**¿Qué es OWASP?**

**OWASP** (acrónimo de Open Web Application Security Project, en inglés ‘Proyecto abierto de seguridad de aplicaciones web’) es un proyecto de código abierto dedicado a determinar y combatir las causas que hacen que el software sea inseguro.

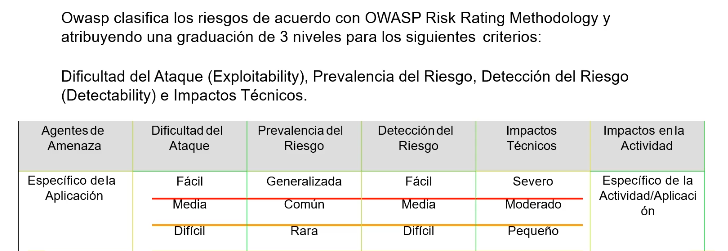
OWASP es un nuevo tipo de entidad en el mercado de seguridad informática. Estar libre de presiones corporativas facilita que OWASP proporcione información imparcial, práctica y redituable sobre seguridad de aplicaciones informáticas. OWASP no está afiliado a ninguna compañía tecnológica, si bien apoya el uso informado de tecnologías de seguridad. OWASP recomienda enfocar la seguridad de aplicaciones informáticas considerando todas sus dimensiones: personas, procesos y tecnologías.

Es un proyecto abierto y una comunidad también abierta, dedicada a analizar, a habilitar, a detectar los riesgos más famosos en nuestras aplicaciones web.

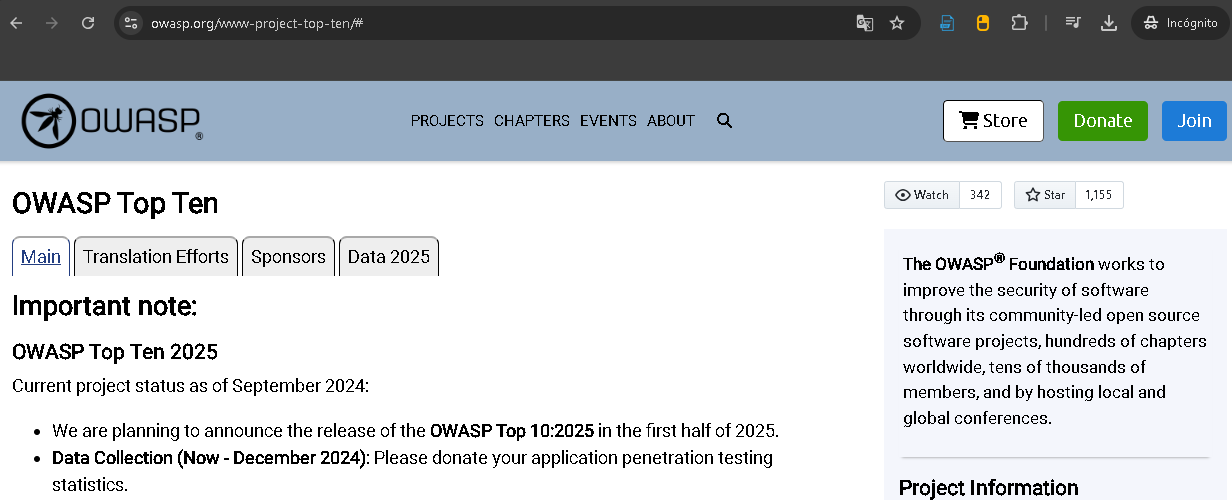
Es una comunidad sin ánimo de lucro y lo que le interesa a la comunidad es mantener sus aplicaciones web lo más confiable posible. La Fundación OWASP es un organismo sin ánimo de lucro que apoya y gestiona los proyectos e infraestructura de OWASP. La comunidad OWASP está formada por empresas, organizaciones educativas y particulares de todo mundo. Juntos constituyen una comunidad de seguridad informática que trabaja para crear artículos, metodologías, documentación, herramientas y tecnologías que se liberan y pueden ser usadas gratuitamente por cualquier persona o entidad.

**Clasificación de OWASP**

Vamos ahora a hacer una clasificación de los riesgos de nuestras aplicaciones.

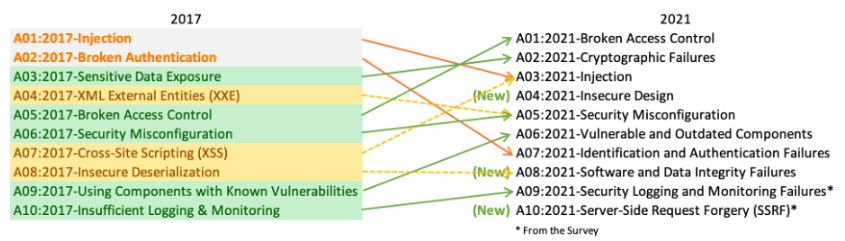


**Página oficial de OWASP**



<https://owasp.org/www-project-top-ten/>

Esta página oficial nos destaca los diez riesgos de seguridad más famosos en la época actual.



Control de Acceso Roto (Broken Access Control)

Descripción: Fallos en el control de acceso que permiten a atacantes acceder a datos o funcionalidades no autorizadas.

Ejemplo: Un usuario puede modificar la URL para acceder a la cuenta de otro usuario.

Fallos Criptográficos (Cryptographic Failures)

Descripción: Debilidades en la implementación de la criptografía que comprometen datos sensibles

Ejemplo: Una institución financiera que no protege adecuadamente la información sensible.

Inyección (Injection)

Descripción: Permitir que datos no sanitizados se interpreten como código.

Ejemplo: Un atacante inyecta código SQL malicioso a través de un formulario.

Diseño Inseguro (Insecure Design)

Descripción: Fallos en el diseño que comprometen la seguridad.

Ejemplo: Un sistema que permite reservas grupales sin límites de cantidad.

Configuración Incorrecta de Seguridad (Security Misconfiguration)

Descripción: Debilidades en la configuración que exponen la aplicación.

Ejemplo: Cuentas predeterminadas sin cambiar sus contraseñas.

Componentes Vulnerables (Vulnerable and Outdated Components)

Descripción: Uso de componentes con vulnerabilidades conocidas.

Ejemplo: Bibliotecas desactualizadas con fallos de seguridad conocidos.

Fallos de Identificación y Autenticación (Identification and Authentication Failures)

Descripción: Debilidades en el proceso de autenticación.

Ejemplo: Permitir contraseñas débiles como "password1".

Fallos de Integridad de Software y Datos (Software and Data Integrity Failures)

Descripción: Debilidades en la actualización y validación de software.

Ejemplo: Deserialización insegura de objetos proporcionados por atacantes.

Fallos de Registro y Monitoreo de Seguridad (Security Logging and Monitoring Failures)

Descripción: Falta de registro adecuado de eventos de seguridad.

Ejemplo: No registrar intentos fallidos de inicio de sesión.

Forged Request (solicitud falsificada) del Lado del Servidor (Server-Side Request Forgery)

Descripción: Permitir que un atacante fuerce al servidor a realizar solicitudes no deseadas.

Ejemplo: Un atacante usa la aplicación para acceder a recursos internos protegidos.