REST API Security Report

Introduction to API Security

APIs are the backbone of modern applications, enabling communication between services, clients, and servers. While REST APIs provide flexibility and scalability, they are also a prime target for attacks such as data leaks, brute force attempts, and man-in-the-middle interceptions. API security involves implementing proper authentication, authorization, encryption, and monitoring to safeguard data integrity and confidentiality.

API Endpoint Documentation

Endpoint	Method	Description	Auth Required
/transactions	GET	Retrieve list of all transactions	Yes
/transactions/<id>	GET	Retrieve details of a specific transaction	Yes
/transactions	POST	Create a new transaction	Yes
/transactions/<id>	DELETE	Delete a transaction by ID	Yes

Results of DSA Comparison

We compared different Digital Signature Algorithm (DSA) key sizes for signing API payloads. Using Python's `cryptography` library, we tested signing and verification times. Results indicated that smaller key sizes (1024 bits) are faster but less secure, while larger key sizes (2048–3072 bits) provide stronger security at the cost of performance. For secure production use, 2048-bit keys are recommended as a balance between speed and safety.

Reflection on Basic Authentication Limitations

Basic Authentication is simple to implement but has significant limitations. Credentials are sent with every request (often only base64 encoded), making them vulnerable to interception. Without HTTPS, attackers can easily steal usernames and passwords. It also lacks token expiration, meaning compromised credentials remain valid until changed. For production APIs, stronger methods such as OAuth2, JWT, or API keys over HTTPS are preferred. Implementing role-based access control and secret storage (e.g., environment variables or vaults) is essential to strengthen API security.