OWASP
Open Web Application Security Project

Standard

Mobile AppSec Verification

Version 1.0

(Spanish Translation)

Project Leaders: Bernhard Mueller and Sven Schleier

This is document is currently under development. We welcome contributions and industry feedback. Contact us on the OWASP Mobile Testing Guide Slack channel:

https://owasp.slack.com/messages/project-mobile_omtg/

You can sign up here:

http://owasp.herokuapp.com/

The Open Web Application Security Project (OWASP) is a worldwide free and open community focused on improving the security of application software. Our mission is to make application security "visible", so that people and organizations can make informed decisions about application security risks. Every one is free to participate in OWASP and all of our materials are available under a free and open software license. The OWASP Foundation is a 501c3 not-for-profit charitable organization that ensures the ongoing availability and support for our work.

Prefacio por Bernhard Mueller, lider del proyecto móvil de OWASP

Las revoluciones tecnológicas pueden suceder rápidamente. Hace menos de una década los celulares eran dispositivos torpes con pequeños teclados - juguetes caros para usuarios empresariales expertos en tecnología. Hoy los celulares son una parte esencial de nuestras vidas. Hemos llegado a confiar en ellos para la búsqueda de información, la navegación y la comunicación y están presentes tanto en los negocios como en nuestra vida social.

Cada nueva tecnología introduce nuevos riesgos de seguridad, y mantenerse al día con estos cambios es uno de los principales retos a los que se enfrenta la industria de la seguridad. El bando defensivo está siempre unos pasos detrás. Por ejemplo, el modo predeterminado para muchos fue aplicar viejas formas de hacer las cosas: los celulares son como pequeños ordenadores y las aplicaciones móviles son como el software clásico, así que seguramente los requerimientos de seguridad son similares, ¿no? Pero no funciona así. Los sistemas operativos para teléfonos inteligentes son diferentes a los sistemas operativos de escritorio, y las aplicaciones móviles son diferentes a las aplicaciones web. Por ejemplo, el método clásico de escaneo de virus basado en firmas no tiene sentido en los sistemas operativos móviles modernos: no sólo es incompatible con el modelo de distribución de aplicaciones móviles, sino que también es técnicamente imposible debido a las restricciones del aisliamiento. Además, algunos tipos de vulnerabilidades, como los desbordamientos de búfer y los problemas XSS, son menos relevantes en el contexto de las aplicaciones móviles que, por ejemplo, en las aplicaciones de escritorio y las aplicaciones web (con excepciones).

Con el tiempo, la industria ha conseguido un mejor control del panorama de amenazas móviles. Resulta que la seguridad móvil tiene que ver con la protección de los datos: las aplicaciones almacenan nuestra información personal, imágenes, grabaciones, notas, datos de cuentas, información empresarial, ubicación y mucho más. Actúan como clientes que nos conectan con los servicios que utilizamos a diario, y como centros de comunicación que procesan todos y cada uno de los mensajes que intercambiamos con otros. Comprometer el celular de una persona, es obtener acceso sin filtros a su vida. Cuando consideramos que los dispositivos móviles se pierden o roban más fácilmente y que el malware para dispositivos móviles está aumentando, la necesidad de protección de datos se hace aún más evidente.

Por lo tanto, un estándar de seguridad para aplicaciones móviles debe centrarse en la forma en que las aplicaciones móviles manejan, almacenan y protegen la información sensible. A pesar de que los sistemas operativos móviles modernos como iOS y Android ofrecen buenas APIs para el almacenamiento y la comunicación de datos seguros, éstas deben ser incluidas en las aplicaciones y usadas correctamente para ser efectivas. El almacenamiento de datos, la comunicación entre aplicaciones, el uso apropiado de las API criptográficas y la comunicación segura a través de la red son sólo algunos de los aspectos que requieren una cuidadosa consideración.

Una cuestión importante que requiere el consenso de la industria es hasta dónde se debe llegar exactamente para proteger la confidencialidad e integridad de los datos. Por ejemplo, la mayoría de nosotros estaríamos de acuerdo en que una aplicación móvil debería verificar el certificado del servidor en una conexión TLS. Pero ¿qué ocurre con la fijación de certificados SSL? ¿No resulta en una vulnerabilidad? ¿Debería ser este un requerimiento si una aplicación maneja datos sensibles, o es contraproducente? ¿Necesitamos cifrar los datos almacenados en bases de datos SQLite, a pesar de que el sistema operativo aísla la aplicación? Lo que es apropiado para una aplicación puede ser poco realista para otra. El MASVS es un intento de

estandarizar estos requerimientos utilizando distintos niveles de verificación que se ajustan a los diferentes escenarios de amenaza.

Además, la aparición del malware y las herramientas de administración remota han creado conciencia de que los propios sistemas operativos móviles tienen fallas, por lo que las estrategias de aislamiento se utilizan cada vez más para proporcionar protección adicional a los datos confidenciales y evitar la manipulación del lado del cliente. Aquí es donde las cosas se complican. Las características de seguridad por hardware y las soluciones de aislamiento a nivel de sistema operativo, como Android for Work y Samsung Knox, existen, pero no siempre están disponibles en diferentes dispositivos. Como una curita, es posible implementar medidas de protección basadas en software - pero desafortunadamente, no hay estándares o procesos de prueba para verificar este tipo de protecciones.

Como resultado, los reportes de pruebas de seguridad de aplicaciones móviles están por todas partes: por ejemplo, algunos testers reportan una falta de ofuscación o detección de root en una aplicación Android como "falla de seguridad". Por otra parte, las medidas como el cifrado de palabras, la detección de depuradores o la ofuscación del flujo de control no se consideran obligatorias. Sin embargo, esta forma binaria de ver las cosas no tiene sentido porque la resistencia no es una proposición binaria: depende de las amenazas particulares en el dispositivo contra las que se quiere defender. Las protecciones de software no son inútiles, pero en última instancia pueden ser eludidas, por lo que nunca deben utilizarse como sustituto de los controles de seguridad.

El objetivo general del MASVS es ofrecer una línea de base para la seguridad de las aplicaciones móviles (MASVS-L1), mientras que también permite la inclusión de medidas de defensa en profundidad (MASVS-L2) y protecciones contra las amenazas del lado del cliente (MASVS-R). El MASVS está pensado para lograr lo siguiente:

- Proveer requerimientos para arquitectos y desarrolladores de software que buscan desarrollar aplicaciones móviles seguras;
- Ofrecer un estándar para que la industria pueda utilizar en las revisiones de seguridad de aplicaciones móviles;
- Clarificar el rol de los mecanismos de protección de software en la seguridad móvil y proporcionar requerimientos para verificar su efectividad;
- Proporcionar recomendaciones específicas sobre el nivel de seguridad que se recomienda para los diferentes casos de uso.

Somos conscientes de que es imposible lograr un consenso del 100% en la industria. No obstante, esperamos que el MASVS sea útil para proporcionar orientación en las fases de desarrollo y prueba de aplicaciones móviles. Como estándar de código abierto, el MASVS evolucionará con el tiempo, y acogemos con agrado cualquier contribución o sugerencia.

Frontispicio

Sobre el Estándar

Bienvenido al Estándar de Verificación de Seguridad de Aplicaciones Móviles (MASVS) 1.0. El MASVS es un esfuerzo comunitario para establecer un marco de requisitos de seguridad necesarios para diseñar, desarrollar y probar aplicaciones móviles seguras en iOS y Android.

El MASVS es la culminación del esfuerzo de la comunidad y la retroalimentación con la industria. Esperamos que este estándar evolucione con el tiempo y agradecemos la retroalimentación de la comunidad. La mejor manera de ponerse en contacto con nosotros es a través del canal OWASP Mobile Project en Slack:

https://owasp.slack.com/messages/project-mobile_omtg/details/

Las cuentas se pueden crear en la siguiente URL:

http://owasp.herokuapp.com/.

Copyright y Licencia

Líderes del proyecto	Autores principales	Colaboradores y revisores
Bernhard Mueller, Sven Schleier	**Bernhard Mueller**	**Abdessamad Temmar, Abhinav Sejpal, Alexander Antukh, Anant Shrivastava, Ben Gardiner, Francesco Stillavato, Jeroen Willemsen, Nikhil Soni, Prabhant Singh, Roberto Martelloni, Stephen Corbiaux, Stephen Reda, Sjoerd Langkemper, Stefaan Seys, Sven Schleier, Yogesh Sharma**
Traducción al español	**---**	**Martín Marsicano**

Este documento comenzó como un bifurcación del Estándar de Verificación de Seguridad de Aplicaciones de OWASP escrito por Jim Manico.

Estándar de Verificación de Seguridad de Aplicaciones Móviles

El MASVS se puede utilizar para establecer un nivel de confianza en la seguridad de las aplicaciones móviles. Los requerimientos fueron desarrollados con los siguientes objetivos en mente:

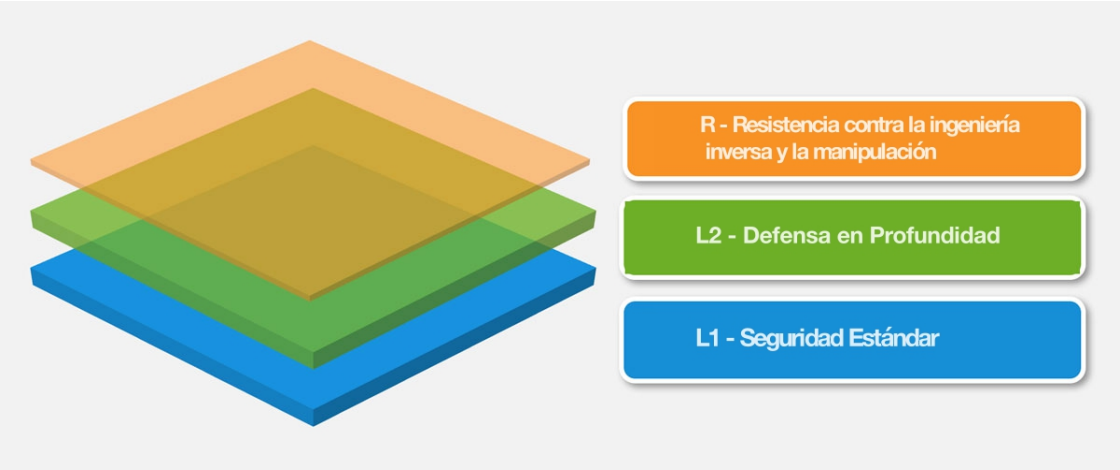
- Usar como una métrica - Para proporcionar un estándar de seguridad contra el cual las aplicaciones móviles existentes pueden ser comparadas por desarrolladores y los propietarios de las aplicaciones;
- Utilizar como guía - Proporcionar una guía durante todas las fases del desarrollo y prueba de las aplicaciones móviles;
- Usar durante la contratación - Proporcionar una línea de base para la verificación de seguridad de aplicaciones móviles.

Modelo de seguridad para una aplicación móvil

El MASVS define dos niveles estrictos de verificación de seguridad (L1 y L2), así como un conjunto de requisitos de resistencia a la ingeniería inversa (MASVS-R) flexible, es decir, adaptable a un modelo de amenaza específico de la aplicación. Los niveles MASVS-L1 y MASVS-L2 contienen requerimientos genéricos de seguridad recomendados para todas las aplicaciones móviles (L1) y para aplicaciones que manejan datos altamente sensibles (L2). MASVS-R cubre los controles de seguridad adicionales que se pueden aplicar si la prevención de las amenazas del lado del cliente son un objetivo de diseño.

Cumplir con los requerimientos de MASVS-L1 resulta en una aplicación segura que sigue las mejores prácticas de seguridad y no sufre de las vulnerabilidades más comunes. MASVS-L2 añade controles adicionales de defensa en profundidad, como la fijación de certificados SSL, lo que resulta en una aplicación resistente a ataques más sofisticados, asumiendo que los controles de seguridad del sistema operativo móvil estén intactos y que el usuario final no sea visto como un adversario potencial. El cumplimiento de todos o de un subconjunto de los requerimientos de protección del software en el nivel MASVS-R ayuda a impedir amenazas específicas del lado del cliente cuando el usuario final es considerado malicioso y/o el sistema operativo móvil está comprometido.

Note que los controles de protección de software listados en el nivel MASVS-R y descritos en la Guía de Pruebas Móviles de OWASP pueden ser evitados y nunca deben ser usados como un reemplazo para los controles de seguridad. En cambio, su objetivo es añadir controles de protección adicionales y específicos a las amenazas que se quieren evitar, sobre las aplicaciones que cumplen los requerimientos de los distintos niveles del MASVS L1 o L2.



Estructura del documento

La primera parte del MASVS contiene una descripción del modelo de seguridad y de los niveles de verificación disponibles, seguido de recomendaciones sobre cómo utilizar el estándar en la práctica. En la segunda parte se detallan los requisitos de seguridad, junto con un mapeo a los distintos niveles de verificación. Los requerimientos se han agrupado en ocho categorías (V1 a V8) basadas en el objetivo/alcance técnico. La siguiente nomenclatura se utiliza a lo largo del MASVS y el MSTG:

- *Categoría de los requerimientos:* MASVS-Vx, ej. MASVS-V2: Almacenamiento de datos y privacidad.
- *Requerimiento:* MASVS-Vx.y, ej. MASVS-V2.2:"No se escribe ningún dato sensible en los registros de la aplicación".

Niveles de verificación detallados

MASVS-L1: Seguridad Estándar

Una aplicación móvil que logra el nivel MASVS-L1 se adhiera a las mejores prácticas de seguridad en aplicaciones móviles. Cumple con los requerimientos básicos en términos de calidad de código, manejo de los datos sensibles e interacción con el entorno móvil. Debe existir un proceso de pruebas para verificar los controles de seguridad. Este nivel es apropiado para todas las aplicaciones móviles.

MASVS-L2: Defensa en Profundidad

MASVS-L2 introduce controles de seguridad avanzados que van más allá de los requisitos estándar. Para cumplir con el nivel L2, debe existir un modelo de amenaza y la seguridad debe ser una parte fundamental de la arquitectura y el diseño de la aplicación. Este nivel es apropiado para aplicaciones que manejan datos sensibles, como las aplicaciones de banca móvil.

MASVS-R: Resistencia contra la ingeniería inversa y la manipulación

La aplicación cuenta con el nivel de seguridad específico para la aplicación y también es resistente a ataques específicos y claramente definidos en el lado del cliente, como alteración, modificación o ingeniería inversa para extraer código o datos sensibles. Esta aplicación

aprovecha las características de seguridad del hardware o bien técnicas de protección de software suficientemente fuertes y verificables. MASVS-R es adecuado para las aplicaciones que manejan datos altamente confidenciales y puede servir como medio para proteger la propiedad intelectual o la manipulación de una aplicación.

Uso recomendado

Las aplicaciones pueden ser verificadas contra el nivel MASVS L1 o L2 de acuerdo con la evaluación previa del riesgo y el nivel general de seguridad requerido. L1 es aplicable a todas las aplicaciones móviles, mientras que L2 se recomienda generalmente para las aplicaciones que manejan datos y/o funciones sensibles. MASVS-R (o partes de él) puede aplicarse para verificar la resistencia frente a amenazas específicas, como el reempaquetado o la extracción de datos sensibles, *además* de una verificación de seguridad adecuada.

En resumen, están disponibles los siguientes tipos de verificación:

- MASVS-L1
- MASVS-L1+R
- MASVS-L2
- MASVS-L2+R

Las diferentes combinaciones reflejan diferentes grados de seguridad y resistencia. El objetivo es permitir la flexibilidad: Por ejemplo, un juego móvil puede no requerir controles de seguridad del MASVS-L2, como la autenticación de 2 factores por razones de usabilidad, pero seguramente deba prevenir la manipulación del código por razones del negocio.

¿Qué tipo de verificación elegir?

La implementación de los requisitos del nivel MASVS L2 aumenta la seguridad, mientras que al mismo tiempo aumenta el costo de desarrollo y potencialmente empeora la experiencia del usuario final (el compromiso clásico). En general, L2 debe utilizarse para aplicaciones siempre que tenga sentido desde el punto de vista del riesgo contra el costo que conlleva (es decir, cuando la potencial pérdida causada por un compromiso de confidencialidad o integridad sea superior al costo que suponen los controles de seguridad adicionales). Una evaluación del riesgo debe ser el primer paso antes de aplicar el MASVS.

Ejemplos

MASVS-L1

- Todas las aplicaciones móviles. El nivel MASVS-L1 enumera las mejores prácticas de seguridad que se pueden seguir con un impacto razonable en el costo de desarrollo y la experiencia del usuario. Aplique los requerimientos de MASVS-L1 para cualquier aplicación que no califique para uno de los niveles superiores.

MASVS-L2

- Industria de la Salud: Aplicaciones móviles que almacenan información personal identificable que puede ser utilizada para el robo de identidad, pagos fraudulentos, o una variedad de esquemas de fraude. Para el sector de la salud en los Estados Unidos, las consideraciones de cumplimiento incluyen la Ley de Portabilidad y Responsabilidad del Seguro Médico (HIPAA, por sus siglas en inglés), Privacidad,

Seguridad, Reglas de Notificación de Violación (Breach Notification Rules) y Reglas de Seguridad del Paciente (Patient Safety Rule).

- Sector Financiero: Aplicaciones que permiten el acceso a información altamente sensible como números de tarjetas de crédito, información personal o que permiten al usuario mover fondos. Estas aplicaciones deben tener controles de seguridad adicionales para prevenir el fraude. Las aplicaciones financieras necesitan asegurar el cumplimiento de las normas de seguridad de datos de la industria de tarjetas de pago (PCI DSS), Gramm Leech Bliley Act y Sarbanes-Oxley Act (SOX).

MASVS L1+R

- Aplicaciones móviles donde la protección de la dirección IP es un objetivo empresarial. Los controles de resistencia listados en MASVS-R se pueden utilizar para aumentar el esfuerzo necesario para obtener el código fuente original e impedir la manipulación / rotura.

- Industria de los juegos: Juegos con una necesidad esencial de evitar la posibilidad de modding y el engaño, como los juegos en línea competitivos. Hacer trampa es un tema importante en los juegos en línea, ya que una gran cantidad de tramposos conduce a un descontento de la base de jugadores y, en última instancia, puede causar que un juego falle. MASVS-R proporciona controles básicos contra la manipulación para ayudar a aumentar el esfuerzo de los tramposos.

MASVS L2+R

- Industria Financiera: Aplicaciones de banca en línea que permiten al usuario mover fondos, donde las técnicas de inyección de código e instrumentación en dispositivos comprometidos suponen un riesgo. En este caso, los controles del MASVS-R se pueden utilizar para impedir la manipulación de código, elevando la barra para los autores de malware.

- Todas las aplicaciones móviles que, por diseño, necesitan almacenar datos sensibles en el dispositivo móvil y, al mismo tiempo, deben soportar una amplia gama de dispositivos y versiones del sistema operativo. En este caso, los controles de resistencia pueden utilizarse como una medida de defensa en profundidad para aumentar el esfuerzo de los atacantes que intentan extraer los datos sensibles.

Evaluación y Certificación

Postura de OWASP sobre las Certificaciones MASVS y Marcas de Confianza

OWASP, como una organización sin fines de lucro e independiente de las empresas, no certifica a ningún proveedor, verificador o software.

Todas estas afirmaciones de garantía, marcas de confianza o certificaciones no son oficialmente examinadas, registradas o certificadas por OWASP, por lo que una organización que confía en tal opinión debe tener la precaución de la confianza depositada en cualquier tercero o marca de confianza que alega la certificación ASVS.

Esto no debe impedir que las organizaciones ofrezcan tales servicios de seguridad, siempre que no reclamen la certificación oficial OWASP.

Guía para certificar aplicaciones móviles

La forma recomendada de verificar la conformidad de una aplicación móvil con el MASVS es realizar una revisión de "libro abierto", lo que significa que los verificadores tienen acceso a recursos claves como arquitectos y desarrolladores de la aplicación, documentación del proyecto, código fuente y acceso autenticado a los terminales, incluyendo acceso a al menos una cuenta de usuario para cada función.

Es importante tener en cuenta que el MASVS sólo cubre la seguridad de la aplicación móvil (en el dispositivo del cliente) y la comunicación de red entre la aplicación y su/s punto/s final/es remoto/s, así como algunos requerimientos básicos y genéricos relacionados con la autenticación del usuario y la gestión de sesiones. No contiene requerimientos específicos para los servicios remotos (por ejemplo, servicios web) asociados a la aplicación, salvo para un conjunto limitado de requerimientos genéricos relacionados con la autenticación y la gestión de sesiones. No obstante, MASVS V1 especifica que los servicios remotos deben ser cubiertos por el modelo general de amenazas, y ser verificados contra los estándares apropiados, como el OWASP ASVS.

Una organización que certifica debe incluir en todos los informes el alcance de la verificación (particularmente si un componente clave está fuera del alcance), un resumen de los resultados de la verificación, incluyendo las pruebas aprobadas y fallidas, con instrucciones claras de cómo resolver las pruebas fallidas. Mantener documentos de trabajo detallados, capturas de pantalla o vídeos, guiones para explotar de forma fiable y repetida un problema y registros electrónicos de las pruebas, como los registros de un proxy y notas asociadas, se consideran práctica estándar de la industria. No basta con simplemente ejecutar una herramienta y presentar un informe sobre los fallos; esto no aporta suficientes evidencias de que se han analizado y probado a fondo todos los aspectos a nivel de una certificación. En caso de controversia, debe haber pruebas suficientes para demostrar que todos los requerimientos verificados han sido efectivamente probados.

Usando la Guía de Pruebas de Seguridad Móvil de OWASP (MSTG)

La OWASP MSTG es una guía para la verificación de la seguridad de las aplicaciones móviles. Describe los procedimientos técnicos para verificar los requerimientos listados en el MASVS.

La MSTG incluye una lista de casos de prueba, cada uno de los cuales se corresponde con un requerimiento del MASVS. Mientras que los requerimientos del MASVS son de alto nivel y genéricos, la MSTG proporciona recomendaciones detalladas y procedimientos de verificación para cada uno de los sistemas operativos móviles.

El papel de las herramientas de pruebas de seguridad automatizadas

Se recomienda el uso de escáneres de código fuente y herramientas de verificación de caja negra para aumentar la eficiencia siempre que sea posible. Sin embargo, no es posible completar la verificación MASVS utilizando únicamente herramientas automatizadas: cada aplicación móvil es diferente, y la comprensión de la arquitectura general, la lógica de negocio y los problemas específicos de las tecnologías y plataformas que se utilizan es un requerimiento obligatorio para verificar la seguridad de la aplicación.

Otros Usos

Como guía detallada para una arquitectura segura

Uno de los usos más comunes del Estándar de Verificación de Seguridad de Aplicaciones Móviles es como recurso para los arquitectos de seguridad. Los dos principales esquemas de seguridad en la arquitectura, SABSA o TOGAF, carecen de una gran cantidad de información necesaria para completar las revisiones de seguridad en la arquitectura de las aplicaciones móviles. El MASVS se puede utilizar para llenar esos vacíos permitiendo a los arquitectos elegir mejores controles para los problemas comunes de seguridad en las aplicaciones móviles.

Como reemplazo de las listas de verificación de codificación segura estándar

Muchas organizaciones se pueden beneficiar de la adopción del MASVS, eligiendo uno de los dos niveles, o bifurcando el MASVS y cambiando lo que se requiere para el nivel de riesgo de cada aplicación de una manera específica al negocio. Fomentamos este tipo de bifurcación siempre y cuando se mantenga la trazabilidad, de modo que si una aplicación cumple el requerimiento 4.1, lo mismo ocurre en las bifurcaciones del estándar cuando éste evoluciona.

Como base para las metodologías de pruebas de seguridad

Una buena metodología de pruebas de seguridad para aplicaciones móviles debe cubrir todos los requerimientos listados en el MASVS. La Guía de Pruebas de Seguridad Móvil de OWASP (MSTG) describe los casos de prueba de caja negra y caja blanca para cada requerimiento de verificación.

Como guía para las automatización de pruebas unitarias y de integración

El MASVS está diseñado para ser altamente verificable, con la única excepción de los requerimientos de la arquitectura. Las pruebas unitarias, de integración y de aceptación automatizadas, basadas en los requerimientos del MASVS, pueden integrarse en el ciclo de vida de un desarrollo continuo. Esto no sólo aumenta la conciencia de seguridad de los desarrolladores, sino que también mejora la calidad general de las aplicaciones desarrolladas, y reduce la cantidad de hallazgos durante las pruebas de seguridad en la fase previa al lanzamiento.

Para el entrenamiento en desarrollo seguro

El MASVS también se puede utilizar para definir características de aplicaciones móviles seguras. Muchos cursos de "codificación segura" son simplemente cursos de hacking ético con algunos consejos de programación segura. Esto no ayuda a los desarrolladores. En su lugar, los cursos de desarrollo seguro móvil pueden utilizar el MASVS, con un fuerte enfoque en los controles proactivos documentados en el MASVS, en lugar de, por ejemplo, el Top 10 móvil de problemas de seguridad del código de OWASP.

V1: Requisitos de Arquitectura, Diseño y Modelado de Amenazas

Objetivo de control

En un mundo perfecto, la seguridad sería considerada en todas las fases del desarrollo. Sin embargo, en la realidad, la seguridad es a menudo sólo considerada en una etapa tardía del desarrollo del software. Además de los controles técnicos, el MASVS requiere que existan procesos que garanticen que la seguridad se ha abordado explícitamente al planificar la arquitectura de la aplicación móvil, y que se conocen los roles funcionales y de seguridad de todos los componentes. Dado que la mayoría de las aplicaciones móviles actúan como clientes de los servicios remotos, debe garantizarse que también se apliquen las medidas de seguridad adecuadas a dichos servicios, no basta con probar la aplicación móvil de forma aislada.

La categoría V1 lista los requerimientos pertinentes a la arquitectura y al diseño de la aplicación. Debido a esto es la única categoría que no se corresponde con casos de test de la Guía de Pruebas Móviles de OWASP. Para cubrir temas tales como el modelado de amenazas, SDLC seguro, gestión de claves, los usuarios del MASVS deben consultar los respectivos proyectos de OWASP y/u otros estándares como los que se encuentran enlazados a debajo.

Requerimientos de Verificación de Seguridad

A continuación se enumeran los requerimientos para MASVS-L1 y MASVS-L2.

#	Descripción	L1	L2
1.1	Todos los componentes se encuentran identificados y asegurar que son necesarios.	✓	✓
1.2	Los controles de seguridad nunca se aplican sólo en el lado del cliente, sino que también en los respectivos servidores remotos.	✓	✓
1.3	Se definió una arquitectura de alto nivel para la aplicación y los servicios y se incluyeron controles de seguridad en la misma.	✓	✓
1.4	Se identificó claramente la información considerada sensible en el contexto de la aplicación mobile.	✓	✓
1.5	Todos los componentes de la aplicación están definidos en términos de la lógica de negocio o las funciones de seguridad que proveen.		✓
1.6	Se realizó un modelado de amenazas para la aplicación mobile y los servicios en el que se definieron las mismas y sus contramedidas.		✓
1.7	La implementación de los controles de seguridad se encuentra centralizada.		✓

1.8	Existe una política explícita para el manejo de las claves criptográficas (si se usan) y se refuerza su ciclo de vida. Idealmente siguiendo un estándar del manejo de claves como el NIST SP 800-57.		✓
1.9	Existe un mecanismo para imponer las actualizaciones de la aplicación móvil.		✓
1.10	Se realizan tareas de seguridad en todo el ciclo de vida de la aplicación.		✓

Referencias

Para más información, ver también:

- Top 10 Móvil de OWASP: M10 - Funcionalidades Extrañas: https://www.owasp.org/index.php/Mobile_Top_10_2016-M10-Extraneous_Functionality
- "Cheat sheet" de la Seguridad en la Arquitectura de OWASP: https://www.owasp.org/index.php/Application_Security_Architecture_Cheat_Sheet
- Modelado de Amenazas de OWASP: https://www.owasp.org/index.php/Application_Threat_Modeling
- "Cheat sheet" para el ciclo de desarrollo seguro de OWASP: https://www.owasp.org/index.php/Secure_SDLC_Cheat_Sheet
- Microsoft SDL: https://www.microsoft.com/en-us/sdl/
- NIST SP 800-57: http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf

V2: Requerimientos en el Almacenamiento de datos y la Privacidad

Objetivo de control

La protección de datos sensibles, como las credenciales del usuario y la información privada, es un aspecto clave de la seguridad móvil. En primer lugar, los datos confidenciales pueden exponerse involuntariamente a otras aplicaciones que se ejecutan en el mismo dispositivo si se utilizan de forma inadecuada mecanismos de comunicación entre procesos del sistema operativo. Los datos también pueden filtrarse involuntariamente en el almacenamiento en la nube, las copias de seguridad o la caché del teclado. Además, los dispositivos móviles pueden perderse o robarse más fácilmente que otros tipos de dispositivos, por lo que un adversario que obtiene acceso físico al mismo es un escenario más probable. En ese caso, se pueden implementar protecciones adicionales para dificultar la recuperación de los datos sensibles.

El MASVS se centra en las aplicaciones y por esto no cubre políticas para el dispositivo como Mobile Device Managment (MDM) (https://gsuite.google.com/products/admin/mobile/) o Enter (EDM). Igualmente se recomienda utilizar estas soluciones en contextos empresariales.

Definición de Datos Sensibles

Los datos sensibles en el contexto del MASVS se refieren tanto a las credenciales de usuario como a cualquier otros datos que se considere sensible en el contexto particular, por ejemplo:

- Información de identificación personal que puede ser usada para el robo de identidad: números de seguro social, números de tarjetas de crédito, números de cuentas bancarias, información médica;
- Datos altamente confidenciales que, en caso de que se comprometieran, ocasionarían daños a la reputación y/o costes financieros: información contractual, información cubierta por acuerdos de confidencialidad, información de gestión;
- Cualquier dato que debe ser protegido por ley o por razones de conformidad.

Requerimientos de Verificación de Seguridad

La gran mayoría de las cuestiones relativas a la divulgación de datos pueden prevenirse siguiendo reglas sencillas. La mayoría de los controles enumerados en este capítulo son obligatorios para todos los niveles de verificación.

#	Descripción	L1	L2
2.1	Las funcionalidades de almacenamiento de credenciales del sistema son utilizadas para almacenar la información sensible, como credenciales del usuario y claves criptográficas.	✓	✓
2.2	No se escribe información sensible en los registros de la aplicación.	✓	✓
2.3	No se comparte información sensible con servicios externos salvo que sea una necesidad de la arquitectura.	✓	✓
2.4	Se desactiva el caché del teclado en los campos de texto donde se maneja información sensible.	✓	✓
2.5	Se desactiva el portapapeles en los campos de texto donde se maneja información sensible.	✓	✓

2.6	No se expone información sensible mediante mecanismos entre procesos (IPC).	✓	✓
2.7	No se expone información sensible como contraseñas y números de tarjetas de crédito a través de la interfaz o capturas de pantalla.	✓	✓
2.8	No se incluye información sensible en los backups generados por el sistema operativo.		✓
2.9	La aplicación remueve la información sensible de la vista cuando la aplicación pasa a un segundo plano.		✓
2.10	La aplicación no conserva la información sensible en memoria más de lo necesario y la memoria es limpiada luego de su uso.		✓
2.11	La aplicación obliga a que exista una política mínima de seguridad en el dispositivo, como que el usuario deba configurar un código de acceso.		✓
2.12	La aplicación educa al usuario acerca de los tipos de información personal que procesa y de las mejores prácticas en seguridad que el usuario debería seguir al utilizar la aplicación.		✓

Referencias

La Guía de Pruebas de Seguridad Móvil de OWASP proporciona instrucciones detalladas para verificar los requisitos listados en esta sección.

- Para Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05d-Testing-Data-Storage.md
- Para iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06d-Testing-Data-Storage.md

Para más información, ver también:

- OWASP Top 10 Móvil: M2 - Almacenamiento de Datos Inseguro: https://www.owasp.org/index.php/Mobile_Top_10_2016-M2-Insecure_Data_Storage
- CWE: https://cwe.mitre.org/data/definitions/922.html

V3: Requerimientos de Criptografía

Objetivo de control

La criptografía es un componente esencial a la hora de proteger los datos almacenados en un dispositivo móvil. También es una categoría en la que las cosas pueden ir terriblemente mal, especialmente cuando no se siguen las convenciones estándar. El propósito de estos controles es asegurarse que la aplicación utiliza criptografía según las mejores prácticas de la industria, incluyendo:

- Uso de librerías conocidas y probadas;
- Configuración y elección de primitivas criptográficas apropiado;
- Cuando se requiere de randomización se selecciona el generador debido.

Requerimientos de Verificación de Seguridad

#	Descripción	L1	L2
3.1	La aplicación no depende de únicamente de criptografía simétrica con "claves a fuego".	✓	✓
3.2	La aplicación utiliza implementaciones de criptografía probadas.	✓	✓
3.3	La aplicación utiliza primitivas de seguridad que son apropiadas para el caso particular y su configuración y sus parámetros siguen las mejores prácticas de la industria.	✓	✓
3.4	La aplicación no utiliza protocolos o algoritmos criptográficos que son considerados deprecados para aspectos de seguridad.	✓	✓
3.5	La aplicación no reutiliza una misma clave criptográfica para varios propósitos.	✓	✓
3.6	Los valores random son generados utilizando un generador de números suficientemente randómicos.	✓	✓

Referencias

La Guía de Pruebas de Seguridad Móvil de OWASP proporciona instrucciones detalladas para verificar los requisitos listados en esta sección.

- Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05e-Testing-Cryptography.md
- iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06e-Testing-Cryptography.md

Para más información, ver también:

- Top 10 Móvil de OWASP: M5 - Criptografía Insuficiente
- CWE: https://cwe.mitre.org/data/definitions/310.html

V4: Requerimientos de Autenticación y Manejo de Sesiones

Objetivo de control

En la mayoría de los casos, una parte esencial de la arquitectura global de aplicaciones móviles es que los usuarios deben inician sesión en un servicio remoto. Aunque la mayor parte de la lógica ocurre en el servidor, MASVS define algunos requerimientos básicos sobre como manejar las cuentas y sesiones del usuario.

Requerimientos de Verificación de Seguridad

#	Descripción	L1	L2
4.1	Si la aplicación provee acceso a un servicio remoto, un mecanismo aceptable de autenticación como usuario y contraseña es realizado en el servidor remoto.	✓	✓
4.2	Si se utiliza la gestión de sesión por estado, el servidor remoto usa tokens de acceso randómicos para autenticar los pedidos del cliente sin requerir el envío de las credenciales del usuario en cada uno.	✓	✓
4.3	Si se utiliza la autenticación basada en tokens sin estado, el servidor proporciona un token que se ha firmado utilizando un algoritmo seguro.	✓	✓
4.4	Cuando el usuario se desloguea se termina la sesión también en el servidor.	✓	✓
4.5	Existe una política de contraseñas y es aplicada en el servidor.	✓	✓
4.6	El servidor implementa mecanismos, cuando credenciales de autenticación son ingresadas una cantidad excesiva de veces.	✓	✓
4.7	La autenticación biométrica, si hay, no está atada a un evento (usando una api que simplemente retorna "true" o "false"). Sino que está basado en el desbloqueo del keychain (iOS) o un keystore (Android).		✓
4.8	Las sesiones y los tokens de acceso expiran luego de un tiempo predefinido de inactividad.		✓
4.9	Existe un mecanismo de segundo factor de autenticación (2FA) en el servidor y es aplicado consistentemente.		✓
4.10	Para realizar transacciones o acciones que manejan información sensible se requiere una re-autenticación.		✓
4.11	La aplicación informa al usuario acerca de los accesos a su cuenta. El usuario es capaz de ver una lista de los dispositivos conectados y bloquear el acceso desde ciertos dispositivos.		✓

Referencias

La Guía de Pruebas de Seguridad Móvil de OWASP proporciona instrucciones detalladas para verificar los requisitos listados en esta sección.

- Para Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05f-Testing-Authentication.md
- Para iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06f-Testing-Authentication-and-Session-Management.md

Para más información, ver también:

- OWASP Top 10 Móvil: M4 - Autenticación Insegura, M6 - Autorización Insegura
- CWE: https://cwe.mitre.org/data/definitions/287.html

V5: Requerimientos de Comunicación a través de la red

Objetivo de control

Los controles enumerados en esta categoría tienen por objetivo asegurar la confidencialidad e integridad de la información intercambiada entre la aplicación móvil y los servicios del servidor. Como mínimo se deben utilizar canales seguros y cifrados utilizando el protocolo TLS con las configuraciones apropiadas. En el nivel 2 se establecen medidas en profundidad como fijación de certificados SSL.

Requerimientos de Verificación de Seguridad

#	Descripción	L1	L2
5.1	La información es enviada cifrada utilizando TLS. El canal seguro es usado consistentemente en la aplicación.	✓	✓
5.2	Las configuraciones del protocolo TLS siguen las mejores prácticas o tan cerca posible mientras que el sistema operativo del dispositivo lo permite.	✓	✓
5.3	La aplicación verifica el certificado X.509 del servidor al establecer el canal seguro y solo se aceptan certificados firmados por una CA válida.	✓	✓
5.4	La aplicación utiliza su propio almacén de certificados o realiza una fijación del certificado o la clave pública del servidor y no establece una conexión con servidores que ofrecen otros certificados o clave por más que estén firmados por una CA confiable.		✓
5.5	La aplicación no depende de un único canal de comunicaciones inseguro (email o SMS) para operaciones críticas como registros o recuperación de cuentas.		✓
5.6	La aplicación sólo depende de bibliotecas de conectividad y seguridad actualizadas.		✓

Referencias

La Guía de Pruebas de Seguridad Móvil de OWASP proporciona instrucciones detalladas para verificar los requisitos listados en esta sección.

- Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05g-Testing-Network-Communication.md
- iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06g-Testing-Network-Communication.md

Para más información, ver también:

- OWASP Top 10 Móvil: M3 - Comunicación Insegura: https://www.owasp.org/index.php/Mobile_Top_10_2016-M3-Insecure_Communication
- CWE: https://cwe.mitre.org/data/definitions/319.html
- CWE: https://cwe.mitre.org/data/definitions/295.html

V6: Requerimientos de Interacción con la Plataforma

Objetivo de control

Estos controles revisan que se utilicen las APIs de la plataforma y componentes estándar de una manera segura. Además se cubre la comuninación entre aplicaciones (IPC).

Requerimientos de Verificación de Seguridad

#	Descripción	L1	L2
6.1	La aplicación requiere la mínima cantidad de permisos.	✓	✓
6.2	Toda entrada del usuario y fuentes externas es validada y si es necesario sanitizada. Esto incluye información recibida por la UI, y mecanismo IPC como los intents, URLs y fuentes de la red.	✓	✓
6.3	La aplicación no exporta funcionalidades sensibles vía esquemas de URL, salvo que dichos mecanismos estén debidamente protegidos.	✓	✓
6.4	La aplicación no exporta funcionalidades sensibles a través de mecanismos IPC salvo que los mecanismos estén debidamente protegidos.	✓	✓
6.5	JavaScript se encuentra deshabilitado en los WebViews salvo que sea necesario.	✓	✓
6.6	Los WebViews se encuentran configurados para permitir el mínimo de los manejadores (idealmente, solo https). Manejadores peligrosos como file, tel y app-id se encuentran deshabilitados.	✓	✓
6.7	Si objetos nativos son expuestos en WebViews, verificar que solo se cargan JavaScripts contenidos del paquete de la aplicación.	✓	✓
6.8	Serialización de objetos, si se realiza, se implementa utilizando API seguras.	✓	✓

Referencias

La Guía de Pruebas de Seguridad Móvil de OWASP proporciona instrucciones detalladas para verificar los requisitos listados en esta sección.

- Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05h-Testing-Platform-Interaction.md
- iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06h-Testing-Platform-Interaction.md

Para más información, ver también:

- OWASP Top 10 Móvil: M1 - Uso inapropiado de la Plataforma
- CWE: https://cwe.mitre.org/data/definitions/20.html
- CWE: https://cwe.mitre.org/data/definitions/749.html

V7: Requerimientos de Calidad de Código y Configuración del Compilador

Objetivo de control

Estos controles buscan asegurar que se siguieron las prácticas de seguridad básicas en el desarrollo de la aplicación. Y que se activaron las funcionalidades "gratuitas" ofrecidas por el compilador.

Requerimientos de Verificación de Seguridad

#	Descripción	L1	L2
7.1	La aplicación es firmada y provista con un certificado válido.	✓	✓
7.2	La aplicación fue liberada en modo release y con las configuraciones apropiadas para el mismo (ej. non-debuggable).	✓	✓
7.3	Los símbolos de debug fueron removidos de los binarios nativos.	✓	✓
7.4	La aplicación no contiene código de prueba y no realiza log de errores o mensajes de debug.	✓	✓
7.5	Todos los componentes de terceros se encuentran identificados y revisados por vulnerabilidades conocidas.	✓	✓
7.6	La aplicación captura y maneja debidamente las posibles excepciones.	✓	✓
7.7	Los controles de seguridad deniegan el acceso por defecto.	✓	✓
7.8	En código no administrado, la memoria es pedida, usada y liberada de manera correcta.	✓	✓
7.9	Funcionalidades de seguridad gratuitas se encuentran activadas.	✓	✓

Referencias

La Guía de Pruebas de Seguridad Móvil de OWASP proporciona instrucciones detalladas para verificar los requisitos listados en esta sección.

- Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05i-Testing-Code-Quality-and-Build-Settings.md
- iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06i-Testing-Code-Quality-and-Build-Settings.md

Para más información, ver también:

- Top 10 Móvil de OWASP: M7 - Calidad del Código en el Cliente
- CWE: https://cwe.mitre.org/data/definitions/119.html
- CWE: https://cwe.mitre.org/data/definitions/89.html
- CWE: https://cwe.mitre.org/data/definitions/388.html
- CWE: https://cwe.mitre.org/data/definitions/489.html

V8: Requerimientos de Resistencia ante la Ingeniería Inversa

Objetivo de control

En esta sección se cubren protecciones recomendadas para aplicaciones que maneja o brindan acceso a información o funcionalidades sensibles. La falta de estos controles no generan vulnerabilidades - sino que, están pensados para incrementar la resistencia contra la ingeniería inversa de la aplicación, dificultandole al adversario el acceso a los datos o el entendimiento del modo de ejecución de la aplicación.

Los controles de esta sección deben aplicarse según sea necesario, basándose en una evaluación de los riesgos causados por la manipulación no autorizada de la aplicación y/o la ingeniería inversa del código. Sugerimos consultar el documento de OWASP "Ingeniería Inversa - Amenazas de la Ingeniería Inversa de OWASP" (vea las referencias a continuación) para obtener una lista de los riesgos del negocio, así como las amenazas técnicas asociadas.

Para que cualquiera de los controles de la lista siguiente sea eficaz, la aplicación debe cumplir al menos todos los requerimientos del nivel MASVS-L1 (es decir, deben existir sólidos controles de seguridad), así como todos los requerimientos de números más bajos en V8. Por ejemplo, los controles de ofuscación listados en la sección "Impedir la comprensión" deben combinarse con el "aislamiento de la aplicación", "impedir el análisis dinámico y la manipulación" y la "atadura al dispositivo".

Tenga en cuenta que los controles de software nunca deben utilizarse como reemplazo de los controles de seguridad. Los controles listados en MASVR-R buscan añadir controles de protección adicionales y específicos contra las amenazas a las aplicaciones que también cumplen con los requerimientos de seguridad del MASVS.

Se aplican las siguientes consideraciones:

i. Debe definirse un modelo de amenaza que defienda claramente las amenazas del lado del cliente. Además, debe especificarse el grado de protección que debe proporcionar el sistema. Por ejemplo, un objetivo podría ser obligar a los autores de malware dirigido que quieren usar la aplicación a que tengan que invertir importantes esfuerzos para realizar la ingeniería inversa.

ii. El modelo de amenaza debe ser sensato. Por ejemplo, ocultar una clave criptográfica en una implementación de caja blanca es un problema si el atacante puede simplemente utilizar la aplicación como un todo.

iii. La eficiencia de la protección siempre debe ser verificada por un experto con experiencia en el testeo de tipos particulares de anti manipulación y ofuscación utilizados (ver también los capítulos "ingeniería inversa" y "evaluación de protecciones del software" en la Guía de Pruebas de Seguridad Móvil).

Impedir el Análisis Dinámico y la Manipulación

#	Descripción	R
8.1	La aplicación detecta y responde a la presencia de un dispositivo ruteado, ya sea alertando al usuario o finalizando la ejecución de la aplicación.	✓

8.2	La aplicación previene el debugging o detecta y responde al debugging de la aplicación. Se deben cubrir todos los protocolos.	✓
8.3	La aplicación detecta y responde a modificaciones de ejecutables y datos críticos de la propia aplicación.	✓
8.4	La aplicación detecta la presencia de las herramientas de ingeniería reversa o frameworks mas utilizados.	✓
8.5	La aplicación detecta y responde al ser ejecutada en un emulador.	✓
8.6	La aplicación detecta y responde ante modificaciones de código o datos en su propio espacio de memoria.	✓
8.7	La aplicación implementa múltiples mecanismos de detección para los puntos del 8.1 al 8.6. Notese que a mayor cantidad y diversidad de mecanismos usados, mayor la resistencia.	✓
8.8	Los mecanismos de detección disparan distintos tipos de respuestas, incluyendo respuestas retardadas y silenciosas.	✓
8.9	La ofuscación es aplicada a las defensas del programa, lo que a su vez impide la des-ofuscación mediante el análisis dinámico.	✓

Atadura del Dispositivo

#	Descripción	R
8.10	– La aplicación implementa un “enlace al dispositivo” utilizando una huella del dispositivo derivado de varias propiedades únicas al mismo.	✓

Impede Comprehension

#	Descripción	R
8.11	Todos los archivos ejecutables y bibliotecas correspondientes a la aplicación se encuentran cifrados, o bien los segmentos importantes de código se encuentran cifrados o empaquetados. De este modo el analisis estático trivial no revela código importante o datos.	✓
8.12	Si el objetivo de la ofuscación es proteger el código propietario, se utiliza un esquema de ofuscación que es apropiado tanto para la tarea particular como robusto contra los métodos de des-ofuscación manual y automatizada, considerando la investigación publicada actualmente. La eficacia del esquema de confusión debe verificarse mediante pruebas manuales. Tenga en cuenta que las características de aislamiento basadas en hardware son preferibles a la ofuscación siempre que sea posible.	✓

Referencias

La Guía de Pruebas de Seguridad Móvil de OWASP proporciona instrucciones detalladas para verificar los requisitos listados en esta sección.

- Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05j-Testing-Resiliency-Against-Reverse-Engineering.md
- iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06j-Testing-Resiliency-Against-Reverse-Engineering.md

Para más información, ver también:

- Top 10 Móvil de OWASP: M8 - Modificación de Código, M9 - Ingeniería Inversa
- Amenazas de la Ingeniería Inversa de OWASP - https://www.owasp.org/index.php/Technical_Risks_of_Reverse_Engineering_and_Unauthorized_Code_Modification
- Ingeniería Inversa y Prevención de Modificación de Código de OWASP - https://www.owasp.org/index.php/OWASP_Reverse_Engineering_and_Code_Modification_Prevention_Project

Apéndice A: Glosario

- **2FA** - La Autenticación de Segundo Factor agrega un segundo nivel de autenticación al proceso de autenticación de una cuenta.
- **Randomización del Espacio de Direcciones (ASLR)** - Una técnica para hacer más difícil la explotación de errores de corrupción de memoria.
- **Seguridad de Aplicación** - La seguridad a nivel de aplicación se centra en el análisis de los componentes que componen la capa de aplicación del modelo de referencia de interconexión de sistemas abiertos (modelo OSI), en lugar de centrarse, por ejemplo, en el sistema operativo subyacente o las redes conectadas.
- **Verificación de la Seguridad de una Aplicación** - La evaluación técnica de una aplicación contra el OWASP MASVS.
- **Reporte de la Verificación de la Seguridad de una aplicación** - Un informe que documenta los resultados generales y el análisis de apoyo producido por el verificador para una aplicación particular.
- **Autenticación** - La verificación de la identidad pretendida por un usuario de una aplicación.
- **Verificación Automatizada** - El uso de herramientas automatizadas (herramientas de análisis dinámico, herramientas de análisis estático o ambas) que utilizan firmas de vulnerabilidades para encontrar problemas.
- **Verificación de Caja Negra** - Es un método de verificación de software que examina la funcionalidad de una aplicación sin fijarse en sus estructuras internas ni en su funcionamiento.
- **Componente** - Una unidad de código autónoma, asociado con un disco e interfaces de red que se comunican con otros componentes.
- **Cross-Site Scripting** (XSS) - Una vulnerabilidad de seguridad que normalmente se encuentra en las aplicaciones web que permiten la inyección de scripts del lado del cliente en el contenido.
- **Módulo Criptográfico** - Hardware, software y/o firmware que implementa algoritmos criptográficos y/o genera claves criptográficas.
- **CWE** - CWE es una lista desarrollada por la comunidad de debilidades comunes de seguridad de software. Sirve como un lenguaje común, un instrumento de medición para las herramientas de seguridad de software, y como una línea base para la identificación de debilidades, mitigación y esfuerzos de prevención.
- **DAST** - Las tecnologías de Pruebas de Seguridad de Aplicaciones Dinámicas (DAST) están diseñadas para detectar condiciones indicativas de una vulnerabilidad de seguridad en una aplicación mientras se está ejecutando.

- **Verificaciones de Diseño** - La evaluación técnica de la seguridad de la arquitectura de una aplicación.
- **Verificación Dinámica** - El uso de herramientas automatizadas que utilizan firmas de vulnerabilidades para encontrar problemas durante la ejecución de una aplicación.
- **Identificado Único Global** (GUID) - Un número de referencia único utilizado como identificador en el software.
- **Hyper Text Transfer Protocol** (HTTP) - Un protocolo de aplicación para sistemas de información distribuidos, colaborativos. Es la base de la comunicación de datos para la World Wide Web.
- **Claves a fuego (Hardcoded keys)** - Claves Criptográficas que se encuentran almacenadas en el código en el dispositivo.
- **IPC** - Comunicación Entre Procesos (Inter Process Communications), En la Comunicación Entre Procesos un proceso se comunica con otro a través del kernel del dispositivo para coordinar sus actividades.
- **Validación de la Entrada** - La canonización y validación de las entradas de usuario no confiables.
- **JAVA Bytecode** - Java bytecode es el conjunto de instrucciones de la máquina virtual Java (JVM). Cada bytecode está compuesto por uno, o en algunos casos dos bytes que representan la instrucción (opcode), junto con cero o más bytes para pasar parámetros.
- **Código Malicioso** - Código introducido en una aplicación durante su desarrollo desconocido para el propietario de la aplicación, que elude la política de seguridad pretendida de la aplicación. ¡No es lo mismo que malware o un virus o gusano!
- **Malware** - Código ejecutable que se introduce en una aplicación durante el tiempo de ejecución sin el conocimiento del usuario o administrador de la aplicación.
- **Open Web Application Security Project** (OWASP) - El Open Web Application Security Project (OWASP) es una comunidad abierta y gratuita a nivel mundial enfocada en mejorar la seguridad del software de aplicaciones. Nuestra misión es hacer que la seguridad de las aplicaciones sea "visible" para que las personas y las organizaciones puedan tomar decisiones informadas sobre los riesgos de seguridad de las aplicaciones. Ver: http://www.owasp.org/
- **Información de Identificación Personal** (Personally Identifiable Information - PII) - es la información que se puede utilizar por sí sola o con otra información para identificar, contactar o localizar a una sola persona, o para identificar a un individuo en su contexto.
- **PIE** - es un código máquina que, al ser colocado en algún lugar de la memoria, se ejecuta correctamente independientemente de su dirección absoluta.
- **PKI** - PKI es un acuerdo que vincula claves públicas con las identidades respectivas de las entidades. La vinculación se establece mediante un proceso de registro y expedición de certificados en y por una autoridad de certificación (CA).
- **SAST** - Las pruebas de seguridad de aplicaciones estáticas (SAST) son un conjunto de tecnologías diseñadas para analizar el código fuente de la aplicación, el bytecode y los binarios del código y las condiciones del diseño que son indicativas de las vulnerabilidades de seguridad. Las soluciones SAST analizan una aplicación desde "dentro hacia fuera" en un estado reposo.
- **SDLC** - Ciclo de desarrollo de una aplicación.
- **Seguridad de la Arquitectura** - Una abstracción del diseño de una aplicación que identifica y describe dónde y cómo se utilizaran los controles de seguridad, e identifica y describe la ubicación y sensibilidad de los datos tanto del usuario como de la aplicación.

- **Configuración de Seguridad** - La configuración en tiempo de ejecución de una aplicación que afecta a la forma en que se utilizan los controles de seguridad.
- **Control de Seguridad** - Una función o componente que realiza un chequeo de seguridad (por ejemplo, una verificación del control de acceso) o cuando se llama produce un evento de seguridad (por ejemplo, al generar un registro de auditoría).
- **Inyección SQL (SQLi)** - Una técnica de inyección de código utilizada para atacar aplicaciones basadas en datos, en la que se insertan instrucciones SQL maliciosas en un punto de entrada.
- **Autenticación SSO** - Single Sign On(SSO) se produce cuando un usuario inicia sesión en un Cliente y luego se conecta a otros Clientes automáticamente, independientemente de la plataforma, tecnología o dominio que esté utilizando el usuario. Por ejemplo, cuando te conectas en Google, automáticamente accedes al servicio de YouTube, Drive y Gmail.
- **Modelado de Amenazas** - Una técnica que consiste en desarrollar arquitecturas de seguridad cada vez más perfeccionadas para identificar agentes de amenazas, zonas de seguridad, controles de seguridad e importantes activos técnicos y empresariales.
- **Seguridad en la Capa de Transporte (TLS)** - Protocolos criptográficos que proporcionan seguridad en las comunicaciones a través de Internet.
- **Fragmentos URI/URL/URL** - Un Identificador Uniforme de Recursos es una cadena de caracteres que se utiliza para identificar un nombre o un recurso web. Un Localizador Uniforme de Recursos se utiliza a menudo como referencia a un recurso.
- **Pruebas de Aceptación de Usuario (UAT)**- Tradicionalmente un entorno de pruebas que se comporta como el entorno de producción donde se realizan todas las pruebas de la aplicación antes de su puesta en marcha.
- **Verificador** - La persona o equipo que está revisando una aplicación contra los requerimientos del MASVS de OWASP.
- **Lista Blanca** - Una lista de datos u operaciones permitidas, por ejemplo, una lista de caracteres que permiten realizar la validación de la entrada.
- **Certificado X.509** - Un certificado X. 509 es un certificado digital que utiliza el estándar internacional de infraestructura de clave pública (PKI) X. 509 ampliamente aceptada para verificar que una clave pública pertenece a la identidad de usuario, computadora o servicio contenida en el certificado.

Apéndice B: Referencias

Los siguientes proyectos de OWASP probablemente sean de utilidad para los usuarios de este estándar:

- Proyecto de Seguridad Móvil de OWASP - https://www.owasp.org/index.php/OWASP_Mobile_Security_Project
- Guía de Pruebas de Seguridad Móvil OWASP - https://www.owasp.org/index.php/OWASP_Mobile_Security_Testing_Guide
- Top 10 Riesgos Móviles de OWASP - https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Top_Ten_Mobile_Risks
- Ingeniería Inversa y Prevención de Modificación de Código de OWASP - https://www.owasp.org/index.php/OWASP_Reverse_Engineering_and_Code_Modification_Prevention_Project

También los siguientes sitios web probablemente sean de utilidad para los usuarios de este estándar:

- Enumeración de debilidades comunes de MITRE - http://cwe.mitre.org/
- Consejo de Normas de Seguridad PCI - https://www.pcisecuritystandards.org
- Estándar de seguridad de datos PCI (DSS) v3.0 Requerimientos y procedimientos de evaluación de la seguridad https://www.pcisecuritystandards.org/documents/PCI_DSS_v3.pdf