Universidad de las Fuerzas Armadas ESPE

Nombre: Joel Taipicaña

Fecha: 03/06/2025

Materia: Ingeniería de Seguridad de Software

Tema: Revisar principales vulnerabilidades en OWASP

Antecedentes

El crecimiento exponencial de las aplicaciones web y móviles ha incrementado los riesgos de seguridad asociados a su desarrollo, despliegue y mantenimiento. Desde 2003, la OWASP Foundation (Open Worldwide Application Security Project) ha proporcionado una lista actualizada de las vulnerabilidades más críticas que afectan a las aplicaciones web, conocida como OWASP Top 10. Esta lista se basa en datos empíricos recolectados de miles de organizaciones y herramientas de escaneo de seguridad, ofreciendo una guía confiable y globalmente reconocida para priorizar riesgos y fortalecer las defensas de las aplicaciones.

Introducción

La seguridad en el desarrollo de software es una responsabilidad compartida que comienza desde el diseño y se extiende hasta el monitoreo en producción. La lista OWASP Top 10 representa una herramienta clave para que desarrolladores, analistas y auditores identifiquen las vulnerabilidades más comunes, comprendan sus causas y adopten medidas preventivas. El último informe OWASP Top 10 (2021) no solo reordenó amenazas conocidas, sino que introdujo categorías modernas como fallas en diseño y software vulnerable.

Objetivo

Revisar y analizar cada una de las principales vulnerabilidades descritas por OWASP en su Top 10 más reciente, con ejemplos prácticos, causas frecuentes, impactos potenciales y estrategias de mitigación.

Desarrollo

OWASP Top 10 (2021)

A01:2021 – Broken Access Control (Control de acceso roto)

- **Descripción:** Usuarios pueden acceder a recursos sin autorización (e.g. modificar roles, ver datos ajenos).
- **Ejemplo:** Un usuario cliente cambia su ID en la URL (/user/123) y accede al perfil de otro (/user/456).
- **Mitigación:** Verificaciones en backend, uso de control de acceso por roles (RBAC).

A02:2021 – Cryptographic Failures (Fallos criptográficos)

- **Descripción:** Uso incorrecto de cifrado o transmisión de datos sensibles sin protección.
- **Ejemplo:** Contraseñas almacenadas sin hash seguro (como SHA1).
- **Mitigación:** Usar HTTPS, algoritmos modernos (Argon2, bcrypt), no almacenar datos sensibles sin necesidad.

A03:2021 – Injection (Inyección)

- **Descripción:** Entrada maliciosa que altera las instrucciones del servidor (SQL, OS, LDAP, etc).
- **Ejemplo:** ‘ OR ‘1’=’1 en un login SQL.
- **Mitigación:** Uso de parámetros preparados (PreparedStatement), validación de entradas.

A04:2021 – Insecure Design (Diseño inseguro)

- **Descripción:** Ausencia de patrones de seguridad en la arquitectura o lógica de negocio.
- **Ejemplo:** App sin límites de intentos de acceso, susceptible a fuerza bruta.
- **Mitigación:** Modelado de amenazas, validación de lógica, pruebas de diseño seguro.

A05:2021 – Security Misconfiguration (Mala configuración de seguridad)

- **Descripción:** Configuraciones por defecto, puertos abiertos, servicios innecesarios.
- **Ejemplo:** Consola de administración expuesta con credenciales por defecto.
- **Mitigación:** Revisiones regulares de configuración, escaneos de seguridad automatizados (Ej: Trivy, Lynis).

A06:2021 – Vulnerable and Outdated Components (Componentes vulnerables u obsoletos)

- **Descripción:** Uso de bibliotecas con CVEs conocidos.
- **Ejemplo:** Log4j 2.14.1 vulnerable al exploit “Log4Shell”.
- **Mitigación:** Uso de herramientas como OWASP Dependency-Check, Snyk o Renovate.

A07:2021 – Identification and Authentication Failures (Fallos de autenticación)

- **Descripción:** Contraseñas débiles, tokens inseguros, sesiones mal gestionadas.
- **Ejemplo:** JWT sin expiración o firma.
- **Mitigación:** MFA, control de sesión, políticas de contraseña, uso de librerías probadas.

A08:2021 – Software and Data Integrity Failures (Fallos de integridad de software y datos)

- **Descripción:** Uso de scripts/módulos sin verificación, cadenas de suministro comprometidas.
- **Ejemplo:** Actualización automática desde fuente no verificada.
- **Mitigación:** Firmas digitales, hashes de integridad, repositorios confiables.

A09:2021 – Security Logging and Monitoring Failures (Fallas de registro y monitoreo)

- **Descripción:** Ausencia de registros o monitoreo que detecten ataques o anomalías.
- **Ejemplo:** No registrar intentos fallidos de login o movimientos sospechosos.
- **Mitigación:** Integración con SIEM, almacenamiento seguro de logs, alertas proactivas.

A10:2021 – Server-Side Request Forgery (SSRF)

- **Descripción:** El atacante hace que el servidor realice solicitudes no autorizadas (internas o externas).
- **Ejemplo:** App permite cargar URLs arbitrarias que acceden a servicios internos (localhost:8080/admin).
- **Mitigación:** Validación de destinos, listas blancas, firewalls de salida.

Conclusiones

- El OWASP Top 10 proporciona una visión actualizada y priorizada de los riesgos más críticos en aplicaciones web, adaptada a la realidad moderna del desarrollo.
- Muchas vulnerabilidades son prevenibles con buenas prácticas de codificación, uso de herramientas automatizadas y una arquitectura centrada en la seguridad.
- Las categorías como “Diseño inseguro” y “Fallos de integridad de software” reflejan la necesidad de adoptar un enfoque integral, no solo técnico, sino también organizacional y estratégico.

Recomendaciones

- Integrar el OWASP Top 10 en capacitaciones de los equipos de desarrollo y QA.
- Usar herramientas Open Source como SonarQube, ZAP, Bandit y Trivy para automatizar la detección de fallos.
- Adoptar un enfoque de seguridad por diseño (Security by Design) y validaciones constantes en todo el ciclo de vida de la aplicación.