Seguridad en Microservicios: Autenticación, Comunicación y Protección

Integrantes: Santiago Gutiérrez, Santiago Urrego, Luis Ramírez

1. Introducción a la Autenticación y Autorización en Microservicios

En un entorno de microservicios, cada servicio está diseñado para ser independiente. Esto genera un reto: no es viable que cada servicio pida credenciales de manera individual, pues afectaría la experiencia del usuario y aumentaría la complejidad. Por eso se emplean mecanismos centralizados que separan la autenticación (verificar identidad) de la autorización (definir los permisos y acciones que puede realizar un usuario).

2. Uso de OAuth2 y OpenID Connect

OAuth2 es un estándar de autorización que permite a un servicio delegar el acceso a recursos sin compartir credenciales directamente. Por ejemplo, cuando un usuario inicia sesión con Google en una aplicación de terceros. Por su parte, OpenID Connect se construye sobre OAuth2 para extenderlo a autenticación, permitiendo confirmar la identidad del usuario. Los elementos clave incluyen: Access Token, Refresh Token, Authorization Server y Resource Server.

3. Seguridad en la Comunicación entre Microservicios (TLS/mTLS)

La comunicación entre microservicios debe estar cifrada para evitar accesos no autorizados o ataques de interceptación. TLS asegura que los datos viajen de forma segura entre cliente y servidor. mTLS (mutual TLS) añade una capa más: ambos extremos, cliente y servidor, se autentican mediante certificados digitales. Esto garantiza que un microservicio solo se comunique con servicios legítimos y no con uno falso.

4. Gestión de Secretos y Configuración Segura en Entornos Distribuidos

Los microservicios requieren contraseñas, claves de API, certificados y otros secretos para operar. Una mala práctica común es almacenarlos en código fuente o en texto plano, lo que abre vulnerabilidades. Las buenas prácticas incluyen el uso de gestores de secretos como HashiCorp Vault, AWS Secrets Manager o Kubernetes Secrets. Además, se recomienda aplicar rotación automática de credenciales y el principio de privilegio mínimo.

5. Protección contra Ataques Comunes

La arquitectura de microservicios amplía la superficie de ataque, exponiendo múltiples endpoints. Por ello, se deben implementar medidas contra amenazas como:
- SQL Injection: validación de entradas y uso de ORM.
- XSS (Cross-Site Scripting): sanitización de datos y Content Security Policy.
- CSRF (Cross-Site Request Forgery): uso de tokens anti-CSRF y cookies SameSite.
Adicionalmente, se aplican técnicas como rate limiting, API Gateway, Zero Trust y uso de WAF (Web Application Firewall).

6. Prueba de Concepto

Como parte de la exposición, se implementó una prueba de concepto (PoC) para demostrar el uso de OAuth2 en microservicios. El frontend en HTML/JS permite al usuario iniciar sesión con Google, obteniendo un token de autenticación. Este token es validado por un backend en FastAPI, que confirma la identidad del usuario antes de permitir el acceso a recursos. Este ejemplo muestra cómo integrar OAuth2 en una arquitectura distribuida de forma segura.