# Cyber Security Internship Report

## Task 13: Secure API Testing & Authorization Validation

**Name:** Mohammed Nihal

**Environment:** Kali Linux (VirtualBox), Postman, cURL

**Objective:** To test REST API endpoints for authentication, authorization, input validation, and rate limiting vulnerabilities and map findings to OWASP API Security Top 10 risks.

## Tools Used

- Postman (Primary API testing tool)
- cURL (Command-line testing tool)
- JSONPlaceholder API (Testing endpoint)

## Testing Methodology

1. Sent GET requests to retrieve user information.
2. Sent POST requests to test data creation and input handling.
3. Modified user IDs to test authorization controls (IDOR testing).
4. Sent DELETE requests to test function-level authorization.
5. Tested malicious inputs for input validation.
6. Tested API without authentication headers.
7. Performed rapid automated requests to evaluate rate limiting.

## Test Results Summary

| Test | Endpoint | Method | Result | Status Code |
|---|---|---|---|---|
| Retrieve user data | /users/1 | GET | Successful access without authentication | 200 OK |
| Create user | /users | POST | Resource created successfully | 201 Created |
| Authorization test | /users/{id} | GET | Access to multiple user records allowed | 200 OK |
| Delete test | /users/1 | DELETE | Deletion request accepted | 200 OK |
| Input validation | /users | POST | Malicious input accepted | 201 Created |
| Rate limiting | /users/1 | GET | No rate limiting detected | 200 OK |

## OWASP API Vulnerability Mapping

| Observation | OWASP Category | Risk Level |
|---|---|---|
| Access without authentication | API2: Broken Authentication | Medium |
| Access to multiple user IDs | API1: Broken Object Level Authorization | High |
| DELETE allowed | API5: Broken Function Level Authorization | High |
| Malicious input accepted | API10: Unsafe Consumption of APIs | Medium |

| | | |
|---|---|---|
| No rate limiting | API4: Unrestricted Resource Consumption | Medium |

## Conclusion

The API testing exercise demonstrated practical techniques for identifying authentication, authorization, input validation, and rate limiting issues. Several security weaknesses were identified, including lack of authentication enforcement, unrestricted access to user resources, and absence of rate limiting controls. These findings highlight the importance of implementing proper authentication, authorization checks, and security configurations to protect API endpoints.