


Determinación del nivel de seguridad requerido por aplicaciones.



Caso práctico


Julián está en un proyecto desarrollo de software en el que uno de sus componentes es una aplicación web. **Julián** nunca había estado en un proyecto de estas características y tiene ciertas preocupaciones con software expuesto a internet, ya que sabe que un fallo no contemplado en el código puede provocar un impacto importante (indisponibilidad, ejecución de código remota, compromiso de datos personales...).

Además las librerías y el código que va a usar está público en [Github](#) , lo cual es bueno puesto que si hay alguna vulnerabilidad hay una gran comunidad de usuarios que la detectará, la reportará y ayudará a mitigar.

Sabe además que hay una fundación ([OWASP](#)) que se encarga de evaluar los riesgos que pueden sufrir estas aplicaciones expuestas a internet por lo que se apoya en estos frameworks para saber a qué se enfrenta y cómo mitigar cualquier futuro problema.



[peggy_marco para Pixabay. Imagen de App. Software y Contorno](#) (Licencia propia de Pixabay)

El desarrollo de software ha evolucionado mucho durante los últimos años, han surgido fundaciones como [OWASP](#)  cuyo objetivo es la mejora en la seguridad del software mediante la identificación de las principales vulnerabilidades existentes y como poder afrontarlas.

También ha evolucionado mucho el desarrollo de software a nivel de exposición, ya que a día de hoy muchas de las líneas de código de las aplicaciones que vemos en internet están disponibles en repositorios públicos, lo cual aporta grandes beneficios como veremos más adelante pero también añade una exposición adicional al código, que cómo veíamos en unidades anteriores hay una gran relación entre el grado de exposición y el número de vulnerabilidades.



[Ministerio de Educación, Formación Profesional y Deportes](#) (Dominio público)

Materiales formativos de FP Online propiedad del Ministerio de Educación, Formación Profesional y Deportes.

[Aviso Legal](#) 

1.- Fuentes abiertas para desarrollo seguro.

Cuando hablamos de software de fuentes abiertas nos referimos a aquel software que podemos encontrar de fácil acceso, publicado y por lo general libre (el código fuente se pone a disposición del público para ser usado y modificado conforme los usuarios o desarrolladores puedan aportar).

Durante los últimos años ha habido una evolución clara, ya que hasta ahora el software que usábamos tanto a nivel doméstico como profesional era software cerrado o propietario, es decir, aquel en el que no se permite el acceso al código fuente.

Estos últimos años se ha demostrado que el software de fuentes abiertas aporta una serie de ventajas tanto a usuarios finales como a empresas y corporaciones de forma clara:

- ✓ **Estabilidad:** al ser el acceso libre, los usuarios y programadores expertos pueden revisar el código y ayudar a mejorarlo, mejorando por tanto la estabilidad del software. Si consideramos que hay sistemas operativos abiertos hace que estos sistemas tengan comunidades detrás que los mejoren día tras día. Mayor estabilidad es también software más seguro, que coteje mejor los datos con los que trabaja la aplicación, por tanto afecta tanto al usuario final como las empresas desarrolladoras.
- ✓ **Coste:** cuando hay una comunidad tan importante de usuarios detrás del código, los costes de desarrollo del mismo se reducen, ya que la comunidad de usuarios interactuando con el código reduce los costes de desarrollo, depuración, testeo, etc.
- ✓ **Favorece la innovación colectiva** ya que incentiva la acción de la comunidad de usuarios detrás del código, revisando, mejorándolo.
- ✓ **Mejoras:** muchos usuarios cuando interactúan con el código lo adaptan y modifican dándole nuevos usos y capacidades, por lo tanto al nivel el software original es depurado y potenciado.

La idea y el movimiento detrás del software de fuentes abiertas ha sido aceptada, aplicada, promovida y difundida por todo el mundo. Haciendo que muchas empresas de desarrollo de software lo hayan incorporado como parte de su ADN y por tanto el paradigma a nivel de desarrollo de software ha cambiado de forma radical. Como hemos comentado, al final este proceso no solamente beneficia a los usuarios finales y empresas sino también



@weareprocreator para Unsplash. Hombre usando laptop (Licencia Unsplash)



Para saber más

El incremento del uso de software abiertas ha sido un autentico cambio de paradigma y, en el vídeo que presentamos a continuación (está en inglés), de la prestigiosa cadena de televisión CNBC, explican todas las causas de este fenómeno y las ventajas que este modelo ha aportado, convirtiéndose casi en el estándar actual para muchas aplicaciones, empresas y servicios.

<https://www.youtube.com/embed/SpeDK1TPbew?si=JM9HaCh8ZFBUCio0>



2.- Listas de riesgos de seguridad habituales.

El riesgo asociado con el uso de software se puede describir conceptualmente de la siguiente manera: un agente de amenaza (**threat agent**, en ingles) interactúa con un sistema (**system**), que puede tener un vulnerabilidad (**vulnerability**) que puede ser explotada (**exploited**) para causar un impacto (**impact**). Un ejemplo podría ser un ladrón (agente de amenaza) pasea por un gimnasio revisando las taquillas de los clientes (el sistema) en busca de taquillas sin candado (la vulnerabilidad) y cuando encuentran una, abre la puerta sin candado (el exploit) y toma lo que sea que haya dentro (el impacto).

Toda aplicación/sistema puede tener fallos que pueden ser explotados, pero detectar y corregir los mismos no es tarea simple. Existen diferentes organismos/empresas que se dedican a detectar y plantear soluciones a las vulnerabilidades que se van detectando. Algunas de las más importantes son:

- **NVD** (La base de datos nacional de vulnerabilidades de E.E.U.U): realiza la recopilación de datos de gestión de vulnerabilidades basados en estándares operados por el gobierno federal.
- La 🇺🇸 Corporación Mitre y el 🇺🇸 Instituto SANS, junto con otros expertos mundiales, mantienen el sistema de categorías **CWE** que cataloga las diversas vulnerabilidades.
- **OWASP** (Proyecto Abierto de Seguridad de Aplicaciones Web), de la que veremos diversos proyectos en profundidad, se encarga de determinar y combatir las causas de que el software no sea seguro.



[incibe.es](https://www.incibe.es) (Captura Pantalla)

En España tenemos el **INCIBE**. Como indica en su página web: "*Trabaja para afianzar la confianza digital, elevar la ciberseguridad y la resiliencia y contribuir al mercado digital de manera que se impulse el uso seguro del ciberespacio en España*". Te recomiendo visitar su página [web](https://www.incibe.es) 📄 en profundidad.

OWASP Foundation

La Fundación OWASP es un organismo sin ánimo de lucro, cuyo único objetivo es trabajar en mejorar la seguridad del software. A través de proyectos de software de código abierto (y fuentes abiertas) liderados por la comunidad, decenas de miles de miembros, conferencias educativas y cientos de cursos anuales de capacitación líderes, la Fundación OWASP es uno de los principales motores en la mejora del software en entornos web.

Uno de los proyectos desarrollados por esta organización es el Top10 de riesgos a nivel software en entornos web y el Top10 de riesgos para entornos móviles. El proceso para confeccionar esta lista no es un proceso sencillo ya que se tienen en cuenta y se evalúan distintos parámetros como las vulnerabilidades, ataques conocidos, impactos, manejo de los datos, etc.

Who is the OWASP® Foundation?

The Open Web Application Security Project® (OWASP) is a nonprofit foundation that works to improve the security of software. Through community-led open-source software projects, hundreds of local chapters worldwide, tens of thousands of members, and leading educational and training conferences, the OWASP Foundation is the source for developers and technologists to secure the web.



- Tools and Resources
- Community and Networking
- Education & Training

[OWASP.org](https://owasp.org) (Captura Pantalla)


Top 10 vulnerabilidades web

En la última versión, de 2021, el Top 10 de vulnerabilidades web es (ver [enlace](#) 📄):

- A1 - Pérdida de Control Acceso
- A2 - Fallos Criptográficos

- A3 - Inyección de código
- A4 - Diseño Inseguro
- A5 - Problemas de configuración a nivel de seguridad
- A6 - Componentes vulnerables y obsoletos
- A7 - Fallos de Autenticación e Identificación
- A8 - Fallos de Software y de Integridad de los Datos
- A9 - Fallos en el registro (logging en inglés) y Monitorización de la Seguridad
- A10 - Falsificación de peticiones en el servidor (Server-Side Request Forgery en inglés)

Top 10 vulnerabilidades móviles


En cuanto a las aplicaciones móviles, la última versión de 2023 que esta a punto de publicarse, ver [enlace](#)  el Top 10 es:

- M1: Uso inadecuado de credenciales
- M2: Seguridad inadecuada de la cadena de suministro (Inadequate Supply Chain Security en inglés)
- M3: Autenticación/autorización insegura
- M4: Insuficiente validación de entrada/salida
- M5: Comunicación insegura
- M6: Controles de privacidad inadecuados
- M7: Protecciones binarias insuficientes
- M8: Configuración errónea de la seguridad (Security Misconfiguration en inglés)
- M9: Almacenamiento de datos inseguro
- M10: Criptografía insuficiente

En las unidades 3 y 4 veremos estas vulnerabilidades en detalle.



Debes conocer

OWASP no solamente trabaja en identificar riesgos a nivel de aplicaciones web y móviles, también tiene otros proyectos muy interesantes. Para aprender más sobre sus otros proyectos pulsa en el [enlace](#) 



Autoevaluación

Identifica si las siguientes frases son verdaderas o falsas

OWASP es una organización que busca obtener beneficios económicos mejorando el software.

☐ Verdadero ☐ Falso

Falso

OWASP es una organización sin ánimo de lucro

El primer riesgo detectado por OWASP es un diseño inseguro

☐ Verdadero ☐ Falso

Falso

Si bien OWASP valora el diseño seguro dentro de su Top10, el primer lugar de la lista es el control del acceso.

OWASP valora vulnerabilidades y sobretodo el acceso a los datos que pudieran provocar.

☐ Verdadero ☐ Falso

Verdadero

Para la elaboración de la lista de OWASP intervienen miles de programadores, empresas y expertos en seguridad y desarrollo.

☐ Verdadero ☐ Falso

Verdadero

2.1.- Vulnerabilidades y exposiciones comunes

Las Vulnerabilidades y exposiciones comunes (**CVE** por sus siglas en inglés), es una lista de información registrada sobre vulnerabilidades de seguridad conocidas, en la que cada referencia tiene:


- ✓ Un número de identificación **CVE-ID**.
- ✓ Descripción de la vulnerabilidad.
- ✓ Que versiones del software están afectadas.
- ✓ Posible solución al fallo (si existe) o como configurar para mitigar la vulnerabilidad.
- ✓ Referencias a publicaciones o entradas de foros o blog donde se ha hecho pública la vulnerabilidad o se demuestra su explotación.

Además, suele también mostrarse un enlace directo a la información de la base de datos de vulnerabilidades (NVD) del **NIST**, en la que pueden conseguirse más detalles de la vulnerabilidad y su valoración.

El **CVE-ID** ofrece una nomenclatura estándar para identificación de la vulnerabilidad de forma inequívoca que es usada en la mayoría de repositorios de vulnerabilidades Normalmente, el formato es **CVE-YYYY-NNNN**, donde YYYY es el año y NNNN un número generado aleatoriamente.



Debes conocer

La lista CVE es definida y mantenida por The MITRE Corporation con fondos de la National Cyber Security Division del gobierno de los Estados Unidos de América. Forma parte del llamado Security Content Automation Protocol. Mira su página web en este [enlace](#)  y verás que sencillo es localizar las vulnerabilidades.


3.- Comprobaciones de seguridad a nivel de aplicación.

La fundación OWASP, como hemos visto, no solamente ayuda a disponer de un software más seguro mediante la identificación de los principales riesgos a los que estamos expuestos, sino que también ha arrancado otros proyectos, entre los que destacan una serie de **guías y herramientas para la construcción y mantenimiento de aplicaciones seguras**. Por ejemplo:

- ✓ La guía de desarrollo (en inglés, **Development Guide**) muestra cómo diseñar y construir una aplicación segura.
- ✓ La guía de verificación de seguridad de la aplicación o **ASVS** muestra cómo verificar la seguridad de la misma.
- ✓ La guía de pruebas o **WSTG** muestra cómo verificar la seguridad de una aplicación web.



[Geralt para Pixabay](#), [gente-google-polaroid-pinterest](#)
(Licencia Propia de Pixabay)

El proyecto del **Estándar de Verificación de Seguridad de Aplicaciones** (ASVS de sus siglas en inglés) de OWASP proporciona información para poder probar los controles técnicos de seguridad de las aplicaciones web y también proporciona una lista de requisitos para un desarrollo correcto y seguro, minimizando el impacto de los riesgos anteriormente detectados. Para ver en detalle este proyecto pulsa en el siguiente [enlace](#)  .

ASVS tiene dos objetivos principales:

- ✓ Ayudar a las organizaciones en el desarrollo y mantenimiento aplicaciones seguras.
- ✓ Permitir la alineación entre las necesidades y ofertas de los servicios de seguridad, proveedores de herramientas de seguridad y consumidores.

Los requisitos se desarrollaron con los siguientes objetivos en mente:

- ✓ Usarlo como **métrica**, proporcionando a los desarrolladores y propietarios de aplicaciones un criterio con el que evaluar el grado de confianza que se puede depositar en sus aplicaciones web.
- ✓ Utilizarlo como **guía**, proporcionando orientación a los desarrolladores de controles de seguridad sobre qué incorporar en los controles de seguridad para satisfacer los requisitos de seguridad de las aplicaciones.
- ✓ Usarlo durante la **compra de software** como **requisito**, proporcionando una base para especificar los requisitos de verificación de seguridad de la aplicación en los contratos.



Debes conocer

OWASP desarrolla el **MASVS** o Estándar de Verificación de Seguridad de Aplicaciones Móviles, que es un equivalente al ASVS pero para aplicaciones móviles. Dicho proyecto ha proporcionado tradicionalmente tres niveles de verificación (L1, L2 y R), pero en la última actualización de 2023 han sido reformulados como "perfiles de pruebas de seguridad" y trasladados al OWASP **MASTG** (en inglés Mobile Application Security Testing Guide).

Para saber más sobre **MASVS** visita el siguiente [enlace](#)  .



Para saber más

Aunque la versión 5.0 ha sido anunciada, la última versión del estándar ASVS es la versión 4.0.3. Esta versión es la que se ha utilizado para crear los contenidos de los apartados 3 y 4 de esta unidad, y que puedes encontrar en el PDF que encontrarás en el siguiente [enlace](#)



3.1.- Niveles de seguridad ASVS

El **ASVS versión 4.0.3** define 3 niveles de Seguridad:

👉 **ASVS Nivel 1 (Opportunistic):**

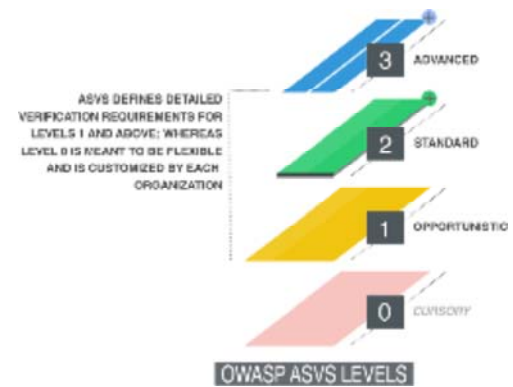
- Deben cumplirlo todas las aplicaciones.
- Contiene 136 controles.
- Una aplicación alcanza ASVS Nivel 1 si logra defenderse contra vulnerabilidades de seguridad de aplicaciones que son fáciles de descubrir, incluido el Top 10 de OWASP y otras listas de comprobación similares.

✔ ASVS Nivel 2 (Standard)

- Deben cumplirlo las aplicaciones que contienen datos sensibles, y es recomendable para todas las aplicaciones.
- Contiene 267 controles
- Una aplicación alcanza ASVS Nivel 2 (o Estándar) si se defiende adecuadamente contra la mayoría de los riesgos asociados con el software hoy en día.

✔ **ASVS Nivel 3 (Advanced).**

- Deben cumplirlo aplicaciones críticas por el alto valor de las transacciones (por ejemplo, la bolsa), o por tratar datos especialmente protegidos (por ejemplo, datos médicos) o cualquier aplicación que requiera el más alto nivel de confianza.
- Contiene 286 controles.
- Una aplicación alcanza ASVS Nivel 3 (o Avanzado) si se defiende adecuadamente contra vulnerabilidades avanzadas de seguridad de aplicaciones y también demuestra principios de buen diseño de seguridad.



[OWASP](#). [OWASP ASVS LEVELS](#) (CC BY-SA)

4.- Requisitos de verificación necesarios asociados al nivel de seguridad establecido.

Como hemos visto cada nivel ASVS versión 4.0.3 contiene una lista de requisitos de seguridad. Cada uno de estos requisitos también se puede asignar a características y capacidades específicas de seguridad que los desarrolladores deben incorporar al software

Cada requisito tiene un identificador en el formato (cada elemento es un número):

v<version>-<capítulo>.<sección>.<requisito>

- ✓ Por ejemplo: **v4.0.2-1.11.3** corresponde a Versión de ASVS 4.0.2
- ✓ El capítulo 1 es de Arquitectura.
- ✓ La sección 11 del capítulo 1: 1.11. es la sección Requisitos de arquitectura de la lógica de negocio del capítulo Arquitectura.
- ✓ El requisito 3 de la sección 11 del capítulo 1, 1.11.3 de esta norma es: Verificar que todos los flujos de lógica de negocio de alto valor, incluyendo la autenticación, la gestión de la sesión y el control de acceso son seguros para los hilos y resistentes a las condiciones de carrera de tiempo de verificación y tiempo de uso.



[thedigitalway para Pixabay. computadora-seguridad-candado](#) (Propia de Pixabay)

El listado de controles generales en ASVS versión 4.0.3 es:

- ✓ V1. Arquitectura, diseño y modelado de amenazas
- ✓ V2. Autenticación
- ✓ V3. Gestión de sesiones
- ✓ V4. Control de acceso
- ✓ V5. Validación, sanitización y codificación
- ✓ V6. Criptografía en el almacenamiento
- ✓ V7. Gestión y registro de errores
- ✓ V8. Protección de datos
- ✓ V9. Comunicaciones
- ✓ V10. Código Malicioso
- ✓ V11. Lógica de negocio
- ✓ V12. Archivos y recursos
- ✓ V13. Servicios Web y API
- ✓ V14. Configuración

Dentro de cada uno de estos capítulos encontraremos secciones con aspectos mas específicos dentro de ese control. Es importante que entiendas no solamente los controles en sí, sino que nivel se necesita cumplir, según el tipo de aplicación que se este desarrollando. Por otra parte el nivel 1 se considera el mínimo que debería de cumplir cualquier aplicación, por lo que es importante que te familiarices con este nivel.



Reflexiona

Es interesante que entiendas la **relación entre el dato y el riesgo**, es decir, cuando nuestra aplicación maneje *datos sensibles, personales o transacciones económicas*, el **nivel del ASVS subirá** y por tanto los controles también lo harán. Por tanto, cuando un programador o empresa de software recibe el encargo de realizar una aplicación que vaya a manejar este tipo de datos, es importante que entienda que deberá seguir unos **controles rigurosos** y

que, entonces, los **tiempos, costes y ciclos de desarrollo y pruebas** variarán.



[storyset para Freepik](#). [Gestión de riesgos](#) (Propia de Freepik)

