OWASP Top 10 2021: Principales Riesgos de Seguridad en Aplicaciones Web

Introducción y Cambios Clave

Este es un extracto de la introducción al OWASP Top 10 de 2021. Aquí se resumen las principales actualizaciones y cambios en las categorías de riesgos de seguridad en aplicaciones web. Algunas de las categorías se han modificado, renombrado o consolidado para reflejar mejor las causas subyacentes de los problemas en lugar de los síntomas. También se hace énfasis en el uso de datos y encuestas de la comunidad para determinar las vulnerabilidades más críticas.

Cambios Clave:

- A01:2021 - Pérdida de Control de Acceso: Ha subido de la quinta posición a la categoría con el mayor riesgo en seguridad de aplicaciones web.
- A02:2021 - Fallas Criptográficas: Ahora ocupa el segundo lugar, reemplazando a la antigua categoría de Exposición de Datos Sensibles.
- A03:2021 - Inyección: Desciende al tercer lugar, pero sigue siendo un problema prevalente con un 94% de aplicaciones analizadas.
- A04:2021 - Diseño Inseguro: Es una categoría completamente nueva que se enfoca en los riesgos relacionados con fallas en el diseño de la arquitectura de aplicaciones.
- A05:2021 - Configuración Incorrecta de Seguridad: Esta categoría ha subido del sexto lugar, afectando al 90% de las aplicaciones analizadas.
- A06:2021 - Componentes Vulnerables y Desactualizados: Ha ascendido desde la novena posición, destacando la importancia de utilizar componentes actualizados.
- A07:2021 - Fallas de Identificación y Autenticación: Ha bajado desde la segunda posición y ahora cubre más fallas relacionadas con la identificación de los usuarios.
- A08:2021 - Fallas en la Integridad del Software y de los Datos: Esta es

una nueva categoría que se enfoca en las vulnerabilidades asociadas con la integridad de los datos y actualizaciones de software.
- A09:2021 - Fallos en el Registro y Monitoreo de Seguridad: Anteriormente en la décima posición, esta categoría ha subido por su impacto en la visibilidad y respuesta ante incidentes de seguridad.
- A10:2021 - Falsificación de Solicitudes del Lado del Servidor (SSRF): Introducida como resultado de la encuesta a la comunidad, destacando su relevancia a pesar de tener una tasa de incidencia relativamente baja.

Metodología

Para esta edición, OWASP ha usado una metodología más orientada a los datos. Han integrado puntuaciones CVSSv3 para evaluar la explotabilidad e impacto de las vulnerabilidades. Además, han aumentado significativamente el número de CWEs (Common Weakness Enumerations) analizadas, pasando de 30 a casi 400.

Cambios en el Enfoque de Categorías
Algunos cambios importantes en la estructura incluyen un enfoque en las causas raíz de las vulnerabilidades (por ejemplo, Falla Criptográfica y Configuración de Seguridad Incorrecta) en lugar de los síntomas (como Exposición de Datos Sensibles). Además, algunas categorías han sido reagrupadas para enfocarse más en los aspectos de causa raíz.

En resumen, el OWASP Top 10 de 2021 presenta una lista más refinada y basada en datos más recientes sobre las amenazas y vulnerabilidades más críticas en aplicaciones web.

El OWASP Top 10 2021 cambió significativamente su estructura con respecto a ediciones anteriores, tanto en la forma en que se agrupan las vulnerabilidades como en la manera de analizar y seleccionar las categorías. A continuación te resumo los principales puntos clave sobre **cómo se estructuran las categorías** y cómo se usan los datos:

🧱 1. Cambios en la estructura de las categorías

- **Antes**: Se partía de unas 30 CWE (Common Weakness Enumeration) predeterminadas, y los datos se limitaban a ese subconjunto.

- **Ahora**: Se solicitaron datos abiertos, sin restricciones de CWE. Se pasó a analizar cerca de **400 CWEs**, lo que requirió **reorganizar las categorías**.

- Se prioriza la agrupación **por causa raíz** en lugar de síntomas. Ejemplo:

    - Causa raíz: *Fallas criptográficas, configuración insegura*.

    - Síntoma: *Exposición de datos, denegación de servicio*.

- En promedio, cada categoría incluye **19,6 CWEs**, con un mínimo de 1 (por ejemplo, en SSRF) y un máximo de 40 (Diseño inseguro).

📊 2. Uso de datos para seleccionar categorías

- **Datos usados**:

    - Número de aplicaciones analizadas.

    - Número de aplicaciones con al menos una instancia de una CWE.

- Se prioriza la **prevalencia (tasa de incidencia)** en lugar de la **frecuencia**, para evitar que ciertos tipos de vulnerabilidades

fáciles de detectar dominen las estadísticas (como XSS).

- Se combinaron datos de distintas fuentes:
    - Herramientas automatizadas.
    - Evaluaciones manuales asistidas por herramientas.
    - Pruebas humanas asistidas por herramientas (HaT y TaH).
- Solo se consideró si **una aplicación tenía o no al menos una instancia** de cierta vulnerabilidad, sin importar si eran 4 o 4.000.

🔍 3. Cálculo del riesgo: explotabilidad + impacto técnico

- Se usaron **puntuaciones CVSS (Common Vulnerability Scoring System)** de la NVD (National Vulnerability Database).
- Se combinaron versiones **CVSSv2 y CVSSv3** para obtener puntuaciones de:
    - **Explotabilidad**.
    - **Impacto técnico**.
- Se asignaron puntuaciones promedio a cada CWE, lo cual permitió ponderar las categorías seleccionadas por riesgo real, no solo por popularidad.

👥 4. Complemento con encuesta a la comunidad

- Solo **8 de las 10 categorías** se seleccionan con base en datos.
- Las otras **2 se eligen por votación** entre expertos de la comunidad de seguridad, para reflejar riesgos emergentes que aún no aparecen con claridad en los datos automatizados.

⚙️ 5. Proceso de recopilación y análisis

- Se formalizó desde 2017 con participación abierta en la comunidad OWASP.
- Se publicaron plantillas de ejemplo y se brindó asistencia a organizaciones para mapear CWEs.
- Se revisaron y normalizaron los datos recibidos de múltiples fuentes: herramientas de escaneo, pruebas internas, programas de recompensas de errores, etc.
- Se analizaron **todas las CWEs con mayor tasa de incidencia**, se agruparon, y luego se seleccionaron las que forman parte del **Top 10 final**.

🔐 OWASP y la Pérdida de Control de Acceso (A01:2021)

🧩 ¿Qué es OWASP?

OWASP (Open Web Application Security Project) es una comunidad abierta que busca mejorar la seguridad de software. Proporciona:

- Herramientas, estándares y bibliotecas abiertas.

- Libros, guías, cheat sheets, videos y entrenamientos gratuitos.
- Eventos, conferencias y grupos locales.
- Contenido sin fines comerciales, mantenido por voluntarios.

🚨 A01:2021 - Pérdida de Control de Acceso

📌 Descripción

Esta vulnerabilidad ocurre cuando los usuarios pueden realizar acciones fuera de los permisos asignados. Generalmente permite:

- Ver o modificar información no autorizada.
- Ejecutar funciones de negocio que no le corresponden.

💥 Ejemplos comunes

- Cambiar parámetros en la URL para acceder a cuentas de otros usuarios.
- Acceder a páginas de administrador sin autenticación.
- Usar identificadores predecibles para editar información de otros.
- Fallas en controles sobre métodos HTTP como POST, PUT o DELETE.
- JWTs no invalidados correctamente después del logout.

- CORS mal configurado que permite acceso desde dominios no confiables.

🧪 Escenarios de Ataque

SQL Injection sin verificación:

java
CopiarEditar
```
pstmt.setString(1, request.getParameter("acct"));
```

- Acceso directo a cuentas de otros usuarios modificando el parámetro.

Acceso directo a URL restringidas:

url
CopiarEditar
```
https://example.com/app/admin_getappInfo
```

-

📉 Estadísticas (2021)

- Tasa de incidencia: **3.81%**
- Afecta al **94.55%** de las apps analizadas.
- **318,487** incidencias totales detectadas.

🛡️ Cómo prevenirla

- **Controles del lado del servidor (o en APIs)**. Nada en el cliente es confiable.
- **Negar acceso por defecto** a todo recurso no público.
- **Invalidar sesiones y tokens** correctamente al cerrar sesión.
- **JWTs de corta duración**, o usar revocación (OAuth).
- **Unificar mecanismos de control** y reutilizarlos en toda la aplicación.
- **Limitar la tasa de accesos** a APIs para prevenir ataques automatizados.
- **Pruebas de control de acceso** tanto unitarias como de integración.
- **Auditoría y registro** de accesos fallidos o sospechosos.

🔐 Principales Categorías y Vulnerabilidades Resumidas

1. CWE Criptográficas y de Comunicación Insegura

Estas CWE están relacionadas con el uso de algoritmos criptográficos débiles, generación de números predecibles o identificación deficiente, y problemas en el transporte seguro:

- **CWE-338, CWE-340, CWE-347**: Problemas en PRNG, firmas mal verificadas.
- **CWE-523, CWE-818**: Transporte de credenciales sin protección.
- **CWE-780**: Uso de RSA sin OAEP.
- **CWE-916**: Hash de contraseñas sin coste computacional suficiente.
- **CWE-720**: Antigua categoría OWASP A9 sobre comunicaciones inseguras.

2. A03:2021 - Inyección

- Tasa de incidencia: Máxima 19%, promedio 3.37%.
- CWEs relacionadas: Incluye más de 30 CWE como CWE-79 (XSS), CWE-89 (SQL Injection), CWE-78 (OS Command Injection), etc.
- Descripción: Ocurre cuando los datos de usuario no se validan y se insertan directamente en consultas, comandos, etc.
- **Prevención**:
    - Uso de consultas parametrizadas (ORM, API segura).
    - Validación con listas blancas.
    - Escapar caracteres especiales solo si es estrictamente necesario.

- Revisar código fuente + herramientas SAST/DAST.

3. A04:2021 - Diseño Inseguro

- **Tasa de incidencia:** Máxima 24.19%, promedio 3.00%.
- **Descripción:** Diferencia clave con defectos de implementación: aquí el sistema **nunca fue diseñado para ser seguro**.
- **Ejemplos CWE:**
  - **CWE-209**: Mensajes de error con información sensible.
  - **CWE-256**: Almacenamiento inseguro de credenciales.
  - **CWE-522**: Credenciales no protegidas.
- **Prevención:**
  - Modelado de amenazas desde el inicio (shift-left).
  - Patrones de diseño seguros.
  - Arquitecturas de referencia y segregación de tenants.

📊 **Estadísticas Clave**

Métrica	Inyección (A03)	Diseño Inseguro (A04)

Tasa incidencia máx	19.09%	24.19%
Tasa incidencia promedio	3.37%	3.00%
Impacto promedio ponderado	7.25	6.46
Cobertura máx	94.04%	77.25%
CVEs totales asociadas	32,078	2,691

📚 Recursos Recomendados OWASP

- Cheat Sheet Injection Prevention
- OWASP ASVS: V5 Input Validation
- OWASP Proactive Controls
- PortSwigger on Server-side template injection

🔐 Configuración Insegura (A05:2021)

Referencias clave:

- OWASP Testing Guide (configuración y manejo de errores)
- Application Security Verification Standard (V14)
- NIST Server Hardening Guide

- CIS Benchmarks

- CWEs relacionadas:
  Incluyen problemas como contraseñas en archivos de configuración (CWE-260, CWE-13), uso de cookies inseguras (CWE-1004, CWE-614), mensajes de error con información sensible (CWE-537), y configuración errónea en ASP.NET (CWE-11, CWE-1174).

Descripción y riesgos:

- Configuraciones por defecto, errores en archivos `web.config`, exposición de información en mensajes de error, archivos temporales accesibles, entre otros.

- Las malas configuraciones pueden permitir que atacantes obtengan acceso o información sensible.

Prevención:

- Revisar configuraciones por entorno (dev/test/prod).

- Eliminar componentes y archivos innecesarios.

- Usar encabezados de seguridad HTTP.

- Aplicar principio de menor privilegio en cuentas y servicios.

🧩 Componentes Vulnerables y Desactualizados (A06:2021)

Referencias clave:

- OWASP Dependency Check, Retire.js
- NVD, CVE, CWE-1104 (componentes sin mantenimiento)
- MITRE CWE mapeadas: CWE-937, CWE-1035

Descripción y riesgos:

- Las aplicaciones que usan librerías, frameworks o sistemas operativos desactualizados pueden ser vulnerables a ataques.
- Muchas organizaciones no tienen control total de sus dependencias, especialmente las anidadas.

Prevención:

- Hacer inventario de versiones y dependencias.
- Usar herramientas de escaneo de vulnerabilidades (como OWASP DC o SCA tools).
- Evitar usar componentes no oficiales o no firmados.
- Mantenerse actualizado con boletines de seguridad y aplicar parches regularmente.

Ejemplos reales:

- CVE-2017-5638 (Struts2) – ejecución remota de código.
- Uso del motor de búsqueda Shodan para encontrar dispositivos vulnerables.

👤 Fallas de Identificación y Autenticación (A07:2021)

Referencias clave:

- NIST 800-63b (autenticación segura)
- CWEs relacionadas:
  CWE-287 (autenticación incorrecta), CWE-384 (fijación de sesión), CWE-297 (certificados inválidos), entre otras.

Descripción y riesgos:

- Contraseñas débiles o por defecto.
- Recuperación de contraseña insegura.
- Falta de MFA o validación débil.
- Almacenamiento inseguro de contraseñas.
- Manejo incorrecto de sesiones (por ejemplo, ID en URL, tokens no invalidados).

Prevención:

- Implementar MFA.
- Bloquear intentos automatizados (fuerza bruta).
- Verificar contraseñas contra listas negras.

- Uso correcto de tokens y expiración de sesiones.
- Evitar mensajes de error distintos que permitan enumeración de usuarios.

A08:2021 - Fallas en el Software y en la Integridad de los Datos

- Descripción: Categoría nueva en 2021. Trata sobre fallas en la verificación de integridad de software, datos y pipelines de CI/CD. Incluye riesgos como descargar código sin verificar, depender de fuentes no confiables o permitir deserialización insegura.
- Problemas típicos:
  - Código de fuentes o CDN no confiables.
  - Actualizaciones sin verificación de integridad (ej. routers o firmwares).
  - Pipelines de CI/CD mal configurados.
  - Uso inseguro de objetos serializados.
- Prevención:
  - Firmas digitales para verificar la integridad.
  - Uso de repositorios confiables (o internos).
  - Herramientas como OWASP Dependency-Check o CycloneDX.

    - Revisión de cambios y control de acceso en CI/CD.
    - Validación de datos serializados.

- Ejemplos destacados:
    - **SolarWinds**: distribución masiva de actualizaciones maliciosas.
    - **Firmware sin firmar** en dispositivos.
    - **Deserialización insegura** con Java Serial Killer.
- CWEs asociadas: CWE-494, CWE-502, CWE-829, CWE-915, entre otras.
- Impacto promedio: 7.94 / Explotabilidad: 6.94 / Cobertura promedio: 45.35%.

📋 A09:2021 - Fallas en el Registro y Monitoreo

- Descripción: Se enfoca en la detección, monitoreo y respuesta a incidentes. Aunque difícil de probar y con pocos CVEs, es clave para la visibilidad y análisis post-incidente.
- Problemas típicos:
    - No se registran eventos importantes (login, errores, transacciones).

    - Falta de monitoreo en tiempo real.
    - Logs insuficientes, locales o sin alertas.
    - Información sensible mal manejada en los logs.
- Prevención:
    - Registro detallado de errores y accesos.
    - Logs en formatos analizables y codificados correctamente.
    - Auditoría de transacciones críticas.
    - Alertas y respuestas rápidas por DevSecOps.
    - Uso de herramientas como ELK Stack o ModSecurity.
- Ejemplos destacados:
    - Proveedor de salud infantil sufrió una brecha masiva sin detectar durante años por no tener monitoreo ni registros.
- CWEs asociadas: CWE-117, CWE-223, CWE-532.
- **Impacto promedio**: 4.99 / Explotabilidad: 6.87 / Cobertura promedio: 39.97%

🔟 OWASP Top 10 - 2021 (Resumen Rápido)

Código	Título	Descripción Breve

A01	**Control de Acceso Roto**	Usuarios acceden a recursos sin autorización; mala validación de permisos o roles.
A02	**Fallos Criptográficos**	Uso incorrecto o débil de cifrado, almacenamiento inseguro de contraseñas o datos sensibles.
A03	**Inyección**	Código malicioso insertado en comandos SQL, NoSQL, LDAP, etc.
A04	**Diseño Inseguro**	Falta de seguridad desde el diseño (ej: sin validaciones, sin límites de acceso desde el inicio).
A05	**Configuración de Seguridad Incorrecta**	Configuraciones por defecto, puertos abiertos innecesarios, headers inseguros, etc.
A06	**Componentes Vulnerables y Desactualizados**	Uso de librerías/frameworks con fallas conocidas.
A07	**Identificación y Autenticación Rota**	Mal manejo de sesiones, tokens inseguros, autenticación débil.
A08	**Fallas en el Software y la Integridad de Datos**	Actualizaciones inseguras, CI/CD comprometido, carga de scripts no verificados.
A09	**Fallas en el Registro y Monitoreo**	No se detectan incidentes por falta de logs, alertas o monitoreo.

A10	**Falsificación de Solicitudes del Lado del Servidor (SSRF)**	El servidor es manipulado para hacer solicitudes a destinos internos o externos.