Task 13: Secure API Testing & Authorization Validation

---

1. Introduction

APIs (Application Programming Interfaces) are widely used for communication between client applications and backend servers. Since APIs handle sensitive data, they are a common target for attacks. Secure API testing helps identify authentication issues, authorization flaws, and misconfigurations that may lead to data exposure or unauthorized access.

---

2. Objectives

The objectives of this task are:

- To understand how REST APIs work
- To test API authentication mechanisms
- To validate authorization controls
- To identify broken authorization issues
- To check input validation and rate limiting
- To analyze HTTP response codes and errors
- To map findings to OWASP API security risks

---

3. Tools Used

- **Postman (Primary tool)**
- **cURL (Alternative)**
- **Insomnia (Alternative)**

---

4. Understanding REST APIs

REST APIs use HTTP methods such as GET, POST, PUT, and DELETE to perform operations on server resources. These methods are used along with endpoints, headers, and request bodies to exchange data between the client and server.

---

5. API Configuration in Postman

The API endpoint URL, headers, and request body were configured in Postman based on API documentation. Proper headers such as authorization tokens and content type were included while sending requests.

---

6. Authentication Testing

API authentication was tested by sending requests with:

- Valid credentials
- Invalid credentials
- Missing authentication headers

The responses were observed to check whether unauthorized access was properly blocked.

---

7. Authorization Validation

Authorization testing was performed by modifying resource identifiers in requests. This helped identify whether users could access resources that they were not authorized to view or modify, indicating broken authorization issues.

---

8. Input Validation Testing

Malformed and unexpected input values were sent in API requests to observe how the API handled incorrect data. Proper input validation helps prevent injection attacks and application crashes.

---

9. Rate Limiting Checks

Multiple rapid requests were sent to the API to check whether rate limiting was enforced. Rate limiting is important to prevent brute-force attacks and abuse of API services.

---

10. HTTP Response Analysis

HTTP response codes and error messages were reviewed carefully. Improper error handling can reveal sensitive information about the backend system and should be avoided.

---

11. Mapping to OWASP API Risks

The identified issues were mapped to OWASP API Security risks such as:

- Broken Object Level Authorization
- Broken Authentication
- Security Misconfiguration
- Improper Input Validation

---

12. Conclusion

This task helped in understanding secure API testing techniques and the importance of proper authentication and authorization. It also provided practical knowledge of identifying common API security vulnerabilities using testing tools.

RIFAH RESLIN MANIYODAN LATHEEF

MSc CYBERSECURITY