

API SECURITY RISK ASSESSMENT REPORT

Assessment Type: Manual Read-Only API Security Evaluation

Prepared By: Harish C

Role: Cybersecurity Intern

Organization: Future Interns

1. Executive Summary

A manual read-only API security assessment was conducted on the Demo API to evaluate authentication enforcement, access control mechanisms, HTTP method restrictions, and response behavior.

The goal of this assessment was to identify possible security risks while staying within ethical testing limits and without attempting any exploitation.

Based on the testing performed, The API correctly blocks unauthorized access and properly validates authentication tokens. No critical vulnerabilities were identified within the defined scope.

The overall security risk is assessed as **Low**, with recommendations focused on strengthening rate limiting and monitoring controls for production environments.

2. Scope of Assessment

2.1 Included in Scope

- Public demo endpoints
- GET requests
- Safe method testing (POST, PUT, DELETE behavior validation)
- Authorization header inspection
- Response and status code analysis
- Documentation-based review

2.2 Excluded from Scope

- Exploitation attempts
- Authentication bypass attacks
- Denial-of-Service (DoS) testing
- Private or production API access
- Automated scanning or brute-force attempts

All testing was performed only on the public demo environment and within approved ethical limits.

3. API Information

API Name: ReqRes Demo API

Base URL: <https://reqres.in>

Assessment Type: Manual Security Evaluation

Testing Method: Postman-based request inspection

4. Methodology

The assessment was performed in the following steps:

Phase 1 – Documentation Review

Reviewed official API documentation to understand endpoint structure, authentication requirements, and expected responses.

Phase 2 – Endpoint Enumeration

Tested publicly available endpoints to determine accessibility and authentication enforcement.

Phase 3 – Access Control Validation

Verified whether user-related endpoints properly enforced authorization controls.

Phase 4 – HTTP Method Testing

Tested POST, PUT, and DELETE methods to validate method-level access restrictions.

Phase 5 – Authentication & Header Analysis

Analyzed API behavior when invalid or malformed Authorization headers were supplied.

Phase 6 – Risk Classification

Each identified issue was rated based on how serious it could be and how it might impact a real business environment.

5. Risk Findings

Risk 1 – Public Base Endpoint Exposure

Endpoint Tested:

GET /api

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' (apis), 'Environments', 'History', and 'Flows'. The main area has tabs for 'Home', 'Workspaces', and 'API Network'. A search bar at the top right says 'Search Postman'. Below it, there are three tabs: 'GET Untitled Request', 'GET Untitled Request', and 'GET https://reqres.in/api'. The third tab is selected. The URL 'https://reqres.in/api' is entered in the address bar. The request type is 'GET'. Under the 'Headers' tab, there are seven entries. In the 'Body' tab, there's a JSON payload: { "name": "ReqRes API", "version": "1.0.0", "description": "Fake data CRUD API with custom endpoints for Pro users", "documentation": "/api-docs", "endpoints": { "free": ["/api/users", "/custom-endpoints", "/health"], "pro": ["/api/users", "/custom-endpoints", "/health"] }, "features": ["Fake data generation", "Custom endpoints (Pro)", "Clerk authentication", "Stripe subscriptions"] }. The response status is '200 OK' with a green checkmark, and a message: 'Request successful. The server has responded as required.' The response body is identical to the JSON in the body tab. At the bottom, there are buttons for 'Runner', 'Capture requests', 'Cloud Agent', 'Cookies', 'Vault', 'Trash', and 'AI'.

Description:

In a real SaaS environment, exposed metadata may provide attackers with information useful for mapping internal API structures.

Evidence:

Endpoint responded successfully without authentication.

Severity: Low

Likelihood: Low

Business Impact:

Public endpoint exposure may assist attackers in reconnaissance and understanding API structure.

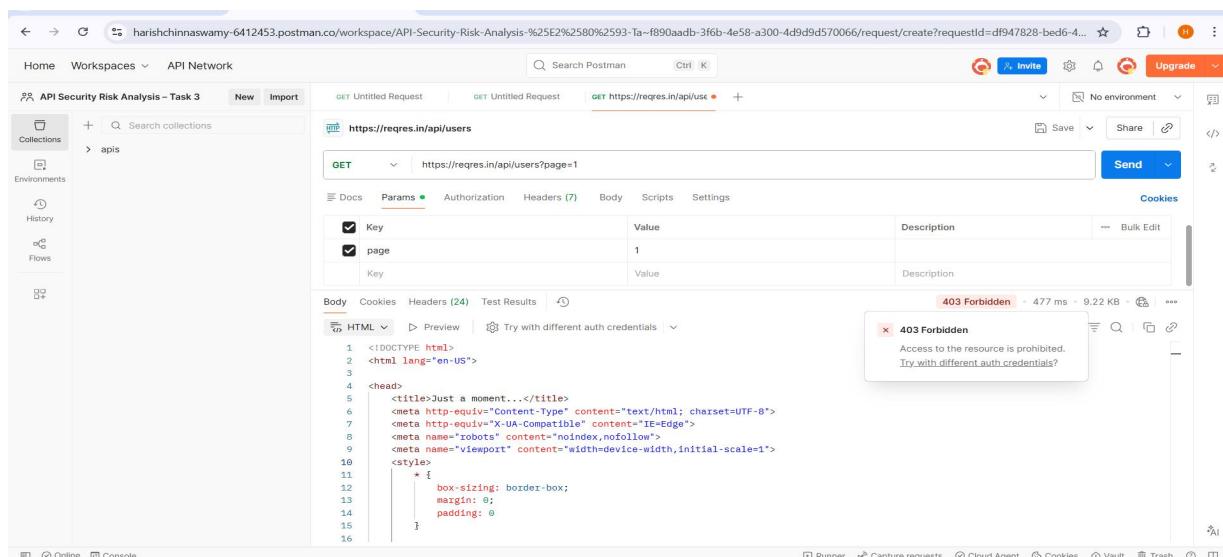
Remediation Recommendation:

Limit unnecessary metadata exposure in production environments and restrict non-essential endpoints.

Risk 2 – Endpoint Access Control Validation

Endpoints Tested:

GET /api/users?page=1



The screenshot shows a Postman interface with the following details:

- Request Method:** GET
- Request URL:** https://reqres.in/api/users?page=1
- Params Tab:** Key: page, Value: 1
- Body Tab:** HTML content:

```
<!DOCTYPE html>
<html lang="en-US">
<head>
<title>Just a moment...</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=Edge">
<meta name="robots" content="noindex, nofollow">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
    * {
        box-sizing: border-box;
        margin: 0;
        padding: 0
    }

```
- Test Results Tab:** Status: 403 Forbidden - 477 ms - 9.22 KB. Message: "Access to the resource is prohibited. Try with different auth credentials?"

GET /api/users/2

The screenshot shows the Postman interface with a collection named "API Security Risk Analysis - Task 3". A request is being made to the endpoint `https://reqres.in/api/users/2`. The response status is 403 Forbidden, with a response time of 487 ms and a size of 8.77 KB. The response body is a long HTML string indicating access is prohibited. The "Body" tab is selected, showing the raw HTML code of the 403 response.

Description:

Endpoints returned 403 Forbidden, indicating access control enforcement.

Evidence:

Unauthorized requests were denied.

Severity: Low

Likelihood: Low

Business Impact:

Proper access control prevents unauthorized user data enumeration.

Remediation Recommendation:

Continue enforcing strict authentication and authorization mechanisms across all endpoints.

Risk 3 – HTTP Method Restriction Testing

Methods Tested: POST /api/users

The screenshot shows the Postman interface with a failed POST request to `https://reqres.in/api/users`. The response status is **403 Forbidden**, with the message: "Access to the resource is prohibited. Try with different auth credentials?".

PUT /api/users/2

The screenshot shows the Postman interface with a failed PUT request to `https://reqres.in/api/users/2`. The response status is **403 Forbidden**, with the message: "Access to the resource is prohibited. Try with different auth credentials?".

DELETE /api/users/2

The screenshot shows the Postman interface with a failed DELETE request to `https://reqres.in/api/users/2`. The request was made with the following parameters:

- Method: `DELETE`
- URL: `https://reqres.in/api/users/2`
- Headers (7): (Listed below)
- Body: (Empty)
- Cookies: (Empty)

The Headers section shows the following:

Key	Value	Description
Key	Value	Description

The response details indicate a `403 Forbidden` status with a duration of `625 ms` and a size of `9.2 KB`.

403 Forbidden
Access to the resource is prohibited.
Try with different auth credentials?

Description:

Modification attempts were denied with 403 responses.

Evidence:

Unauthorized data manipulation attempts were blocked.

Severity: Low

Likelihood: Low

Business Impact:

Prevents unauthorized data modification and privilege escalation.

Remediation Recommendation:

Maintain method-level authorization and implement role-based access control (RBAC).

Risk 4 – Authorization Header Validation

Headers Tested:

Authorization: Bearer invalidtoken

The screenshot shows a Postman interface with a task titled "API Security Risk Analysis - Task 3". A GET request is made to "https://reqres.in/api/users/2". The "Authorization" tab is selected, showing "Bearer Token" as the auth type and a placeholder for the token. The response status is "401 Unauthorized", with the message "The request is unauthenticated. Pass the correct auth credentials." The response body is a JSON object with an "error" field indicating a missing API key.

```
1 {
2     "error": "missing_api_key",
3     "message": "The x-api-key header is required for this endpoint.",
4     "hint": "You sent a Bearer token but /api/* endpoints require x-api-key. Bearer tokens are for /app/* endpoints (app user access to scoped data).",
5     "next_steps": [
6         "For admin operations: add x-api-key header instead of Bearer",
7         "For app user data access: use /app/collections/:slug/records with your Bearer token",
8         "Get your API key at app.reqres.in/api-keys"
9     ],
10    "docs_url": "https://app.reqres.in/docs#authentication",
11    "example_curl": "curl -H \"x-api-key: YOUR_API_KEY\" https://api.reqres.in/api/collections",
12    "details": "Authorization: Bearer was provided but this endpoint expects x-api-key header",
13    "_meta": {
14        "powered_by": "ReqRes",
15        "docs_url": "https://app.reqres.in/documentation"
16    }
17}
```

Description:

Invalid tokens were rejected without verbose error disclosure.

Evidence:

API responded with access denial and did not leak authentication details.

Severity: Low

Likelihood: Low

Business Impact:

Reduces risk of authentication bypass and token forgery.

Remediation Recommendation:

Ensure production APIs use properly signed JWT tokens with expiration, signature validation, and token revocation mechanisms.

Risk 5 – Rate Limiting & Abuse Prevention (Conditional)

Description:

Rate limiting mechanisms were not observable during manual testing.

Severity: Medium (Conditional for production environments)

Likelihood: Medium

Business Impact:

If rate limiting is not implemented in production environments, it may increase the risk of automated scraping or abuse.

Remediation Recommendation:

Implement request throttling, IP-based rate limits, and anomaly detection monitoring.

6. Risk Level Summary

Risk Area	Level
Endpoint Exposure	Low
Access Control	Low
Method Restriction	Low
Authentication Handling	Low
Rate Limiting (Production)	Medium

7. OWASP API Security Alignment

This assessment considered common risk categories aligned with industry standards, including:

- Broken Object Level Authorization
- Broken Authentication
- Excessive Data Exposure
- Security Misconfiguration
- Lack of Rate Limiting

Observed testing did not indicate violations of the above OWASP API risk categories within the permitted scope.

8. Monitoring & Detection Recommendations

For production deployment, the following controls are recommended:

- Centralized API logging
- Monitoring failed authentication attempts
- Alerting on abnormal traffic patterns
- Detection of excessive requests from single IP addresses
- Implementation of automated abuse detection mechanisms

9. Overall Business Impact

Based on the manual security evaluation performed, the API demonstrates structured access control enforcement and appropriate authentication validation.

No unauthorized data access or modification was observed within the permitted scope.

For production-grade deployment, implementation of advanced rate limiting, logging, and monitoring controls is recommended to mitigate automated attack risks.

10. Final Conclusion

From a defensive security perspective, the assessed API reflects a structured and controlled access model with effective authentication validation.

Based on the read-only testing performed, no high-risk issues were found.

The overall security posture is assessed as **Low Risk**, with conditional improvements recommended for production traffic control and monitoring mechanisms.