# OWASP

Open Web Application Security Project- Los 10 principales riesgos de seguridad de las aplicaciones web

Presentado por: Joseph Guerrero y Analía Solís

# Introduccion

Para ayudar a los equipos de desarrollo, operaciones y seguridad a entender y mitigar los riesgos más comunes, la organización OWASP (Open Web Application Security Project) elabora y actualiza periódicamente una lista conocida como el OWASP Top 10. Esta lista recopila los diez riesgos de seguridad más críticos que afectan a las aplicaciones web, basada en datos reales recopilados a nivel global y en el análisis de expertos en ciberseguridad.

# Broken Access Control (Control de Acceso Roto)

- Descripción:
  - Ocurre cuando los usuarios pueden acceder o modificar recursos que no deberían.
- Importancia:
  - Permite robo de datos, modificación de cuentas o acciones maliciosas si no se controla adecuadamente quién puede hacer qué.

Ejemplos:

- Cambiar la URL /usuario/123 a /usuario/124.
- Acceder a funciones administrativas sin autorización.

# Cryptographic Failures (Fallos Criptograficos)

1

Descripción:

- Mal uso o ausencia de cifrado en datos sensibles.

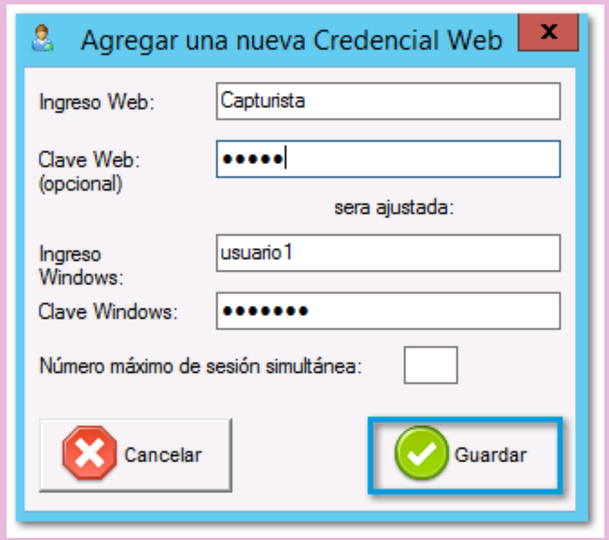
Importancia:

- Exponen información crítica como contraseñas o tarjetas a posibles ataques.

2

Ejemplos:

- Enviar credenciales por HTTP.
- Usar MD5 o guardar contraseñas en texto plano.



# Injection (Inyecciones)

- Descripción:
  - Los datos del usuario se ejecutan como código en consultas o comandos.
- Importancia:
  - Permite acceso no autorizado, corrupción de datos o control del servidor.

Ejemplos:

- Inyección SQL: ' OR '1'='1.
- Comandos insertados en formularios ejecutados en el servidor.

# Insecure Design (Diseño Inseguro)

- Descripción:
  - Falta de medidas de seguridad desde la fase de diseño.
- Importancia:
  - Deja vulnerabilidades estructurales difíciles de corregir más adelante.

Ejemplos:

- No limitar la cantidad de solicitudes por usuario.
- No realizar análisis de amenazas al diseñar la arquitectura.

# Security Misconfiguration (Configuracion Insegura)

- Descripción:
  - Errores en la configuración de servidores, software o servicios.
- Importancia:
  - Común fuente de brechas de seguridad, incluso con código correcto.

Ejemplos:

- Consolas de administración expuestas.
- Mensajes de error con detalles internos.

# Vulnerable and Outdated Components

# (Componentes Vulnerables y Obsoletos)

- Descripción:
  - Uso de librerías o frameworks con vulnerabilidades conocidas.
- Importancia:
  - Pueden explotarse sin necesidad de fallos en el propio código.

Ejemplos:

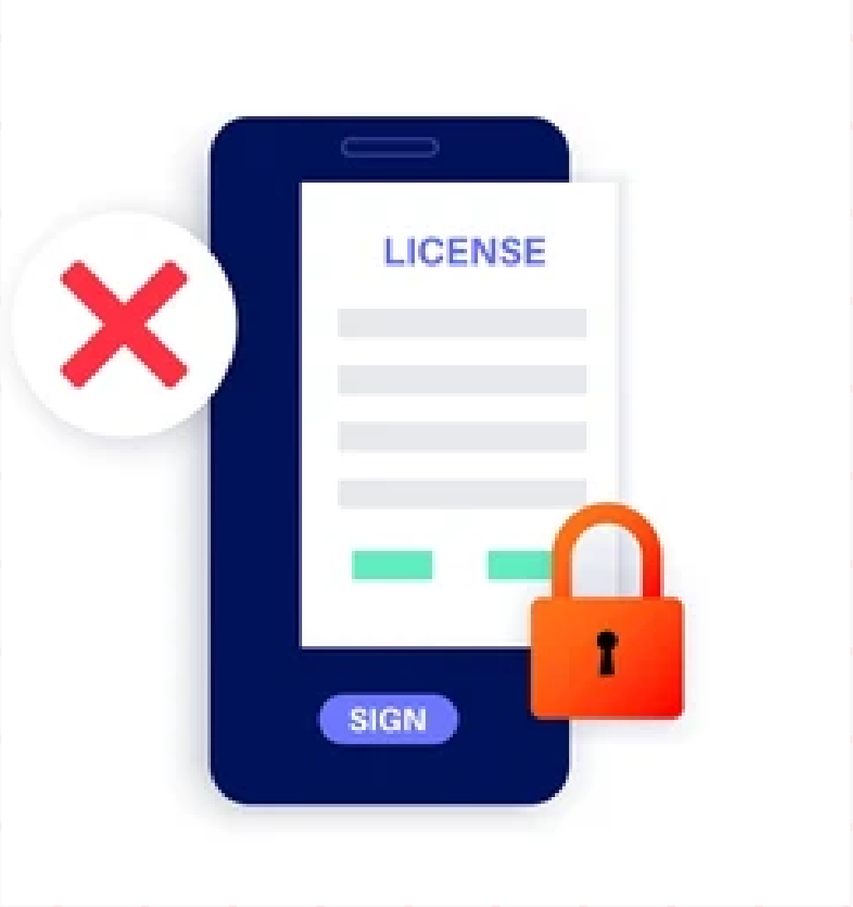
- Uso de Log4j con vulnerabilidad Log4Shell.
- jQuery obsoleto con fallas XSS.

# Identification and Authentication Failures (Fallos de Autenticacion)

- Descripción:
  - Errores en la gestión de identidad y sesiones de usuarios.
- Importancia:
  - Permite que un atacante suplante a otros usuarios o acceda como administrador.

Ejemplos:

- No hay límite de intentos de login.
- Tokens de sesión inseguros o sin expiración.



# Software and Data Integrity Failures

# (Fallos de Integridad de Software y Datos)

- Descripción:
  - Falta de validación de la integridad del software y datos críticos.
- Importancia:
  - Permite introducir código malicioso en actualizaciones o procesos internos.

Ejemplos:

- CI/CD descargando software de fuentes no verificadas.
- Inyección de código en repositorios sin control.

# Security Logging and Monitoring Failures

# (Fallos de Monitoreo y Registro)

- Descripción:
  - Ausencia de registros o sistemas de alerta sobre actividades sospechosas.
- Importancia:
  - Dificulta detectar y responder a ataques en tiempo real.

Ejemplos:

- No registrar accesos fallidos o cambios importantes.
- Sin alertas ante comportamiento anómalo.

# Server-Side Request Forgery (SSRF)

- Descripción:
  - El servidor realiza peticiones HTTP arbitrarias provocadas por el usuario.
- Importancia:
  - Puede usarse para explorar redes internas o acceder a servicios sensibles.

Ejemplos:

- Formularios que descargan URLs sin validación.
- Acceso a metadatos de servidores en la nube (ej. AWS).

# Conclusion

Los riesgos del OWASP Top 10 no son teóricos, son vulnerabilidades reales y frecuentes con impactos graves si no se abordan.
Incluyen errores como control de acceso débil, mala configuración, falta de cifrado y uso de componentes obsoletos.
La seguridad no es solo tarea del equipo de seguridad: debe integrarse desde el diseño hasta la operación de la aplicación.
Más que una lista de errores, el OWASP Top 10 es una guía para aplicar buenas prácticas, formar equipos y reforzar la seguridad en todo el ciclo de vida del software.

# Gracias