackers to enumerate valid accounts.

Remediation

Return generic responses: "If this email is registered, you will receive further instructions."

V-05: Publicly Exposed API Documentation [Medium — CVSS 5.3]

CWE: CWE-200 (Exposure of Sensitive Information)

Description

Interactive API documentation endpoints are accessible without authentication, exposing the complete API structure, endpoints, and parameters to potential attackers.

Affected Endpoints

- `/docs` — Swagger UI
- `/redoc` — ReDoc documentation
- `/openapi.json` — OpenAPI specification

Remediation

Disable or restrict access to documentation endpoints in production environments.

V-06: Active Debug Endpoints in Production [Medium — CVSS 5.3]

CWE: CWE-489 (Active Debug Code)

Description

Debug endpoints are active in the production environment, potentially exposing sensitive internal information.

Remediation

Remove all debug endpoints before production deployment. Implement environment-based configuration.

V-07: Unauthenticated Access to User Ratings [Medium — CVSS 4.3]

CWE: CWE-284 (Improper Access Control)

Description

User rating data is accessible without authentication, enabling profiling and data harvesting.

Remediation

Require authentication or limit exposed information to non-sensitive aggregates.

7. Passed Security Controls

The following security controls were tested and found to be properly implemented:

Control	Status	Notes
HTTPS Enforcement	✓ **Passed**	TLS 1.3
Password Policy	✓ **Passed**	Min 8 chars
Admin Privilege Escalation	✓ **Passed**	Blocked
SQL Injection	✓ **Passed**	Not vulnerable
Path Traversal	✓ **Passed**	Not vulnerable
JWT Token Manipulation	✓ **Passed**	Properly signed

8. Remediation Roadmap

Immediate (Within 7 Days) — Critical/High Severity

7. Implement rate limiting on authentication endpoints
8. Add input sanitization for all user-supplied fields
9. Disable or secure API documentation in production
10. Remove all debug endpoints

Short-term (Within 30 Days) — Medium Severity

11. Implement email verification workflow
12. Normalize error messages to prevent enumeration
13. Audit all public endpoints for appropriate access controls

9. Conclusion

This assessment identified several vulnerabilities that require remediation before the API can be considered secure for production use. While foundational controls such as HTTPS, SQL injection prevention, and JWT implementation are properly configured, the critical and high-severity findings present significant risk to user accounts and data integrity.

A follow-up assessment is recommended after remediation to verify that fixes have been properly implemented and no regression has occurred.

Appendix A: CVSS Scoring Reference

This report uses CVSS v3.1 (Common Vulnerability Scoring System) for severity ratings:

Score	Severity	Description
9.0–10.0	**Critical**	Immediate exploitation likely with severe impact
7.0–8.9	**High**	Exploitation probable with significant impact
4.0–6.9	**Medium**	Exploitation possible with moderate impact
0.1–3.9	**Low**	Limited exploitability or impact

— End of Report —

Benbo | reda@benbo.se