



API Security Risk Analysis Report

Security risk report

Prepared by:
RASANI LAKSHMI CHARANYA

Future Interns Project

Cybersecurity Task 3

OBJECTIVE

The objective of this task is to perform a read-only API Security Risk Analysis using public or demo APIs. The analysis focuses on identifying common security risks such as unauthenticated access, excessive data exposure, and missing security controls without exploiting or attacking the system. This task helps understand how API security audits are performed in real-world security consulting environments.

API SELECTED

API Name:JSONPlaceholder

Base URL:

<https://jsonplaceholder.typicode.com>

Why this API?

Public and safe for testing

Designed for learning purposes

No real sensitive data

Suitable for beginner security analysis

TOOLS & TESTING ENVIRONMENT

The following tools were used during analysis:

Postman

- Sending API requests
- Viewing responses
- Header inspection

Kali Linux Terminal

- Header analysis using curl command

Command Used:

- curl -I
<https://jsonplaceholder.typicode.com/posts>

TESTING PROCESS

- Reviewed API documentation in browser.
- Selected safe public API endpoints.
- Sent GET requests using Postman.
- Observed response data structure.
- Checked authentication requirements.
- Inspected headers and status codes.
- Identified potential security risks.
- Classified risk severity and documented observations.

ENDPOINTS TESTED

./posts Endpoint

Method: GET

Returns post objects with userId, title, body.

Accessible without authentication.

./users Endpoint

Method: GET

Returns user details such as name, email, phone, address, and company info.

./comments Endpoint

Method: GET

Returns comment records linked to posts.

SECURITY FINDINGS

⚠ Finding 1 — Unauthenticated Endpoint

All tested endpoints are publicly accessible without authentication.

Risk Level: Medium

Impact: Unauthorized users can access API data.

⚠ Finding 2 — Excessive Data Exposure

The users endpoint returns full user objects including contact and address details.

Risk Level: Medium

Impact: In real APIs this could lead to information leakage.

⚠ Finding 3 — Missing Rate Limiting Indicators

No clear rate-limiting information observed in responses.

Risk Level: Medium

Impact: API may be vulnerable to abuse through excessive requests.

⚠ Finding 4 — Public Demo Data

Data appears to be test/demo information.

Risk Level: Low

Impact: No real sensitive data exposure.

RISK SEVERITY TABLE

RISK	SEVERITY	IMPACT
Unauthenticated endpoint access	Medium	Unauthorized access
Excessive data exposure	Medium	Information leakage
Missing rate limiting	Medium	API abuse possible
Demo/Public data	Low	Minimal impact

REMEDIATION RECOMMENDATIONS

To improve API security, the following practices are recommended:

- Implement authentication tokens (API keys / OAuth).
- Return only required fields in responses.
- Enable rate limiting to prevent abuse.
- Apply input validation for request data.
- Use security headers consistently.

SCREENSHOTS

```
{
  "posts": [
    {
      "userId": 1,
      "id": 1,
      "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
      "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
    },
    {
      "userId": 1,
      "id": 2,
      "title": "qui est esse",
      "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi nulla"
    },
    {
      "userId": 1,
      "id": 3,
      "title": "ea molestias quasi exercitationem repellat qui ipsa sit aut",
      "body": "et iusto sed quo iure\nvoluptatem occaecati omnis eligendi aut ad\nvoluptatem doloribus vel accusantium quis pariatur\nmolestiae porro eius odio et labore et velit aut"
    },
    {
      "userId": 1,
      "id": 4,
      "title": "eum et est occaecati",
      "body": "ullam et saepe reiciendis voluptatem adipisci\nsit amet autem assumenda provident rerum culpa\nquis hic commodi nesciunt rem tenetur doloremque ipsam iure\nquis sunt voluptatem rerum illo velit"
    },
    {
      "userId": 1,
      "id": 5,
      "title": "nesciunt quas odio",
      "body": "repudiandae veniam quaerat sunt sed\nnihil aut fugiat sit autem sed est\nvoluptatem omnis possimus esse voluptatibus quis\nest aut tenetur dolor neque"
    },
    {
      "userId": 1,
      "id": 6,
      "title": "dolorem eum magni eos aperiam quia",
      "body": "ut aspernatur corporis harum nihil quis provident sequi\nmollitia nobis aliquid molestiae\nperspiciatis et ea nemo ab reprehenderit accusantium quas\nvoluptate dolores velit et doloremque molestiae"
    }
  ]
}
```

Getting started GET https://jsonplaceholder. GET https://jsonplaceholder. + No environment

https://jsonplaceholder.typicode.com/posts Unsaved changes Save Share

GET https://jsonplaceholder.typicode.com/posts Send

Docs Params Authorization Headers (6) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (25) Test Results 200 OK · 386 ms · 8.03 KB

{ } JSON Preview Visualize

```
1 {
2   {
3     "userId": 1,
4     "id": 1,
5     "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
6     "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
7   },
8   {
9     "userId": 1,
10    "id": 2,
11    "title": "qui est esse",
12    "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi nulla"
13  },
14  {
15    "userId": 1,
16    "id": 3,
17    "title": "ea molestias quasi exercitationem repellat qui ipsa sit aut",
18    "body": "et susto sed quo iure\nvoluptatem occaecati omnis eligendi aut ad\nvoluptatem doloribus vel accusantium quis pariatur\nmolestiae porro eius odio et labore et velit aut"
19  },
20  {
21    "userId": 1,
22    "id": 4,
23    "title": "eum et est occaecati",
24    "body": "ullam et saepe reiciendis voluptatem adipisci\nsit amet autem assumenda provident rerum culpa\nquis hic commodi nesciunt rem tenetur doloremque ipsam iure\nquis sunt voluptatem rerum illo velit"
25  },
26  {
27    "userId": 1,
28    "id": 5,
29    "title": "nesciunt quas odio",
```

```

(root@kali)~[/home/kali]
# curl -i https://jsonplaceholder.typicode.com/posts
HTTP/2 200
date: Tue, 17 Feb 2026 13:02:00 GMT
content-type: application/json; charset=utf-8
access-control-allow-credentials: true
cache-control: max-age=43200
etag: W/"6b80-Ybsq/K6GwwqrYkAsFxqDXGC7DoM"
expires: -1
nel: {"report_to": "heroku-nel", "response_headers": ["Via"], "max_age": 3600, "success_fraction": 0.01, "failure_fraction": 0.1}
pragma: no-cache
report-to: {"group": "heroku-nel", "endpoints": [{"url": "https://nel.heroku.com/reports?s=9PWz1wYqSXF1%2B0kpvCXHzdvru%2FCLEF%2Fz1HL197eq80g%3D%5Cu0026sid=e11707d5-02a7-43ef-b45e-2cf4d2036f7d%5Cu0026ts=1771161155"}], "max_age": 3600}
reporting-endpoints: heroku-nel="https://nel.heroku.com/reports?s=9PWz1wYqSXF1%2B0kpvCXHzdvru%2FCLEF%2Fz1HL197eq80g%3D%5Cu0026sid=e11707d5-02a7-43ef-b45e-2cf4d2036f7d%5Cu0026ts=1771161155"
server: cloudflare
vary: Origin, Accept-Encoding
via: 2.0 heroku-router
x-content-type-options: nosniff
x-powered-by: Express
x-ratelimit-limit: 1000
x-ratelimit-remaining: 999
x-ratelimit-reset: 1771161197
age: 28132
cf-cache-status: HIT
cf-ray: 9cf57a065a3dfd1f-SIN
alt-svc: h3=":443"; ma=86400

```

The screenshot shows the Postman interface with a GET request to `https://jsonplaceholder.typicode.com/users`. The response is a 200 OK status with a JSON body. The JSON body contains two user objects. The first user is Glenn Reichert, and the second user is Clementina DuBuque.

```

{
  "id": 9,
  "name": "Glenn Reichert",
  "username": "Delphine",
  "email": "Chaim_McDermott@dana.io",
  "address": {
    "street": "Dayna Park",
    "suite": "Suite 449",
    "city": "Bartholomewbury",
    "zipcode": "76495-3109",
    "geo": {
      "lat": "24.6463",
      "lng": "-168.8889"
    }
  },
  "phone": "(775)976-6794 x41206",
  "website": "conrad.com",
  "company": {
    "name": "Vost and Sons",
    "catchPhrase": "Switchable contextually-based project",
    "bs": "aggregate real-time technologies"
  }
},
{
  "id": 10,
  "name": "Clementina DuBuque",
  "username": "Moriah Stanton",
  "email": "May_Padberg@kailua.biz",
  "address": {
    "street": "Kattie Turnpike",
    "suite": "Suite 198",

```

← → Home Workspaces API Network

Search Postman Ctrl K

Getting started GET https://json GET https://json GET https://json POST Check for Val GET https://api.jk GET Check for Val GET Test Respon Integration Te POST Test Respor + No environment

Contract Testing / Test Response Unsaved changes Save Share

GET https://jsonplaceholder.typicode.com/comments Send

Overview Params Authorization Headers (8) Body Scripts Settings Cookies

Key	Value	Description
Key	Value	Description

Body Cookies Headers (25) Test Results (1/5) 200 OK · 470 ms · 40.81 KB Save Response

{ } JSON Preview Visualize

```
198 {
199   "postId": 6,
200   "id": 29,
201   "name": "eum distinctio amet dolor",
202   "email": "Jennings_Pouros@erica.biz",
203   "body": "tempora voluptatem est\nmagnam distinctio autem est dolorem\net ipsa molestiae odit rerum itaque corporis nihil nam\nneaque rerum error"
204 },
205 {
206   "postId": 6,
207   "id": 30,
208   "name": "Quasi nulla ducimus facilis non voluptas aut",
209   "email": "Lurline@marvin.biz",
210   "body": "consequuntur quia voluptate assumenda et\nautem voluptatem reiciendis ipsum animi est provident\nnearum aperiam sapiente ad vitae iste\naccusantium aperiam eius qui
211 },
212 {
213   "postId": 7,
214   "id": 31,
215   "name": "ex velit ut cum eius odio ad placeat",
216   "email": "Buford@shaylee.biz",
217   "body": "Quia incidunt ut\naliquid est ut rerum deleniti iure est\nnisi quia ea sint et\nvoluptatem quaerat eaque repudiandae eveniet aut"
218 },
219 {
220   "postId": 7,
221   "id": 32,
222   "name": "dolorem architecto ut pariatur quae qui suscipit",
223   "email": "Marie@laurel.name",
224   "body": "nihil ea itaque libero illo\nofficiis quo quo dicta inventore consequatur voluptas voluptatem\nncorporis sed necessitatibus velit tempore\nrerum velit et temporibus"
225 },
226 {
227   "postId": 7,
228   "id": 33,
229   "name": "voluptatum totam vel voluptate omnis",
```

CONCLUSION

This assessment demonstrated a basic API security review using safe and ethical testing methods. The API endpoints were openly accessible and highlighted common risks such as unauthenticated access and excessive data exposure. Although the API contains demo data, the analysis reflects real-world API security considerations and provides practical experience in documenting security risks professionally.