# Desarrollo de un Protocolo de Seguridad en el Desarrollo

El desarrollo de software seguro no es solo algo que se recomienda hacer, es algo totalmente necesario para que el sistema funcione bien y no haya problemas con la privacidad de los usuarios ni con la estabilidad de todo el sistema en general. Desde que se empieza a desarrollar el software, hay que seguir un protocolo de seguridad bien estructurado que ayude a anticiparse a los problemas antes de que se conviertan en algo serio y cueste más solucionarlos.

Antes de escribir una sola línea de código hay que ver qué amenazas pueden afectar al sistema porque si no se tienen en cuenta desde el principio luego pueden ser un problema. Es importante identificar los puntos por donde podrían atacar, ver qué impacto tendrían y decidir cómo evitarlo. Para esto existen metodologías como STRIDE y DREAD que dan una estructura para analizar estos riesgos desde el principio y tomar buenas decisiones para reducir los posibles fallos de seguridad.

Desde el primer momento en que se empieza a desarrollar un programa la seguridad tiene que ser una parte fundamental y no algo que se mete al final deprisa y corriendo. Si se aplica el principio de privilegios mínimos cada parte del programa tendrá solo los permisos que realmente necesite y así se evitan posibles ataques que aprovechen permisos innecesarios. Además, confiar solo en una medida de seguridad no es buena idea, por eso es mejor usar un sistema con varias capas de defensa para que si una falla no se quede todo el sistema expuesto. También es muy importante controlar bien los datos que entran al sistema para evitar ataques como inyecciones SQL o XSS que pueden hacer mucho daño. Respecto a la autenticación y la autorización no basta con usar contraseñas porque hoy en día se pueden hackear fácilmente por lo que lo mejor es usar autenticación en dos pasos como MFA y protocolos de acceso seguros como OAuth 2.0 o OpenID Connect para que nadie pueda acceder sin permiso.

Para asegurarnos de que el código es seguro hay que hacer revisiones constantes para detectar posibles vulnerabilidades antes de que lleguen a producción y usarse en el mundo real. Existen herramientas como SAST y DAST que analizan el software en distintas fases para buscar fallos y corregirlos. Usar bibliotecas seguras y bien mantenidas con herramientas como Dependabot o Snyk ayuda a no depender de código que pueda ser inseguro sin darnos cuenta. También es importante que el entorno de desarrollo esté aislado con contenedores o máquinas virtuales para que no haya filtraciones de datos o errores que comprometan la seguridad.

Los datos tienen que estar siempre protegidos tanto si están guardados como si están en tránsito. Para eso hay que usar cifrado fuerte como AES-256 para datos guardados y TLS 1.3 para datos que se envían por la red. También es muy importante no guardar contraseñas ni claves en el código porque es un fallo muy grave que se puede evitar usando herramientas como HashiCorp Vault o AWS Secrets Manager. Además, hay que establecer reglas para que cada persona dentro del sistema solo tenga acceso a la información que necesita y no más.

Para asegurarse de que todo está funcionando bien y detectar posibles problemas es importante tener un registro de eventos centralizado con herramientas como ELK Stack o Splunk que permiten analizar lo que está pasando en tiempo real. También se pueden usar sistemas de detección de intrusos como Snort o Wazuh que avisan si hay actividad sospechosa en el sistema. Y por si algo sale mal hay que tener protocolos de respuesta bien definidos para reaccionar rápido y minimizar los daños en caso de ataque.

El software tiene que estar siempre actualizado porque si no con el tiempo aparecen vulnerabilidades que pueden ser explotadas. Es clave mantener actualizadas las dependencias de manera periódica y aplicar parches de seguridad siempre que sea necesario. Además, hay estrategias como Blue-Green Deployment o Canary Releases que permiten actualizar sin interrumpir el servicio y reduciendo riesgos.

La seguridad no es solo cosa de herramientas y código sino también de quien lo desarrolla. Por eso es importante que los desarrolladores reciban formación continua para estar al día de nuevas amenazas y mejores prácticas. Además, es útil hacer simulacros y pruebas como pentesting y ejercicios de Red Teaming para comprobar si las medidas de seguridad realmente funcionan.