## Introduction

In today's digital world, Application Programming Interfaces (APIs) are widely used to connect web applications, mobile apps, and backend services. APIs allow different software components to communicate with each other using standard HTTP methods such as GET, POST, PUT, and DELETE. Since APIs often handle sensitive data and critical operations, securing them is extremely important.

Many security breaches occur due to weak API security controls such as improper authentication, broken authorization, poor input validation, and lack of rate limiting. Secure API testing helps identify these weaknesses before attackers can exploit them. This task focuses on testing REST APIs using Postman to identify common security misconfigurations and authorization flaws, following the OWASP API Security Top 10 guidelines.

---

## Objective of the Task

The objective of this task is to understand and perform API security testing to identify vulnerabilities related to authentication, authorization, input handling, and rate limiting. The task aims to verify whether APIs correctly restrict access to authorized users and prevent misuse. By performing this task, practical knowledge of secure API implementation and common security risks is gained.

---

## Tools Used

The primary tool used for this task is **Postman**, which is a widely used API testing tool that allows users to send HTTP requests and analyze responses easily.
Other tools such as **cURL** and **Insomnia** can also be used for API testing, but Postman is preferred due to its simple interface and ease of use.

---

## Overview of REST APIs

REST APIs work using standard HTTP methods:

- **GET** is used to retrieve data from the server.
- **POST** is used to send data or create new resources.
- **PUT** is used to update existing resources.
- **DELETE** is used to remove resources.

Each API request includes an endpoint URL, headers, and sometimes a request body. Headers usually contain authentication details such as tokens and content type information.

---

**API Security Testing Process**

**1. API Configuration**

The API endpoint is configured in Postman by selecting the appropriate HTTP method and entering the endpoint URL. Required headers such as Authorization and Content-Type are added based on the API documentation. Proper configuration ensures that the API behaves as expected during testing.

---

**2. Authentication Testing**

Authentication testing is performed to verify whether the API properly checks user identity. Requests are sent using valid credentials to confirm successful access. Additional requests are sent using invalid or missing authentication tokens to verify that unauthorized access is denied. If the API allows access without authentication, it indicates a serious security vulnerability.

---

**3. Authorization Testing**

Authorization testing ensures that authenticated users can access only the resources they are permitted to use. This is tested by modifying resource identifiers such as user IDs in API requests. If a user is able to access or modify another user's data, it indicates a broken authorization issue.

---

**4. Input Validation Testing**

Input validation testing checks whether the API properly handles unexpected or invalid input. Various malformed inputs, incorrect data types, and abnormal values are sent in API requests. A secure API should reject such inputs and return appropriate error messages. Poor input validation can lead to injection attacks or system instability.

---

**5. Rate Limiting Testing**

Rate limiting testing is conducted to verify whether the API restricts excessive requests within a short time period. Multiple rapid requests are sent using Postman to simulate brute-force or denial-of-service attacks. If the API does not enforce rate limits, it becomes vulnerable to abuse and performance issues.

---

### 6. Response and Error Analysis

During testing, HTTP response codes such as 200, 401, 403, and 500 are carefully reviewed. Error messages are analyzed to ensure that sensitive information such as system details or stack traces is not exposed. Detailed error messages can provide attackers with useful information about the system.

---

## Identified Vulnerabilities and OWASP Mapping

| Vulnerability Identified | OWASP API Risk |
|---|---|
| Broken Authorization | API1: Broken Object Level Authorization |
| Missing Rate Limiting | API4: Lack of Resources & Rate Limiting |
| Improper Input Validation | API8: Injection |
| Information Disclosure | API7: Security Misconfiguration |

---

## Interview Questions – Important Points

**What is API Authentication?**
API authentication is the process of verifying the identity of a user or application before granting access to API resources.

**What is Broken Authorization?**
Broken authorization occurs when a user can access data or perform actions beyond their permitted privileges.

**Why is Rate Limiting Important?**
Rate limiting helps prevent brute-force attacks, API abuse, and denial-of-service attacks.

**Difference Between GET and POST:**
GET is used to retrieve data, while POST is used to send data securely in the request body.

**Common API Security Risks:**
Broken authentication, broken authorization, excessive data exposure, lack of rate limiting, and improper input validation.

### Recommendations

To improve API security, strong authentication mechanisms such as token-based authentication should be implemented. Role-based access control should be enforced to restrict unauthorized access. All user inputs must be validated on the server side. Rate limiting should be enabled to prevent misuse. Error messages should be kept generic to avoid information leakage.

### Final Outcome

This task helped in gaining practical experience in API security testing using Postman. It improved understanding of authentication, authorization, and common API vulnerabilities. The task also provided knowledge on mapping security issues to OWASP API risks and understanding best practices for secure API design.

**Prepared by:**

**Megaraj S**