API Security Risk Analysis

Project Report

Submitted By Sachin Kumar

Table of Contents

1. Executive Summary

This report presents a detailed, read-only API Security Risk Analysis of publicly accessible REST API endpoints. The primary objective of this assessment is to identify common security weaknesses related to authentication, authorization, data exposure, and abuse risks that are frequently observed in modern SaaS (Software as a Service) applications.

The analyzed APIs return structured JSON responses and are accessible without authentication. While such configurations are often acceptable for testing or demo environments, they may introduce serious security, privacy, and compliance risks if similar designs are deployed in production systems.

All findings documented in this report are based solely on passive observation techniques, including API response review, HTTP header inspection, pagination behavior analysis, and documentation review. No exploitation, bypass attempts, or intrusive testing activities were conducted.

This assessment demonstrates a security-first mindset focused on identifying risks, understanding business impact, and recommending practical remediation steps aligned with industry best practices.

2. Scope of Assessment

The scope of this assessment was strictly limited to public and openly accessible API endpoints.

✓ **Included in Scope**

- Public REST API endpoints
- Read-only HTTP methods GET **(Read-only)**
- Response body analysis
- Header and metadata inspection
- **Testing Style: Passive / Non-intrusive**
- Authentication and authorization behavior

❌ **Excluded from Scope**

- Exploitation or vulnerability attacks
- Authentication bypass attempts
- Rate-limit flooding or DoS testing
- Access to private or production systems

This controlled scope ensures ethical, legal, and professional security assessment practices.

3. Methodology

This API Security Risk Analysis was conducted using a read-only, non-intrusive approach focused on observation and documentation rather than exploitation. The assessment was performed with standard API testing tools to review endpoint behavior, response data, and security-relevant headers.

Approach

1. **Endpoint Selection**
   Selected publicly accessible REST endpoints for review:
   /public/v2/users, /public/v2/posts, /public/v2/comments, /public/v2/todos.
2. **Read-Only Request Execution**
   Sent controlled **GET** requests using Postman to ensure no data modification occurred.
3. **Response Body Analysis**
   Reviewed JSON responses to identify potential **data exposure** issues (e.g., presence of user identifiers and user-related fields).
4. **Authentication Verification**
   Checked whether endpoints required authentication (API key, token, OAuth). Documented access behavior when **No Auth** was used.
5. **Authorization Behavior Review**
   Observed whether access appeared restricted by role, ownership, or identity (based on response visibility and accessible records).
6. **Header and Metadata Inspection**
   Inspected HTTP response headers for security signals such as pagination headers (x-pagination-*), caching, and any visible rate-limiting indicators (e.g., X-RateLimit-*).
7. **Risk Identification and Severity Rating**
   Mapped observations to common API security risk categories and assigned severity levels based on potential impact and likelihood in a SaaS production context.
8. **Evidence Collection and Documentation**
   Captured screenshots of key evidence (request, status code, authorization settings, response body, and headers) and documented findings in a structured report format.

Ethical Statement

All testing was limited to publicly available endpoints and passive inspection. No exploitation, bypass attempts, stress testing, or denial-of-service activity was performed.

4. API Endpoints Reviewed

The following API endpoints were included in the assessment:

Endpoint	Reason (Security Point of View)
/public/v2/users	User data exposure, authentication check
/public/v2/posts	Content exposure, ownership validation
/public/v2/comments	Data over-exposure & rate-limit analysis
/public/v2/todos	Authorization & user-mapping analysis

All endpoints returned JSON-formatted responses and were accessible **without authentication**.

5. Authentication Analysis 🔐

No authentication mechanisms such as API keys, access tokens, or OAuth were required to access the tested endpoints. This allows any user or automated script to retrieve data without identity verification.

Check Point	Observation
API Key Required	❌ No
Token Required	❌ No
OAuth Used	❌ No
Access Control	Open

📌 **Security Impact:**
In real SaaS environments, missing authentication controls can lead to unauthorized access, data scraping, and misuse of backend resources.

- **Risk Level: Medium**

6. Authorization Analysis

Check Point	Observation
User Identity Validation	❌ Not enforced
Role-Based Access	❌ Not present
Ownership Check	❌ Missing

📌 **Security Impact:**
Lack of authorization controls may result in horizontal access issues, privacy violations, and regulatory non-compliance in production systems.
- **Risk:** Users can access data belonging to other users.

- **Risk Level: Medium**

7. Data Exposure Analysis 📂

The API responses exposed multiple data fields, including identifiers and user-related information. Excessive exposure of data increases the risk of user enumeration, phishing, spam campaigns, and information misuse.

Data Field	Exposed	Risk
User ID	✓ Yes	Enumeration
Name	✓ Yes	Privacy
Email	✓ Yes	Phishing / Spam

📌 **Security Impact:**

APIs should follow the principle of data minimization by exposing only necessary fields required for functionality.

Risk: Excessive data exposure increases privacy and abuse risks.

Risk Level: **Medium**

7.1 ⚠️ Risk Impact Assessment

Risk Area	Impact Description
Privacy Risk	Exposure of personal details may violate privacy principles
Abuse Risk	Email data can be harvested for malicious campaigns
Enumeration	Predictable IDs enable large-scale data collection
Compliance	May conflict with data protection regulations in production

7.2 Severity Rating

Category	Level
Overall Risk Severity	Medium
Likelihood	Medium
Business Impact	Medium

📌 **Key Observation**

The API exposes more user-related information than required for public consumption. In production SaaS environments, such exposure increases the attack surface and creates unnecessary privacy and compliance risks.

8. Security Headers & Cloud Indicators

The API responses included certain security-related headers and cloud infrastructure indicators, suggesting the use of CDN and basic protective measures.

📌 **Positive Observations:**

- Use of cloud-based infrastructure
- Basic response security headers present

📌 **Areas for Improvement:**

- Lack of visible API abuse prevention headers
- No explicit rate-limit enforcement indicators

Header / Indicator	Status
CDN / Cloud Protection	✅ Present
Content Security Headers	⚠️ Partial
API Abuse Protection	❌ Not visible

❖ **Observation:** Basic cloud protection exists, but API-level security controls are limited.

9. Risk Summary

Risk Area	Observation (from your Postman evidence)	Severity
Authentication	Endpoints accessible without login/token (e.g., /public/v2/users)	Medium
Authorization	No role/ownership control visible for returned records	Medium
Data Exposure	User fields like name, email, status exposed in response	Medium
Rate Limiting	No visible X-RateLimit-* headers; pagination headers present (x-pagination-*)	Medium

9.1 Recommendations ✅

- ✓ Enforce authentication using secure tokens (JWT) or OAuth 2.0 for non-public resources
- ✓ Apply authorization checks (RBAC + ownership validation per resource)
- ✓ Minimize exposed fields (hide/mask PII such as email on public endpoints)
- ✓ Implement rate limiting + throttling; return rate-limit headers
- ✓ Enable logging/monitoring to detect scraping and abnormal pagination requests

9.2 Conclusion

This read-only API security review identified common SaaS API risks: open access, missing authorization enforcement, excessive data exposure, and limited abuse controls. Applying strong authentication, authorization, data minimization, and rate limiting can significantly reduce privacy and misuse risks in production environments.

9.3 Skills Demonstrated

- ✓ API Security Analysis
- ✓ Authentication & Authorization Assessment
- ✓ Risk Identification
- ✓ Security Documentation
- ✓ SaaS Security Fundamentals