OWASP: TOP 10 VULNERABILIDADES PRINCIPALES

ALEJANDRO DE LA HUERGA FERNANDEZ

FUNDAMENTOS DE CIBERSEGURIDAD

19/01/2026

Tabla de contenido

OWASP (Open Web Application Security Project)

OWASP es un proyecto de **código abierto** el cual su función consta de determinar y compartir las **causas** que hacen que el **software** sea **inseguro**.

La **fundación OWASP** es una fundación sin ánimo de lucrio que apoya y gestiona los proyectos e infraestucturas de OWASP.

Projectos en los que trabaja OWASP

Como organización OWASP trabaja en diversos projectos relacionados con la ciberseguridad y la seguridad informática colaborando asi en la difusión y elaboración de sitios web seguros.

Dentro de los diferentes proyectos en los que trabaja OWASP se pueden categorizar en 3 bloques bien diferenciados:

- **Proyectos emblemáticos**
- **Proyectos de producción**
- **Otros proyectos**

Proyectos emblemáticos

- **OWASP Amass**: Framework que ayuda a los profesionales de la seguridad de la información a realizar el mapeo de redes de las superficies de ataque y descubrimiento de activos externos.
- **Norma de verificación de seguridad de aplicaciones:** Marco de requisitos de seguridad que se centrar en definir los controles de seguridad requeridos al diseñar, desarrollar y probar aplicaciones web y servicios web modernos.
- **OWASP Defectdojo:** Herramienta líder de gestión de vulnerabilidades de aplicaciones de código abierto creada para DevOps y la integración continua de seguridad.

Proyectos de producción:

- **Proyecto de seguridad de la API de OWASP**: El proyecto API Security se centra en estrategias y soluciones para comprender y mitigar las vulnerabilidades únicas y los riesgos de seguridad de las interfaces de programación de aplicaciones (API).
- **Firewall de aplicaciones web de OWASP Coraza**: OWASP Coraza es un marco WAF de nivel empresarial de golang compatible con Modsecurity y OWASP Core Ruleset.

OWASP TOP 10 2025

El OWASP Top 10 es un documento de concienciación estándar para desarrolladores y seguridad de **aplicaciones web**. Representa un amplio consenso sobre los **riesgos de seguridad más críticos para las aplicaciones web.**

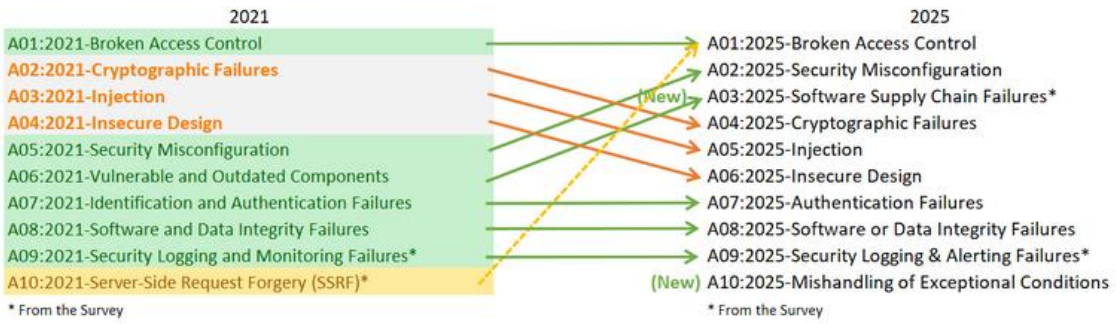
Este 2025 será la **octava entrega** de dicho documento por parte de OWASP.

Presentación del TOP 10 2025 OWASP:

1. Control de acceso roto.
2. Configuración incorrecta de seguridad.
3. Fallas en la cadena de suministro del software.
4. Fracasos criptográficos.
5. Inyección.
6. Diseño Inseguro.
7. Fallas de autenticación.
8. Fracasos de integridad de software o datos.
9. Registro de seguridad y fallas de alertas.
10. Mal manejo de condiciones exceptionales.

Cambios producidos en el Top 10 para 2025:

Se han agregado **dos nuevas categorías** y una consolidación para el top 10 en este 2025, también se ha trabajado en mantener el **enfoque** en el **problema raíz** categorizando así las diferentes vulnerabilidades que aparecen en el top 10.



La siguiente representa los cambios realizados este 2025 a diferencia del 2021 en el top 10.

Análisis detallado Top 10 OWASP:

En el siguiente apartado vamos a ver un **análisis detallado** de cada una de las vulnerabilidades que componen el top 10 de OWASP para este 2025.

A01:2025 – Control de acceso roto

Manteniendo su posición en el top 1 respecto al pasado año. Los datos aportados indican que el promedio, el **3,73%** de las aplicaciones probadas tenían una o mas de las 40 vulnerabilidades comunes en cuanto a esta categoría se refiere, entre estas dichas vulnerabilidades comunes encontramos:

- **CWE-200:** Exposición de información confidencial a un actor no autorizado.
- **CWE-201:** Exposición de información confidencial a través de datos enviados.
- **CWE-918 (SSRF):** Falsificación de solicitudes del lado del servidor.
- **CWE-352**: Falsificación de petición en sitios cruzados.

COMO PREVENIR: El control de acceso solo es efectivo cuando se implementa en código del lado del servidor o en API sin servidor de confianza, donde el atacante no puede modificar la comprobación del control de acceso o los metadatos.

- Implementar los mecanismos de control de acceso una sola vez y reutilizarlos a lo largo de toda nuestra aplicación web.
- Desactivar la lista de directorios del servidor web y asegurarnos que los **metadatos** y los archivos, por ejemplo **“.git”** junto con los archivos de **copia de seguridad** no estén presentes dentro de las raíces de la web.
- Desactivar los identificadores de sesión deben invalidarse en el servidor después de la conexión.
- Utilizar frameworks o patrones bien establecidos que proporcionen controles de acceso simples y declarativos.

EJEMPLO DE ATAQUE:

Cuando la aplicación utiliza datos no verificados en una llamada SQL que esta accediendo a la información de la cuenta:

```
pstmt.setString(1, request.getParameter("acct"));
ResultSet results = pstmt.executeQuery( );
```

El atacante puede simplemente modificar el parámetro ‘acct’ del navegador para enviar cualquier número de cuenta deseado.

```
https://example.com/app/accountInfo?acct=notmyacct
```

Página Control de Acceso OWASP

A02:2025 – Configuración incorrecta de seguridad

Subiendo del puesto número 5 en el año 2021 al puesto número 2 en este 2025. Las configuraciones erróneas son más frecuentes en este 2025 ya que los datos aportados indican que un 3% de las aplicaciones probadas tenían una o más de las 40 vulnerabilidades comunes:

- **Configuración CWE-16.**
- **CWE-611: Restricción incorrecta.**

Esto se produce cuando un sistema, aplicación o servicio en la nube se configura incorrectamente desde una perspectiva de seguridad, creando vulnerabilidades.

COMO PREVENIR: Se deben de implementar procesos de instalación seguros que incluyan:

- Configuración de los entornos de desarrollo, control de calidad y producción de manera idéntica con diferentes credenciales utilizadas en cada entorno.
- Retirar o eliminar los marcos no utilizados.
- Una arquitectura de aplicación segmentada proporciona una separación efectiva y segura entre componentes.
- Agregar de forma proactiva una configuración central para interceptar mensajes de error.

EJEMPLOS DE ATAQUE:

El servidor de aplicaciones viene con aplicaciones de ejemplo no eliminadas del servidor de producción, estas aplicaciones de ejemplo vienen con defectos de seguridad que los atacantes utilizan para comprometer el servidor. El atacante encuentra y descarga las clases Java compiladas que luego utiliza para hacer ingeniería inversa.

Página Configuración incorrecta OWASP

A03:2025 – Fallos en la cadena de suministro del software

Esta fue la vulnerabilidad mejor clasificada en la encuesta de la comunidad TOP 10 con un porcentaje del 50%. Apareciendo por primera vez en el top 10 de 2013 como A9.

El uso de componentes con vulnerabilidades conocidas ha crecido en cuento al riesgo ya que incluye todos los fallos producidos en la cadena de suministro. Entre las vulnerabilidades más comunes que afectan a este término tenemos:

- **CWE-477:** Uso de una función obsoleta.
- **CWE- 1104:** Uso de componentes de terceros no mantenidos.
- **CWE-1329:** Dependencia de componentes que no es actualizable.
- **CWE- 1395:** Dependencia de componentes de terceros vulnerable.

Los fallos de la cadena de suministro de software son averías u otros compromisos en el proceso de construcción, distribución o actualización del software.

COMO PREVENIR: Para poder prevenirlo debe de haber un proceso de gestión de parches para:

- Generar y gestionar de forma centralizada la lista de materiales del software.
- Rastrear tanto las dependencias directas como de terceros o transitivas.
- Reducir la superficie de ataque eliminado dependencias no utilizadas.
- Obtener únicamente componentes de fuentes oficiales.

EJEMPLOS DE ATAQUE:

Un proveedor de confianza se ve comprometido con el malware, lo que lleva a que sus sistemas informáticos se vean comprometidos cuando se actualizan, un ejemplo claro de todo esto es:

El compromiso de 2019 de SolarWinds llevó a que ~ **18,000 organizaciones se vieran comprometidas**

https://www.npr.org/2021/04/16/985439655/a-worst-nightmare-cyberattack-the-untold-story-of-the-solarwinds-hack

Página fallos en la cadena de suministro OWASP

A04:2025 – Fallos Criptográficos

Esta vulnerabilidad desciende del puesto número dos al cuatro teniendo en cuenta la versión de 2021. Los datos aportados indican que, en promedio, el **3,80%** de las aplicaciones tienen una o más de las vulnerabilidades comunes de esta categoría.

Esta vulnerabilidad se basa en que todos los datos de tránsito deben **cifrarse en la capa de transporte (Capa 4 del OSI).**

- **CWE - 261:** Codificación débil para la contraseña.
- **CWE - 319:** Transmisión de información sensible de texto claro.
- **CWE - 322:** Intercambio de claves sin autenticación de entidad.
- **CWE - 327:** Uso de un algoritmo criptográfico roto o arriesgado.

COMO PREVENIR: Hacer lo siguiente como mínimo:

- Clasificar y etiquetar los datos procesados, almacenados o transmitidos por una aplicación.
- Almacenar las **claves mas sensibles** en un hardware o HSM basado en la nube.
- Asegurarnos de cifrar todos los datos confidenciales en reposo.
- No utilizar **protocolos no cifrados** como FTP y STARTTLS.
- Deshabilitar el almacenamiento en caché para las respuestas que contienen datos confidenciales.

EJEMPLOS DE ATAQUE:

Un sitio no utiliza **no aplica TLS** para todas las páginas o admite un cifrado débil. Un atacante monitorea el tráfico de red (por ejemplo, en una red inalámbrica insegura), degrada las conexiones de HTTPS a HTTP, intercepta solicitudes y **roba la cookie de sesión del usuario**.

El atacante **reproduce esta cookie** y secuestra la sesión (autenticada) del usuario.

Página Fallos Criptográficos OWASP

A05:2025 - Inyección

Esta vulnerabilidad desciende del puesto número 3 al 5 respecto a la versión de 2021. Esta vulnerabilidad es una de las categorías mas probadas con **el 100% de las aplicaciones probadas para algún tipo de Inyección.**

Tiene el mayor número de vulnerabilidades comunes para cada categoría: Mas de **14k CWEs para inyección SQL** y por encima de los **30k CWEs para Cross-Site-Scripting**.

- **CWE - 74:** Validación de entrada inadecuada.
- **CWE - 89:** Neutralización incorrecta de elementos especiales utilizando una consulta SQL.
- **CWE - 80:** Neutralización incorrecta de etiquetas HTML relacionadas con scripts en una página web (XSS básico).

COMO PREVENIR: Los mejores medios para prevenir la inyección requieren mantener los datos alejados lo máximo posible de los comandos y consultas.

- Una de las mejores opciones es usar una API segura, que evite usar el intérprete por completo.

Cuando no es posible separar los datos de los comandos, puede reducir las amenazas utilizando las siguientes técnicas.

- Utilizar una buena validación de entrada del lado del servidor.
- Para cualquier consulta residual, elimine los caracteres especiales utilizando sintaxis bloqueante.
- Uso constante de consultas parametrizadas para evitar la inyección SQL.

EJEMPLOS DE ATAQUE:

Una aplicación utiliza datos no confiables en la construcción de la siguiente llamada SQL vulnerable:

```
String query = "SELECT * FROM accounts WHERE custID='" +
request.getParameter("id") + "'";
```

El atacante modifica el valor del parámetro ‘id’ en su navegador para enviar “ OR ‘1’ = ‘1’ “.Por ejemplo:

```
http://example.com/app/accountView?id=' OR '1'='1
```

Esto cambia el significado de la consulta para devolver todos los registros de la tabla de cuentas. Los ataques más peligrosos podrían modificar o eliminar datos o incluso invocar procedimientos almacenados.

Pagina Inyección OWASP

A06:2025 – Diseño inseguro

Esta categoría se introdujo en **2021**, y hemos visto mejoras notables en la industria relacionadas con el modelado de amenazas y un mayor énfasis.

Esta vulnerabilidad incluye fallas en la lógica de negocio de una aplicación como por ejemplo la falta de definición de cambios de estados no deseados o inesperados.

Entre las **vulnerabilidades mas comunes** tenemos las siguientes:

- **CWE-256**: Almacenamiento no protegido de credenciales.
- **CWE- 269**: Gestión de privilegios inapropiados.
- **CWE – 434**: Carga no restringida de archivo peligroso.
- **CWE – 501**: Infracción de límites de confianza.
- **CWE – 522**: Credenciales insuficientes protegidas.

COMO PREVENIR:

- Establecer y utilizar un ciclo de vida de desarrollo seguro con los profesionales de **AppSec**.
- Establecer y utilizar una biblioteca de **patrones de diseños seguros** o componentes pavimentados.
- Escribir pruebas de unidad e integración para validar que todos los flujos críticos son resistentes al modelo de amenaza.
- Integre las comprobaciones de plausibilidad en cada nivel de su aplicación (desde **frontend** hasta **backend**).

EJEMPLOS DE ATAQUE:

ESCENARIO #1

Un flujo de trabajo de recuperación de credenciales podría incluir **“preguntas y respuestas”**, que está prohibido por NIST 800-63b, el OWASP ASVS y el OWASP Top 10.

No se puede confiar en las preguntas y respuestas como evidencia de identidad, ya que más de una persona puede conocer las respuestas.

ESCENARIO #2

El sitio web de comercio electrónico de una cadena minorista no tiene protección contra los **bots administrados por scalpers** que compran tarjetas de video de alta gama para revender en sitios web de subastas.

Esto crea una publicidad terrible para los fabricantes de tarjetas de video y los propietarios de la cadena minorista.

Página Diseño Inseguro OWASP

A07:2025 – Fallos de autenticación

Esta vulnerabilidad se produce cuando un **atacante** es capaz de **engañar a un sistema** para que reconozca a un **usuario inválido o incorrecto** como legítimo, esta vulnerabilidad está presente.

Entre las vulnerabilidades más comunes que incluye tenemos:

- **CWE – 259**: Uso de contraseña codificada.
- **CWE – 297**: Validación incorrecta del certificado con desajuste de host.
- **CWE – 287**: Autenticación incorrecta.
- **CWE – 384**: Fijación de sesión *y* uso.
- **CWE – 798**: Credenciales codificadas.

COMO PREVENIR:

- Cuando sea posible, implemente y aplique el uso de la autenticación multifactor para evitar el relleno de credenciales automatizados, la fuerza bruta y los ataques de reutilización de credenciales robados.
- No envíe ni implemente con credenciales predeterminadas, especialmente para los usuarios administradores.
- Durante la creación de nueva cuenta y los cambios de contraseña validan contra listas de credenciales violadas conocidas.
- Limite o demore cada vez más los intentos de inicio de sesión fallidos, pero tenga cuidado de no crear un escenario de denegación de servicio.

EJEMPLOS DE ATAQUE:

ESCENARIO #1

El relleno de credenciales, el uso de listas de combinaciones de nombre de usuario y contraseña conocidos, es ahora un ataque muy común. Más recientemente, se ha encontrado que los atacantes **“incrementan” o ajustan contraseñas, basadas en un comportamiento humano común.**

ESCENARIO #2

La mayoría de los ataques de autenticación exitosos se producen debido al **uso continuado** de **contraseñas** como el **único factor de autenticación**.

Web Fallos de autenticación OWASP

A08-2025 – Fallos en la integridad de datos o software

Esta categoría se centra en la **falta de mantenimiento** de los límites de confianza y **verificar la integridad del software**, el código y los artefactos de datos a un nivel más bajo que los fallos de la cadena de suministro de software.

Un ejemplo de esto es cuando una aplicación se basa en **plugins, bibliotecas o módulos de fuentes no confiables**.

Entra las vulnerabilidades mas comunes en esta categoría se incluyen:

- **CWE-502:** Deserialización de datos no confiables.
- **CWE-915:** Modificación controlada incorrectamente de atributos de objetos.

COMO PREVENIR:

- Utilizar firmas digitales o mecanismos similares para verificar que el software o los datos provienen de la fuente esperada y no se ha modificado.
- Asegurarse de que las bibliotecas y dependencias, como npm o Maven, solo consumen repositorios de confianza.
- Asegurarse de que los datos serializados no firmados o no cifrados no se reciban de clientes no confiables.

EJEMPLOS DE ATAQUE:

ESCENARIO #1

Inclusión de la funcionalidad web de una fuente no confiable: una empresa utiliza un proveedor de servicios externo para proporcionar funcionalidad de soporte. Para mayor comodidad, tiene un **mapeo DNS** para myCompany.SupportProvider.com a support.myCompany.com

Cualquier persona con acceso a la infraestructura del proveedor de asistencia **puede robar las cookies de todos los usuarios** que han visitado support.myCompany.comY realizar un ataque de secuestro de sesión.

Web Fracaso en integridad de software o datos OWASP

A09:2025 – Fallo en el login de registro y en las alertas.

Sin registro y monitoreo, los ataques y las brechas no se pueden detectar, y sin alertarlo es muy difícil responder de manera rápida y efectiva durante un incidente de seguridad.

Esta categoría incluye problemas con el manejo adecuado de la codificación de salida a los archivos de registro:

- **CWE – 532 :** La inserción de datos confidenciales en los archivos de registro.
- **CWE – 778:** Registro insuficiente.

COMO PREVENIR:

Los desarrolladores deben implementar algunos o todos los siguientes controles, dependiendo del riesgo de la aplicación:

- Asegurarse de que todos los fallos de inicio de sesión, control de acceso y validación de entrada del lado del servidor se puedan registrar con suficiente contexto de usuario para identificar cuentas sospechosas o maliciosas.
- Asegurarse de que cada parte de su aplicación que contiene un control de seguridad esté registrada, ya sea que tenga éxito o falle.
- Asegurarse de que los datos de registro estén codificados correctamente para evitar inyecciones o ataques en los sistemas de registro o monitoreo.
- Si su aplicación o sus usuarios se comportan de manera sospechosa, emita una alerta.

EJEMPLOS DE ATAQUE:

ESCENARIO #1

Una importante aerolínea india tuvo **una violación de datos que involucró más de diez años de datos personales de millones de pasajeros**, incluidos datos de pasaportes y tarjetas de crédito.

La **violación de datos** ocurrió en un **proveedor de alojamiento cloud** de terceros, que **notificó** a la aerolínea la violación **después de algún tiempo**.

Web de fallos de registro y alertas OWASP

A10:2025 – Manejo incorrecto de condiciones excepcionales

El mal manejo de las condiciones excepcionales en el software ocurre **cuando los programas no previenen, detectan y responden a situaciones inusuales e impredecibles.**

El mal manejo de las condiciones excepcionales es una nueva categoría para 2025. Esta categoría contiene **24 CWE** y se centra en el manejo indebido de errores, errores lógicos, fallas abiertas y otros escenarios.

Esta categoría tiene algunos CWE que anteriormente se asociaban con una **mala calidad del código.** CWEs notables incluidos en esta categoría:

- **CWE-209:** Generación de mensajes de error que contienen información confidencial.
- **CWE-234:** Fallo en manejar el parámetro faltante.
- **CWE-274:** Manejo inapropiado de privilegios insuficientes.

COMO PREVENIR:

Para manejar una condición excepcional correctamente debemos **planificar tales situaciones (esperar lo peor).**

Debemos **"atrapar" todos los posibles errores** del sistema directamente en el lugar donde ocurren **y luego manejarlo** (lo que significa hacer algo significativo para resolver el problema y asegurarnos de que nos recuperamos del problema).

La captura y el manejo de condiciones excepcionales aseguran que la infraestructura subyacente de nuestros programas no se deje en práctica con situaciones impredecibles.

EJEMPLOS DE ATAQUE:

ESCENARIO #1:

El agotamiento de los recursos a través del mal manejo de condiciones excepcionales (**Denegación de servicio)** podría ser causado si la aplicación detecta excepciones cuando se cargan los archivos.

ESCENARIO #2:

Exposición de datos confidenciales a través de errores de manejo incorrectos o de base de datos que revelan el error completo del sistema al usuario.

Web Mal manejo de condiciones excepcionales OWASP