Evaluación de seguridad en aplicaciones web

Dirección General de Sistemas de Información y Comunicaciones

Áreas de Gobierno Tecnológico de SSII y del Dato

Servicio Andaluz de Salud
Consejería de Sanidad, Presidencia y Emergencias

Contenido

Resumen

- **Versión:** v01r01
- **Fecha publicación:** 13 de noviembre de 2024
- **Entrada en vigor desde:** 13 de noviembre de 2024
- **Contacto Dpto:** **Oficina de Calidad**

Seguridad Dinámicas (SPSD)

- Informe de resultados

Histórico de cambios

Los cambios en la normativa vendrán acompañados de un registro de las modificaciones. De este modo se podrá realizar un seguimiento y consultar su evolución.

Versión	v01r01	**Fecha publicación**	13 de noviembre de 2024	**Fecha entrada en vigor**	13 de noviembre de 2024
Alcance					
• Primera versión publicada					

Introducción

El objetivo de este artículo es establecer un método y unos criterios de evaluación para la revisión de la seguridad en las aplicaciones web desarrolladas por los proveedores de la STIC. Pretende servir de guía para la realización de las pruebas de seguridad de las aplicaciones y servicios web.

Los proveedores deberán tener en cuenta el Marco Normativo de Seguridad TIC y Protección de Datos publicado por la Unidad de Seguridad TIC en las aplicaciones software desarrolladas para la STIC para que sea posible la promoción de las aplicaciones a entornos productivos.

Pruebas de seguridad web

Las pruebas de seguridad son un conjunto de actividades destinadas a identificar fallos y vulnerabilidades en las aplicaciones web. Su finalidad es descubrir y mitigar la mayoría, si no todas, de las posibles vulnerabilidades para garantizar la protección de la aplicación ante posibles amenazas. Estas pruebas son cruciales para salvaguardar la confidencialidad, disponibilidad e integridad de los datos y funciones que maneja el software.

> **Confidencialidad**: Asegurar que la información sensible esté protegida contra accesos no autorizados.
>
> **Disponibilidad**: Garantizar que los servicios y funciones de la aplicación estén disponibles para los usuarios cuando los necesiten.
>
> **Integridad**: Proteger la exactitud y consistencia de los datos contra modificaciones no autorizadas.

Las pruebas de seguridad deben llevarse a cabo en todas las fases del ciclo de desarrollo del software. Esto asegura que las vulnerabilidades sean identificadas y abordadas desde las etapas iniciales, reduciendo así el riesgo de problemas de seguridad graves en las fases posteriores.

OWASP (Open Web Application Security Project)

Tal y como indica su nombre, OWASP es un proyecto de código abierto dedicado a determinar y combatir los problemas de seguridad en el software. Constituidos como fundación desde 2004, su labor se centra en proporcionar metodologías, documentación y herramientas de forma gratuita que contribuyan a la mejora de la seguridad de las aplicaciones web.

La metodología de pruebas de OWASP se basa en las distintas categorías de las vulnerabilidades de las aplicaciones web y sobre esto se construye el marco que se presenta en la guía. El enfoque de OWASP, además de ser orientada a las pruebas de seguridad en todo el ciclo de desarrollo, también se centra en las pruebas de intrusión o caja negra, que es la manera en la que la mayoría de los atacantes puede tener acceso a las aplicaciones.

En la realización de las pruebas, se dará prioridad a la detección de vulnerabilidades según la escala Top 10 de vulnerabilidades de aplicaciones publicada por OWASP en 2021.

Tareas
A1 – Pérdida de Control de Acceso
A2 – Fallas Criptográficas
A3 – Inyección
A4 – Diseño Inseguro
A5 – Configuración de Seguridad Incorrecta
A6 – Componentes Vulnerables y Desactualizados
A7 – Fallas de Identificación y Autenticación
A8 – Fallas de Software y en la Integridad de los Datos

A9 – Fallas en el Registro y Monitoreo
A10 – Falsificación de Solicitudes del Lado del Servidor

Las pruebas se encontrarán divididas en dos grandes bloques:

- Pruebas estáticas (SAST) **es un método de prueba de caja blanca, donde se examina el código fuente** para encontrar fallos y debilidades del software, como las inyecciones de SQL y otras que figuran en el Top 10 de OWASP.
- Pruebas dinámicas (DAST) **son una metodología de pruebas de caja negra que examinan una aplicación mientras se ejecuta**, para encontrar vulnerabilidades que un atacante podría aprovechar.

Análisis estático de la seguridad (SAST)

El análisis estático evalúa el código de una aplicación sin ejecutarlo. Su propósito es identificar problemas de seguridad, rendimiento y sintaxis en la codificación, con el fin de prevenir futuros problemas en el software.

Las herramientas SAST (Static Application Security Testing) se integran en el pipeline CI/CD, revisando el código en busca de patrones y flujos que determinen vulnerabilidades, malas prácticas de programación y otros problemas potenciales. A diferencia del análisis dinámico, que implica la ejecución del código, el análisis estático se realiza en el código tal como está escrito, permitiendo identificar problemas antes de que el programa se ejecute.

Durante el desarrollo de una aplicación, es común emplear librerías y dependencias que son fundamentales para crear el producto que se desea. Estas librerías suelen ser de terceros y, en muchas ocasiones, suponen un 60% o más el uso de ellas en los programas. Las herramientas SAST analizan las dependencias de estas librerías en busca de vulnerabilidades, conectándose a bases de datos que proporcionan la información necesaria al respecto.

El Servicio Andaluz de Salud utiliza Sonarqube para el análisis estático de código, integrado en el ciclo CI/CD.

La versión 9.9.2 de Sonarqube, actualmente instalada, incorpora la metodología Clean as You Code, que promueve un enfoque más eficiente para la entrega de un "código limpio". La adopción de esta metodología conlleva la incorporación de una nueva métrica (Security Hotspots Reviewed) dentro de los umbrales de calidad. En materia de seguridad, Clean as You Code garantiza que:

- No se introduce ninguna nueva vulnerabilidad.
- Se revisan todos los nuevos *"security hotspots"*.

A partir de la publicación de esta normativa, los problemas de seguridad tendrán un tratamiento diferencial dentro de la revisión que realiza Sonarqube de la calidad estática del código.

Análisis Dinámico de Seguridad (DAST)

Las herramientas DAST (Dynamic Application Security Testing) son tecnologías diseñadas para detectar vulnerabilidades de seguridad en aplicaciones web en ejecución. Estas herramientas se comunican con la aplicación a través del front-end web y realizan pruebas de caja negra, revelando debilidades de seguridad sin necesidad de comprender o visualizar la arquitectura o el código fuente interno de la aplicación.

DAST lleva a cabo escaneos complejos para identificar una amplia gama de fallos y prevenir brechas de seguridad, tales como ataques distribuidos de denegación de servicio (DDoS), secuencias de comandos entre sitios (XSS), inyecciones de SQL, pruebas de penetración y/ o pruebas de API. Se utiliza al final del ciclo de vida de desarrollo de software, ya que requiere una versión en ejecución de la aplicación para comenzar su análisis.

Procedimiento para la ejecución de pruebas de seguridad dinámicas (DAST)

En nuestro caso, las pruebas se ejecutarán de forma automática en entornos de preproducción. El plan de pruebas será el que nos proporciona OWASP ZAP (Open Web Application Security Project - Zed Attack Proxy).

Las pruebas de seguridad dinámicas se lanzarán bajo dos supuestos:

1. Como parte del procedimiento de seguridad de la USTIC para la implantación de nuevas aplicaciones.
2. A demanda de la dirección de proyectos, solicitando en Jira la ejecución de un ***"Servicio de Pruebas de Seguridad Dinámicas" (SPSD)***, uno de los que actualmente ofrece la Oficina de Calidad.

Descripción del Servicio de Pruebas de Seguridad Dinámicas (SPSD)

Este servicio está incluido en el catálogo que ofrece la Oficina de Calidad. Como el resto de los Servicios, quedará documentado en Jira, la herramienta de gestión de proyectos utilizada por la STIC.

La información de entrada necesaria para la ejecución de las pruebas:

- Entorno de ejecución.
- Conjunto de pruebas:
    - Análisis Base de Seguridad. Este tipo de pruebas realizará un análisis de las alertas OWASP mediante un escaneo pasivo. Actualmente, son 51 pruebas, aunque este número podrá variar con las actualizaciones de OWASP.

      El objetivo principal de las alertas pasivas es proporcionar una evaluación no intrusiva de la seguridad de una aplicación web. Al monitorear el tráfico que fluye entre el cliente y el servidor, OWASP Zap puede identificar posibles problemas de seguridad, como la exposición de información confidencial en respuestas HTTP, el uso de prácticas inseguras de codificación y la falta de encabezados de seguridad HTTP, entre otros, sin interrumpir el servicio ni perturbar la operación normal de la aplicación.

Alertas/pruebas de escaneo pasivo

Alerta (Nomenclatura OWASP) - Prueba
[90030] - Detección de archivos WSDL

[10098] - Desconfiguración entre Dominios
[10097] - Divulgación de "Hash"
[10096] - Divulgación de Fecha y Hora
[10062] - Divulgación de la IIP
[10061] - Cabecera de Respuesta X-AspNet-Version
[10057] - Se ha encontrado un "Hash" de Nombre de Usuario
[10056] - Fuga de Información de X-Debug-Token
[10055] - Política de Seguridad de Contenidos (CSP)
[10054] - Cookie sin atributo "SameSite"
[10052] - Fuga de Información de la Cabecera "X-ChromeLogger-Data" (XCOLD)
[10050] - Recuperado de la Caché
[10044] - Gran Redirección Detectada (posible fuga de información sensible)
[10043] - Evento JavaScript (XSS) Controlable por el Usuario
[10042] - HTTPS a HTTP Transición Insegura en el Formulario Post
[10041] - HTTP a HTTPS Transición Insegura en el Formulario Post
[10040] - Las Páginas Seguras Incluyen Contenido Mixto
[10039] - Fuga de Información de la Cabecera del Entorno del Servidor X
[10038] - Encabezado de Directiva de Seguridad de Contenido (CSP) no establecido
[10037] - El Servidor filtra Información a través del Campo de Encabezado de Respuesta HTTP 'X-Powered-By'
[10036] - Cabecera de Respuesta del Servidor HTTP
[10035] - Cabecera Seguridad de Transporte Estricta (HSTS).
[10034] - Vulnerabilidad de "Heartbleed" OpenSSL (Indicativo)
[10033] - Exploración de Directorios
[10032] - Viewstate
[10031] - Atributo de Elemento HTML Controlable por el Usuario (XSS Potencial)
[10030] - Conjunto de Caracteres Controlables por el Usuario
[10029] - Envenenamiento de Cookies
[10028] - Redirección Abierta
[10027] - Revelación de Información: Comentarios Sospechosos
[10025] - Revelación de Información - Información Confidencial en "HTTP Referrer Header"
[10024] - Revelación de Información: Información Confidencial en URL
[10023] - Revelación de Información: Mensajes de Error de Depuración
[10021] - Falta Cabecera X-Content-Type-Options

[10020] - Cabecera Anti-Clickjacking
[10019] - Falta Cabecera del Tipo de Contenido
[10017] - Inclusión de Archivos Fuente JavaScript entre Dominios
[10015] - Reexaminar las Directivas de Control de Caché
[90033] - Cookie de Ámbito Flexible
[90022] - Divulgación de Errores de Aplicación
[10011] - Cookie sin Bandera Segura
[10010] - Bandera Cookie No HttpOnly
[10003] - Librería Javascript Vulnerable
[90011] - Discordancia en el Conjunto de Caracteres
[90001] - ViewState de JSF Inseguro
[3] - ID de Sesión en URL reescrita
[2] - Divulgación de IP privada
[10109] - Aplicación Web Moderna
[10108] - Tabulación Inversa
[10105] - Método de Autenticación Débil
[10202] - Ausencia de Anti-CSRF Tokens

- Análisis Completo de Seguridad. Se realizará un análisis exhaustivo mediante **escaneos pasivos y activos,** incluyendo las pruebas base y un conjunto total de **99 pruebas**.

  A diferencias de las pruebas base de seguridad, éstas conllevan un mayor tiempo de ejecución.

  Escaneo activo: El objetivo de las alertas activas en OWASP Zap es simular ataques reales contra una aplicación web para identificar posibles vulnerabilidades de seguridad como inyecciones SQL, ataques de cross-site scripting (XSS), ataques de fuerza bruta, entre otros. A diferencia de las alertas pasivas, que observan el tráfico sin modificarlo, las alertas activas interactúan directamente con la aplicación al enviar solicitudes especialmente diseñadas para explorar posibles puntos débiles.

Alertas/pruebas de escaneo pasivo y activo

Alerta (Nomenclatura OWASP) - Prueba
[90026] - Suplantación de acción SOAP
[90029] - Inyección XML SOAP
[90030] - Detección de archivos WSDL
[40017] - Secuencia de Comandos Entre Sitios (XSS Persistente - Spider)
[40016] - Secuencia de Comandos Entre Sitios (XSS Persistente - Prime)
[10104] - "Fuzzer" de Agente de Usuario

[10098] - Desconfiguración entre Dominios
[10097] - Divulgación de "Hash"
[10096] - Divulgación de Fecha y Hora
[10062] - Divulgación de la IIP
[10061] - Cabecera de Respuesta X-AspNet-Version
[10058] - GET por POST
[10057] - Se ha encontrado un "Hash" de Nombre de Usuario
[10056] - Fuga de Información de X-Debug-Token
[10055] - Política de Seguridad de Contenidos (CSP)
[10054] - Cookie sin atributo "SameSite"
[10052] - Fuga de Información de la Cabecera "X-ChromeLogger-Data" (XCOLD)
[10050] - Recuperado de la Caché
[10045] - Divulgación del Código Fuente. Carpeta /WEB-INF
[10044] - Gran Redirección Detectada (posible fuga de información sensible)
[10043] - Evento JavaScript (XSS) Controlable por el Usuario
[10042] - HTTPS a HTTP Transición Insegura en el Formulario Post
[10041] - HTTP a HTTPS Transición Insegura en el Formulario Post
[10040] - Las Páginas Seguras Incluyen Contenido Mixto
[10039] - Fuga de Información de la Cabecera del Entorno del Servidor X
[10038] - Encabezado de Directiva de Seguridad de Contenido (CSP) no establecido
[10037] - El Servidor filtra Información a través del Campo de Encabezado de Respuesta HTTP 'X-Powered-By'
[10036] - Cabecera de Respuesta del Servidor HTTP
[10035] - Cabecera Seguridad de Transporte Estricta (HSTS).
[10034] - Vulnerabilidad de "Heartbleed" OpenSSL (Indicativo)
[10033] - Exploración de Directorios
[10032] - Viewstate
[10031] - Atributo de Elemento HTML Controlable por el Usuario (XSS Potencial)
[10030] - Conjunto de Caracteres Controlables por el Usuario
[10029] - Envenenamiento de Cookies
[10028] - Redirección Abierta
[10027] - Revelación de Información: Comentarios Sospechosos
[10025] - Revelación de Información - Información Confidencial en "HTTP Referrer Header"
[10024] - Revelación de Información: Información Confidencial en URL

[10023] - Revelación de Información: Mensajes de Error de Depuración
[10021] - Falta Cabecera X-Content-Type-Options
[10020] - Cabecera Anti-Clickjacking
[10019] - Falta Cabecera del Tipo de Contenido
[10017] - Inclusión de Archivos Fuente JavaScript entre Dominios
[10015] - Reexaminar las Directivas de Control de Caché
[90034] - Metadatos en la Nube Potencialmente Expuestos
[90033] - Cookie de Ámbito Flexible
[90024] - Ataque Genérico de Padding Oracle
[90023] - Ataque de Entidad Externa XML
[90022] - Divulgación de Errores de Aplicación
[10011] - Cookie sin Bandera Segura
[10010] - Bandera Cookie No HttpOnly
[90020] - Inyección Remota de Comandos del Sistema Operativo
[10003] - Librería Javascript Vulnerable
[90019] - Inyección de Código del Lado del Servidor
[7] - Inclusión de Ficheros Remotos
[6] - Salto de Directorios
[90017] - Inyeción XSLT
[90011] - Discordancia en el Conjunto de Caracteres
[90001] - ViewState de JSF Inseguro
[40035] - Archivo Oculto Encontrado
[3] - ID de Sesión en URL reescrita
[40034] - Fuga de Información de .env
[40032] - Fuga de Información de .htaccess
[2] - Divulgación de IP privada
[40029] - Fuga de Información de trace.axd
[40028] - Fuga de Información de ELMAH
[40027] - Inyección SQL - MsSQL
[40026] - Secuencia de Comandos Entre Sitios (XSS basado en DOM)
[40024] - Inyección SQL - SQLite
[40022] - Inyección SQL - PostgreSQL
[40021] - Inyección SQL - Oracle
[40020] - Inyección SQL - Hypersonic SQL

[40019] - Inyección SQL - MySQL
[40018] - Inyección SQL
[40014] - Secuencia de Comandos Entre Sitios (XSS Persistente)
[40012] - Secuencia de Comandos Entre Sitios (XSS Reflejado)
[40009] - Inclusión del Lado del Servidor
[40008] - Manipulación de Parámetros
[40003] - Inyección de CRLF
[30002] - Error de Formato de Cadena
[30001] - Desbordamiento del Búfer
[20019] - Redireccionamiento Externo
[20018] - Ejecución Remota de Código (CVE-2012-1823)
[20017] Divulgación del Código Fuente (CVE-2012-1823)
[20015] - Vulnerabilidad de Hemorragia de Corazón en OpenSSL
[10109] - Aplicación Web Moderna
[10108] - Tabulación Inversa
[10105] - Método de Autenticación Débil
[0] - Exploración de directorios
[10202] - Ausencia de Anti-CSRF Tokens

- Análisis de APIs. Se realizará un análisis de APIs activo. Este proceso requiere que la URL a analizar sea en efecto una API y no una aplicación web. El análisis lanzará el conjunto completo de alertas previamente enumeradas. (OWASP ZAP – ZAP - API Scan (zaproxy.org))

En todos estos análisis se incluirán además **8 análisis** adicionales especfíficos para **Websocket** (Navegador de Registros - Jira (juntadeandalucia.es)). Estos análisis se activarán si la herramienta detecta comunicación entre Websockets. Las alertas correspondientes a estos análisis tienen el código **110xxx**.

Alertas/pruebas de websocket

Alerta (Nomenclatura OWASP) - Prueba
[110008] - Divulgación de Información - Comentarios Sospechosos en XML vía WebSocket
[110007] - Hash de Usuario encontrado en el mensaje WebSocket
[110006] - Divulgación de IP Privada vía WebSocket
[110005] - Información de Identificación Personal a través de WebSocket
[110004] - Dirección email encontrada en mensaje de WebSocket

[110003] - Divulgación de Información - Mensaje de Depuración de Error via WebSocket
[110002] - Divulgación de Mensaje de WebSocket en Base64
[110001] - Divulgación de errores de aplicación vía WebSocket

- URL de la aplicación web a analizar: Se refiere a la dirección web específica que se va a someter al análisis de seguridad.
- Formato de API: Esta información es relevante únicamente para escaneos de APIs. Aquí se especifica el formato de la API a analizar, como OpenAPI, SOAP o GraphQL. Este parámetro no es aplicable a otros tipos de escaneos y debe ser ignorado en esos casos.
- Entrega: Literal de la entrega asociada a la aplicación a analizar. Deben coincidir los tres primeros dígitos con la versión asociada previamente (p.ej. 4.5.0.1)
- Componente/s: Nombre y entrega del componente que será objeto del análisis (aplica en caso de aplicación /componente).

El servicio de pruebas se realiza a nivel de aplicación, por lo que en una solicitud deben detallarse todos los componentes web que conforman la versión de la aplicación que será sometidas a las pruebas.

Informe de resultados

Una vez completadas las pruebas, el Servicio proporcionará un informe detallado de la ejecución, en el cual se indicarán los siguientes aspectos:

- Nº total de pruebas ejecutadas.
- Pruebas en las que se detectaron.
- Pruebas finalizadas con éxito.
- Pruebas abortadas.

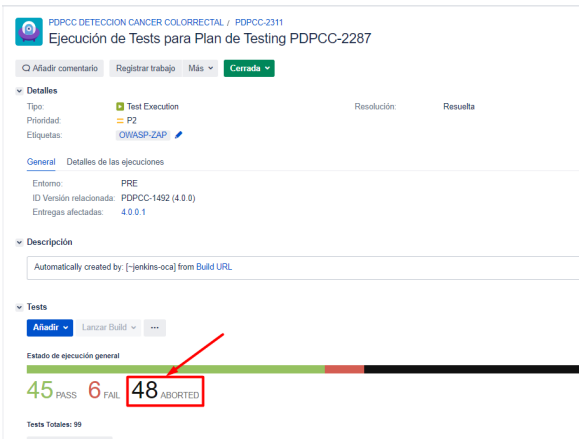
Es importante resaltar que la herramienta parte siempre del conjunto completo de pruebas. Por lo tanto, si se selecciona sólo la ejecución de pruebas dinámicas de seguridad base, aquellas que no pertenezcan al conjunto de los tests case Base, se mostrarán como **'ABORTED'**, lo que indica que no fueron ejecutadas.

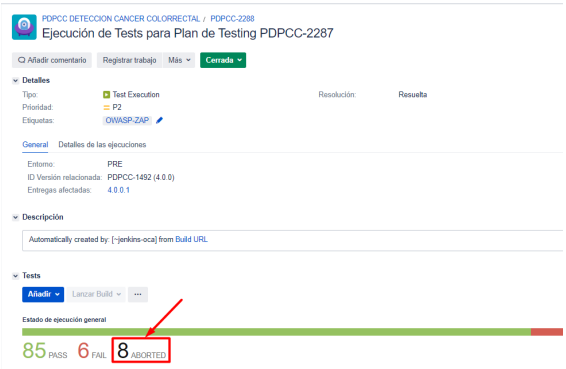
Ejemplo de un test execution (escaneo pasivo).

- 48 **ABORTED.** Corresponden a las 40 pruebas del escaneo activo (que no se lanza en este caso), más las 8 relativas a las pruebas de websocket.
- 45 **PASS** y 6 **FAIL**. Corresponden a las 51 pruebas del escaneo pasivo.

Ejemplo de un test execution (escaneo activo).

- 8 **ABORTED.** Corresponden a las pruebas de websocket que no se han lanzado en este caso.
- 85 **PASS** y 6 **FAIL**. Corresponden a las 51 pruebas del escaneo pasivo más las 40 del escaneo activo.





Las No Conformidades derivadas de cada fallo serán debidamente registradas y clasificadas conforme a su nivel de criticidad e impacto. Cada una de ellas incluirá una descripción detallada de las vulnerabilidades identificadas, así como información relevante para su resolución.

Adicionalmente, se entregará un análisis exhaustivo, junto con las conclusiones elaboradas por la USTIC, basadas en el informe previamente mencionado.

La USTIC llevará a cabo una revisión de los errores detectados y especificará las acciones correctivas que el proveedor deberá priorizar e implementar para mitigar los riesgos identificados.