EXECUTIVE SECURITY ASSESSMENT · WEB & API

crAPI

Executive Report — API Security Assessment

Assessment date: 19 February 2026
Prepared by: Eric Muturi and David Munene

> **Lab assessment / demonstration.** Performed against **crAPI**, an intentionally vulnerable, open-source application published by the OWASP Foundation for security training. This is **not** a client engagement and contains no real customer data.

Executive Summary

This report summarises the findings of a web and API security assessment conducted against the crAPI application. Testing identified severe security vulnerabilities that would, in a production system, pose an immediate and critical risk to application stability, user data, and business operations.

The application has fundamental flaws in user authentication and data access controls which, if exploited, could lead to catastrophic data breaches and full administrative takeover. We identified **6 vulnerabilities** — **4 CRITICAL** and **2 HIGH**.

Overall Risk Level: CRITICAL

Total Vulnerabilities Identified	6
Critical Risk Items	4
High Risk Items	2
Medium / Low Risk Items	0
API Endpoints Tested	18

Application & Testing Context

Application Target	crAPI Application (http://127.0.0.1:8888)
Application Type	Web Application
Key Technologies	Node.js, MongoDB, PostgreSQL
Authentication	JWT Token
Testing Method	Structured methodology based on OWASP, PTES and NIST, focused on API security weaknesses

Summary of Findings

The assessment focused on the OWASP API Security Top 10 and revealed significant failures across the tested areas.

1. Complete Administrative System Takeover

CRITICAL CVSS 9.1 | API2:2023 Broken Authentication & API5:2023 Broken Function Level Authorization

Description: There is a fundamental failure in how the application validates user identity tokens (JWTs). The system accepts a completely fake, unsigned token, allowing an attacker to trick the application into granting the highest level of administrative access.

Business impact: System-wide compromise. An attacker can gain full administrative control, steal, modify or permanently delete all stored data, shut down services, and establish backdoors. This threatens major financial loss, severe regulatory fines, and total business interruption.

Recommendation: Immediately harden token validation. Enforce strong signature verification, explicitly reject unsigned tokens, and perform authorization checks against trusted server-side records rather than user-supplied token data.

2. Broken Authorization & Mass Data Theft

CRITICAL CVSS 9.1 | API1:2023 Broken Object Level Authorization (BOLA)

Description: The application fails to check whether a user is authorized to view specific data. Any authenticated user can access the private service reports and order details of any other user simply by changing an ID number in a request.

Business impact: This compromises all user privacy, leading to a mass breach of personally identifiable information (PII) including email addresses, phone numbers, vehicle information, and order history. Consequences include loss of customer trust, regulatory penalties (e.g. GDPR, CCPA), and potential class-action lawsuits.

Recommendation: Immediately implement strict server-side controls that verify user ownership for every data request; deny any request for data not belonging to the authenticated user. Use non-sequential IDs (e.g. UUIDs) to prevent bulk enumeration.

3. Unintended Sensitive Data Leakage

HIGH CVSS 7.5 | Excessive Data Exposure

Description: The community-features API accidentally includes private details belonging to other users in its standard responses. When a user views a community post, the response contains unauthorized data such as other users' email addresses and vehicle IDs.

Business impact: A continual, low-level leakage of customer PII that is not authorized for public view, significantly increasing the privacy risk profile and the likelihood of regulatory scrutiny and penalties.

Recommendation: Review and strictly filter all API responses so they include only the minimum data required for the specific function being performed.

4. Remote Code Execution Gateway

HIGH CVSS 8.8 | Unrestricted File Upload

Description: The user-picture upload feature performs no proper file-type validation. An attacker can upload dangerous, executable code disguised as an image file.

Business impact: This creates a critical pathway for a full-scale attack, including Remote Code Execution (RCE). Malicious code executing on the server could lead to complete compromise of the underlying infrastructure and all hosted data.

Recommendation: Restrict uploads to a whitelist of safe image formats (e.g. JPEG, PNG) and implement rigorous server-side validation of true file content and signature before storage.

Conclusion & Required Actions

The identified vulnerabilities indicate a fundamentally insecure application that would require immediate and comprehensive remediation in a production setting.

- **Crisis response (24 to 48 hours):** address all CRITICAL findings to prevent active takeover and large-scale data theft.
- **High-priority fixes (7 days):** implement HIGH severity fixes to close data-leakage and compromise pathways.
- **Mandatory security review:** conduct a full code review focused on JWT validation and object-level authorization.
- **Process integration:** embed security testing into the SDLC.
- **Verification:** retest after remediation to confirm all issues are fully resolved.

crAPI is a public OWASP practice target; this document is a demonstration of methodology and business-risk communication, not a confidential client engagement.