API Security Risk Analysis Report

Cybersecurity Internship – Task 3

Name: Aryan Sanjay Salunke
API Tested: JSONPlaceholder Test API
Endpoint Tested: https://jsonplaceholder.typicode.com/users
Tools Used: Postman Web
Testing Method: Read-only ethical API testing
Date: February 11, 2026

---

1. Introduction

Modern applications rely heavily on APIs for communication between services such as mobile apps, dashboards, and SaaS platforms. If APIs are not properly secured, attackers may access sensitive data, bypass authentication, or overload systems.

This report analyzes a public test API to identify potential security risks, evaluate access control, and recommend remediation measures to improve API security.

2. API Overview

The API tested was JSONPlaceholder, a public REST API designed for testing and learning purposes. It provides simulated data such as users, posts, and comments.

The following endpoint was tested using a GET request:

GET /users

This endpoint returns user information in JSON format.

3. Testing Methodology

The API was tested using Postman Web with read-only requests. The following aspects were analyzed:

- Authentication requirements
- Response data structure
- Response headers
- Public accessibility of endpoints

No exploitation, attack simulation, or harmful activity was performed. Testing remained within ethical and allowed scope.

4. Identified Security Risks

Risk 1 — No Authentication Required

Severity: High
Observation: The API allows unrestricted access without login credentials or authentication tokens.
Impact: Anyone can retrieve data without verification, leading to unauthorized data access.

Risk 2 — Excessive Data Exposure

Severity: Medium
Observation: User information is fully visible in the response.
Impact: If used in production, sensitive user data could be exposed publicly, leading to privacy risks.

Risk 3 — No Rate Limiting Observed

Severity: Medium
Observation: No request limit or throttling behavior was observed during testing.
Impact: Attackers could send large numbers of requests, potentially causing performance issues or service disruption.

5. Business Impact

If these vulnerabilities existed in a real SaaS or production system, they could result in:

- Unauthorized access to sensitive data
- Privacy violations and compliance risks
- API abuse or service disruption
- Loss of user trust and reputational damage

6. Remediation Recommendations

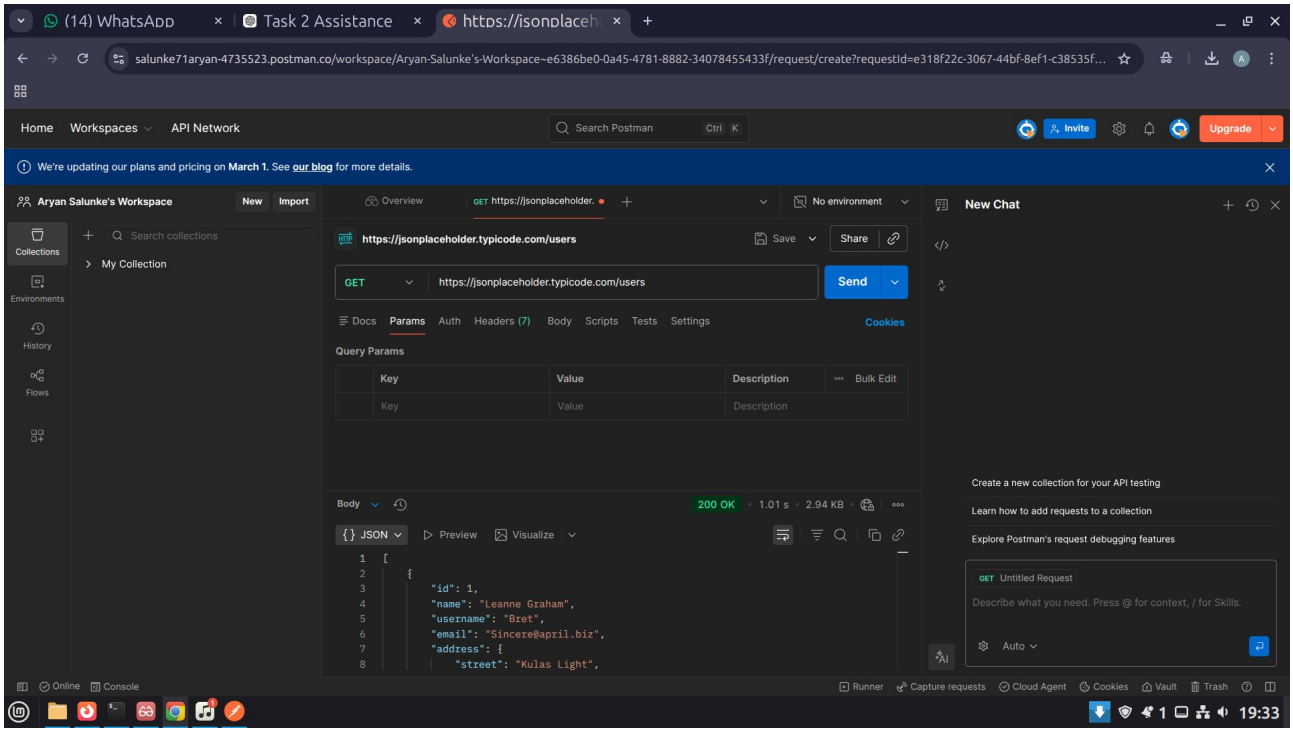
To improve API security, the following controls should be implemented:

- Require authentication using API keys, tokens, or OAuth
- Enforce authorization to restrict data access
- Implement rate limiting to prevent request abuse
- Minimize exposure of sensitive user data in responses

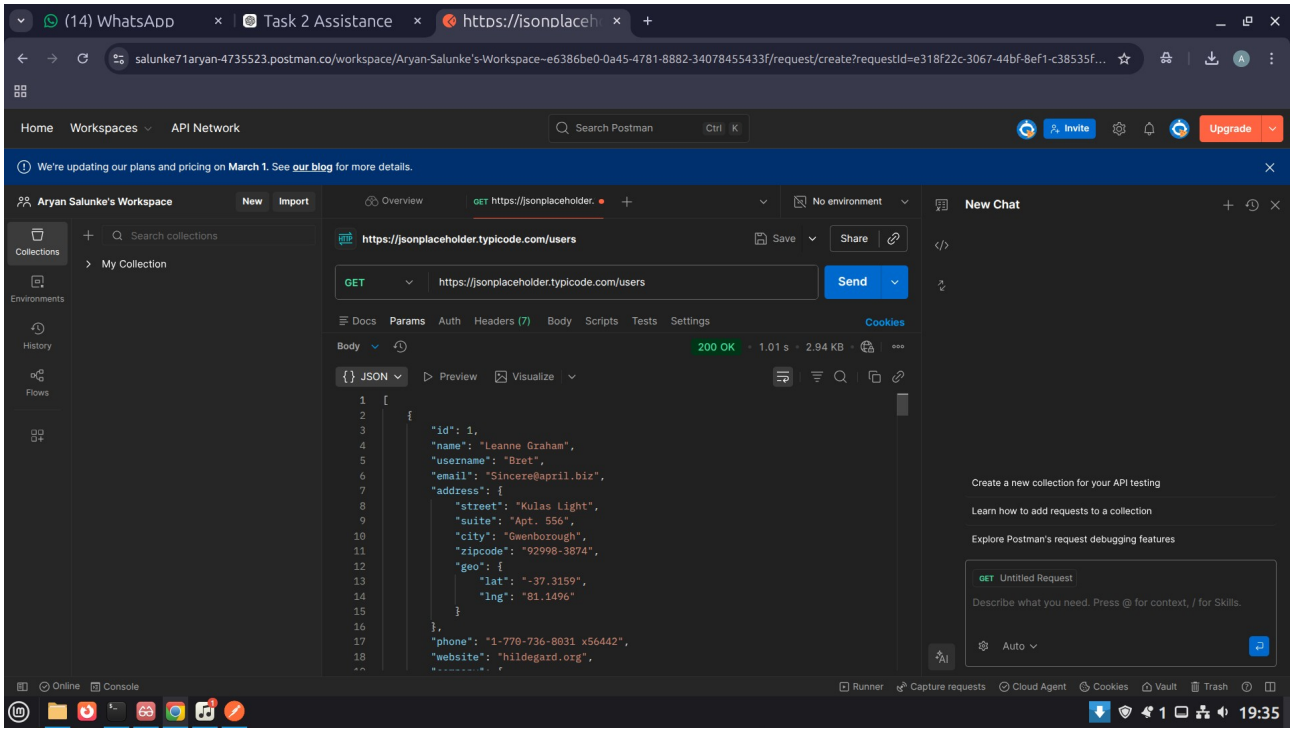
7. Testing Evidence (Screenshots)

The following screenshots provide visual proof of API testing conducted in Postman, including request execution, response data, and headers inspection.

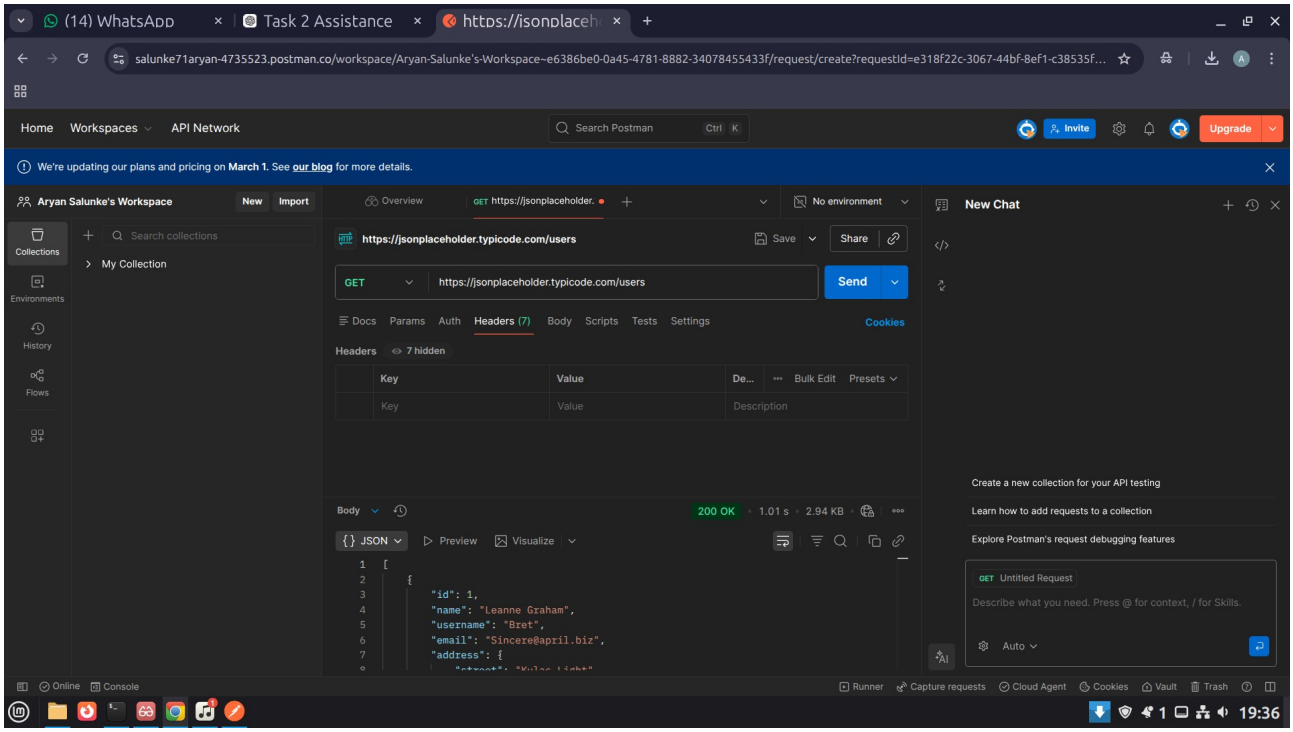
Screenshot 1 — API Request and Response



Screenshot 2 — Response Body



Screenshot 3 — Response Headers



---

8. Conclusion

The API demonstrates common security weaknesses found in publicly accessible services. Proper authentication, access control, and response protection are necessary to ensure secure API deployment in production environments.