Guía esencial de buenas prácticas para la programación segura

La seguridad en el desarrollo de software no es una característica opcional, sino un requisito fundamental. En un mundo hiperconectado, proteger nuestras aplicaciones desde la base es crucial. Basándome en fundamentos de programación segura y las tendencias actuales, aquí tienes una guía esencial para elevar la seguridad de tus proyectos.

1. Prácticas fundamentales de programación segura

El primer paso es la mentalidad del desarrollador. No debemos confiar ciegamente en la entrada del usuario ni en entornos externos.

- **Limpieza de datos (Input Sanitization):** Nunca confíes en los datos recibidos. Valida, limpia y filtra siempre cualquier entrada, ya sea desde formularios, URLs o parámetros de línea de comandos.
- **Comprobación de límites:** Evita desbordamientos de búfer (buffer overflows) asegurándote de que los datos se ajusten al espacio reservado.
- **Principio de menor privilegio:** Ejecuta tu aplicación con los permisos mínimos necesarios.

2. Seguridad en aplicaciones web: El punto crítico

Las aplicaciones web son el blanco principal de los atacantes. Algunas medidas clave incluyen:

Amenaza	Acción de seguridad
Manipulación de URLs	No utilices URLs para pasar datos sensibles o variables de control.

Amenaza	Acción de seguridad
Campos HTML ocultos	Nunca confíes en ellos para lógica de servidor; son fácilmente modificables por el usuario.
Cookies inseguras	Utiliza flags de seguridad (HttpOnly, Secure) y no almacenes información sensible.

3. Qué evitar a toda costa

Existen prácticas que, aunque facilitan el desarrollo a corto plazo, abren puertas a vulnerabilidades graves:

1. No confíes nunca en que el usuario no es malicioso.
2. No guardes contraseñas en texto plano (usa algoritmos de hashing fuertes como Argon2 o BCrypt).
3. No envíes información confidencial por e-mail sin cifrar.
4. No inviertas en el uso de programas o librerías de terceros sin verificar su fiabilidad.

4. Perspectiva actual: Más allá de los fundamentos

Hoy en día, la seguridad ha evolucionado hacia un modelo más integrado:

- **DevSecOps:** La seguridad debe ser parte del ciclo de vida de desarrollo desde el día uno, no un paso final.
- **Escaneo automático (SAST/DAST):** Implementa herramientas que analicen tu código (SAST) y tu aplicación en ejecución (DAST) automáticamente en tu pipeline de CI/CD.
- **Gestión de dependencias:** Usa herramientas (como Dependabot) para mantener tus librerías de terceros actualizadas y libres de vulnerabilidades conocidas.

La seguridad es un proceso continuo, no una meta final. ¡Mantén tu código limpio y protegido!