

# API SECURITY RISK ANALYSIS Report

Prepared by: Olaniyan Sodiq Abiodun

Date: 17-February-2026

For: Future-interns / Internal Training

## Executive Summary

We assessed the /users and /posts endpoints of the public API. Our audit identified three high-severity risks, including a complete lack of authentication and broken access control, which could allow an attacker to access all data. We recommend implementing strict authorization checks and rate limiting.

## Scope

- Target: JSONPlaceholder API ([typicode.com](https://typicode.com)).
- Endpoints Tested: /posts, /users, /comments.
- Methodology: Read-only security analysis based on OWASP API Top 10.
- Tools Used: Postman, Browser DevTools, Google Docs.

## Findings

Risk Category	HTTP METHOD and Endpoint Tested	Observation	Business Impact	Severity	Remediation
Broken Authentication	GET /users/1	API accessible without any API key or token.	Unauthorized access to all user data, leading to data breaches and loss of customer trust.	High	Implement OAuth 2.0 or API Key authentication. Require valid tokens for every request.
Excessive Data Exposure	GET /users/1	Response includes lat and lng (geo-coordinates) when unnecessary.	Expose user location data unnecessarily.	Medium	Ensure the API only returns fields explicitly.

		only name/email is needed.	y, increasing the impact of a data breach.		needed by the client. Use schema-based responses.
<b>Broken Object Level Authorization (BOLA)</b>	GET /posts/2	Changing the ID in the URL from 1 to 2 allows access to another user's post.	Users can view/edit other users' private data. Could lead to major privacy scandals	High	Implement server-side access control checks. Verify the authenticated user owns the resource ID before returning data.
<b>Missing Rate Limiting</b>	GET /posts	Sent 30 rapid requests; all were successful.	System vulnerable to Denial of Service (DoS) and brute-force attacks, causing potential downtime.	Medium	Implement rate limiting (e.g., 100 requests per minute per IP). Return 429 Too Many Requests status code.
<b>Input Validation</b>	POST /posts	API accepted a script tag (<script>alert('XSS')</script>) in the request body.	If this data were stored and displayed, it could lead to Cross-Site Scripting (XSS) attacks against other users.	High	Implement strict input validation and output encoding. Sanitize all user inputs on the server side.

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Olanlyan Sodiq's Workspace' containing 'Collections' (My Collection), 'Environments', 'History', 'Flows', and 'Files'. The main area has tabs for 'Get data' and 'Post data'. A 'Get data' tab is active, showing a request to 'http://jsonplaceholder.typicode.com/users/1'. The response status is '200 OK' with a response time of '34 ms'. The response body is a JSON object representing a user:

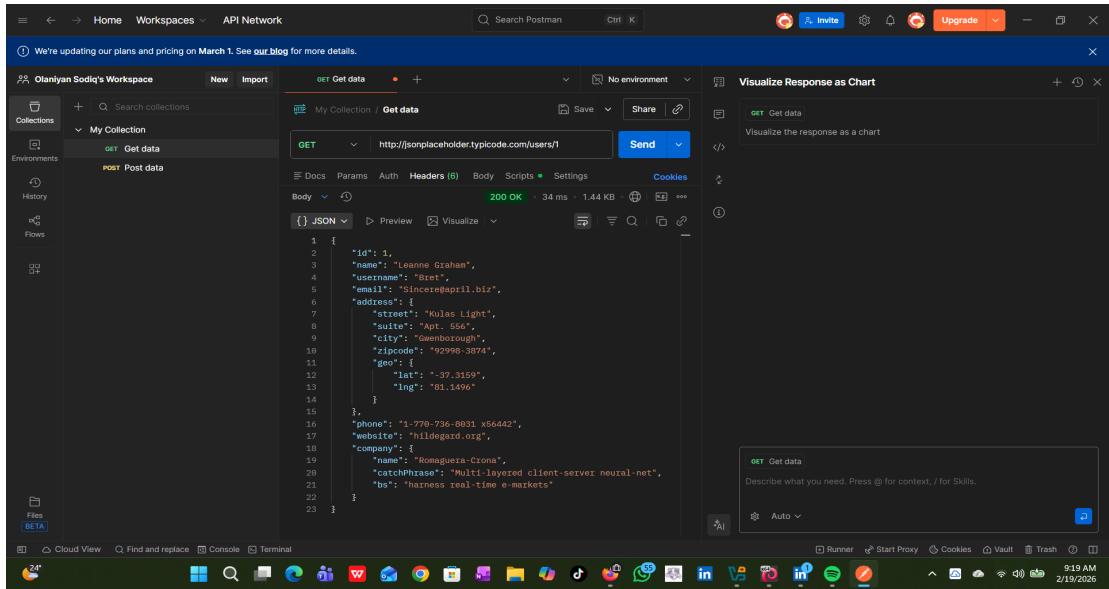
```
1 {  
2   "id": 1,  
3   "name": "Leanne Graham",  
4   "username": "Bret",  
5   "email": "Sincere@april.biz",  
6   "address": {  
7     "street": "Kulas Light",  
8     "suite": "Apt. 556",  
9     "city": "Gwenborough",  
10    "zipcode": "92998-3874",  
11    "geo": {  
12      "lat": "-37.3159",  
13      "lng": "81.1496"  
14    }  
15  },  
16  "phone": "1-770-736-8831 x56442",  
17  "website": "hildegard.org",  
18  "company": {  
19    "name": "Romaguera-Crona",  
20    "catchPhrase": "Multi-layered client-server neural-net",  
21    "bs": "harness real-time e-markets"  
22  }  
23 }
```

Auth-test.png

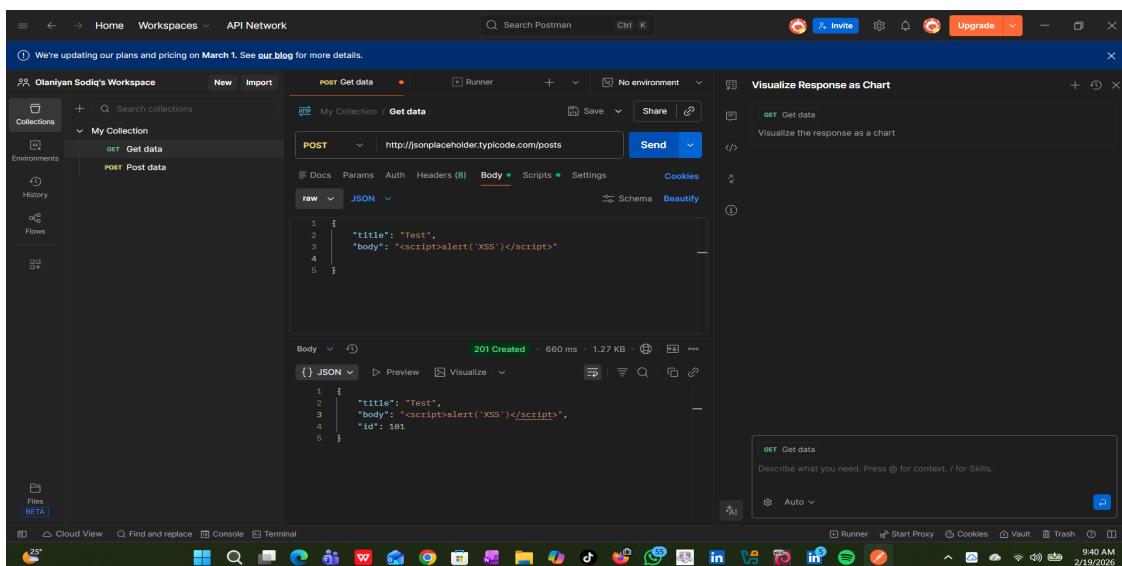
The screenshot shows the Postman application interface. The sidebar is identical to the first one. The main area has tabs for 'Get data' and 'Post data'. A 'Get data' tab is active, showing a request to 'http://jsonplaceholder.typicode.com/posts/2'. The response status is '200 OK' with a response time of '1.05 s'. The response body is a JSON object representing a post:

```
user_id: 1  
id: 2  
title: qui est esse  
body: est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis qui aperiam non debitis possimus qui neque nisi nulla
```

BOLA-test.png



Data-Exposure.png



Input-Validation.png

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Olanayan Sodiq's Workspace' containing sections for Collections, Environments, History, and Flows. The main area displays 'My Collection - Run results' with an 'ERROR' status. It shows a single test named 'Get data' with a 'GET' request to 'http://jsonplaceholder.typicode.com/posts/1'. The test passed with a status code of 200, response time of 46 ms, and size of 1.341 KB. Below this, there are sections for 'Iteration 1', 'Iteration 2', and 'Iteration 3', each showing the same successful 'Get data' request. A 'Visualize Response as Chart' tab is open on the right, showing a chart for the 'Get data' test. The bottom of the screen shows a taskbar with various icons and the date/time '2/19/2026 9:54 AM'.

Rate-Limit.png