API

SECURITY

RISK ANALYSIS REPORT

2024-2025

PREPARED BY :

GIRIDHARAN S

INTRODUCTION

- API Tested: JSONPlaceholder (jsonplaceholder.typicode.com in Bing)
- Tool Used: Postman
- Objective: Identify common API security risks in a safe demo environment.
- Scope: Focused on endpoints /posts, /users, /comments.

KEY RISKS IDENTIFIED

Open or Unauthenticated Endpoints

- Observation: Endpoints like GET /posts and GET /users are accessible without authentication.
- Impact: Anyone can retrieve data without restrictions.
- Remediation: Require authentication (API keys, OAuth).

EXCESSIVE DATA EXPOSURE

- Observation: GET /users exposes sensitive fields such as emails, addresses, phone numbers.
- Impact: Risk of privacy violations and data leakage.
- Remediation: Limit response fields, mask sensitive data

WEAK OR MISSING AUTHENTICATION TOKENS

- Observation: No authentication required for any endpoint.
- Impact: APIs are vulnerable to abuse.
- Remediation: Implement token-based authentication.

AUTHORIZATION ISSUES

- Observation: POST /posts allows setting arbitrary userId.
- Impact: Attackers can impersonate other users.
- Remediation: Enforce authorization checks.

MISSING RATE LIMITING

- Observation: Unlimited requests can be sent without restriction.
- Impact: Risk of denial-of-service or resource abuse.
- Remediation: Implement rate limiting (e.g., 100 requests/minute).

INPUT VALIDATION ISSUES

- Observation: POST /posts accepts malformed JSON and fake data.
- Impact: Risk of data pollution and injection attacks.
- Remediation: Enforce strict schema validation.

BUSINESS IMPACT

- Unauthorized access to user data.
- Privacy risks due to exposed personal information.
- Potential abuse of API resources (spam, fake data injection).

CONCLUSION

- The analysis of JSONPlaceholder demonstrates common API risks:
- Open endpoints
- Data exposure
- Missing authentication/authorization
- Lack of rate limiting
- Weak input validation
- While JSONPlaceholder is intentionally insecure for demo purposes, these findings highlight why secure design, authentication, authorization, and validation are critical in real-world APIs.