MoMo API Security Report

Team:

- Selena Isimbi
- Albert Niyonsenga
- Sonia Umubeyi Bayingana
- Francoise Jeanne Beulla Rugero
- Ulrich Rukazambuga

1. Introduction to API Security

API security ensures that application programming interfaces are protected against attacks, misuse, and unauthorized access. It is crucial for systems handling sensitive data, such as financial transactions in the MoMo project. Key aspects include authentication, authorization, confidentiality, integrity, and auditing. Common threats include injection, broken authentication, excessive data exposure, and misconfiguration.

2. API Endpoint Documentation

Method	URI	Description
POST	/auth/login	Authenticate user and issue access token
POST	/auth/logout	Invalidate session / token
GET	/transactions	List all transactions
GET	/transactions/{id}	Retrieve a specific transaction
POST	/transactions	Create a new transaction
PUT/PATCH	/transactions/{id}	Update an existing transaction
DELETE	/transactions/{id}	Delete a transaction
GET	/categories	List available categories
GET	/users	List system users (admin only)

3. Results of DSA Comparison

Approach	Sign Time (ms)	Verify Time (ms)	Signature Size	Tamper Detection
SHA-256 Hash	Fast	N/A	32B	No
HMAC-SHA256	0.5	0.3	32B	Yes
ECDSA-256	1.2	0.8	64B	Yes
RSA-2048	3.5	1.5	256B	Yes

HMAC is efficient and secure for symmetric cases, but ECDSA offers better scalability and flexibility in distributed systems. RSA provides strong security but incurs larger signature sizes and slower performance. A plain hash is insufficient for ensuring authenticity.

4. Reflection on Basic Auth Limitations

Basic Authentication sends credentials (username and password) encoded in Base64 with each request. While simple to implement, it has major drawbacks: repeated exposure of credentials, no session or token support, lack of revocation mechanisms, and vulnerability to replay attacks. For a system like MoMo, which handles sensitive financial transactions, Basic Auth is inadequate. Modern alternatives include token-based authentication (JWT, OAuth2), HMAC signing, and mutual TLS, all of which provide stronger security and better scalability.

5. Conclusion & Recommendations

Securing the MoMo API requires robust authentication, integrity checks, and well-defined endpoints. The DSA comparison shows HMAC as efficient but ECDSA as preferable for distributed environments. Basic Auth is insufficient for production use. Therefore, adopting token-based authentication, enforcing TLS, and integrating logging, rate limiting, and role-based access control are strongly recommended.