CYBER SECURITY TASK 03

API SECURITY RISK ANALYSIS REPORT

Public Demo API – JSONPlaceholder

Assessment Type:	**Read-Only API Security Risk Analysis**
Prepared by:	**Razi Ammari**
Environment:	**Kali Linux**
Prepared for:	**Future Interns**

January 2026

Table of Contents

1. Executive Summary

Modern software applications increasingly rely on APIs to deliver core functionality, enable integrations, and expose business data. While APIs provide flexibility and scalability, insecure API implementations introduce significant business risks including data leakage, privacy violations, service abuse, and regulatory exposure.

This report presents a professional API Security Risk Analysis conducted on a public demo API to simulate how real-world SaaS platforms are assessed by security consultants. The objective of this engagement was risk identification and impact analysis, not exploitation.

The assessment identified multiple high-impact security risks commonly observed in production environments, including unauthenticated access, excessive data exposure, missing authorization controls, and lack of rate limiting.

2. Scope & Methodology

2.1 Scope

- API Tested: JSONPlaceholder (Public Demo API)
- Endpoint Assessed: GET /users
- Base URL: https://jsonplaceholder.typicode.com
- Authentication Testing: Observational only
- Request Type: Read-only (GET)

2.2 Ethical Boundaries

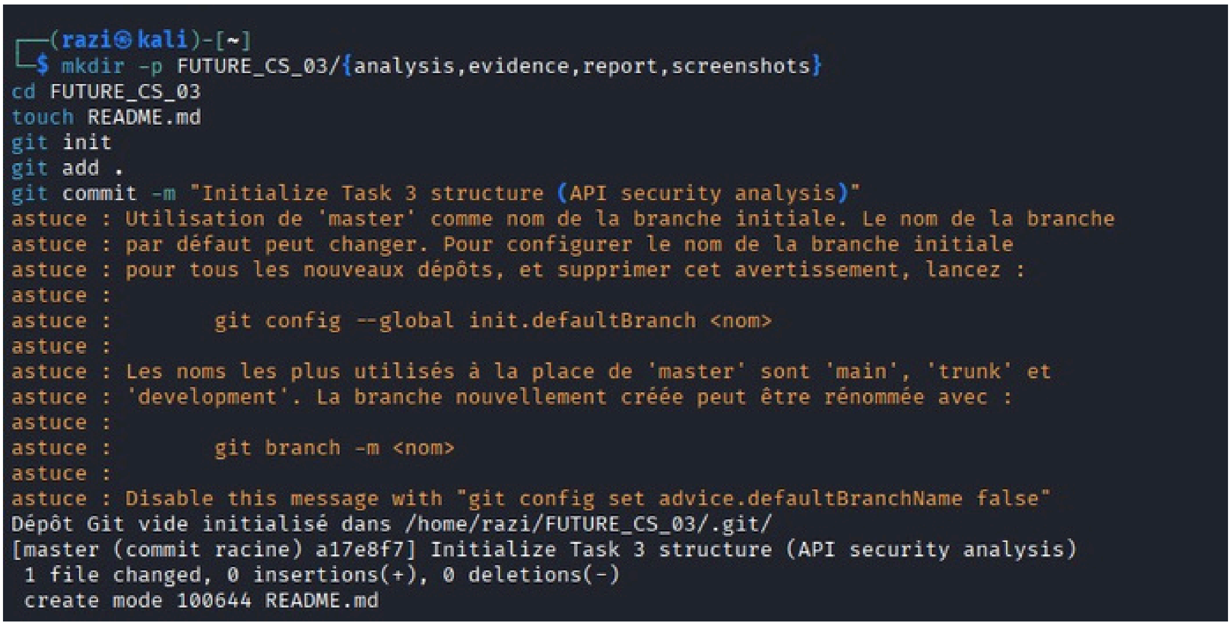
✓ Public API only
✓ No exploitation attempts
✓ No data modification
✓ No stress or denial-of-service testing
This assessment complies with ethical security research standards.

3. Tools Used

Tool	Purpose
Postman	API request execution and response inspection
Browser Developer Tools	Header and response analysis
Markdown Documentation	Risk reporting
GitHub	Transparent evidence storage

4. API Discovery & Documentation Review

The JSONPlaceholder API provides publicly documented endpoints intended for testing and prototyping. A documentation review revealed multiple openly accessible endpoints without authentication requirements.



```
┌──(razi㉿kali)-[~]
└─$ mkdir -p FUTURE_CS_03/{analysis,evidence,report,screenshots}
cd FUTURE_CS_03
touch README.md
git init
git add .
git commit -m "Initialize Task 3 structure (API security analysis)"
astuce : Utilisation de 'master' comme nom de la branche initiale. Le nom de la branche
astuce : par défaut peut changer. Pour configurer le nom de la branche initiale
astuce : pour tous les nouveaux dépôts, et supprimer cet avertissement, lancez :
astuce :
astuce :        git config --global init.defaultBranch <nom>
astuce :
astuce : Les noms les plus utilisés à la place de 'master' sont 'main', 'trunk' et
astuce : 'development'. La branche nouvellement créée peut être rénommée avec :
astuce :
astuce :        git branch -m <nom>
astuce :
astuce : Disable this message with "git config set advice.defaultBranchName false"
Dépôt Git vide initialisé dans /home/razi/FUTURE_CS_03/.git/
[master (commit racine) a17e8f7] Initialize Task 3 structure (API security analysis)
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
```

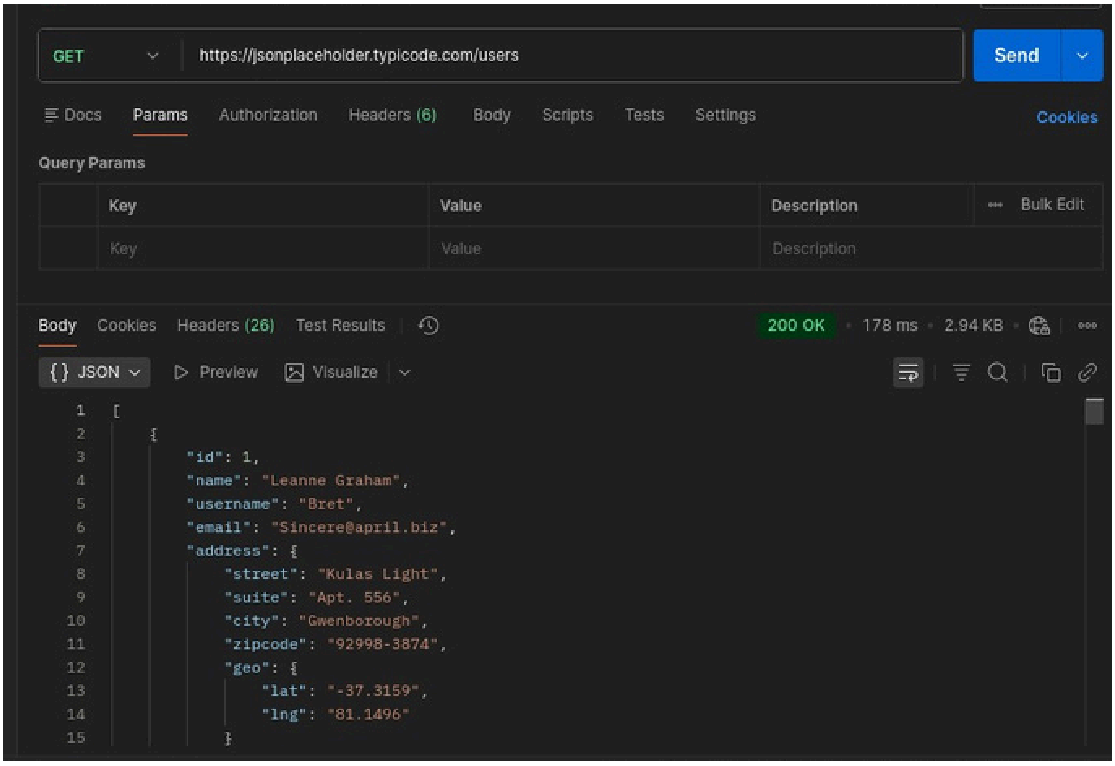
Official API documentation and available endpoints

5. Endpoint Testing & Observation

The following endpoint was tested using Postman:

GET https://jsonplaceholder.typicode.com/users

The endpoint returned a 200 OK response with a full list of user records.



```
[
    {
        "id": 1,
        "name": "Leanne Graham",
        "username": "Bret",
        "email": "Sincere@april.biz",
        "address": {
            "street": "Kulas Light",
            "suite": "Apt. 556",
            "city": "Gwenborough",
            "zipcode": "92998-3874",
            "geo": {
                "lat": "-37.3159",
                "lng": "81.1496"
            }
```

Postman request and full JSON response

6. Identified Security Risks

6.1 Risk 1 – Open / Unauthenticated Endpoint

Severity: 🔴 High

Observation:

The /users endpoint is publicly accessible without any authentication mechanism such as API keys, OAuth tokens, or session validation.

Security Risk:

Any external party can retrieve user-related data without verification.

Business Impact:

- Unauthorized data access
- Large-scale data scraping
- Increased exposure to privacy incidents

6.2 Risk 2 – Excessive Data Exposure

Severity: 🔴 High

Observation:

The API response exposes full user profiles, including:

- Full name
- Email address
- Phone number
- Physical address
- Geo-location coordinates
- Company information

Security Risk:

The API violates the principle of least privilege by returning more data than required.

Business Impact:

- User privacy violations
- Increased phishing and social engineering risk
- Potential regulatory non-compliance (GDPR, data protection laws)

6.3 Risk 3 – Missing Authorization Controls

Severity: 🟠 Medium-High

Observation:

There are no role-based or user-specific access controls applied to the endpoint.

Security Risk:

All users' data is accessible regardless of requester identity or permissions.

Business Impact:

- Horizontal data exposure
- Unauthorized access to entire datasets

6.4 Risk 4 – Missing Rate Limiting

Severity: 🟡 Medium

Observation:

The API does not indicate any rate limiting or request throttling mechanisms.

Security Risk:

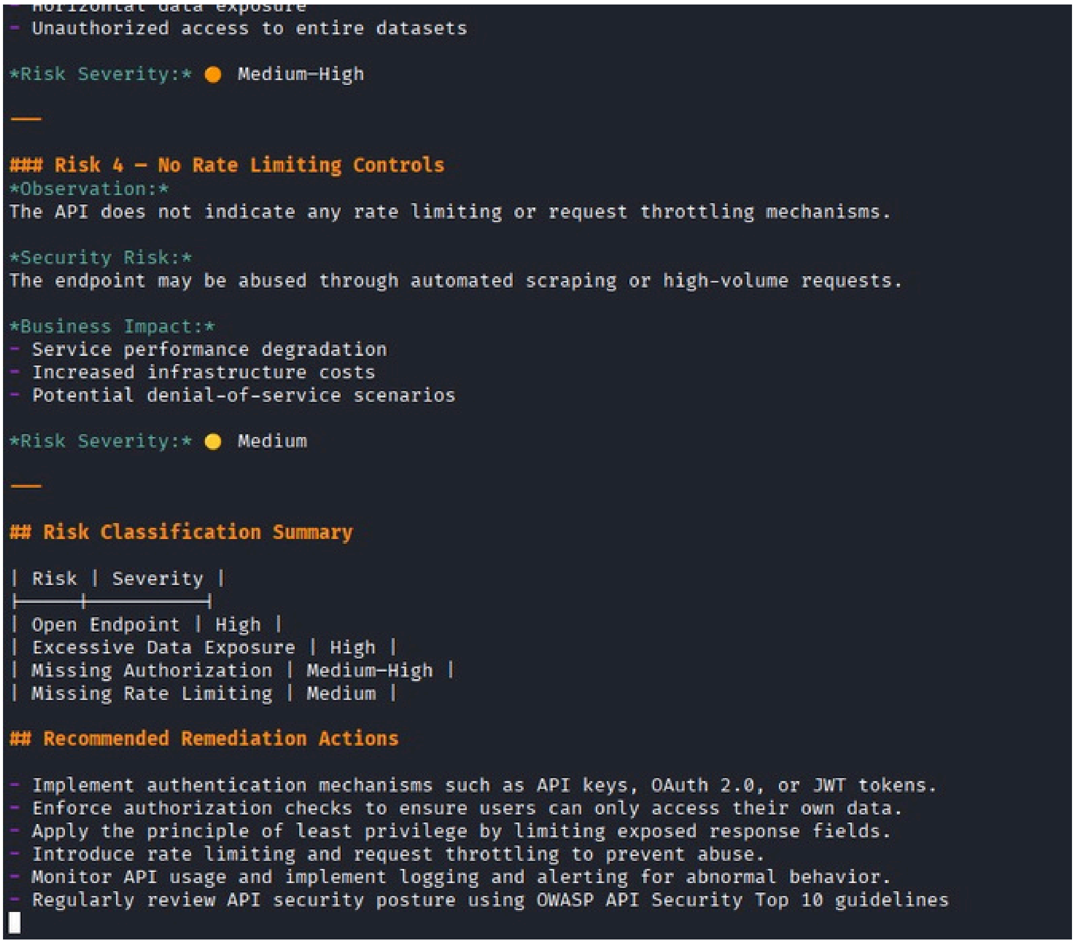
The endpoint may be abused through automated scraping or high-volume requests.

Business Impact:

- Service performance degradation
- Increased infrastructure costs
- Potential denial-of-service scenarios

7. Risk Classification Summary

Risk Category	Severity
Open / Unauthenticated Endpoint	**High**
Excessive Data Exposure	**Excessive Data Exposure**
Missing Authorization Controls	**Medium-High**
Missing Rate Limiting	**Medium**



```
- Horizontal data exposure
- Unauthorized access to entire datasets

*Risk Severity:* 🟠 Medium–High

—

### Risk 4 – No Rate Limiting Controls
*Observation:*
The API does not indicate any rate limiting or request throttling mechanisms.

*Security Risk:*
The endpoint may be abused through automated scraping or high-volume requests.

*Business Impact:*
- Service performance degradation
- Increased infrastructure costs
- Potential denial-of-service scenarios

*Risk Severity:* 🟡 Medium

—

## Risk Classification Summary

| Risk | Severity |
├──────┼──────────┤
| Open Endpoint | High |
| Excessive Data Exposure | High |
| Missing Authorization | Medium–High |
| Missing Rate Limiting | Medium |

## Recommended Remediation Actions

- Implement authentication mechanisms such as API keys, OAuth 2.0, or JWT tokens.
- Enforce authorization checks to ensure users can only access their own data.
- Apply the principle of least privilege by limiting exposed response fields.
- Introduce rate limiting and request throttling to prevent abuse.
- Monitor API usage and implement logging and alerting for abnormal behavior.
- Regularly review API security posture using OWASP API Security Top 10 guidelines
```

risk analysis in nano editor

8. Recommended Remediation Actions

To reduce the identified risks, the following actions are recommended:

- Implement authentication mechanisms (API keys, OAuth 2.0, JWT)
- Enforce authorization checks to ensure users access only permitted data
- Apply least-privilege principles to API responses
- Introduce rate limiting and request throttling
- Enable API logging and monitoring
- Align API security posture with OWASP API Security Top 10

9. Assessment Conclusion

This assessment demonstrates how commonly overlooked API design choices can introduce serious security and business risks, even in non-production environments. The findings align closely with vulnerabilities frequently reported in real SaaS security incidents.

The methodology used in this project mirrors industry-standard API security assessments, focusing on risk identification, business impact analysis, and remediation guidance.