API Security and Project Analysis Report

Table of Contents

Introduction to API Security

API security is the practice of protecting Application Programming Interfaces (APIs) from attacks. As APIs are the backbone of modern applications, securing them is critical. Key concepts in API security include:

- **Authentication:** Verifying the identity of a user or application. Common methods include API keys, OAuth 2.0, and JWT.
- **Authorization:** Determining what an authenticated user is allowed to do.
- **Data Encryption:** Protecting data in transit (using TLS/HTTPS) and at rest.
- **Rate Limiting:** Preventing abuse by limiting the number of requests an entity can make.
- **Input Validation:** Preventing injection attacks by validating all incoming data.

Common vulnerabilities include broken authentication, broken object-level authorization (BOLA), excessive data exposure, and security misconfigurations.

API Endpoint Documentation

This documentation is based on the `docs/api_docs.md` file.

Authentication

This API uses HTTP Basic Authentication. * **Username:** `admin` * **Password:** `password123`

Endpoints

`GET /transactions` List all transactions.

`GET /transactions/{id}` Get a single transaction by ID.

`POST /transactions` Add a new transaction.

- **Request Body:** `json { "type": "PAYMENT", "amount": 5000, "sender": "+2507xxxxxxx", "receiver": "+2507yyyyyyy", "timestamp": "2025-09-01T10:00:00", "note": "..." }`

`PUT /transactions/{id}` Replace an existing transaction.

`DELETE /transactions/{id}` Delete a transaction by ID.

Error Codes

- `400 Bad Request`
- `401 Unauthorized`
- `404 Not Found`

DSA Comparison Results

The `dsa/dsa_compare.py` script was executed to compare the performance of linear search versus dictionary lookup.

- **Records:** 1691
- **Sample IDs:** 20
- **Linear search total time for 20 searches:** 0.000033 sec
- **Dict lookup total time for 20 searches:** 0.000010 sec
- **Dict lookup is faster by factor:** 3.39

These results demonstrate that dictionary lookup is significantly faster for this use case.

Reflection on Basic Auth Limitations

While simple to implement, Basic Authentication has significant limitations, especially for production environments:

- **Credentials Sent in Plain Text (without HTTPS):** Credentials are only Base64 encoded, not encrypted. Without HTTPS, they can be easily intercepted.

- **Vulnerability to Brute-Force Attacks:** Basic Auth is susceptible to brute-force attacks. Measures like rate limiting are necessary to mitigate this risk.
- **No Session Management:** Credentials must be sent with every request, which is inefficient and does not support session management or Single Sign-On (SSO).
- **Limited Granular Control:** It is difficult to implement fine-grained access control. It typically grants all-or-nothing access.
- **Poor User Experience:** Repeated credential prompts can be frustrating for users.

For these reasons, more secure authentication methods like **OAuth 2.0** or **JWT (JSON Web Tokens)** are recommended for production APIs.