

Task 13

SECURE API TESTING & AUTHORIZATION VALIDATION

JASHMI KS

Secure API testing is the process of verifying that an Application Programming Interface (API) is protected against unauthorized access, data leakage, and misuse. It ensures that only legitimate users and applications can interact with backend services in a controlled and secure manner.

Secure API testing focuses on validating authentication, authorization, input validation, rate limiting, and error handling mechanisms implemented in the API.

Objectives of Secure API Testing

- Prevent unauthorized access to sensitive data
- Ensure proper identity verification
- Protect backend systems from attacks
- Detect security misconfigurations
- Validate compliance with security standards

REST API Requests in Postman:

GET:

The screenshot shows the Postman application interface. The top navigation bar includes Home, Workspaces, API Network, Search Postman, Invite, Upgrade, and various icons. The main workspace shows a GET request to <https://api.github.com>. The 'Params' tab is selected under the request details. Below the request, the 'Body' tab is active, displaying a JSON response. The response body is a large object containing numerous URLs for GitHub's API endpoints, such as 'current_user_url', 'current_user_authorizations_html_url', 'authorizations_url', etc. The status bar at the bottom indicates a successful 200 OK response with a duration of 69 ms and a size of 1.73 KB.

```

1 {
2   "current_user_url": "https://api.github.com/user",
3   "current_user_authorizations_html_url": "https://github.com/settings/connections/applications{/client_id}",
4   "authorizations_url": "https://api.github.com/authorizations",
5   "code_search_url": "https://api.github.com/search/code?q={query}{&page,per_page,sort,order}",
6   "commit_search_url": "https://api.github.com/search/commits?q={query}{&page,per_page,sort,order}",
7   "emails_url": "https://api.github.com/user/emails",
8   "emojis_url": "https://api.github.com/emojis",
9   "events_url": "https://api.github.com/events",
10  "feeds_url": "https://api.github.com/feeds",
11  "followers_url": "https://api.github.com/user/followers",
12  "following_url": "https://api.github.com/user/following{/target}",
13  "gists_url": "https://api.github.com/gists{/gist_id}",
14  "hub_url": "https://api.github.com/hub"
  
```

POST:

The screenshot shows the Postman interface with a failed POST request to <https://api.github.com>. The status bar indicates a 404 Not Found error. The response body contains JSON with a message about the 'Accept' header.

```
1 {  
2   "message": "Not Found",  
3   "documentation_url": "https://docs.github.com/rest",  
4   "status": "404"  
5 }
```

Configuring an API Request in Postman:

The screenshot shows the Postman interface with a failed GET request to <https://api.github.com>. The status bar indicates a 415 Unsupported Media Type error. The response body contains JSON with a message about the 'Accept' header.

```
1 {  
2   "message": "Unsupported 'Accept' header: 'application/vnd.api+json'. Must accept 'application/json'.",  
3   "documentation_url": "https://docs.github.com/v3/media",  
4   "status": "415"  
5 }
```

Testing Authentication (Valid vs Invalid Token):

The screenshot shows the Postman application interface. A collection named "Collections" is selected on the left sidebar. In the main workspace, a request to "https://api.github.com/user" is being tested via a GET method. The "Headers" tab is active, displaying the following configuration:

Key	Value	Description
Accept	application/vnd.api+json	
Authorization	JWT token (REDACTED)	

The "Body" tab shows a JSON response with the following content:

```
1 {  
2   "message": "Requires authentication",  
3   "documentation_url": "https://docs.github.com/rest",  
4   "status": "401"  
5 }
```

The status bar at the bottom indicates a 401 Unauthorized error with a response time of 884 ms and a size of 1.02 KB.

Testing Unauthenticated Access:

This screenshot shows the same Postman interface as the previous one, but with a different configuration. The "Authorization" header is now populated with a valid GitHub API token: "ghp_JxpRxit3HKbbbFwSaKjrLpPOOU1UV0V5m6k". The "Headers" tab is still active, and the "Body" tab shows the same JSON response as before:

```
1 {  
2   "message": "Requires authentication",  
3   "documentation_url": "https://docs.github.com/rest",  
4   "status": "401"  
5 }
```

The status bar at the bottom indicates a 401 Unauthorized error with a response time of 1.73 s and a size of 1.16 KB.

Testing Authorization (IDOR Check):

The screenshot shows the Postman interface with a successful API call to `https://api.github.com/users/octocat`. The response status is `200 OK` with a response time of `2.14 s` and a size of `1.76 KB`. The response body is a JSON object containing detailed information about the user `octocat`, including login, id, node_id, avatar_url, gravatar_id, url, html_url, followers_url, following_url, gists_url, starred_url, subscriptions_url, and organizations_url.

```
1 {  
2   "login": "octocat",  
3   "id": 583231,  
4   "node_id": "MDQ6VXNlcjU4MzIzMQ==",  
5   "avatar_url": "https://avatars.githubusercontent.com/u/583231?v=4",  
6   "gravatar_id": "",  
7   "url": "https://api.github.com/users/octocat",  
8   "html_url": "https://github.com/octocat",  
9   "followers_url": "https://api.github.com/users/octocat/followers",  
10  "following_url": "https://api.github.com/users/octocat/following{other_user}",  
11  "gists_url": "https://api.github.com/users/octocat/gists{gist_id}",  
12  "starred_url": "https://api.github.com/users/octocat/starred{owner_repo}",  
13  "subscriptions_url": "https://api.github.com/users/octocat/subscriptions",  
14  "organizations_url": "https://api.github.com/users/octocat/orgs"  
}
```

Private:

The screenshot shows the Postman interface with an unauthorized response for the `https://api.github.com/user/emails` endpoint. The status is `401 Unauthorized` with a response time of `1.36 s` and a size of `1.15 KB`. The response body indicates that authentication is required, with a documentation URL provided.

```
1 {  
2   "message": "Requires authentication",  
3   "documentation_url": "https://docs.github.com/rest",  
4   "status": "401"  
5 }
```

Input Validation Testing:

The screenshot shows the Postman interface with a 404 Not Found response for the `https://api.github.com/users/<>?` endpoint. The status is `404 Not Found` with a response time of `465 ms` and a size of `1.19 KB`. The response body indicates that the user was not found, with a documentation URL provided.

```
1 {  
2   "message": "Not Found",  
3   "documentation_url": "https://docs.github.com/rest",  
4   "status": "404"  
5 }
```

Rate Limiting Test:

The screenshot shows the Postman application interface. A GET request is made to `https://api.github.com/user`. In the Headers tab, there is an `Accept` header set to `application/vnd.api+json`. The response status is `403 rate limit exceeded`, which is highlighted with a red box. The response body contains the message `application/json; charset=utf-8`. Other headers listed include `Content-Type`, `Referrer-Policy`, `X-GitHub-Media-Type`, `X-RateLimit-Limit` (60), `X-RateLimit-Remaining` (0), `X-RateLimit-Reset` (1770393280), `X-RateLimit-Resource` (core), and `X-RateLimit-Used` (60).

Review HTTP Response Codes

Common GitHub API Codes:

Code	Meaning
200	Success
401	Unauthorized
403	Forbidden / Rate limit
404	Not Found
422	Validation error