**Module 8 Activity 1: Modern Security Best Practices Technologies**

Saurabh Kale

IFT 458/554: Middleware Programming & Database Security

Dinesh Sthapit

November 26th, 2023

**Comparative Analysis of Application Security Technologies: Current System vs. OAuth 2.0 and mTLS**

**Current Application Security Technology:**

The current application security technology in use is characterized by a traditional authentication model, often relying on username/password combinations for user access. While this model may provide a basic level of security, it lacks the advanced features offered by more modern and robust security protocols like OAuth 2.0 and mTLS. Here, we delve into the technical aspects and potential technical debt associated with the existing system.

**Technical Aspects of Current System:**

**1. User Credential-Based Authentication:**

- The system relies on traditional username/password authentication.

- User credentials are exchanged between the client and server for validation.

- Limited flexibility in managing third-party access without compromising user credentials.

**2. Lack of Token-Based Authorization:**

- Authorization is not token-based; the absence of access tokens can lead to security vulnerabilities.

- Scopes and granular access control are not implemented, limiting the ability to define specific permissions for different resources.

**Technical Debt Accumulated:**

**1. Limited Security Scope:**

- The system lacks the advanced security features necessary for protecting against modern cyber threats.

- Exposure to potential credential theft or unauthorized access due to the absence of token-based authorization.

**2. Scalability Challenges:**

- Difficulty in scaling the system for third-party integrations securely.

- Over-reliance on user credentials can lead to performance bottlenecks and increased vulnerability surface.

**Adapting to OAuth 2.0:**

**Proposed Changes and Benefits:**

**1. Integration of OAuth 2.0:**

- Implement OAuth 2.0 for token-based authorization.

- Facilitate secure third-party access to resources without exposing user credentials.

- Enhance security by adopting access tokens with defined scopes, allowing granular control over resource access.

**2. Diversification of Authorization Grant Types:**

- Depending on the use case, implement appropriate OAuth 2.0 grant types (e.g., Authorization Code, Implicit, Resource Owner Password Credentials, Client Credentials).

- Enable more secure and flexible authorization flows tailored to specific scenarios.

**Mitigating Technical Debt:**

**1. Enhanced Security:**

- Mitigate security concerns related to user credentials by adopting OAuth 2.0's token-based approach.

- Address potential vulnerabilities and reduce the risk of unauthorized access.

**2. Improved Integration Capabilities:**

- Enable smoother integration with third-party services.

- Alleviate scalability challenges associated with traditional user credential-based authentication.

**Integrating mTLS:**

**Proposed Changes and Benefits:**

**1. Implementation of Mutual TLS:**

- Introduce mTLS for secure communication between the client and server.

- Mutual authentication during the TLS handshake ensures both parties' identities are verified.

**2. Certificate Exchange:**

- Implement the exchange of certificates during the TLS handshake to establish mutual trust.

- Strengthen the integrity of the communication channel, preventing unauthorized access.

Mitigating Technical Debt:

**1. Enhanced Communication Security:**

- Alleviate potential vulnerabilities associated with traditional TLS by adopting mutual authentication through mTLS.

- Ensure a more robust defense against man-in-the-middle attacks.

**2. Identity Verification:**

- Confirm the identities of both the client and server, reducing the risk of unauthorized entities participating in the communication.

- Address any technical debt related to insecure communication channels.

**B2B Sites:**

1. **Alibaba (B2B):**
   * **Reference:** [Alibaba](https://www.alibaba.com/)
2. **ThomasNet (B2B):**
   * **Reference:** [ThomasNet](https://www.thomasnet.com/)

**B2C Sites:**

1. **Amazon (B2C):**
   * **Reference:** [Amazon](https://www.amazon.com/)
2. **Walmart (B2C):**
   * **Reference:** [Walmart](https://www.walmart.com/)

**Before Implementation of OAuth 2.0 and mTLS**

A diagram of a server

Description automatically generated

**After implementation –**

A diagram of a software flow

Description automatically generated