Guidelines and Recommendations for Handling OAuth API Access Tokens

Introduction

This document provides guidelines and recommendations for handling OAuth API tokens for the Metadata Bank API. Security is paramount, and the following practices ensure that tokens are managed securely and efficiently. This document is in adherence to CISA/DOD and the IETF best practice documentation ^[PBS's Cybersecurity team].This guideline will mitigate OWASP's top 10 CI/CD security risks ^[PBS's Cybersecurity team]. While the majority of the content was authored by the Metadata Bank API Team, it incorporates valuable contributions and expert security guidance from our Cybersecurity Team.

General Guidelines

1. **Secure Storage of Consumer Key and Secret**

- **Environment Variables:** Store the consumer key and secret in environment variables or a secure configuration manager.
- **Automate the discovery of secrets:** Use a centralized management tool and a secret vault to store various credentials^[PBS's Cybersecurity team].
- **Avoid Hardcoding and embedded secrets:** Never hardcode these values within your codebase.
- **Use Encryption:** If storing in a file, make sure the file is encrypted and accessible only to authorized applications and users.

2. **Token Encryption**

- **Use Strong Encryption Algorithms:** Encrypt tokens using robust algorithms like AES.

- **Key Management:** Manage encryption keys securely, rotating them periodically.

3. Token Transmission

- **Use HTTPS:** Always transmit tokens over HTTPS to ensure they are encrypted during transit^[Token Best Practices].
- **Use TLS:** Use a minimum of TLS 1.2 to send keys, secrets, and tokens^[PBS's Cybersecurity team].

4. Token Validation

- **Validate Tokens:** Validate tokens on the server-side to ensure they are not tampered with.

Frequent Refreshing of OAuth Tokens

1. Short-Lived Access Tokens

- **Use Short Expiration Times:** Set short expiration times (e.g., one hour).
- **Implement Refresh Tokens:** Use refresh tokens for obtaining new access tokens.
- **Privilege Limitation:** Access tokens should have the minimum privileges required for the specific application or use case^[PBS's Cybersecurity team].
- **Resource & Action Restriction:** Access tokens should be limited to particular resource servers and actions on those resources^[PBS's Cybersecurity team].

2. Token Rotation

- **Rotate and Revoke:** Rotate tokens regularly and revoke old ones.

3. Automated Refresh Mechanism

- **Auto-Refresh Logic:** Implement logic to refresh tokens automatically.
- **Dynamic Token Refresh:** Implement dynamic token refresh^[PBS's Cybersecurity team].

4. Handling Refresh Errors

- **Graceful Error Handling:** Handle refresh errors gracefully.

5. Monitoring and Logging

- **Monitor and Log Securely:** Monitor token refresh activities and log securely.

6. Consider User Experience

- **Balance Security with Usability** Ensure a smooth user experience.

7. **Compliance with Provider Guidelines**

- **Follow Provider's Recommendations:** Adhere to specific OAuth provider guidelines.

Conclusion

By following these guidelines and recommendations, you can ensure that OAuth API access tokens are handled securely and efficiently. Frequent token refreshing, secure storage, encryption, and compliance with best practices are essential to maintaining the integrity and confidentiality of the tokens.

Prepared by: Patrick Roch

Public Broadcasting Service(PBS)