UTT
UNIVERSIDAD TECNOLÓGICA DE TIJUANA
GOBIERNO DE BAJA CALIFORNIA

SECURE CODING PRINCIPLES SPECIFICATION

Alumno
Gomez Perez Manuel de Jesus

Grupo
10 B

Docente
Ray Bruett Parra Galaviz

Materia
Desarrollo Movil integral

8 Mejores Prácticas de Codificación Segura

OWASP ofrece una lista de verificación de 14 áreas clave en el ciclo de vida del desarrollo de software. Aquí presentamos las ocho principales prácticas para proteger tu software:

1. **Seguridad por Diseño**
   La seguridad debe ser prioritaria desde el inicio del desarrollo. Aunque puede entrar en conflicto con la rapidez de desarrollo, priorizar la seguridad desde el diseño evita futuros costos asociados a errores y vulnerabilidades. Implementa revisiones del código y herramientas de automatización de seguridad durante el ciclo de vida del software.
2. **Gestión de Contraseñas**
   Las contraseñas son un punto débil común. Asegúrate de que sean complejas y almacénalas como hashes criptográficos con sal, nunca como texto plano. Establece políticas de longitud, complejidad y bloqueo tras varios intentos fallidos.
3. **Control de Acceso**
   Aplica un enfoque de "denegar por defecto". Restringe los privilegios y permite acceso solo a usuarios autorizados. Valida cada solicitud de acceso a datos sensibles para garantizar que el usuario tenga autorización.
4. **Manejo de Errores y Registro**
   Manejar errores y registrar fallos ayuda a mitigar vulnerabilidades. Documenta y analiza errores, excepciones y fallos en un sistema seguro para identificar y corregir sus causas.
5. **Configuración del Sistema**
   Mantén el software actualizado y elimina componentes innecesarios. Las actualizaciones regulares corrigen vulnerabilidades conocidas, y un sistema de gestión de parches puede ayudarte a mantenerte al día.
6. **Modelado de Amenazas**
   Este proceso identifica y aborda áreas de riesgo en el software. Desde el

desarrollo hasta la producción, evalúa las posibles amenazas y valida las soluciones implementadas.

7. **Prácticas Criptográficas**

   Utiliza algoritmos criptográficos modernos y sigue las mejores prácticas de gestión de claves para proteger los datos, incluso en caso de una brecha.

8. **Validación de Entradas y Codificación de Salidas**

   Verifica todas las entradas de datos y clasifica las fuentes como confiables o no confiables. Implementa rutinas estándar para validar entradas y codificar salidas de forma segura.