TECNOLÓGICA DEL ORIENTE
INSTITUCIÓN DE EDUCACIÓN SUPERIOR

SISTEMAS DISTRIBUTIVOS

SEGURIDAD Y BUENAS PRÁCTICAS EN LA COMUNICACIÓN

SEGURIDAD Y BUENAS PRÁCTICAS EN LA COMUNICACIÓN

En un entorno distribuido, donde múltiples componentes intercambian información a través de redes en muchos casos públicas o compartidas, la seguridad en la comunicación adquiere una relevancia crítica. Cualquier vulnerabilidad en la transmisión de datos puede comprometer la integridad de la información, permitir accesos no autorizados, o incluso detener el funcionamiento del sistema completo (Muñoz Escoí, 2013).

Por esta razón, la seguridad en la comunicación no debe considerarse una capa secundaria o complementaria, sino un componente estructural del diseño de sistemas distribuidos. Su implementación adecuada garantiza que los datos sean transmitidos de forma confidencial, íntegra y sólo entre entidades autorizadas.

Principios de seguridad en la comunicación

Para lograr una comunicación segura entre componentes distribuidos, se deben garantizar los siguientes principios fundamentales:

- **Confidencialidad:** la información transmitida solo puede ser leída por los destinatarios autorizados.
- **Integridad:** los datos no deben ser alterados durante el tránsito.
- **Autenticación:** ambas partes deben confirmar la identidad de su contraparte.
- **Autorización:** después de autenticar, se debe verificar qué recursos está permitido usar.
- **No repudio:** una parte no puede negar que envió o recibió cierta información.
- **Disponibilidad:** los canales de comunicación deben permanecer operativos y resilientes a ataques.

Mecanismos de protección en la comunicación

Para materializar los principios anteriores, los sistemas distribuidos utilizan una variedad de mecanismos técnicos, entre los cuales se destacan:

1 Cifrado

El cifrado de datos garantiza que la información transmitida no pueda ser interpretada por terceros. Se emplean dos tipos principales:

- **Cifrado simétrico:** la misma clave se usa para cifrar y descifrar (ejemplo, AES).
- **Cifrado asimétrico:** se utilizan claves públicas y privadas (ejemplo, RSA).

En la práctica, muchos sistemas utilizan cifrado simétrico para los datos y cifrado asimétrico para intercambiar claves de sesión.

2 Protocolos seguros: HTTPS y TLS

El protocolo HTTPS (HTTP sobre TLS) es la base de la comunicación segura en la web. TLS (Transport Layer Security) proporciona un canal cifrado, resistente a espionaje y manipulación (Coulouris et al., 2012).

En sistemas distribuidos modernos, toda comunicación entre servicios internos y externos debe usar HTTPS/TLS como estándar mínimo de seguridad.

3 Firmas digitales

Las firmas digitales permiten validar la autenticidad e integridad del contenido, garantizando que no ha sido modificado y que proviene de un emisor confiable.

4 Tokens de acceso: JWT y OAuth

Los tokens JWT (JSON Web Tokens) se utilizan para representar identidades de forma segura en una sesión. Pueden incluir datos como nombre de usuario, roles y tiempos de expiración.

El protocolo OAuth 2.0 permite delegar el acceso sin compartir contraseñas. Es ampliamente usado para autorizar servicios de terceros (por ejemplo, iniciar sesión con Google).

Buenas prácticas en la comunicación segura

Más allá de los mecanismos técnicos, existen buenas prácticas ampliamente reconocidas para fortalecer la seguridad en la comunicación de sistemas distribuidos:

- **Implementar cifrado de extremo a extremo**. Toda comunicación debe estar cifrada, incluso dentro de redes internas, para evitar ataques de tipo man-in-the-middle. No se debe asumir que una red privada es segura por sí sola (Coulouris et al., 2012).
- **Validar datos entrantes**. Todo mensaje recibido debe pasar por mecanismos de validación: tipo de dato, formato, longitud, origen. Esto previene ataques como inyección de código o desbordamientos de búfer.
- **Uso de certificados digitales válidos**. Los certificados TLS deben ser emitidos por autoridades confiables. Se debe evitar el uso de certificados autofirmados en producción y renovar los certificados antes de su vencimiento.
- **Uso de firewalls y control de acceso a nivel de red**. Segmentar la red y restringir los puntos de acceso, reduce la superficie expuesta a ataques. Esto debe combinarse con firewalls lógicos y físicos que controlan el tráfico entre componentes.
- **Control de versiones y obsolescencia**. Es fundamental mantener las bibliotecas criptográficas y protocolos actualizados. Protocolos antiguos como SSL o TLS 1.0 deben ser desactivados por vulnerabilidades.
- **Prevención contra ataques de denegación de servicio (DoS/DDoS)**. Deben implementarse mecanismos de rate limiting, uso de CAPTCHAs, y herramientas de mitigación automática que detecten patrones anómalos de tráfico.

Ejemplo de implementación segura: REST con HTTPS y JWT

Un sistema RESTful moderno puede implementar una arquitectura segura, de la siguiente manera:

- El cliente solicita acceso mediante un login, y el servidor responde con un token JWT.
- Todas las llamadas posteriores se realizan vía HTTPS, incluyendo el token en la cabecera Authorization: Bearer <token>.
- El servidor valida la firma del token, verifica los permisos asociados y responde.
- Los tokens tienen expiración y pueden ser renovados mediante mecanismos de refresh token.

Este enfoque combina autenticación, autorización, confidencialidad e integridad, todo en un esquema relativamente sencillo y estandarizado.

Tabla 1. Amenazas comunes y cómo prevenirlas

Amenaza	Descripción	Prevención recomendada
Man-in-the-Middle (MITM).	Interceptor se posiciona entre cliente y servidor.	Uso obligatorio de TLS/HTTPS, certificados válidos.
Replay Attack.	Reenvío de una solicitud legítima para repetir una operación.	Tiempos de expiración, tokens únicos, nonces.
Inyección de código.	Inclusión de comandos maliciosos en datos enviados.	Validación exhaustiva, escape de caracteres.
Robo de sesión (session hijacking).	Suplantación de identidad usando un token robado.	Tokens firmados, expiración corta, IP bind, revocación de sesiones.
DdoS.	Saturación del sistema con tráfico masivo.	Rate limiting, proxies inversos, protección en la nube.