Universidad Cenfotec

Maestría en Ciberseguridad

Documento final de Proyecto de Investigación Aplicada 2

Propuesta de una metodología de auditoría de seguridad en aplicaciones móviles para PyMEs con políticas de BYOD

Gamboa Porras Erick
Morales Solano Daniela

Agosto, 2020

Declaratoria

Se declara que el presente proyecto de investigación fue realizado por los autores Erick Gamboa Porras y Daniela Morales Solano, fundamentando los diferentes capítulos del trabajo en diferentes fuentes bibliográficas, literatura citada, las cuales tienen su respectiva referencia, respetando los derechos de autor de dichos trabajos.

Se autoriza la reproducción total o parcial de este trabajo para ser usados como referencia de trabajos futuros de tipo académico y científico, en este caso, se solicita incorporar la referencia de este trabajo tomando en consideración los derechos de sus autores.

Agradecimiento

Le agradecemos a Dios por habernos acompañado y guiado a lo largo de nuestras carreras, por ser nuestra fortaleza en los momentos de debilidad y por brindarnos una vida llena de aprendizajes, experiencias y sobre todo felicidad.

Queremos agradecer a nuestras familias por su apoyo, comprensión y por la motivación para culminarla nuestra Maestría de manera exitosa.

Muy especialmente, agradecemos a nuestro profesor tutor Carlos Calvo Muñoz, por su orientación y recomendaciones durante todo el proceso de investigación y desarrollo del trabajo final de graduación, las cuales fueron claves para su exitosa culminación.

Dedicatoria

Nos gustaría dedicar esta tesis a nuestras familias.

A nuestros padres por su comprensión y ayuda en momentos malos y buenos. Nos han enseñado a encarar las adversidades sin perder nunca la fé ni desfallecer en el intento. Nos han dado todo lo que somos como persona, nuestros valores, principios, perseverancia y empeño y todo ello con una gran dosis de amor y sin pedir nunca nada a cambio.

universidad cenfotec_
tecnologías digitales

Universidad Cenfotec
Carrera de Postgrado
Maestría en Ciberseguridad

TRIBUNAL EXAMINADOR

Este proyecto fue aprobado por el Tribunal Examinador de la carrera: **Maestría en Ciberseguridad,** requisito para optar por el título de grado de **Maestría,** para los estudiantes: **Erick Alberto Gamboa Porras y Daniela Arausi Morales Solano**

M. Sc. Carlos Calvo Muñoz
Tutor

MSEG. Esteban Alpírez Monge
Lector 1

IGNACIO TREJOS ZELAYA (FIRMA)
Firmado digitalmente por IGNACIO TREJOS ZELAYA (FIRMA)
Fecha: 2020.09.07 10:27:26 -07'00'

M. Sc. Ignacio Trejos Zelaya
Lector 2

San José, Costa Rica, 03 de setiembre de 2020

Firmado digitalmente, de conformidad con la Ley de Certificados, Firmas Digitales y Documentos Electrónicos N° 8454, destacando el artículo 9°-.

Tabla de contenidos

Tabla de figuras

Tablas

Abstract

Los avances tecnológicos suceden rápidamente. Hace unos años atrás, los celulares eran dispositivos “tontos” con pequeños teclados. Actualmente, 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. 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. 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.

Como es natural, la competitividad y crecimiento de las PyMEs van de la mano con las tecnologías en informática, comunicación y las aplicaciones móviles, gracias a la manera en que estas mejoran la eficiencia de muchos procesos en favor de los pequeños empresarios.

Las pequeñas y medianas empresas, PyMEs, representan el grueso del ecosistema empresarial del país, ya que conforman el 75,5% de las compañías formales que hay en Costa Rica. De ahí la importancia de asegurar su información.

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, estas deben ser incluidas en las aplicaciones y usadas correctamente para ser efectivas.

Palabras clave: investigación, auditoría, ciberseguridad, aplicaciones móviles, OWASP, seguridad móvil, riesgos de seguridad, PyMEs, BYOD.

Capítulo 1. Introducción

1.1 Generalidades

Este proyecto se presenta como una iniciativa personal de los miembros que conforman el grupo de investigación, al observar el avance de la tecnología móvil en los últimos años, la facilidad de adquirir los dispositivos móviles y la alta demanda de aplicaciones para mejorar las actividades de las PyMEs.

Para tener un mejor entendimiento de esta investigación, es importante tomar en cuenta las siguientes abreviaturas:

APP/Apps: Aplicaciones.

APIs: Application Programming Interface

APK: Android application package

BYOD: Bring your own device

MASVS: Mobile Application Security Verification Standard

OWASP: Open Web Application Security Project

PyMEs: Pequeña y mediana empresa

1.2 Antecedentes del problema

Las pequeñas y medianas empresas, PyMEs, tienen particular importancia para la economía nacional, no solo por sus aportaciones a la producción y distribución de bienes y servicios, sino también por la flexibilidad de adaptarse a los cambios tecnológicos y el gran potencial de generación de empleos.

Con los avances tecnológicos enfocados en la informática y los dispositivos móviles, el uso de aplicaciones móviles en las PyMEs se ha incrementado considerablemente. Esto permite que cada vez se expanda más el sector, atrayendo a su vez a un mayor número de clientes; sin embargo, no todo son ventajas cuando se habla de estos avances e incorporación de dispositivos móviles a las empresas PyMEs, ya que lamentablemente la seguridad de dichos dispositivos y sus aplicaciones, es un tema al que hoy no se le presta mucha atención.

1.3 Definición y descripción del problema

Este proyecto tiene como objetivo implementar auditorías de seguridad en los dispositivos móviles utilizados en las PyMEs con políticas de BYOD. En la actualidad, las PyMEs realizan el mercadeo de sus productos por medio de aplicaciones o redes sociales ya que son parte fundamental del día a día y les facilita poder entrar en contacto con el cliente en cualquier lugar y momento del día.

Una empresa que no conozca suficiente sobre el uso que se hace de los dispositivos móviles por parte de sus empleados (los dispositivos de uso corporativo) es una compañía en riesgo.

La manera más estratégica de contrarrestar este fenómeno en las organizaciones es hacer auditorías de seguridad de los dispositivos móviles y sus aplicaciones. Asimismo, usar herramientas de gestión dispositivos móviles sin convertirse en una herramienta restrictiva para los usuarios.

1.4 Justificación

La principal razón por la que se busca desarrollar un plan de auditorías de seguridad de aplicaciones móviles, es para detectar vulnerabilidades y riesgos, y a su vez, realizar recomendaciones sobre las medidas de seguridad que deben de tomar en cuenta las PyMEs con políticas de BYOD y sus empleados, para que de esta forma se pueda asegurar la información confidencial de la empresa y sus clientes.

Adicionalmente, se espera que el presente proyecto sirva como referencia para que cualquier empresa con un perfil PyME, pueda realizar proactivamente un análisis de seguridad y conocer el estado de los dispositivos móviles, antes de que se presente un posible incidente de seguridad, generando un impacto negativo, pérdidas económicas o daños en la imagen de la empresa, afectando las oportunidades de negocio y/o clientes.

1.5 Viabilidad

1.5.1 Punto de vista técnico

Como profesionales y futuros másteres en Ciberseguridad, los autores cuentan con el conocimiento y experiencia en la implementación y evaluación de auditorías de seguridad, tratamiento de riesgos y conocimientos para llevar a cabo pruebas prácticas en dispositivos móviles.

Igualmente, se espera que, como parte del proceso investigativo, se recopile la experiencia de diversos autores en la implementación de auditorías de seguridad para dispositivos móviles que puedan ser aplicados al problema que se pretende abordar.

1.5.2 Punto de vista operativo

Esta investigación no altera de ninguna manera el funcionamiento normal de las empresas debido a tratarse de un enfoque en la seguridad de la información de las PyMEs de Costa Rica, a través de auditorías de seguridad de dispositivos móviles y considerando que la mayoría utiliza aplicaciones móviles ya existentes y gratuitas.

1.5.3 Punto de vista económico

Se utilizarán para el desarrollo de esta investigación, herramientas de escaneos de aplicaciones gratuitas que permitirán identificar las vulnerabilidades de las aplicaciones móviles, sistemas operativos de código libre y ambientes de virtualización gratuitos, por lo que se espera llevar a cabo la investigación sin ningún costo económico.

1.6 Objetivos

1.6.1 Objetivo general

Proponer una metodología para realizar auditorías de seguridad de aplicaciones móviles, con el fin de detectar riesgos y posibles vulnerabilidades de seguridad en PyMEs con policitas de BYOD.

1.6.2 Objetivos específicos

- Conocer las herramientas disponibles para realizar auditorías en aplicaciones móviles.
- Identificar riesgos y vulnerabilidades actuales de las aplicaciones móviles.
- Investigar los frameworks de seguridad para aplicaciones móviles.
- Desarrollar una prueba práctica para identificar las vulnerabilidades y riesgos en una aplicación existente.
- Establecer controles para reducir el impacto de los riesgos identificados en aplicaciones móviles.
- Recomendar una metodología que se adapte como base para realizar auditorías de seguridad en aplicaciones móviles para las PyMEs con políticas de BYOD.

1.7 Alcances y limitaciones

1.7.1 Alcances

El alcance de la investigación se centra en proponer una metodología de auditorías de seguridad para aplicaciones móviles con base en los estándares MASVS y MSTG de OWASP para las PyMEs.

- Metodología de auditorías de seguridad de aplicaciones móviles basada en los estándares MASVS y MSTG de OWASP.
- Recomendación de herramientas gratuitas que se pueden utilizar para realizar escaneos de aplicaciones móviles.
- Resultados del caso práctico (análisis de vulnerabilidades de aplicaciones móviles).
- Recomendar controles para reducir el impacto de los riesgos de seguridad de aplicaciones móviles.

1.7.2 Limitaciones

- No se considera como parte del proyecto, desarrollar una herramienta que realice de manera automática la auditoría de una aplicación móvil.
- Se limita el desarrollo de la prueba práctica para aplicaciones desarrolladas para el sistema operativo Android.
- La metodología propuesta no se va a implementar en una PyME como parte de este proyecto.

1.8 Estado de la cuestión

Actualmente, la tecnología avanza muy rápido y con ella los dispositivos móviles, por lo que cada vez son más sus usuarios alrededor de todo el mundo. De acuerdo con las estadísticas recopiladas de la página web Newzoo (2020), el número de usuarios de celulares inteligentes sobrepasaba los 3.6 billones (casi la mitad de la población del mundo). De ellos, más de la mitad de los usados son ya dispositivos inteligentes y de acuerdo con ese mismo estudio se estima que en promedio un dispositivo inteligente en Estados Unidos tiene unas 41 aplicaciones descargadas por cada usuario. Las aplicaciones que tienen acceso a datos sensibles e información de los usuarios que no están siendo auditados o revisados por los usuarios ni por las empresas para las que laboran.

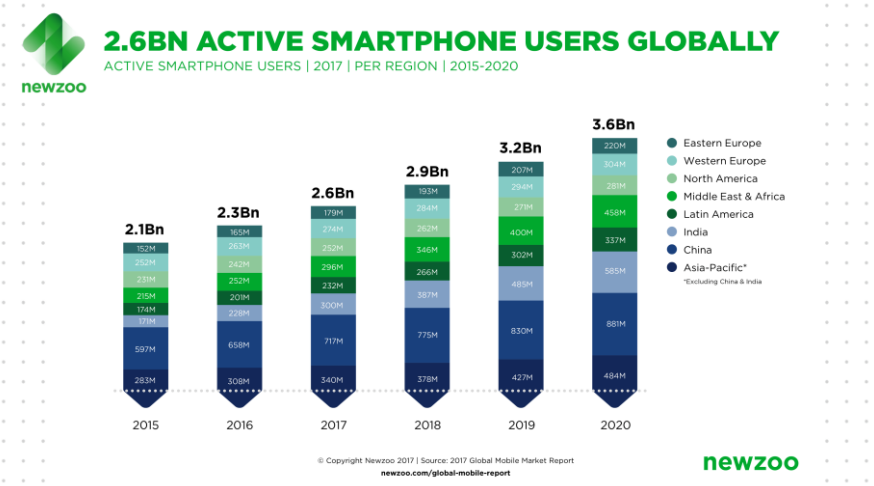


Figura 1. Usuarios activos de teléfonos inteligentes.

Fuente Newzoo

1.8.1 Planificación de la revisión

Se formula en esta etapa, una pregunta clara y bien definida del tema de investigación. Se hace una búsqueda de la documentación existente sobre el tema con el objetivo de conocer el desarrollo académico que existe, las posibles áreas débiles que se puedan ampliar y verificar que no se estén duplicando los estudios realizados en otras investigaciones.

1.8.1.1 Formulación de la pregunta

La formulación de la pregunta ayuda a delimitar los esfuerzos de búsqueda de información e investigación. El objetivo es encontrar respuestas que demuestren la contribución de este trabajo al campo de investigación y las relaciones entre ideas, teoría y aplicación práctica.

1.8.1.1.1 Foco de la pregunta

Se requiere para la presente investigación centralizar la búsqueda de documentos técnicos que especifiquen las mejores prácticas, estándares y herramientas para el análisis de vulnerabilidades en los dispositivos móviles.

1.8.1.1.2 Amplitud y calidad de la pregunta

Se establece en esta sección la pregunta de investigación que se desea responder de forma clara y concisa, basados en un problema por resolver. Se hace un listado de términos clave, relevantes para la búsqueda de información y se consideran componentes clave como los tipos de aplicaciones móviles, seguridad móvil y estándares. Se definen las medidas por utilizar para medir el efecto con base en la pregunta por responder y el diseño de los estudios.

1. Problema

Las tendencias de crecimiento de descargas de aplicaciones móviles en los últimos años aumentan el crecimiento exponencial de las amenazas de seguridad para los dispositivos móviles. Amenazas que no están siendo auditadas o revisadas por las PyMEs para las cuales laboran los usuarios. El gran problema radica en que los usuarios conectan sus dispositivos móviles a las redes de las compañías y a su vez, descargan a sus dispositivos, aplicaciones utilizadas con fines laborales como WhatsApp e Instagram, las cuales manejan información personal y sensible de la compañía.

2. Pregunta

Se formula la siguiente pregunta de investigación con la anterior definición del problema:

¿Qué trabajos se han llevado a cabo en el área de seguridad en los dispositivos móviles para la identificación de amenazas y análisis de vulnerabilidades en las PyMEs?

3. Palabras claves y sinónimos

Se hace un listado de palabras clave que se van a utilizar para la búsqueda e identificación de documentos y trabajos relacionados con la investigación. Algunas de estas palabras están en el idioma inglés, ya que hay un grueso de publicaciones en ese idioma, las cuales se muestran en la Tabla 1. A continuación, un listado de estas palabras:

Palabra	Equivalente en inglés
Auditorías	Audtis
Seguridad móvil	Mobile security
Aplicaciones móviles	Mobile Applications
OWASP	OWASP
Análisis	Analysis
Vulnerabilidades	Vulnerabilities
Amenazas	Threats

Tabla 1. Palabras claves.
Fuente: elaboración propia.

4. Intervención

Extraer los artículos y documentos de mayor relevancia para la investigación y analizar los resultados obtenidos.

5. Control

Al iniciar la investigación, no se cuenta con ninguna base de información. Se empieza con una búsqueda de cero a partir de las palabras clave definidas.

6. Efectos

Se espera tener documentación suficiente con las búsquedas realizadas para entender cuáles son las amenazas a las que se exponen los usuarios de dispositivos móviles a la hora de descargar aplicaciones móviles y a su vez, definir cuál es el mejor estándar para poder auditar los dispositivos móviles.

7. Medida de salida

Se realiza para la documentación encontrada una revisión de la calidad en sitios web especializados para tal fin.

8. Aplicación

Este tipo de investigación puede resultar de utilidad para personas en el área de Ingeniería en Computación, Ingeniería en Sistemas, tecnología, Ciberseguridad y estudiantes que tengan interés en conocer cómo se puede auditar los dispositivos móviles y las amenazas a los que estos están expuestos.

9. Diseño experimental

Se hace un análisis y clasificación de los estudios obtenidos durante el diseño experimental, basándose en la calidad del contenido y relevancia para la investigación. Con lo anterior, se garantiza no solo contar con la documentación de mayor confianza para la investigación, sino también la suficiente, con lo cual se evita tener un rango demasiado amplio de estudios que pudieran generar resultados no deseados.

1.8.1.2 Selección de fuentes

Se especifican en esta sección las fuentes para la identificación de estudios primarios que se utilizarán para la investigación.

1.8.1.2.1 Definición del criterio de selección de fuentes

Se han tomado en cuenta para la selección de fuentes, en general, varios aspectos como la popularidad entre investigadores y el respaldo teórico con que cuenta la fuente. También se consideran las que cuenten con gran variedad de documentación y con relevancia vigente. Asimismo, algo que se considera importante es la facilidad de acceso de la información y poder contar con credenciales para acceder a los documentos.

1.8.1.2.2 Lenguaje de estudio

Se utiliza para el estudio tanto el idioma español como el inglés en lo que respecta a las búsquedas, de esta manera, se puede incrementar el rango de posibles resultados.

1.8.1.2.3 Identificación de fuentes

Se describe en este apartado la selección de fuentes para la documentación primaria, se hace una descripción acerca de cómo se ejecutan las búsquedas y se provee una lista de fuentes.

1. **Método de selección de fuentes**

El método de selección de fuentes se basa principalmente en el respaldo con el que cuenta la fuente en el área de tecnología con respecto a la publicación de estudios y documentos investigativos. Además, se considera la facilidad de acceso al sitio y para hacer las búsquedas.

2. **Cadena de búsqueda:**

Cadenas de búsqueda utilizadas tienen combinación de “OR” y “AND”.
(“Mobile Applications” AND “audits”) AND (("audits" OR "auditorías" OR “mobile security”) AND (“vulnerabilidades” OR “vulnerabilities”)) OR (("audits" OR “auditorías " OR “mobile security”) AND (“amenazas” OR “threats”)) OR (("audits" OR "auditorías" OR “mobile security”) AND (“OWASP” OR “OWASP”))

3. **Lista de fuentes**

Debido a la calidad de los artículos académicos y cantidad de artículos recientes, de acuerdo con el tema investigado, se considera la utilización de las siguientes fuentes:

1. ACM Digital
2. Google Scholar

1.8.1.2.4 Selección de fuentes después de la evaluación

Los elementos para refinar la lista de fuentes dependen de la facilidad de aplicación de las cadenas de búsqueda y la calidad de los documentos brindados. Un aspecto que se toma en cuenta es la facilidad que se tiene para acceder al material.

1.8.1.2.5 Comprobación de las fuentes

No se cuenta en este momento con criterio experto para la selección de las fuentes; sin embargo, se escogieron las más utilizadas para obtener documentación relacionada con tecnología, entre otras ramas. Se espera obtener un criterio experto para refinar la lista o agregar más, según sea necesario.

1.8.1.3 Selección de los estudios

Ya definidas las fuentes, se define cuáles trabajos recuperados en las búsquedas van a ser incluidos en el análisis final.

1.8.1.3.1 Definición del criterio de inclusión y exclusión de estudios

Se utilizan los criterios detallados en la tabla 2, para incluir o excluir un documento. Los artículos que cumplan los requisitos son candidatos por incluir.

Pregunta de investigación	Criterio de inclusión	Criterio de exclusión
¿Qué trabajos se han llevado a cabo en el área de seguridad en los dispositivos móviles para la identificación de amenazas y análisis de vulnerabilidades en las PyMEs?	"Auditorías" o "Audits" "Seguridad móvil" o "Mobile security" "Aplicaciones móviles" o "Mobile Applications" "OWASP" "Análisis" o "Analysis" "Vulnerabilidades" o "Vulnerabilities" "Amenazas" o "Threads"	*Documentos de Dispositivos móviles que no tengan relación con seguridad móvil. * Documentos de auditorías que no tengan relación con dispositivos móviles. *Análisis de vulnerabilidades de seguridad que no tengan relación con los dispositivos móviles.

Tabla 2. Criterio de inclusión y exclusión de estudios.
Fuente: elaboración propia.

1.8.1.3.2 Definición de tipos de estudio

La definición de los tipos de estudios está relacionada con la pregunta de investigación, se crea la siguiente tabla para determinar los requisitos para definir los artículos de interés.

Pregunta de investigación	¿Quién?	¿Qué?	¿Cómo?	¿Dónde?
¿Qué trabajos se han llevado a cabo en el área de seguridad en los dispositivos móviles para la identificación de amenazas y análisis de vulnerabilidades en las PyMEs?	Los usuarios de dispositivos móviles y las PyMEs con políticas BYOD.	Aplicaciones móviles, vulnerabilidades y riesgos.	Análisis de vulnerabilidades y auditorías móviles.	Dispositivos móviles y PyMEs.

Tabla 3. Tipos de estudios.
Fuente: elaboración propia.

1.8.1.3.3 Procedimiento para la selección de los estudios

Se realizó el siguiente proceso iterativo por cada fuente para la selección de los estudios relevantes:

1. Utilizar la opción de búsqueda avanzada o búsqueda general disponible en las fuentes seleccionadas.
2. Con base en la cantidad de resultados obtenidos, emplear las respectivas cadenas de búsqueda aplicables para obtener resultados considerados de interés o que satisfagan los criterios de inclusión.
3. Si al ejecutar el paso dos, la lista de resultados es superior a los 50, aplicar filtros o cadenas adicionales para disminuirla, por ejemplo, aplicando rangos de fechas de los estudios.
4. Evaluar los resultados obtenidos y aplicar los criterios de exclusión basados en el abstract y palabras claves del artículo.
5. Seleccionar los resultados considerados relevantes para la fuente consultada y repetir el proceso con las demás fuentes disponibles.

1.8.2 Ejecución de la revisión

Se presenta a continuación, el proceso de selección llevado a cabo para las diferentes fuentes.

1.8.2.1 Ejecución de la selección en la fuente ACM

1.8.2.1.1 Selección de estudios iniciales

Siguiendo las recomendaciones provistas por Blanco et al. (2007), se realiza la búsqueda de estudios iniciales de la siguiente manera:

Búsqueda basada en los siguientes parámetros:

- Mobile applications
- Audits
- OWASP

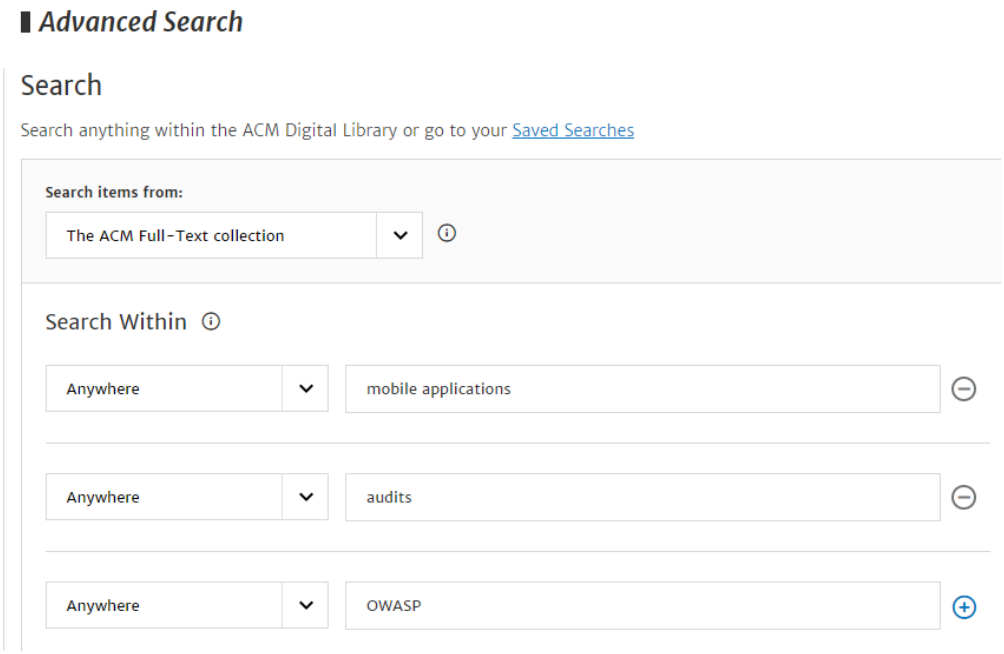


Figura 2. Ejecución de la selección.
Fuente ACM

Tras realizar la búsqueda utilizando los parámetros mencionados, se encontraron ciento cincuenta y nueve resultados, de los cuales fueron seleccionados cinco artículos tras aplicar el método de exclusión propuesto. A continuación, se presenta el detalle:

#	Título	Autores	Año	URL
1	Security During Application Development: An Application Security Expert Perspective	*Tyler W Thomas *Madiha Tabassum *Bill Chu *Heather Lipford	2018	https://dl.acm.org/doi/pdf/10.1145/3173574.3173836
2	Causality-based Sensemaking of Network Traffic for Android Application Security∗	*HaoZhang *Danfeng *Yao *Naren Ramakrishnan	2017	https://dl.acm.org/doi/pdf/10.1145/2996758.2996760
3	Security in Mobile Communications: Challenges and Opportunities	*Audun Jøsang *Gunnar Sanderud	2018	https://dl.acm.org/doi/pdf/10.5555/827987.827993
4	The Current Practices	*Ameerah Muhsinah	2017	https://dl.acm.org/doi/pdf

	of Changing Secure Software	*Lotfi ben *Altaz Valani		/10.1145/3341105.3373922
5	A Data Mining Approach to Assess Privacy Risk in Human Mobility Data	*Roberto Pellungrin *Uca Pappalardo *Francesca Pratesi	2017	https://dl.acm.org/doi/pdf/10.1145/3106774

Tabla 4. Estudios encontrados ACM.
Fuente: elaboración propia.

1.8.2.1.2 Evaluación de la calidad de los estudios

Se presume la calidad de los artículos mencionados, con base en la cantidad de filtros y evaluaciones realizados por ACM.

1.8.2.1.3 Revisión de la selección

La selección de estudios primarios se realiza tras llevar a cabo una revisión de los abstracts y contenido incluido en cada artículo. Para su revisión, fueron ordenados con base en su relevancia.

1.8.2.1.4 Extracción de información

Se consideran los siguientes elementos para la extracción de la información relevante de los estudios primarios y el cumplimiento de los objetivos de la investigación:

- Seguridad de aplicaciones móviles.
- Riesgos y vulnerabilidades actuales de las aplicaciones móviles.
- OWASP.
- Auditorías de seguridad.

Repositorio	**ACM Digital Library**
Título	Security During Application Development: An Application Security Expert Perspective
Año de publicación	2018
Autores	Tyler W Thomas, Madiha Tabassum, Bill Chu and Heather Lipford
Resumen	Muchos de los problemas de seguridad a los que se enfrenta hoy la gente, como las brechas de seguridad y el robo de datos, son causados por vulnerabilidades de seguridad en el código fuente de las aplicaciones. Por lo tanto, existe la necesidad de entender y mejorar las experiencias de aquellos que pueden prevenir tales

	vulnerabilidades, en primer lugar, los desarrolladores de software, así como los expertos en seguridad de aplicaciones.
Aspectos importantes	El artículo destaca que es de vital importancia que los desarrolladores incorporen las medidas de seguridad en las primeras fases del desarrollo de las aplicaciones. Además, realizan un estudio que comprueba lo establecido.

Tabla 5. Extracción fuente 1.
Fuente: elaboración propia.

Repositorio	**ACM Digital Library**
Título	Causality-based Sensemaking of Network Traffic for Android Application Security∗
Año de publicación	2017
Autores	Hao Zhang, Danfeng (Daphne) Yao, and Naren Ramakrishnan
Resumen	Las aplicaciones maliciosas de Android representan una serie de amenazas para la seguridad móvil. Amenazan la confidencialidad de los datos y la integridad del sistema en los dispositivos Android.
Aspectos Importantes	El paper propone un modelo de relación de activación para analizar dinámicamente el tráfico de red en dispositivos Android.

Tabla 6. Extracción fuente 2.
Fuente: elaboración propia.

Repositorio	**ACM Digital Library**
Título	Security in Mobile Communications: Challenges and Opportunities
Año de publicación	2018
Autores	Audun Jøsang and Gunnar Sanderud
Resumen	Este paper analiza algunas de las dificultades que enfrentan los arquitectos de sistemas, así como algunas ventajas que ofrecen las redes móviles al diseñar soluciones de seguridad para sus comunicaciones.
Aspectos Importantes	Plantea requerimientos de seguridad básicos que las aplicaciones debe cumplir. Explica las medidas de seguridad que se deben tomar en las conexiones de red.

Tabla 7. Extracción fuente 3.
Fuente: elaboración propia.

Repositorio	ACM Digital Library
Título	The Current Practices of Changing Secure Software
Año de publicación	2017
Autores	Ameerah Muhsinah Jamil, Lotfi ben Othmane, Altaz Valani, Moataz Abdelkhalek and Ayhan Tek
Resumen	Este paper informa un estudio cualitativo de las prácticas de cambio a software seguro en la industria. El estudio involucra entrevistas con desarrolladores y expertos en seguridad. A través de dichas entrevistas identifican que los principales aspectos de seguridad son: vulnerabilidades, autenticación y autorización, y OWASP Top 10.
Aspectos importantes	Identificación de los aspectos de seguridad que los profesionales buscan al cambiar al software seguro. Identificación de las técnicas utilizadas en la industria para garantizar que los mecanismos de seguridad no se rompan después de un cambio de software. Identificación de los desafíos que enfrentan los profesionales al cambiar al software seguro.

Tabla 8. Extracción fuente 4.
Fuente: elaboración propia.

Repositorio	ACM Digital Library
Título	A Data Mining Approach to Assess Privacy Risk in Human Mobility Data
Año de publicación	2017
Autores	Roberto Pellungrin, Uca Pappalardo, Francesca Pratesi and Anna Monreale
Resumen	Este paper define que es la información, qué son los riesgos de privacidad aparte de las medidas que se deben tomar para prevenir dichos riesgos.
Aspectos importantes	Los marcos existentes para la evaluación de riesgos de privacidad proporcionan a los proveedores de datos herramientas para controlar y mitigar los riesgos de privacidad, pero sufren dos defectos principales: Tienen una alta complejidad computacional.

	El riesgo de privacidad debe volverse a calcular cada vez que haya nuevos registros de datos disponibles y para cada usuario.

Tabla 9. Extracción fuente 5.
Fuente: elaboración propia.

1.8.2.2 Ejecución de la selección en la fuente Google Scholar

1.8.2.2.1 Selección de estudios iniciales

Siguiendo las recomendaciones provistas por Blanco et al. (2007), se realiza la búsqueda de estudios iniciales, basada en los siguientes parámetros:

- Seguridad de aplicaciones móviles.
- OWASP.
- Auditorías de seguridad.

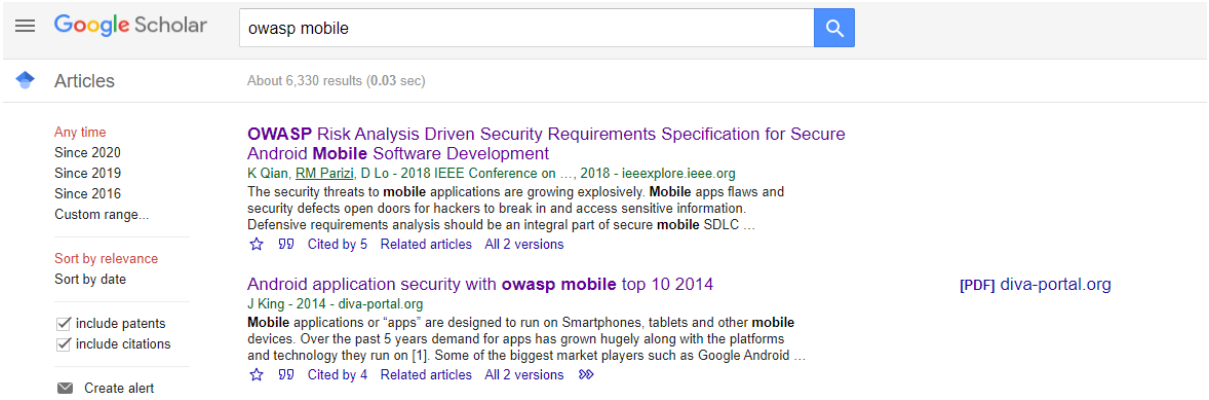


Figura 3. Ejecución de la búsqueda en Google Scholar.
Fuente: Google Scholar

Tras realizar la búsqueda utilizando los parámetros mencionados, se encontraron tres documentos tras aplicar el método de exclusión propuesto. A continuación, se presenta el detalle de los mismos por medio de la tabla 10.

#	Título	Autores	Año	URL
1	Analysis of Security Vulnerabilities for Mobile Health Applications	Y. Cifuentes, L. Beltrán, L. Ramírez	2015	https://publications.waset.org/10002458/analysis-of-security-vulnerabilities-for-mobile-health-applications
2	A Survey of Android	Bahman Rashidi and Carol	2016	https://www.researchgat

	Security Threats and Defenses	Fung		e.net/profile/Bahman_Rashidi2/publication/282365848_A_Survey_of_Android_Security_Threats_and_Defenses/links/560ec06908ae6b29b499a51f/A-Survey-of-Android-Security-Threats-and-Defenses.pdf
3	Audit Process during Projects for Development of New Mobile IT Applications	Marius POPA	2014	https://www.researchgate.net/profile/Popa_Marius/publication/47278135_Audit_Process_during_Projects_for_Development_of_New_Mobile_IT_Application/links/0fcfd50bcb7e55adb5000000/Audit-Process-during-Projects-for-Development-of-New-Mobile-IT-Application.pdf

Tabla 10. Estudios encontrados en Google Scholar.

Fuente: elaboración propia.

1.8.2.2.2 Evaluación de la calidad de los estudios

Se presume la calidad de los artículos mencionados, con base en la cantidad de filtros y evaluaciones realizados por Google Scholar.

1.8.2.2.3 Revisión de la selección

La selección de estudios primarios se hace tras realizar una revisión de los abstracts y contenido incluido en cada artículo.

1.8.2.2.4 Extracción de información

Se consideran los siguientes elementos para la extracción de la información relevante de los estudios primarios y el cumplimiento de los objetivos de la investigación:

- Seguridad de aplicaciones móviles.

- Riesgos y vulnerabilidades actuales de las aplicaciones móviles.
- OWASP.
- Auditorías de seguridad.

Repositorio	**Google Scholar**
Título	Analysis of Security Vulnerabilities for Mobile Health Applications
Año de publicación	2015
Autores	Y. Cifuentes, L. Beltrán, L. Ramírez
Resumen	La habilidad para implementar aplicaciones móviles aumenta a diario a través de diferentes tiendas de aplicaciones móviles. Pero dentro de estas capacidades, el número de ataques de piratería también ha aumentado. El objetivo de la investigación es analizar las vulnerabilidades detectadas en aplicaciones móviles de acuerdo con los estándares de factores de riesgo definidos por OWASP en 2014.
Aspectos Importantes	Las medidas de seguridad que se deben tomar en cuenta al manejar información personal de los usuarios. Requerimientos de seguridad para de desarrollar una aplicación móvil. Factores de riesgo detectados por el proyecto de seguridad móvil de OWASP"

Tabla 11. Extracción fuente 1.
Fuente: elaboración propia.

Repositorio	**Google Scholar**
Título	A Survey of Android Security Threats and Defenses
Año de publicación	2016
Autores	Bahman Rashidi and Carol Fung
Resumen	Con los miles de millones de personas que usan teléfonos inteligentes y el crecimiento exponencial de las aplicaciones para teléfonos inteligentes, es casi imposible que los mercados de aplicaciones, como Google App Store, verifiquen a fondo si una aplicación es legítima o maliciosa. Como resultado, los usuarios de las aplicaciones móviles deben decidir si dichas aplicaciones son malas o buenas.

Aspectos importantes	Explica el sistema operativo Android y la arquitectura que cumplen sus aplicaciones. Profundiza en los mecanismos de seguridad para Android. Problemas y amenazas de seguridad en Android.

Tabla 12. Extracción fuente 2.
Fuente: elaboración propia.

Repositorio	**Google Scholar**
Título	A Survey of Android Security Threats and Defenses
Año de publicación	2016
Autores	Bahman Rashidi and Carol Fung
Resumen	Este paper presenta características del proceso de auditoría de la computadora durante el ciclo de vida de desarrollo de software en aspectos específicos de las aplicaciones móviles.
Aspectos importantes	Define la seguridad con la que debe contar el proceso de desarrollo de una aplicación móvil. Explica los problemas del proceso de auditoría durante el ciclo de vida de las aplicaciones móviles.

Tabla 13. Extracción fuente 3.
Fuente: elaboración propia.

Capítulo 2. Marco teórico o conceptual

2.1 PyMEs

PyME, según Guillermo Westreicher (2016), es el acrónimo utilizado a la hora de hablar de pequeñas y medianas empresas. Estas, generalmente suelen contar con un bajo número de trabajadores y de un volumen de negocio e ingresos moderados en comparación con grandes corporaciones industriales o mercantiles.

Tradicionalmente, las empresas se clasifican según su tamaño en pequeñas, medianas y grandes. Así pues, al conjunto de las dos primeras se le denomina de forma abreviada PyMEs (pequeñas y medianas empresas). Además, con el paso de los años se les ha sumado a estos tres grupos un cuarto: las microempresas, que también se incluyen en las PyMEs. Es decir, las PyMEs son organizaciones con fines de lucro (que buscan generar beneficios) y cuyas operaciones son de baja escala.

A continuación, se muestra una tabla que resalta las ventajas y desventajas de las PyMEs:

Ventajas	Desventajas
Es más sencillo que puedan cambiar el nicho o modelo de negocio. Es decir, existe mayor flexibilidad.	Al no tener un gran volumen de transacciones no alcanzan economías de escala. Es decir, sus operaciones podrían tener un menor costo unitario si el número de ventas fuera mayor.
La relación entre el empresario y sus clientes es cercana. Esto, ya que usualmente el trato es directo entre ambos.	Es más complicado para una PyME conseguir financiamiento, en comparación a una gran empresa, en vista de que sus ingresos y su respaldo financiero son menores.
Lo anterior genera un vínculo entre el negocio y los clientes que se traduce muchas veces en fidelidad, es decir, el comprador va al establecimiento no por el menor costo, sino por la amabilidad o la simpatía del dueño de la tienda.	Es difícil que acceda a financiamiento y grandes capitales, probablemente la pequeña empresa no pueda (o vea dificultades para) invertir en campañas publicitarias masivas o en desplegar una extensa red de ventas.
Son empresas que pueden encontrar nichos de mercado no atendidos	

Tabla 14. Ventajas y desventajas de PyMEs.
Fuente: Economipedia (2016).

El uso de apps en PyMEs como clave para el desarrollo

Las aplicaciones para dispositivos móviles son un gran avance que las PyMEs tienen a su alcance para llegar a sus clientes. Su uso está creciendo cada día más, es la gran oportunidad de las empresas para realizar un marketing eficaz.

Las aplicaciones móviles tienen la gran ventaja de poder entrar en contacto con el cliente en cualquier lugar y momento del día. No es necesario que la empresa o tienda esté abierta para que esto suceda. Tienen la capacidad de mejorar la productividad y llevar un control minucioso de todos los movimientos económicos de la empresa.

2.2 BYOD

BYOD tal como lo define Mila Lavín (2013) proviene de las siglas de “Bring Your Own Device”, una expresión en inglés que podríamos traducir al español como “Trae Tu Propio Dispositivo”. Se trata de una política empresarial en la que la organización permite o anima

a sus trabajadores a que utilicen sus propios dispositivos móviles, tanto en el centro de trabajo como en las actividades laborales que realizan fuera de la oficina. De esta forma, el trabajador utiliza sus tabletas, smartphones, computadoras portátiles, etc., para conectarse con la intranet de su empresa.

El objetivo principal de esta nueva modalidad es disminuir los costos directos de la empresa, optimizar los recursos y mejorar la eficiencia y productividad del equipo humano.

Ventajas	Desventajas
Mayor productividad de los empleados	Riesgos en la seguridad y la privacidad de la información corporativa.
Posibilidad de trabajar con más flexibilidad	Requiere nuevas políticas de control de accesos.
Uso del dispositivo en cualquier momento y lugar.	Precisa recursos de red suficientes para soportar la llegada de los dispositivos.
Permite a los empleados dar un mejor servicio al cliente.	Necesidad de soporte TI para una diversidad de dispositivos, aplicaciones y software.

Tabla 15. Ventajas y desventajas de BYOD.
Fuente: Computerhoy (2013).

¿Cómo regular el Bring Your Own Device?

Deberemos dejar constancia al empleado de las normas que rigen esta práctica en el momento de la incorporación de un usuario con la práctica del BYOD implementada:

1. Consentimiento y colaboración del trabajador por escrito, en el que se recopilen las medidas de seguridad por aplicar.
2. Las políticas de uso y seguridad específicas para regular los requisitos básicos de seguridad.
3. Regulación técnica en la que se detalle cómo se implementarán las herramientas de monitorización o bloqueo de los dispositivos, cifrado de información.
4. Los controles de seguridad se centrarán únicamente en la parte corporativa, nunca se aplicarán a aplicaciones, áreas o información privada que cada usuario tenga en su dispositivo.

2.3 Definición y cómo funcionan las aplicaciones móviles

Es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles. Este tipo de aplicaciones permiten al usuario efectuar un variado conjunto de tareas profesional, de ocio, educativas, de acceso a servicios, etc., facilitando las gestiones o actividades por desarrollar.

2.3.1 Tipos de aplicaciones móviles

El diseño de una aplicación se puede enfocar de distintas maneras, ya sea para que sean ejecutadas directamente sobre una plataforma en concreto, sobre un navegador web o mediante una combinación de ambos, por lo que es importante conocer las particularidades de cada uno de estos enfoques como punto de partida para definir el enfoque de una auditoría de seguridad de una aplicación móvil.

2.3.1.1 Aplicación nativa

Una aplicación nativa, según LanceTalent (2014), es la que se desarrolla de forma específica para un determinado sistema operativo, llamado Software Development Kit o SDK. Cada una de las plataformas, Android, iOS o Windows Phone, tienen un sistema diferente, por lo que si quieres que tu app esté disponible en todas las plataformas se deberán crear varias apps con el lenguaje del sistema operativo seleccionado.

Por ejemplo:

- Las apps para iOS se desarrollan con lenguaje Objective-C
- Las apps para Android se desarrollan con lenguaje Java
- Las apps en Windows Phone se desarrollan en .Net

Ventajas y desventajas

Ventajas	Desventajas
Acceso completo al dispositivo	Diferentes habilidades, idiomas, herramientas para cada plataforma de destino.
Mejor experiencia del usuario	Tienden a ser más caras de desarrollar.
Visibilidad en App Store	Requieren de una aprobación para ser publicadas.
Envío de notificaciones o “avisos” a los usuarios	

La actualización de las aplicaciones es constante.	

Tabla 16. Ventajas y desventajas aplicación nativa.
Fuente: elaboración propia, basada en la información de Softcorp.

2.3.1.2 La aplicación web

Una aplicación web o webapp es la desarrollada con lenguajes muy conocidos por los programadores, como HTML, Javascript y CSS, según LanceTalent (2014). La principal ventaja con respecto a la nativa es la posibilidad de programar independiente del sistema operativo en el que se usará la aplicación. De esta forma, se pueden ejecutar en diferentes dispositivos sin tener que crear varias aplicaciones.

Las aplicaciones web se ejecutan dentro del propio navegador web del dispositivo a través de una URL. Por ejemplo, en Safari, si se trata de la plataforma iOS. El contenido se adapta a la pantalla adquiriendo un aspecto de navegación APP.

Ventajas y desventajas

Ventajas	Desventajas
El mismo código base es reutilizable en múltiples plataformas.	Requiere conexión a Internet.
Proceso de desarrollo más sencillo y económico.	Acceso muy limitado a los elementos y características del hardware del dispositivo.
No necesitan ninguna aprobación externa para publicarse (a diferencia de las nativas para estar visibles en las apps store)	La experiencia del usuario (navegación, interacción…) y el tiempo de respuesta es menor que en una app nativa.
El usuario siempre dispone de la última versión.	Requiere de mayor esfuerzo en promoción y visibilidad.

Tabla 17. Ventajas y desventajas aplicación web.
Fuente: elaboración propia, basada en la información de Softcorp.

2.3.1.3 La aplicación híbrida

Cómo y como nos indica LanceTalent (2014), una aplicación híbrida es una combinación de las dos anteriores, se podría decir que recoge lo mejor de cada una de ellas. Las apps híbridas se desarrollan con lenguajes propios de las webapp, es decir, HTML, Javascript y CSS, por lo que permite su uso en diferentes plataformas, pero también dan la posibilidad

de acceder a gran parte de las características del hardware del dispositivo. La principal ventaja es que, a pesar de estar desarrollada con HTML, Java o CSS, es posible agrupar los códigos y distribuirla en app store.

Ventajas y desventajas

Ventajas	Desventajas
Es posible distribuirla en las tiendas de iOS y Android.	Experiencia del usuario más propia de la aplicación web que de la app nativa.
Instalación nativa pero construida con JavaScript, HTML y CSS.	Diseño visual no siempre relacionado con el sistema operativo en el que se muestre.
El mismo código base para múltiples plataformas.	
Acceso a parte del hardware del dispositivo	

Tabla 18. Ventajas y desventajas aplicación hibrida.
Fuente: elaboración propia, basada en la información de Softcorp.

2.3.2 Sistemas operativos móviles

Entre los más importantes podemos nombrar los siguientes:

- Android
- IOS
- Windows Phone
- Blackberry
- Ubuntu Touch
- Otros.

A continuación, se detallará los dos sistemas operativos más relevantes a nivel mundial:

Android

Como lo menciona TecnologiaFacil (2016), Android es un sistema operativo para dispositivos móviles como tablets, smartphones, relojes inteligentes y más, y es el principal creador de la popularización de este tipo de dispositivos, fundamentalmente debido a su facilidad de uso, la enorme cantidad de aplicaciones que podemos ejecutar en él, su

estabilidad, constante modernización y gratuidad, tanto para el propio sistema operativo como para las aplicaciones que contiene.

Originalmente desarrollado por una empresa llamada Android, en el año 2006 fue adquirida por Google, quienes tomaron las riendas del proyecto y lo convirtieron en el poderoso sistema operativo que disfrutamos hoy en todo tipo de artefactos.

A partir de la adquisición de esta pequeña empresa por parte de Google, el sistema no paró de mejorar, ya que desde el HTC Dream, primer teléfono que incorporó a Android como sistema operativo, su desarrollo no se detuvo en ningún momento, llegando a otros mercados como el de los relojes inteligentes, los televisores y los automóviles, con total éxito y aceptación.

Principales características de Android

Si bien, Android no es el único sistema operativo para móviles del mercado, ya que también existen iOS, Symbian, Blackberry OS y Windows Phone, entre otros, lo cierto es que se diferencia de ellos debido a que su núcleo está basado en Linux, con las consiguientes ventajas de seguridad y estabilidad que ello provee.

Este hecho permite que el sistema sea seguro y estable, ya que el usuario no tiene los permisos necesarios como para permitir el ingreso de virus, malwares y otras amenazas al sistema, logrando de este modo un funcionamiento sin sobresaltos.

iOS

Como lo menciona Rocío García (2020), iOS es un sistema operativo lanzado y utilizado por Apple. Su nombre proviene de iPhone OS, es decir, iPhone Operative System o Sistema Operativo de iPhone. Se lanzó originalmente para el teléfono de la marca, aunque también se ha utilizado durante años en otros dispositivos de la compañía como en algunos de los reproductores de música iPod o en las tabletas iPad (hasta la llegada de iPadOS).

Se trata de un sistema cerrado que no puedes utilizar salvo en dispositivos de marca Apple. La gran diferencia con Android es esta: el sistema operativo de Google puede instalarse en infinidad de teléfonos de todas las marcas, pero iOS es un sistema cerrado y exclusivo para los aparatos de la marca de Cupertino, no para los demás. Al igual que otros sistemas operativos móviles, iOS nos permite instalar aplicaciones para añadir funciones a las que vienen por defecto en el smartphone, es decir, más allá del teléfono o los mensajes puedes visitar la App Store en busca de aplicaciones que cumplan alguna función que necesitas, aprender inglés o hacer la compra.

2.3.3 Tiendas de aplicaciones

Las principales tiendas de aplicaciones para dispositivos móviles son las siguientes:

- Google Play desarrollada por Google Inc.
- App Store de Apple
- Windows Phone Store de Microsoft
- Amazon Appstore
- ApkPure
- UptoDown
- F-Droid

2.3.4 Datos que pueden acceder las aplicaciones móviles

Cuando descargas aplicaciones móviles, en el proceso de instalación, te suelen indicar a qué tipos de datos necesita acceder. Importante que le prestes atención a esto, puede ser delicado incluso inseguro en casos particulares. Alguno de estos tipos de datos puede ser:

- Lista de contactos de teléfono y de email.
- Registro de llamadas.
- Datos transmitidos por Internet.
- Información del calendario.
- Datos de localización del dispositivo móvil.
- Código de identificación exclusivo del dispositivo móvil.
- Información que indica la manera en que usted usa la aplicación propiamente.
- Otros.

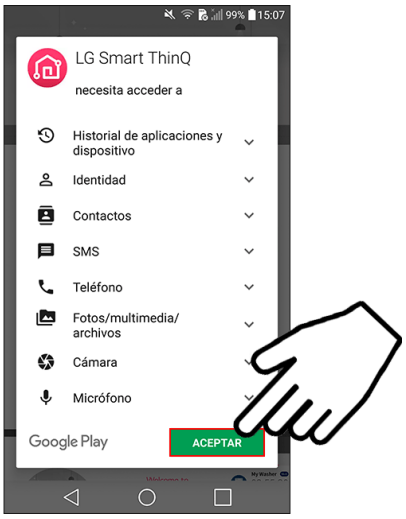


Figura 4. Términos y Condiciones.
Fuente LG

2.4 Estado de los dispositivos móviles en el mercado

Como lo menciona Yi Min (2020), el mundo móvil se ha ido expandiendo, desde que los seres humanos comprendieron que necesitaban más acceso, más información, más

conocimiento y procesar de forma rápida todo en un equipo pequeño, accesible y fácil de trasladar.

Todo esto trae consigo el desarrollo de los nuevos equipos electrónicos que tienen cada usuario actualmente desde laptops, tablets, relojes inteligentes y el smartphone (que es nuestra oficina portátil), todos estos equipos facilitan la vida de los usuarios y más para una globalización tan interconectada y que conecten en su máxima expresión a cada persona en el mundo.

Actualmente hay 5190 millones de usuarios únicos en dispositivos móviles, en los que no se diferencia el tipo de teléfono, esto cubre el 67% de la población. Lo más interesante de esto es que hay más de 7950 millones de números de teléfonos (excluyendo el IOT), que cubre el 103% de la población, dando un promedio por usuario móvil de 1,53%, esto se ha vuelto común por la flexibilidad de los equipos en permitir tener dos chips de telefonía similar o diferente.

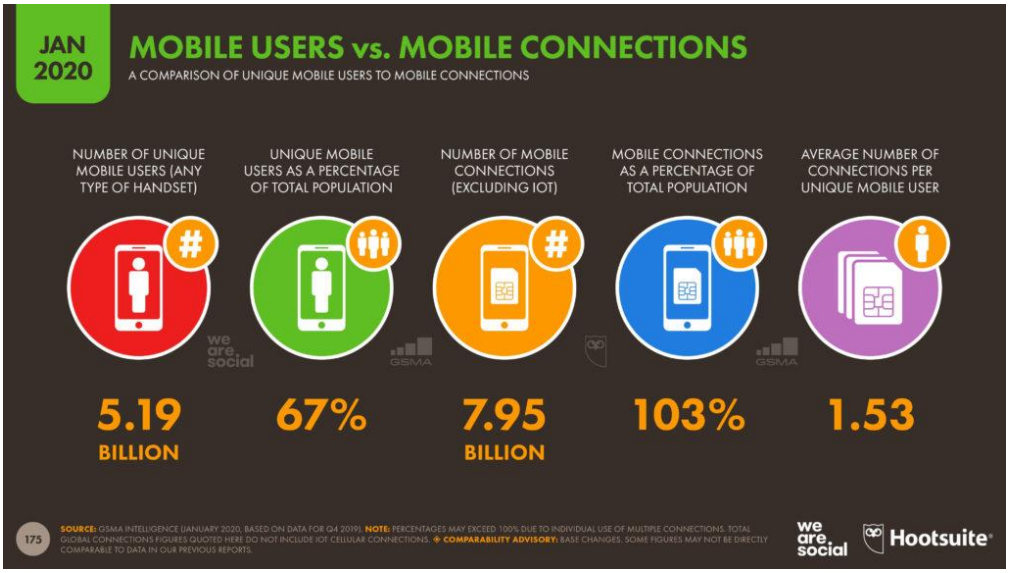


Figura 5. Usuarios de móviles vs conexiones móviles.
Fuente GSMA Intelligence y Yiminshum.com

2.4.1 Clasificación de conectividad móvil

El promedio de los países conectados es del 103%, pero existen diez países que superan el 170% y hay países que duplican el número promedio y son los siguientes: Macao (295%), Islas Vírgenes U. S. (198%), Antigua & Barbuda (195%), Montenegro (191%), U. A. E. (187%), Hong Kong (181%), Finlandia (179%), Costa Rica (178%), Seychelles (176%) y Sudáfrica (176%).

Y los últimos diez países más desconectados son Isla Marshall (11%), Corea del Norte (18%), Sur de Sudán (20%), Eritrea (20%), Micronesia (22%), Papúa Nueva Guinea (32%), Madagascar (33%), Chad (37%), El Congo (40%) y Etiopía (41%).

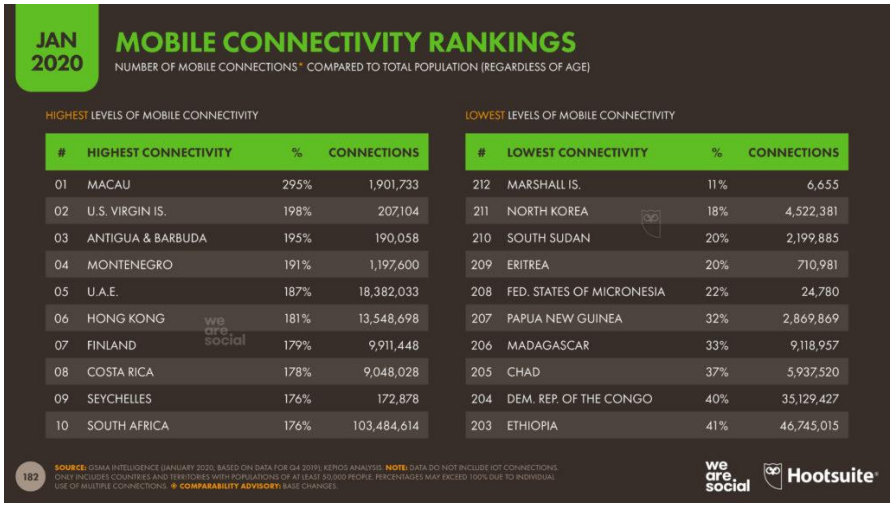


Figura 6. Clasificación de conectividad móvil.
Fuente GSMA Intelligence y Yiminshum.com

2.4.2 Aplicaciones con más usuarios activos

Las aplicaciones móviles con más usuarios activos son WHASTAPP MESSENGER, FACEBOOK, FACEBOOK MESSENGER, WECHAT, INSTAGRAM, TIKTOK, ALIPAY, QQ, TAOBAO Y BAIDU. De las aplicaciones mencionadas, cuatro pertenecen al grupo de Facebook, liderado por Mark Zuckerberg y cinco a China, esto demuestra la globalización digital que quiere lograr poco a poco a través de aplicaciones y conquistar Occidente.

Figura 7 **.** Clasificación global de aplicaciones móviles.
Fuente App Annie y Yiminshum.com

2.5 Vulnerabilidades, amenazas y riesgos en las aplicaciones móviles

Antes de realizar cualquier auditoría de seguridad, es importante entender bien los términos de vulnerabilidad, riesgo y amenaza, además de su relación, con el fin comprender la base de la gestión de riesgos y de cualquier programa o actividad que se lleve adelante respecto a la protección de la información.

- Una vulnerabilidad, según INCIBE (2017), es una debilidad o fallo en un sistema de información que pone en riesgo la seguridad de la información, pudiendo permitir que un atacante comprometa la integridad, disponibilidad o confidencialidad de la misma.
- Una amenaza es toda acción que aprovecha una vulnerabilidad para atentar contra la seguridad de un sistema de información, menciona INCIBE (2017). Las amenazas pueden proceder de ataques (fraude, robo, virus), sucesos físicos (incendios, inundaciones) o negligencia y decisiones institucionales (mal manejo de contraseñas, no usar cifrado). Desde el punto de vista de una organización pueden ser tanto internas como externas.
- Una vez que tenemos clara la diferencia entre amenaza y vulnerabilidad, es importante introducir el concepto de riesgo.
- El riesgo es la probabilidad de que se produzca un incidente de seguridad, materializándose una amenaza y causando pérdidas o daños, así lo menciona INCIBE (2017).

El riesgo depende entonces de los siguientes factores: la probabilidad de que la amenaza se materialice aprovechando una vulnerabilidad y produciendo un daño o impacto. El producto de estos factores representa el riesgo.

Figura 8 . Amenaza vs vulnerabilidad.
Fuente INCIBE.

2.5.1 Tipos de amenazas de los dispositivos móviles

Cada aspecto de tu dispositivo móvil, desde su software hasta el hardware, la conexión telefónica, la conexión a Internet y lo más importante, las aplicaciones que usas proporcionan una vía para las amenazas de seguridad. Cuando se trata de aplicaciones, Internet y la conexión a Internet, hay dos tipos de amenazas: amenazas generadas por el usuario o internas y amenazas externas.

2.5.1.1 Amenazas internas

Representan brechas de seguridad que pasan inadvertidas durante la fase de desarrollo de la aplicación, por ejemplo, al confiar en librerías de terceras partes que podrían incorporar comportamientos inesperados como conexiones a servidores desconocidos o vulnerabilidades.

2.5.1.2 Amenazas externas

Basadas en la habilidad de un hacker de acceder a la aplicación, alterar su funcionamiento o derivarla hacia actividades maliciosas, en lo que se conoce como tampering. También se incluirían dentro de esta categoría los ataques que aprovechen las conexiones de red o las vulnerabilidades del sistema operativo, así como la publicación de malware de cualquier tipo.

2.5.2 Vectores de ataque

Se hace referencia en este capítulo a las amenazas que actualmente afectan a los dispositivos móviles y se realizará una descripción de estas amenazas.

Malware

Como lo cita Andaluciasmart (2016), un malware es un software malicioso instalado sin autorización. Suele ser el vector más utilizado en la actualidad -junto con los Exploit Kits-, según la mayor parte de las fuentes especializadas, sirviendo en muchos casos como mecanismo de entrada de otros tipos de ataques, ya que, una vez instalado, normalmente se convierte en la puerta trasera para intentar tomar el control de sistemas y entornos de forma silenciosa.

Figura 9. Tipos de Malware.

Fuente: Andaluciasmart

Ransomware:

Como lo define Andaluciasmart (2016), un ransomware es un software cuyo fin es comprometer la disponibilidad de la información y los sistemas. En la mayor parte de las ocasiones se requiere un desembolso económico por parte del afectado para la recuperación de los datos, ya que la operativa de la organización afectada puede verse comprometida por completo, en función de la gravedad de la infección.

A efectos didácticos, podría simplificarse considerando el ransomware como un malware que infecta todos los sistemas de información que tenga a su alcance, con la finalidad de cifrar la información y extorsionar económicamente a las víctimas si quieren recuperar dicha información. Aunque, dependiendo de la fuente utilizada, el número de empresas que paga el rescate, pero que nunca recupera sus datos, suele rondar entre el 15 y el 25%.

Exploit kits

Andaluciasmart (2016) menciona que un exploit kit se trata de paquetes de software, que incluyen un conjunto de programas y utilidades ya desarrollados, destinados a explotar diferentes vulnerabilidades de programas y sistemas, afectando a la disponibilidad, integridad y/o confidencialidad del objetivo. Este tipo de paquetes incluyen módulos para la explotación de vulnerabilidades, distribución de malware, control del sistema objetivo, etc., en definitiva, constituyen un esquema completo de herramientas que permiten automatizar en gran medida los ciberataques.

Ingeniería social

La ingeniería social se refiere según Andaluciasmart (2016), a un ataque en el que se utilizan las habilidades sociales para obtener información, ya sea personal como acerca de los sistemas informáticos. Las personas que practican la ingeniería social (estafadores) utilizan la interacción social, ya sea para engañar a sus víctimas para que entreguen la

información o manipularlos para que confíen en él o ella y compartan información con el atacante. Podría haber una interacción cara a cara, por teléfono, mensajería, e-mail, etc.

Phishing

Andaluciasmart (2016) cita que phishing es un ataque que utiliza la suplantación de servicios y webs, engañando al usuario para -entre otros fines- robar información confidencial, así como inyectar software malicioso.

Desde hace unos años, se ha detectado un notable incremento de los ciberataques mediante phishing, como consecuencia de la escasa complejidad y de la poca preparación que necesitan, unido a la falta de concienciación por parte de los usuarios. En el caso de ataques dirigidos a objetivos concretos (Spear Phising), que suelen ser más efectivos al personalizar el engaño, la publicación en redes sociales de datos de toda índole permite a los atacantes disponer de la información necesaria mediante técnicas de inteligencia de fuentes abiertas (Open Source Intelligence - OSINT).

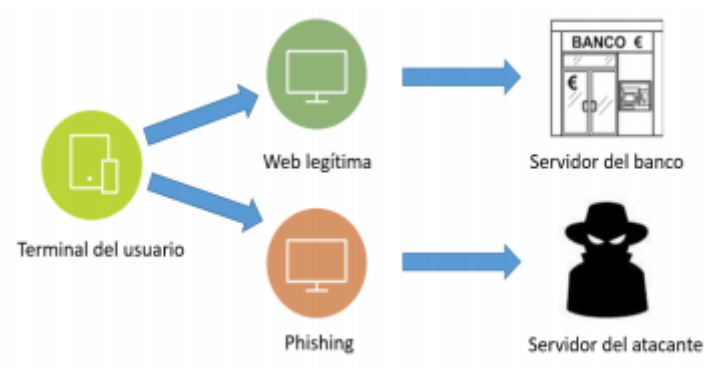


Figura 10. Ataque phishing.
Fuente: Andaluciasmart

Interceptación, robo o sabotaje de datos

Como lo menciona Andaluciasmart (2016), este vector de ataque, los ciberdelincuentes explotan las vulnerabilidades de protocolos inseguros de comunicación y de transmisión de información entre dos dispositivos o sistemas de información, o en otros casos, errores en la configuración de protocolos.

Para la interceptación, el ataque denominado “Man in the Middle” suele ser el más habitual. En él, los ciberdelincuentes aprovechan las vulnerabilidades de los diferentes protocolos utilizados durante el proceso de comunicación, consiguiendo posicionarse en medio de los intercambios de información, obligando a que todo el flujo, ya sea unidireccional o bidireccional, pase por el ciberdelincuente. A través de estos ataques se puede comprometer tanto la integridad, como la confidencialidad de los datos.

Figura 11. Ataque “Man in the Middle”.
Fuente: Andaluciasmart

Jailbreak en iOS

Los dispositivos iOS parecen ser bastante seguros hasta ahora; sin embargo, esta declaración sólo se aplica a los dispositivos que no han sido “jailbroken”. Se conoce por Jailbreak según Andaluciasmart (2016), al método de modificar el sistema iOS con la finalidad de saltarse medidas de seguridad impuestas por Apple en su sistema y acceder a algunas funciones que no están permitidas de manera normal, como pueden ser: realizar algunas modificaciones visuales del sistema (personalizar más el sistema con fuentes de letra diferentes, iconos) y la más interesante, habilitar la descarga de aplicaciones que no han sido aceptadas en la App Store o aplicaciones de pago gratuitas vía tiendas no oficiales. Un dispositivo al cual se le realiza el proceso de jailbreak, ya no está limitado a usar la App Store oficial de Apple para conseguir aplicaciones, sino que puede utilizar repositorios de aplicaciones externos

Root en Android

Root, Rooting o “Rootear” según Andaluciasmart (2016), se le conoce al método utilizado para darle al usuario los privilegios de administrador o súper usuario (superuser) del sistema operativo Android. Este concepto proviene de los SO Unix como Linux y hace referencia al mismo comportamiento en Android, ya que este es en esencia un Linux modificado. El proceso de Root es necesario cuando se quiere ejecutar ciertas aplicaciones que necesitan privilegios especiales para realizar tareas que entran en conflicto con directivas de seguridad impuestas por el desarrollador del SO (en este caso, Google). Al “rootear” un dispositivo Android, lo que generalmente ocurre es que se instala el programa su [35], que permite elevar los permisos de usuario transformándose en el usuario root del sistema.

Cada vez que se intenta correr algún programa que necesite utilizar privilegios de súper usuario, se invocará su. En algunas versiones la instalación de su viene acompañada de otra aplicación llamada SuperSu o Superuser, que agrega algunos controles y algunas

funcionalidades, como hacer aparecer una pantalla pidiendo autorización para correr el proceso que requiere privilegios, guardar la decisión del usuario para el futuro, etc.

2.6 Metodologías de análisis de vulnerabilidades

Ostec (2019) menciona que el análisis de vulnerabilidad consiste en definir, identificar, clasificar y priorizar las debilidades de las aplicaciones para proporcionar una evaluación de las amenazas previsibles y reaccionar de manera apropiada.

Dentro del análisis de vulnerabilidades en dispositivos móviles hay dos principales tipos, los cuales son: el análisis estático y el análisis dinámico, que se pueden combinar con el objetivo de resguardar la mayor área posible de ataques que puede recibir una aplicación móvil.

2.6.1 Análisis estático

El análisis estático o también llamado por sus siglas en inglés, SAST (Static Application Security Testing), según A2Secure (2019), analiza la fuente de la aplicación, el código de bytes o el código binario para detectar vulnerabilidades de seguridad, generalmente en las fases de programación y/o prueba del ciclo de vida del desarrollo de software (SDLC). Las herramientas SAST se pueden considerar como pruebas de White-Hat o White-Box, en las que el probador conoce información sobre el sistema o el software que se está probando, incluido un diagrama de arquitectura, acceso al código fuente, etc. Las herramientas SAST examinan el código fuente (en reposo) para detectar y reportar las debilidades que pueden conducir a vulnerabilidades de seguridad.

2.6.2 Análisis dinámico

También conocido como DAST por sus siglas en inglés (Dynamic Application Security Testing), según A2Secure (2019), analiza las aplicaciones en su estado dinámico de ejecución durante las fases de prueba o de operación. Simula ataques contra una aplicación (generalmente aplicaciones y servicios habilitados para la web) y analiza las reacciones de la aplicación para determinar si es vulnerable. A diferencia de las herramientas SAST, las herramientas DAST se pueden considerar como pruebas de Black-Hat o Black-Box, en las que el probador no tiene conocimiento previo del sistema. Detectan condiciones que indican una vulnerabilidad de seguridad en una aplicación en su estado de ejecución. Las herramientas DAST se ejecutan en el código operativo para detectar problemas con interfaces, solicitudes, respuestas, secuencias de comandos, inyección de datos, sesiones, autenticación y más. Las herramientas DAST emplean fuzzing: lanzar casos de prueba conocidos, no válidos e inesperados, en una aplicación, a menudo en gran volumen.

2.7 Herramientas para realizar análisis estáticos y dinámicos

2.7.1 Mobile Security Framework (MobSF)

Mobile Security Framework (MobSF) es un marco de pruebas de penetración de aplicaciones móviles (Android / iOS / Windows) automatizado, de código abierto y todo en uno, capaz de realizar análisis estáticos, dinámicos y de malware.

OWASP MSTG lo sugiere para el análisis estático de seguridad en aplicaciones móviles. Se puede usar para un análisis de seguridad efectivo y rápido de las aplicaciones móviles de Android, iOS y Windows y es compatible tanto con binarios (APK, IPA y APPX) como con código fuente comprimido. MobSF puede realizar pruebas dinámicas de aplicaciones en tiempo de ejecución para aplicaciones de Android y tiene capacidades de fuzzing de API web impulsadas por CapFuzz, un escáner de seguridad específico de API web. MobSF está diseñado para hacer que su integración de canalización CI / CD o DevSecOps sea perfecta. Tiene una interfaz gráfica de usuario en forma de servicio web. El servicio web consiste en un panel que presenta los resultados del análisis, su propio sitio de documentación, un emulador integrado y una API que permite a los usuarios activar el análisis automáticamente. Está alojado en un entorno local, por lo que los datos confidenciales nunca interactúan con la nube.

2.7.2 APK Extractor

APK Extractor extraerá los APK instalados en los dispositivos móvil Android y los copiará en su tarjeta SD. Algunas de sus características:

- Rápido y fácil de usar.
- Extrae casi todas las aplicaciones, incluidas las aplicaciones del sistema.
- No se requiere acceso ROOT.
- Por defecto, los Apk se guardarán en / sdcard / ExtractedApks /.
- Proporcionó la opción de búsqueda para buscar aplicaciones.
- Compatible con la última versión de Android 7.0
- formato apk guardado:
- AppName_AppPackage_AppVersionName_AppVersionCode.apk.
- Puede extraer múltiples / todos los APK manteniendo presionado el botón largo sobre cualquier elemento.

2.7.3 OWASP Zed Attack Proxy (ZAP)

Como lo menciona OWASP (2020), Zed Attack Proxy (ZAP) es una herramienta gratuita de prueba de penetración de código abierto que se mantiene bajo Open Web Application Security Project (OWASP). ZAP está diseñado específicamente para probar aplicaciones web y es flexible y extensible.

En esencia, ZAP es lo que se conoce como un "proxy del hombre en el medio". Se encuentra entre el navegador del probador y la aplicación web para que pueda interceptar e inspeccionar los mensajes enviados entre el navegador y la aplicación web, modificar el contenido si es necesario y luego reenviar esos paquetes al destino. Se puede usar como una aplicación independiente y como un proceso de demonio.

2.7.4 Fortify on Demand

Fortify on Demand es un servicio de seguridad como servicio (SaaS), una solución de prueba que permite a cualquier organización probar la seguridad del software de forma rápida, precisa y accesible, y sin ningún software para instalar o administrar. Este servicio automatizado bajo demanda ayuda a las organizaciones con dos desafíos clave:

- Garantizar la seguridad de las aplicaciones con licencia de terceros.
- Incrementar la velocidad y la eficiencia de la construir seguridad en un ciclo de vida de desarrollo

Fortify on Demand cumple la función de un sistema de registro de terceros, llevando a cabo un análisis imparcial de una aplicación y proporcionar un informe detallado a prueba de manipulaciones al equipo de seguridad.

Los usuarios simplemente cargan los binarios de su aplicación y/o proporcione una URL para la prueba. HP Fortify on Demand puede realizar una prueba estática y/o dinámica, verificar todos los resultados y presentar los correlacionados en una interfaz detallada o por medio de reporte.

2.7.5 Virtual Box

Yubal (2020) describe Virtual Box como una aplicación que sirve para hacer máquinas virtuales con instalaciones de sistemas operativos. Esto quiere decir que, si se tiene un ordenador con Windows, GNU/Linux o incluso macOS, se puede crear una máquina virtual con cualquier otro sistema operativo para utilizarlo dentro del que se esté usando, en otras palabras, este software de virtualización va a permitir instalar otros sistemas operativos o el mismo que se tiene.

2.7.6 Genymotion

Genymotion es un entorno virtual que le permite simular teléfonos en su computadora. Puede crear el teléfono y ejecutar aplicaciones a través de él como si las estuviera reproduciendo en un dispositivo móvil. Los desarrolladores lo utilizan para probar sus productos. El sistema incluye tecnología en la nube que le permite consultar sitios web y trabajar con otros.

2.8 Marcos de referencia para aplicaciones móviles

Los estándares de seguridad de aplicaciones son establecidos por los principales organismos de investigación y estándares de la industria, para ayudar a las organizaciones a identificar y eliminar vulnerabilidades de seguridad de aplicaciones en sistemas de software complejos. Las siguientes organizaciones establecen estándares de seguridad para aplicaciones nacionales e internacionales.

2.8.1 OWASP

CastSoftware (2017), menciona que OWASP es una organización internacional y la Fundación OWASP apoya los esfuerzos de OWASP en todo el mundo. OWASP es una comunidad abierta dedicada a permitir que las organizaciones conciban, desarrollen, adquieran, operen y mantengan aplicaciones confiables.

2.8.2 CWE/SANS Top 25

El Common Weakness Enumeration, según Templarbit (2018), es una lista de vulnerabilidades de seguridad de software que se encuentran en toda la industria del desarrollo de software. Es un proyecto impulsado por la comunidad mantenido por MITRE, un grupo de investigación y desarrollo sin fines de lucro. Para cada entrada, el CWE proporciona una descripción de la vulnerabilidad y los pasos para mitigarla.

2.8.3 OWASP VS CWE

OWASP top 10	CWE top 25
Cubre conceptos más generales y se centra en aplicaciones web.	Cubre una gama más amplia de problemas que los que surgen de la vista centrada en la Web del OWASP Top Ten.
Algunos problemas en el Top Ten tienen aplicaciones generales para todas las clases de software.	Uno de los objetivos del Top 25 era estar en un nivel directamente accionable para los programadores, por lo que contiene problemas más detallados que las categorías que se utilizan en el Top Ten.

Tabla 19. OWASP vs CWE.
Fuente: elaboración propia.

2.9 Marco de referencia y sistema operativo móvil seleccionados

2.9.1 OWASP

OWASP, de acuerdo con Redeszone (2020), responde a las siglas Open Web Application Security Project. Es una fundación sin fines de lucro formada por empresas, organizaciones educativas y particulares de todo mundo. Juntos constituyen una comunidad de seguridad informática que trabaja para crear artículos, metodologías, documentación, herramientas y tecnologías que se liberan y pueden ser usadas gratuitamente por cualquiera.

Uno de los proyectos que más destaca de la fundación es el OWASP Top Ten, un documento estandarizado de concienciación acerca del desarrollo de código seguro para todo tipo de software, además de las aplicaciones web. Este tiene como raíz un consenso amplio respecto a los principales riesgos y amenazas de seguridad. Por este motivo, es altamente recomendado implementar las recomendaciones que realiza OWASP Top Ten para minimizar los riesgos y amenazas.

2.9.1.1 OWASP Mobile Application Verification Standard (MASVS)

Tal y como lo define OWASP (2019), el Estándar de Verificación de Seguridad de Aplicaciones Móviles (MASVS) es un esfuerzo comunitario para establecer un marco de requisitos de seguridad necesarios para diseñar, desarrollar y probar la seguridad de aplicaciones móviles iOS y Android.

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:

- Uso como una métrica - proporciona un estándar dirigido tanto a desarrolladores como a propietarios de aplicaciones, el cual puede ser utilizado para comparar aplicaciones en términos de seguridad;
- Uso como guía - hace de guía durante todas las fases del desarrollo y pruebas de seguridad de las aplicaciones móviles;
- Uso durante la compra o contratación - proporcionar una línea base para la verificación de la seguridad de aplicaciones móviles.

El MASVS define dos niveles de verificación de seguridad (MASVS-L1 y MASVS-L2), así como un conjunto de requisitos de resistencia a la ingeniería inversa (MASVS-R).

- El nivel MASVS-L1 contiene requerimientos genéricos de seguridad recomendados para todas las aplicaciones móviles, mientras que MASVS-L2 debería aplicarse a aplicaciones que manejan datos altamente sensibles. MASVS-R cubre los controles de seguridad adicionales que se pueden aplicar si la prevención de las amenazas

del lado del cliente es un objetivo de diseño. El cumplimiento de los requerimientos de MASVS-L1 dará como resultado 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, tales como SSL certificate pinning, haciendo la aplicación resistente a ataques más sofisticados, siempre y cuando los controles de seguridad del sistema operativo móvil estén intactos y que el usuario final no sea considerado como un potencial adversario.
- El cumplimiento de todos o de un subconjunto de los requerimientos de protección de software del nivel MASVS-R ayuda a prevenir amenazas específicas del lado del cliente cuando el usuario final es considerado malicioso y/o el sistema operativo móvil ha sido comprometido.

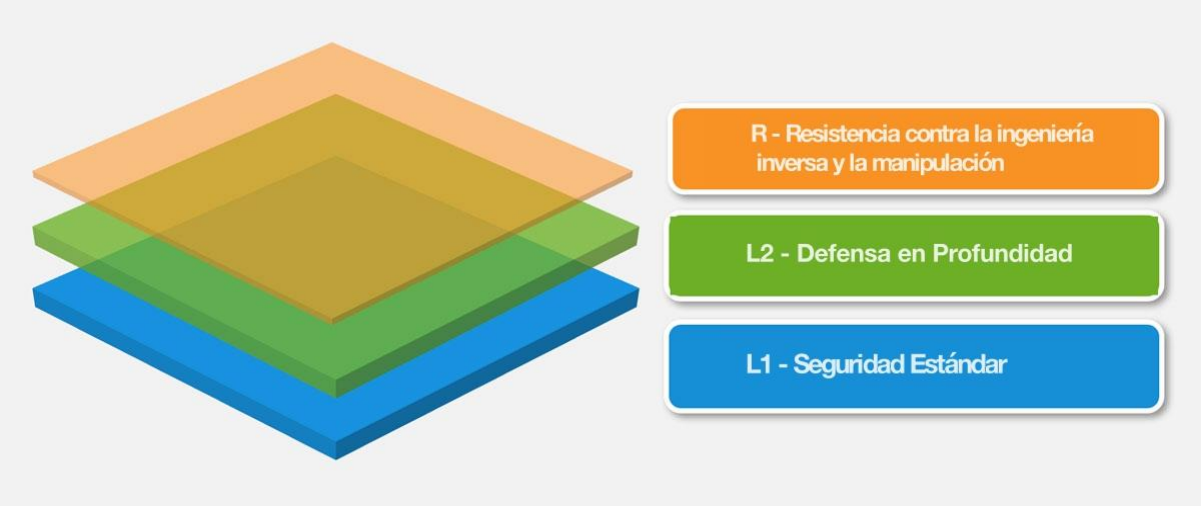


Figura 12. Niveles de verificación MASVS.
Fuente OWASP

A continuación, se enumeran los requisitos de verificación sugeridos por MASVS L1 y L2 que tratan de cubrir o ayudar a mitigar el Top 10 de riesgos de OWASP.

V1. Arquitectura, diseño y modelado de amenazas: la categoría V1 lista los requerimientos pertinentes a la arquitectura y al diseño de la aplicación. Los cuales buscan garantizar que el diseño de la arquitectura de la aplicación haya considerado los principales aspectos de seguridad en todos los componentes. Esta es la única categoría que no se corresponde con casos de OWASP Mobile Security Testing Guide (MSTG).

V2: Requerimientos en el almacenamiento de datos y la privacidad: 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 los 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

V3: Requerimientos de criptografía: 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 estándares. El propósito de estos controles es asegurarse de que la aplicación utiliza criptografía, siguiendo las mejores prácticas de la industria, incluyendo:

- Uso de librerías criptográficas reconocidas y probadas;
- Configuración y elección apropiada de primitivas criptográficas;
- Uso de generadores de números aleatorios suficientemente seguros.

V4: Requerimientos de autenticación y manejo de sesiones: en la mayoría de los casos, una parte esencial de la arquitectura global de aplicaciones móviles es que los usuarios deben iniciar la sesión en un servicio remoto. Aunque la mayor parte de la lógica ocurre en el servidor, MASVS define algunos requerimientos básicos sobre cómo manejar las cuentas y sesiones del usuario.

V5: Requerimientos de comunicación a través de la red: 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.

V6: Requerimientos de interacción con la plataforma: estos controles revisan que se utilicen las APIs de la plataforma y componentes estándares de una manera segura. Además, se cubre la comunicación entre aplicaciones.

V7: Requerimientos de calidad de código y configuración del compilador: 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.

V8: Requerimientos de resistencia ante la ingeniería inversa: 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 genera vulnerabilidades, sino que están pensados para incrementar la resistencia contra la ingeniería inversa de la aplicación, dificultándole 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.

2.9.1.2 OWASP Mobile Security Testing Guide (MSTG)

Como lo menciona OWASP (2020), el MSTG es un manual completo para pruebas de seguridad de aplicaciones móviles e ingeniería inversa. Describe los procesos técnicos para verificar los controles enumerados en el estándar de verificación de aplicaciones móviles de OWASP (MASVS).

2.9.1.2.1 OWASP Top 10 Mobile Risks

OWASP Top 10 Mobile Risks identifican los riesgos más críticos de seguridad que afectan a las aplicaciones móviles. Fernando Saavedra (2019) define el Top 10 Mobile Risk de OWASP como:

M1 – Uso inadecuado de la plataforma

Esta categoría cubre el uso indebido de características de la plataforma o el no uso de los controles de seguridad, permisos de plataforma, uso indebido de TouchID, servicios de contraseñas (KeyChain IOS) o algún otro control de seguridad que sea parte del sistema operativo móvil. Lo que se valora en el ataque es la seguridad de cualquier API expuesta.

M2 – Almacenamiento de datos inseguros

La valoración y, por ende, el vector de ataque varía mucho. Desde las aplicaciones de terceros que utilizan caché, cookies y otra información para recopilar datos protegidos, hasta que un adversario pueda obtener físicamente el dispositivo y ver información, debe manejar el almacenamiento de datos correctamente de varias maneras. Esto incluye la

autenticación, el cifrado y el manejo adecuado de todas las funciones de almacenamiento en caché.

En resumen, lo que se valora de manera más importante es que el almacenamiento de datos inseguros puede ser extremadamente fácil de explotar, ya que, dependiendo de la aplicación, el impacto en el negocio puede ser bastante severo.

M3 – Comunicación insegura

Esta categoría cubre los protocolos inseguros de enlace, versiones SSL incorrectas, negociación débil, la comunicación sin cifrar de datos sensibles, etc., por lo que hay que comprobar que los desarrolladores (a menudo muy preocupados en cuanto a la protección del procedimiento de autenticación y los datos en reposo) hayan implementado un cifrado de los datos que se transmiten correctamente.

M4 – Autenticación insegura

Esta categoría captura las nociones de autenticación del usuario final o gestión de sesión incorrecta. Esto puede incluir:

- No identificar al usuario en absoluto cuando sea necesario
- No mantener la identidad del usuario cuando se requiera
- Debilidades en el manejo de sesiones

M5 – Criptografía insuficiente

El uso incorrecto del cifrado es extremadamente común en las aplicaciones móviles. Los algoritmos de cifradas débiles, así como el procedimiento de cifrado / descifrado defectuoso, pueden ser fácilmente explotados y, por ende, lo que se valorará son los problemas en los cuales se intentó la criptografía, pero no se hizo correctamente.

M6 – Autorización insegura

Esta es una categoría para capturar cualquier fallo en la autorización (por ejemplo: decisiones de autorización en el lado del cliente, navegación forzada, etc.). Es distinto de los problemas de autenticación (por ejemplo: inscripción de dispositivos, identificación de usuarios, etc.).

M7 – Calidad del código en el lado del cliente

Se refiere a la captura de todos los problemas de implementación a nivel de código en el cliente móvil. Esto es distinto de los errores de codificación del servidor. Entrarían dentro de ella, cosas como desbordamientos de búfer, vulnerabilidades de cadena de formato y varios

otros errores de nivel de código en el que la solución es reescribir algún código que se esté ejecutando en el dispositivo móvil. Es decir, se centra en las vulnerabilidades creadas debido a errores de codificación.

M8 – Adulteración del código

Cubre parches binarios, modificación de recursos locales, incorrecta utilización de métodos y modificación de memoria dinámica, por ende, esta categoría cubre cualquier modificación que el adversario pueda realizar en el código de la aplicación.

M9 – Ingeniería inversa

Incluye el análisis del núcleo binario final para determinar su código fuente, bibliotecas, algoritmos y otros activos, esto puede utilizarse para explotar otras vulnerabilidades nacientes en la aplicación, así como para revelar información sobre los servidores backend, las claves criptográficas y la propiedad intelectual, por lo que hay que valorar que el propio código se encuentre correctamente ofuscado, ya que es la mejor manera de paliar con la ingeniería inversa.

M10 – Funcionalidad extraña

Esta vulnerabilidad surge cuando los desarrolladores no eliminan funciones adicionales, creadas durante el proceso de desarrollo para facilitar la prueba de la aplicación y que pueden ser explotadas para realizar ataques contra la aplicación.

2.9.2 Sistema Operativo Android para dispositivos móviles

Android, según Adslzone (2020), es un sistema operativo móvil diseñado para dispositivos móviles con pantalla táctil como teléfonos inteligentes o tablets, pero que también lo encontramos en otros dispositivos como relojes inteligentes, televisores o incluso en los sistemas multimedia de algunos modelos de coches. Un sistema operativo desarrollado por Google y basado en el Kernel de Linux y otros softwares de código abierto y que se ha convertido en el principal responsable de la popularización de muchos dispositivos inteligentes por el hecho de facilitar el uso de una gran cantidad de aplicaciones de forma sencilla.

2.7.2.1 Historia

Android Inc. fue fundada en el año 2003 por Andry Rubin, Rich Miner, Nick Sears y Chris White. Ellos fueron los que anunciaron Android como un sistema operativo orientado inicialmente para las cámaras digitales y que permitía conectarlas con el PC sin la necesidad de cables; sin embargo, puesto que no era un tema con mucho futuro, se decantaron por los teléfonos móviles. En tan solo dos años, el propio Google se interesa por esta compañía y decide comprar Android Inc. por la cantidad de 50 millones de dólares e incorporar a los cuatro fundadores a las filas de la compañía.

Dos años después, concretamente el 5 de noviembre de 2007, se anuncia la primera versión del sistema operativo, Android 1.0 Apple Pie, junto con la creación de la Open Handset Alliance, un consorcio de 78 compañías de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para dispositivos móviles, lo que hizo que Google liberase la mayoría del código de Android bajo licencia de Apache, una licencia libre y de código abierto; empero, hubo que esperar un poco hasta que comenzásemos a ver dispositivos con esta versión del sistema, algo que llegó en la segunda mitad de 2008.

Durante el segundo y tercer trimestre de 2010, según Adslzone (2020), Android consigue una cuota de mercado del 43.6% en Estados Unidos y es capaz de superar el 50% durante el cuarto trimestre de 2011 a nivel mundial, superando de esta manera al todo poderoso iOS de Apple.

Desde entonces y durante estos últimos años, el desarrollo de Android no ha parado y han sido muchas las versiones lanzadas del sistema con mejoras a nivel de rendimiento y seguridad, así como de soporte de muchas tecnología y nuevas funciones. Además, la gran comunidad de desarrolladores detrás del entorno de Google ha permitido extender la funcionalidad de los dispositivos. A principios de 2018, ya se superaban los dos millones de aplicaciones disponibles en la tienda oficial de aplicaciones para Android, Google Play. Incluso hemos visto cómo han ido apareciendo otras tiendas de aplicaciones no oficiales con gran cantidad de aplicaciones para el sistema operativo de Google.

2.7.2.2 Versiones y actualizaciones

Android ha recibido, desde su lanzamiento y hasta el día de hoy, numerosas actualizaciones. Diferentes versiones del sistema que han ido arreglando fallos detectados, añadiendo nuevas funciones, soportes para nuevas tecnologías, etc. Unas versiones que curiosamente han sido desarrolladas bajo un nombre en código de un elemento relacionado con postres o dulces y que además ha ido respetando rigurosamente un orden alfabético.

OS Name	Versión	Fecha de lanzamiento
Android Apple Pie	versión 1.0	23 de septiembre de 2008.
Android Banana Bread	versión 1.1	9 de febrero de 2009.
Android Cupcake	versión 1.5	25 de abril de 2009.
Android Donut	versión 1.6	15 de septiembre de 2009.
Android Éclair	versión 2.0-2.1	26 de octubre de 2009.
Android Froyo	versión 2.2-2.3	20 de mayo de 2010.
Android Gingerbread	versión 2.3-2.7	6 de diciembre de 2010.
Android Honeycomb	versión 3.0-3.2.6	22 de febrero de 2011.
Android Ice Cream Sandwich	versión 4.0-4.0.5	18 de octubre de 2011.
Android Jelly Bean	versión 4.1-4.3.1	9 de julio de 2012.
Android Kitkat	versión 4.4-4.4.4	31 de octubre de 2012.
Android Lollipop	versión 5.0-5.1.1	12 de noviembre de 2014.
Android Marshmallow	versión 6.0-6.0.1	5 de octubre de 2015.
Android Nougat	versión 7.0-7.1.2	15 de junio de 2016.
Android Oreo	versión 8.0-8.1	21 de agosto de 2017.
Android Pie	versión 9.0	6 de agosto de 2018.
Android 10	versión 10.0	3 de septiembre de 2019.

Tabla 20**.** Que es Android.

Fuente: elaboración propia, basada en la información de Adslzone.

2.7.2.3 Principales componentes del sistema operativo Android

Dentro de la propia arquitectura del sistema, podemos destacar los principales componentes de Android:

- **Núcleo Linux:** el núcleo del sistema es Linux y actúa como una capa de abstracción entre el hardware del dispositivo y las aplicaciones instaladas. Además, el sistema operativo de Google depende de Linux para otros servicios básicos como la seguridad, gestión de memoria, gestión de procesos, pila de red o controladores.
- **Runtime:** el sistema operativo de Google para dispositivos móviles incluye un conjunto de bibliotecas que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje de programación Java. Cada aplicación Android corre su propio proceso con su instancia a la máquina virtual Dalvik. Esta máquina ejecutaba hasta la versión 5.0 archivos en formato. dex, pero a partir de esa versión se utilizar el ART, que compila totalmente al momento de instalación de la aplicación.

- **Bibliotecas:** el sistema operativo Android incluye un conjunto de bibliotecas de C o C++ utilizadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de las aplicaciones de Android. Entre estas bibliotecas, caben destacar a System C, bibliotecas de medios, de gráficos, 3 D y SQLite, entre otras.
- **Marco del trabajo de aplicaciones:** el entorno de Google permite que los desarrolladores tengan acceso a las mismas API del entorno de trabajo utilizadas por las aplicaciones base y es que la arquitectura de Android está diseñada para simplificar la reutilización de componentes, es decir, cualquier aplicación puede publicar sus capacidades y que otras aplicaciones puedan reutilizarlas dentro de unas reglas de seguridad.
- **Aplicaciones:** Android cuenta con ciertas aplicaciones base que permiten el uso de las funciones básicas de un dispositivo como: correo electrónico, mensajes de texto SMS, calendario, mapas, navegador, contactos y otros. Aplicaciones desarrolladas en lenguaje Java.

El marco teórico explica y contrasta las teorías base del área que está siendo investigada. Es más común ver un marco teórico en investigaciones de tipo puro o básico.

Cuando se trata de investigaciones de tipo aplicado o evaluativo, con frecuencia se requiere de un marco conceptual, el cual es una aclaración de conceptos. Se escribe de manera hilvanada, para no confundirlo con un glosario y es recomendable que su redacción se haga luego de elaborar un mapa conceptual.

Tanto un marco teórico como uno conceptual derivan de manera natural de los hallazgos en el estado de la cuestión. Es responsabilidad del investigador identificar si el debate en su área gira alrededor de discusiones teóricas o si, por el contrario, se concentra en las mejores maneras de usar conceptos para encarar el problema y proponer escenarios de solución.

2.7.2.4 Aplicaciones Android

Si hay algo que caracterice al sistema operativo Android, es su gran libertad para el desarrollo de aplicaciones. Tanto es así que cuenta con una gran tienda de apps disponibles para que cualquier usuario pueda instalarlas en su smartphone o tablet y en el que los desarrolladores pueden subir sus creaciones y esté disponible para todos los usuarios del sistema.

Google Play es la tienda o plataforma online desarrollada por el gigante buscador y donde podemos encontrar todo tipo de apps para Android. Su app está instalada por

defecto en todos los dispositivos Android para que los usuarios tengan un fácil acceso a todo el catálogo de apps de música, juegos, noticias, clima, educación, compras, salud, utilidades, mapas, deporte, bancarias, etc.

Para poder usar la tienda de aplicaciones, es necesario asociar una cuenta de Gmail, pero lo cierto es que, debido a la gran popularidad del sistema operativo de Google, hoy podemos encontrar otras tiendas de aplicaciones alternativas a Google Play en las que encontraremos otras muchas herramientas y aplicaciones.

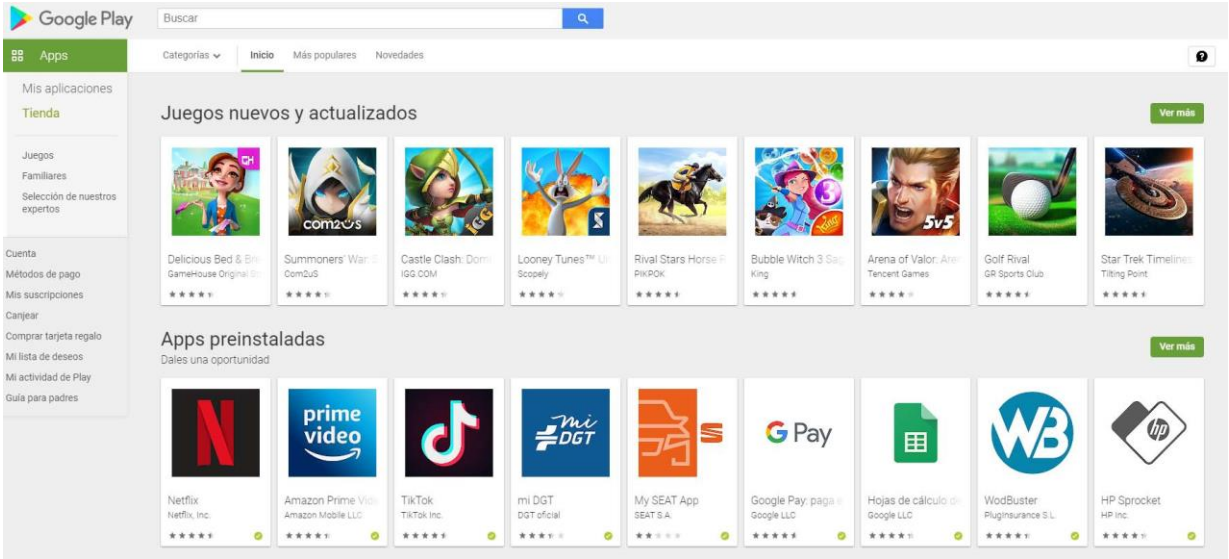


Figura 13. Que es Android.
Fuente Adslzone.

De cualquier forma, las aplicaciones para Android están comprimidas en formato APK, lo que facilita su instalación desde cualquier explorador de archivos en la mayoría de los dispositivos. Eso sí, es posible que antes que nada, tengamos que acceder a los ajustes del sistema para activar la opción que nos permite instalar aplicaciones de fuentes desconocidas y es que Android viene configurado para que sea en Google Play desde donde se realice la instalación de todo tipo de aplicaciones; no obstante, activando esta opción podremos instalar el APK que nos hayamos descargado sin mayor problema con el simple hecho de ir a la carpeta donde tenemos guardada la app y tocar sobre el archivo APK. Ahora bien, es importante tener extremada precaución y elegir bien el sitio desde donde vamos a descargar las apps, ya que son muchos los sitios que se aprovechan de esta gran popularidad para ofrecer apps con malware.

Capítulo 3 Marco metodológico

3.1 Tipo de investigación

Se utilizará para efectos de este trabajo, la investigación de tipo aplicada, ya que se emplearán las mejores prácticas de seguridad para aplicaciones móviles propuestas por OWASP, con el objetivo de recomendar una metodología que se adapte como base para realizar auditorías de seguridad en aplicaciones móviles para las PyMEs con políticas de BYOD.

3.2 Alcance investigativo

Se utilizará el alcance investigativo descriptivo, tomando como referencia la definición presenta por Vargas (2004), en la que define el alcance investigativo descriptivo como: “el tipo de estudio que busca especificar las propiedades, las características y los perfiles importantes de personas, grupos, comunidades o cualquier otro fenómeno que se someta a análisis”, ya que se utilizará el perfil de las pequeñas y medianas empresas (PyMEs) que trabajan con políticas de BYOD y no tienen establecidas una metodología para auditar los dispositivos móviles.

El fin de este trabajo es que cualquier empresa que cuente con el perfil de una PyME con políticas de BYOD, pueda implementar la metodología para realizar auditorías de los dispositivos móviles.

3.3 Enfoque

Se utilizará el enfoque alternativo propuesto por Luis Naranjo (2020), en el que se indica que se “ubica en el paradigma pragmático y se hacen explícitas las dimensiones epistemológica, ontológica y axiológica de la investigación”. A continuación, se presenta el detalle sobre cómo se abarcarán las tres dimensiones fundamentales.

Para el punto de vista epistemológico, que se refiere a la postura del investigador frente a su objeto de estudio, los investigadores la abordarán desde una perspectiva aplicada, debido a que se desarrollarán casos prácticos que permitan realizar análisis estáticos y dinámicos de vulnerabilidades en aplicaciones utilizadas en las PyMEs con políticas de BYOD.

Para el punto de vista ontológico definido por Gruber (1993) como: “lo que existe es exactamente aquello que puede ser representado”. El trabajo presenta las ontologías o conceptualización de áreas como: PyMEs, aplicaciones móviles, escaneo de vulnerabilidades, OWASP, Android, auditorías. La siguiente figura muestran las ontologías creadas para el conjunto de datos de auditoría de dispositivos móviles.

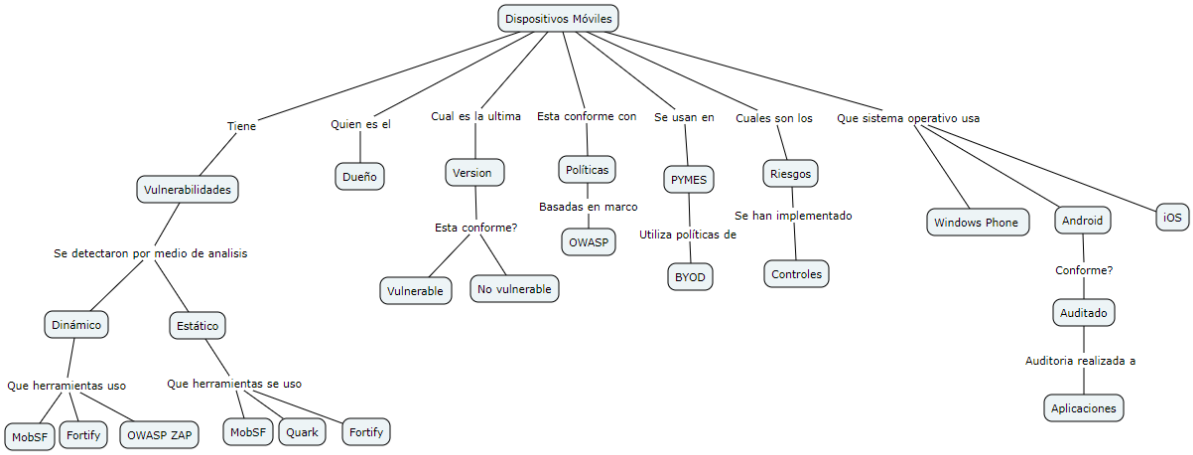


Figura 14. Ontología de auditorías de dispositivos móviles.
Fuente propia.

Finalmente, para el punto de dimensión axiológica, el cual se refiere a la escala de valores sobre lo que se va a medir, se pretende abordar desde el entendimiento y comprensión para determinar cuándo la metodología ofrece un nivel de evaluación aceptable, para lo mismo, el esfuerzo de los investigadores se centra en evaluar la precisión de acuerdo con las siguientes escalas.

Porcentaje total	Definición
Menos 25%	La metodología no ayudó reducir las vulnerabilidades/riesgos de un dispositivo móvil.
26% - 50%	La metodología ayuda a eliminar algunas vulnerabilidades, pero no lo suficiente para tener un dispositivo móvil seguro.
51% - 79%	El uso de la metodología es adecuado para asegurar un dispositivo móvil.
80% - 100%	La metodología ayuda a garantizar un dispositivo móvil seguro, al menos a partir de las 10 vulnerabilidades más críticas según OWASP.

Tabla 21. Criterio axiológico para garantizar un dispositivo seguro.
Fuente: elaboración propia.

3.4 Diseño

Este trabajo emplea un enfoque alternativo utilizando una investigación evaluativa. Las características de la investigación evaluativa se adaptan para la implementación de una metodología que se utilice como base para realizar auditorías de seguridad en aplicaciones móviles para las PyMEs con políticas de BYOD, para la toma de decisiones a la hora de

implementar nuevos controles, dispositivos y software de seguridad y a su vez, permite evaluar la eficacia de una organización en asegurar su información.

3.5 Población y muestreo

La población que ha sido tomada como base de estudio y análisis para la recolección de datos que sirve como modelo para el diseño de la metodología, son todas aquellas pequeñas y medianas empresas (PyMEs) que tengan políticas de BYOD.

Tomando en cuenta lo anterior, el muestreo utilizado en esta investigación es no probabilístico, basado en método intencional o por juicio. Los investigadores pueden utilizar una muestra intencional porque los entrevistados cumplen con una descripción o propósito específico que es necesario para realizar la investigación, en este caso, solo aplica a las PyMEs que tengan políticas de BYOD.

3.6 Instrumentos de recolección de datos

Es este apartado se utilizarán herramientas de escaneo de vulnerabilidades para recopilar información sobre las amenazas y fallos de seguridad que afectan a los dispositivos móviles. Las herramientas de escaneo de vulnerabilidades para dispositivos móviles a utilizar serán: MobSF, Fortfy OnDemand y OWASP ZAP.

3.7 Técnicas de análisis de la información

Se realizará un análisis a partir de la información extraída de los escaneos de vulnerabilidades a través de análisis estáticos y dinámicos, para mostrar descubrimientos relevantes acerca de ataques basados en vulnerabilidades, además de las técnicas y herramientas utilizadas actualmente para detectar vulnerabilidades en aplicaciones móviles, esta información servirá de base para la realización de la metodología.

De acuerdo con los resultados obtenidos, también se propondrán medidas para mitigar las vulnerabilidades, políticas de seguridad para dispositivos móviles y controles para reducir el riesgo según la vulnerabilidad detectada.

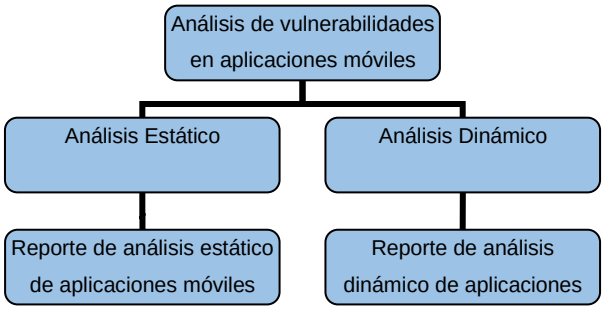


Figura 15. Métodos de análisis de vulnerabilidades en aplicaciones móviles.

Fuente: propia.

3.8 Estrategia de desarrollo de la propuesta

La metodología que se presentará a la PyME estará basada en las mejores prácticas de seguridad para aplicaciones móviles de OWAS MASVS y MSTG. Para asegurar la mayor compatibilidad y facilidad de uso, se analizarán varios escenarios y se escogerá el que permita mayor flexibilidad.

Capítulo 4. Caso práctico

Se desarrollará el caso práctico en este capítulo, el cual se basa en auditar dos aplicaciones móviles para Android de acuerdo con las metodologías y apartados descritos en Capítulo 2.

El desarrollo del caso práctico cuenta con un análisis de vulnerabilidades estático y dinámico con el fin de identificar las vulnerabilidades críticas y los riesgos que posee la aplicación.

- El análisis estático nos permitirá realizar la revisión del código de la aplicación y las librerías utilizadas. Dicho análisis lo realizaremos con la herramienta MobSF.
- El análisis dinámico nos permitirá evaluar la aplicación mientras se ejecuta en tiempo real, para el cual utilizaremos un dispositivo móvil Android.

Debido a que la aplicación no utiliza información de carácter médico o bancario, se utilizará el estándar MASVS-L1 (nivel de seguridad estándar) que contiene las mejores prácticas de seguridad para todas las aplicaciones móviles y que, además, previene los riesgos descritos en el Mobile TOP 10 de OWASP.

Antes de realizar los análisis, se recopila información básica de la aplicación (funcionalidad, permisos requeridos para la instalación, autenticación, componentes hardware (bluetooth, GPS, etc.), conexiones de red (datos móviles / Wi-Fi) y datos requeridos para realizar sus funciones (contactos, calendario, etc.), con el fin de identificar las posibles áreas más críticas. Al final de este capítulo se presentan los resultados más relevantes obtenidos de ambos análisis.

4.1 Recopilación de información de las aplicaciones a auditar

Las aplicaciones seleccionadas para llevar a cabo la auditoría de seguridad son WhatsApp y Instagram para efectos del caso práctico, las cuales cumplen con las siguientes funciones:

- WhatsApp es una aplicación de mensajería gratuita disponible para los dispositivos móviles.

- Instagram es una red social y una aplicación móvil al mismo tiempo, la cual permite a sus usuarios subir imágenes y vídeos con múltiples efectos fotográficos como filtros, marcos, colores retro, etc.

Se ha decidido analizar WhatsApp e Instagram debido a que son las aplicaciones más utilizadas en dispositivos móviles a nivel mundial y a su vez, por las PyMEs.

Figura 16. WhatsApp vs Instagram.
Fuente: Google Imágenes.

WhatsApp

- **Funcionalidad:** permite enviar y recibir mensajes, llamadas, fotos, videos, documentos y mensajes de voz.
- **Permisos requeridos para la instalación:** cámara, contactos, micrófono, almacenamiento, locación, registro de llamadas, cámara, acceso a los mensajes.

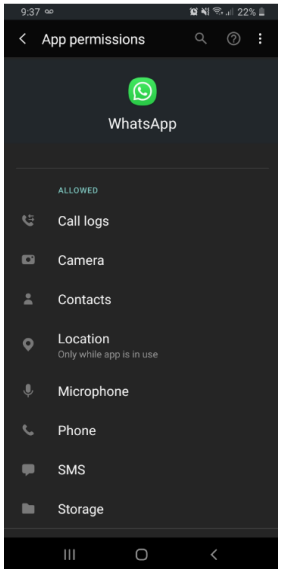


Figura 17. Permisos de WhatsApp.
Fuente propia.

- **Autenticación:** autenticación multifactor.

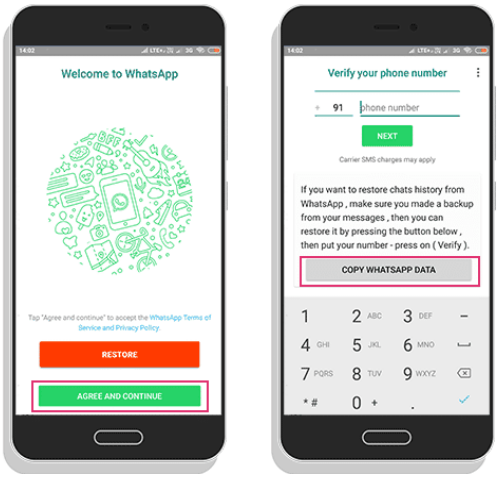


Figura 18**.** Pantalla de inicio de WhatsApp.

Fuente: The Next Pro.

- **Componentes hardware:** micrófono, cámara, almacenamiento, bluetooth.
- **Conexiones de red**: datos móviles / Wi-Fi

Instagram

- **Funcionalidad:** puede compartir información, noticias y contenidos audiovisuales
- **Permisos requeridos para la instalación:** cámara, contactos, micrófono, almacenamiento, locación, teléfono.

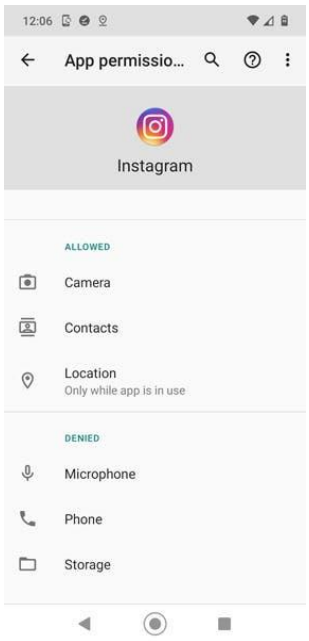


Figura 19**.** Permisos de Instagram**.**

Fuente: imágenes de Google.

- **Autenticación:** autenticación multifactor.

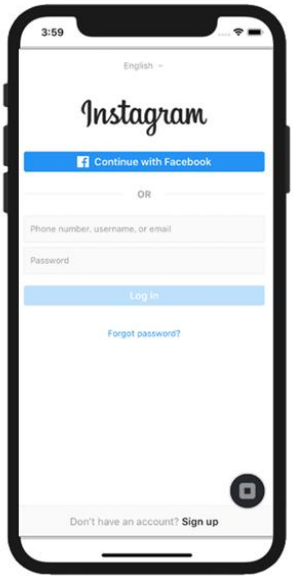


Figura 20. Pantalla de inicio Instagram.
Fuente: imágenes de Google.

- **Componentes hardware:** micrófono, cámara, GPS, almacenamiento.
- **Conexiones de red**: datos móviles / Wi-Fi

4.2 Herramientas utilizadas para el análisis estático

Se específica a continuación, las herramientas que utilizamos para la elaboración del análisis estático de ambas aplicaciones móviles.

4.2.1 VirtualBox

Debido a que se necesita correr la herramienta de análisis en un servidor Linux, instalamos VirtualBox que sirve para hacer máquinas virtuales con instalaciones de sistemas operativos, esto nos permite instalar otros sistemas operativos dentro de la computadora. El sistema operativo que utilizamos para la máquina virtual es Ubuntu 18.04.4 debido a que es la versión recomendada para instalar las herramientas de análisis de vulnerabilidades.

4.2.2 APK Extractor

Descargamos la aplicación APK Extractor para obtener el APK de las dos aplicaciones por analizar, la cual que se instala en el dispositivo móvil Android para extraer cualquier APK desde todas las aplicaciones que se han instalado en el dispositivo móvil.

Una vez que se extrajo los APK, se compartieron por medio de correo electrónico para iniciar así el análisis en la máquina virtual de Ubuntu. Los pasos de instalación de esta herramienta se encuentran en el Apéndice 2.

4.2.3 MobSF

Utilizamos la herramienta Mobile Security Framework (MobSF) para realizar el análisis estático de las dos aplicaciones, es una aplicación móvil automatizada para Android, iOS y Windows, la cual permite hacer análisis de malware y evaluación de seguridad. MobSF admite binarios de aplicaciones móviles (APK, IPA y APPX) junto con el código fuente comprimido. Los pasos y requisitos de instalación de esta herramienta se encuentran en el Apéndice 3.

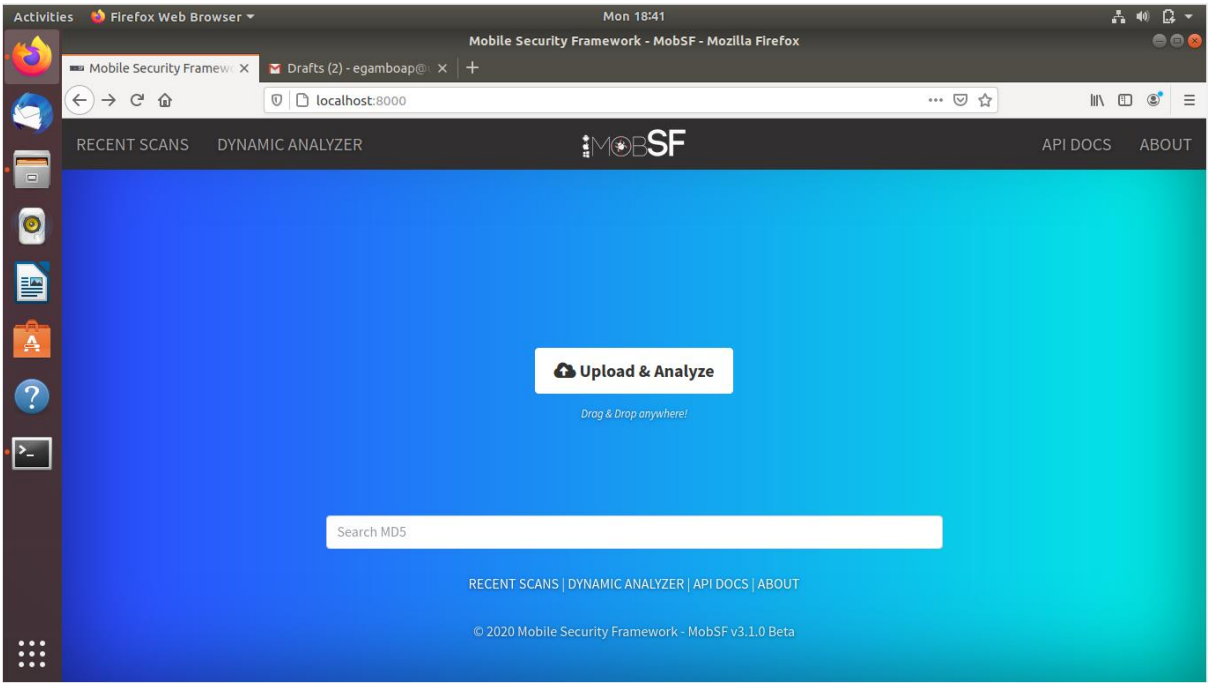


Figura 21. MobSF página principal.
Fuente: propia con base en la página principal de MobSF.

Tal y como lo define Denise Giusto (2016), los resultados del análisis estático de MobSF se categorizan en las siguientes secciones:

- **Información del archivo:** nos mostrará un resumen de sus características más sobresalientes que podrán permitirnos su posterior identificación. Entre ellas encontramos el nombre de la muestra, su tamaño y los hashes resultados de diferentes funciones hash (MD5, SHA1, SHA256).

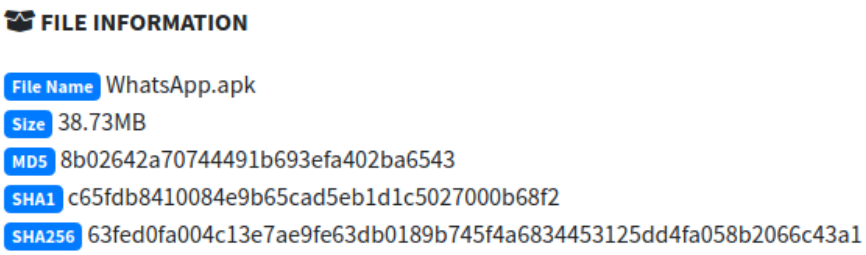


Figura 22. Información de archivo WhatsApp.
Fuente: propia con base en el escaneo de MobSF.

- **Información de la aplicación:** aquí encontraremos detalles de la aplicación mayormente obtenidos del Android Manifest, como el nombre del paquete, el nombre de clase de la actividad principal y atributos referentes a los requisitos de la plataforma para la cual la aplicación fue desarrollada.

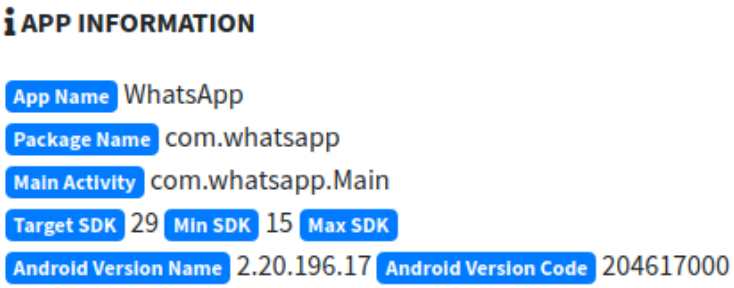


Figura 23. Información de la aplicación WhatsApp.
Fuente: propia con base en el escaneo de MobSF.

- **Posibles elementos vulnerables:** seguidamente encontraremos en la pantalla cuatro recuadros que nos resumen la información referente a las actividades, servicios, receptores de intentos y proveedores de contenidos, indicando cuántos de ellos son exportados. La identificación de estos cuatro elementos es un paso rutinario en cualquier proceso de análisis de malware o pentesting de aplicaciones, ya que nos permitirá no solo saber cómo se comporta la aplicación, sino también distinguir posibles puntos de explotación.

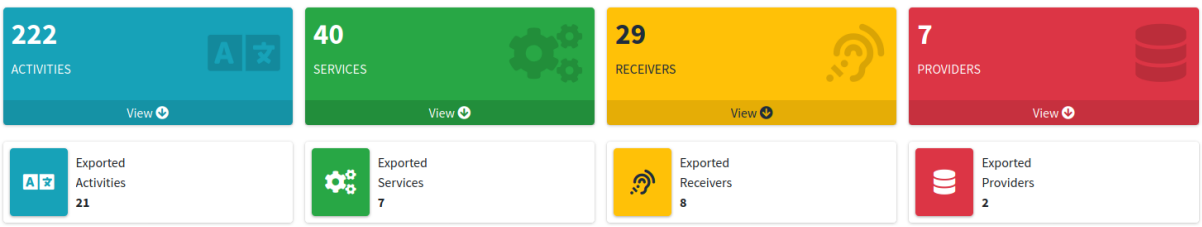


Figura 24. Posibles elementos vulnerables de WhatsApp.
Fuente: propia con base en el escaneo de MobSF.

- **Análisis del código de compilado:** dentro de las opciones para el análisis del código fuente, MobSF nos permite acceder a un listado de las clases tanto en formato java como en smali y también al archivo manifiesto. Además, encontramos dos opciones: una para escanear nuevamente la muestra y otra para iniciar su análisis dinámico.

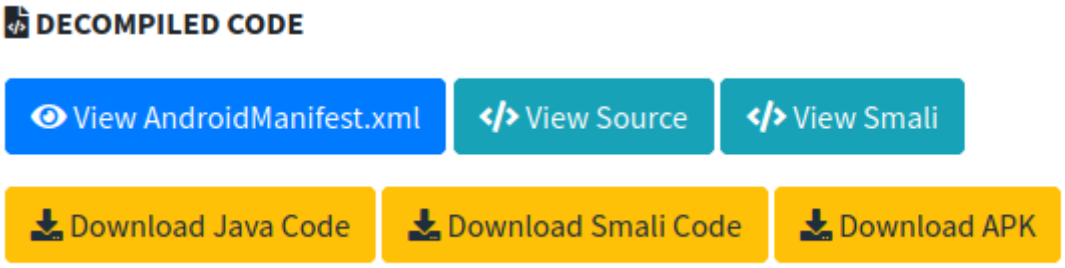


Figura 25**.** Análisis de código de compilado de WhatsApp.
Fuente: propia con base en el escaneo de MobSF.

- **Información del certificado:** el análisis del certificado de un APK puede arrojar datos muy interesantes en cuanto a quién ha desarrollado la aplicación y qué otras muestras maliciosas se han encontrado con el mismo certificado.

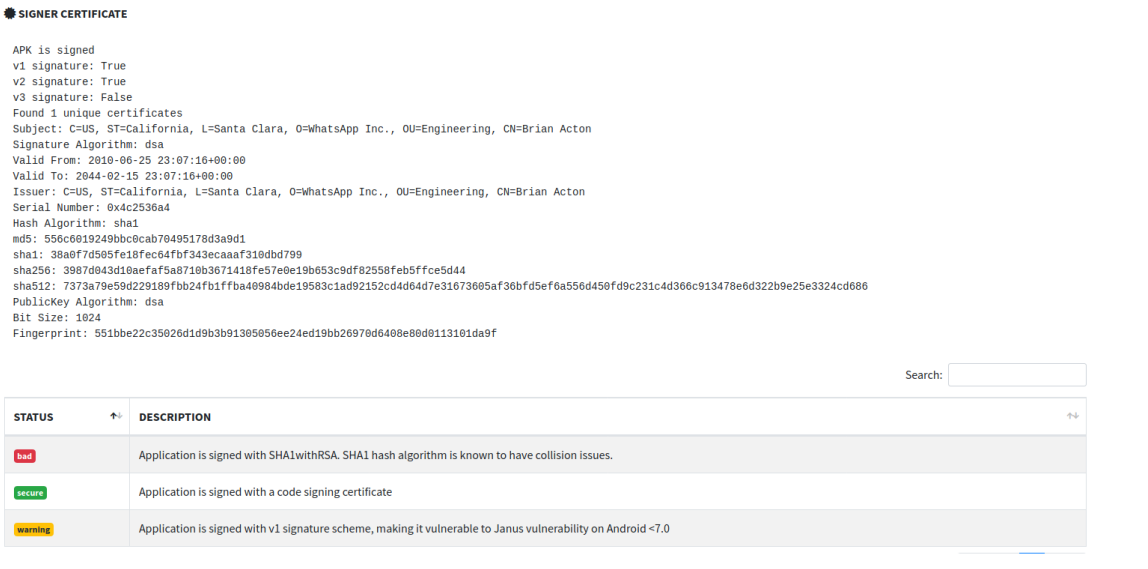
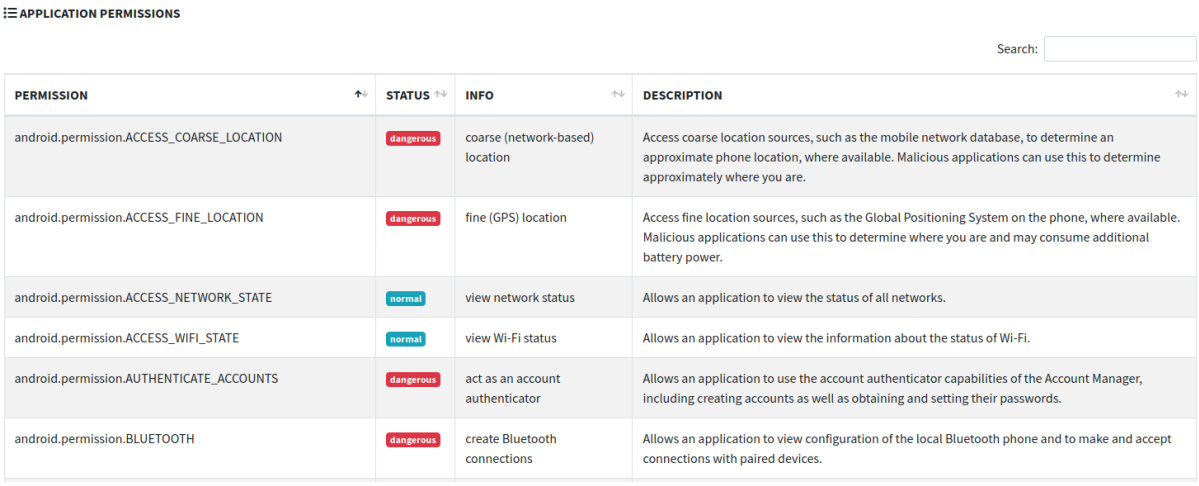


Figura 26**.** Información de certificado de WhatsApp.
Fuente: propia con base en el escaneo de MobSF.

- **Listado de permisos**: en esta sección podremos observar una lista de los permisos declarados en el manifiesto de la aplicación, juntamente con su descripción y una

categorización según la peligrosidad que puede representar para el sistema al acceder a información o funcionalidad sensible.



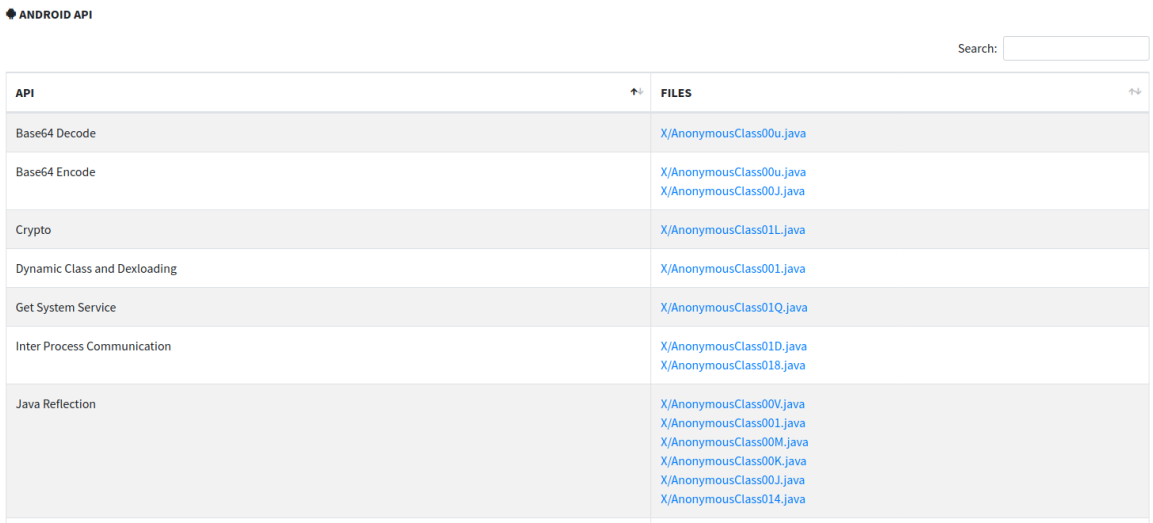
APPLICATION PERMISSIONS

Search:

PERMISSION	STATUS	INFO	DESCRIPTION
android.permission.ACCESS_COARSE_LOCATION	dangerous	coarse (network-based) location	Access coarse location sources, such as the mobile network database, to determine an approximate phone location, where available. Malicious applications can use this to determine approximately where you are.
android.permission.ACCESS_FINE_LOCATION	dangerous	fine (GPS) location	Access fine location sources, such as the Global Positioning System on the phone, where available. Malicious applications can use this to determine where you are and may consume additional battery power.
android.permission.ACCESS_NETWORK_STATE	normal	view network status	Allows an application to view the status of all networks.
android.permission.ACCESS_WIFI_STATE	normal	view Wi-Fi status	Allows an application to view the information about the status of Wi-Fi.
android.permission.AUTHENTICATE_ACCOUNTS	dangerous	act as an account authenticator	Allows an application to use the account authenticator capabilities of the Account Manager, including creating accounts as well as obtaining and setting their passwords.
android.permission.BLUETOOTH	dangerous	create Bluetooth connections	Allows an application to view configuration of the local Bluetooth phone and to make and accept connections with paired devices.

Figura 27. Listado de permisos de WhatsApp.
Fuente: propia con base en el escaneo de MobSF.

- **Android API:** esta sección resulta muy útil para un analista, ya que permite identificar rápidamente qué funcionalidades de la API del sistema son accedidas por cada clase de la aplicación. De este modo, es muy sencillo identificar qué función realiza cada clase y podremos concentrarnos en aquello que realmente nos interese.



ANDROID API

Search:

API	FILES
Base64 Decode	X/AnonymousClass00u.java
Base64 Encode	X/AnonymousClass00u.java X/AnonymousClass00J.java
Crypto	X/AnonymousClass01L.java
Dynamic Class and Dexloading	X/AnonymousClass001.java
Get System Service	X/AnonymousClass01Q.java
Inter Process Communication	X/AnonymousClass01D.java X/AnonymousClass018.java
Java Reflection	X/AnonymousClass00V.java X/AnonymousClass001.java X/AnonymousClass00M.java X/AnonymousClass00K.java X/AnonymousClass00J.java X/AnonymousClass014.java

Figura 28. Android API de WhatsApp.
Fuente: propia con base en el escaneo de MobSF.

- **Extras de seguridad:** además de las secciones antes discutidas, podremos encontrar otras categorías con detalles de elementos a ser considerados en cualquier análisis. Por ejemplo, podremos ver una sección en la que se especifica

con detalle, cuáles son las actividades, servicios, broadcast receivers y content providers especificados en la aplicación o acceder a un listado de las strings encontradas dentro del código fuente.

4.2.4 Fortify on Demand

Fortify on Demand al ser servicio (SaaS), ningún software se tuvo que instalar o administrar. Los usuarios simplemente cargan los binarios de su aplicación y/o proporcione una URL para la prueba. Fortify on Demand puede realizar una prueba estática y/o dinámica, verificar todos los resultados y presentar resultados correlacionados en una interfaz detallada o por medio de reporte.

4.3 Herramientas utilizadas para el análisis dinámico

Se define a continuación las herramientas que utilizamos para la elaboración del análisis dinámico de ambas aplicaciones móviles.

4.3.1 Genymotion

Es un emulador de Android que aprovecha la arquitectura x86 para ejecutar de forma fluida y rápida distintos dispositivos Android. Olvidando la lentitud del emulador nativo de Android se puede ejecutar todo tipo de aplicaciones y juegos.
Los sistemas operativos con soporte oficial son:

- Windows 32/64 bits.
- Mac OS X 64 bits.
- Linux Ubuntu 32/64 bits.
- Linux Debian 64 bits.

Los pasos y requisitos de instalación de esta herramienta se encuentran en el Apéndice 4.

4.3.2 OWASP ZAP Proxy

Otra herramienta que utilizaremos para el análisis dinámico es OWASP Zed Attack Proxy (ZAP), debido a que nos permite analizar el flujo de respuestas HTTP de ambas aplicaciones. Los pasos y requisitos de instalación de esta herramienta se encuentran en el Apéndice 5.

4.4 Resultados del análisis estático

4.4.1 MobSF

4.4.1.1 WhatsApp

Se realizó un análisis estático de la aplicación WhatsApp a través de MobSF con el fin de identificar las vulnerabilidades y riesgos que posee la aplicación.

La versión de WhatsApp que se analizó fue 2.20.196.17, publicada el 4 de agosto de 2020, la cual tiene como requisitos mínimos:

- Sistema operativo Android 4.0.3 o superior
- Se recomienda un plan ilimitado de datos de Internet
- Las tablets no son compatibles

Los resultados del análisis revelaron que en la aplicación se ejecutan 222 actividades de las cuales se exportan 21; 40 servicios, 7; 29 receptores, 8 y 7 proveedores de contenidos, 2.

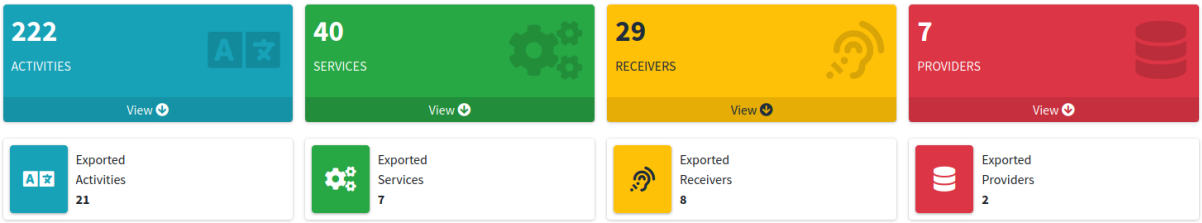


Figura 29. Posibles elementos vulnerables de WhatsApp.

Fuente: propia con base en el escaneo de MobSF.

Puntuación de la aplicación

La aplicación recibe una aplicación de CVSS de 5.9 y de seguridad de 5/100. Se detecta un problema con severidad alta, debido a que la aplicación utiliza un certificado que firma con SHA1 de RSA. Se sabe que el algoritmo hash SHA1 tiene problemas de colisión y está obsoleto. El reporte completo con los hallazgos del análisis realizado se adjunta en la sección de Apéndice 1.

Permisos

Se presentan en esta sección, los diez permisos más relevantes de los 32 encontrados a través del escaneo y de acuerdo con la severidad asignada por MobSF.

Permisos	Severidad	Información	Descripción
android.permission.AUTHENTICATE_ACCOUNTS	Peligroso	actuar como un autenticador de cuenta	Permite que una aplicación utilice las capacidades de autenticación de cuentas del administrador de cuentas, incluida la creación de cuentas y la obtención y configuración de sus contraseñas.
android.permission.USE_CREDENTIALS	Peligroso	utilizar las credenciales de autenticación de una cuenta	Permite que una aplicación solicite tokens de autenticación.
android. permission. MANAGE_ACCOUNTS	Peligroso	administrar la lista de cuentas	Permite que una aplicación realice operaciones como agregar y eliminar cuentas y eliminar su contraseña.
android. permission. ACCESS_FINE_LOCATION	Peligroso	ubicación (GPS)	Acceda a fuentes de ubicación precisas, como el sistema de posicionamiento global en el teléfono, donde esté disponible. Las aplicaciones maliciosas pueden usar esto para determinar dónde se encuentra y pueden consumir energía adicional de la batería.
android. permission. BLUETOOTH	Peligroso	crear conexiones Bluetooth	Permite que una aplicación vea la configuración del teléfono Bluetooth local y realice y acepte conexiones con dispositivos emparejados.
android. permission. CAMERA	Peligroso	tomar fotos y videos	Permite que la aplicación tome fotos y videos con la cámara. Esto permite que la aplicación recopile imágenes que la

			cámara está viendo en cualquier momento.
android.permission.CHANGE_WIFI_STATE	Peligroso	Cambiar estado Wi-Fi	Permite que una aplicación se conecte y se desconecte de puntos de acceso Wi-Fi y realice cambios en las redes Wi-Fi configuradas.
com.huawei.android.launcher.permission.WRITE_SETTINGS	Peligroso	modificar la configuración global del sistema	Permite que una aplicación modifique los datos de configuración del sistema. Las aplicaciones maliciosas pueden dañar la configuración de su sistema.
android.permission.REQUEST_INSTALL_PACKAGES	Peligroso	Permite que una aplicación solicite la instalación de paquetes.	Las aplicaciones maliciosas pueden usar esto para intentar engañar a los usuarios para que instalen paquetes maliciosos adicionales.
android.permission.READ_CONTACTS	Peligroso	leer datos de contacto	Permite que una aplicación lea todos los datos de contacto (direcciones) almacenados en tu teléfono. Las aplicaciones maliciosas pueden usar esto para enviar sus datos a otras personas.

Tabla 22**.** Top 10 permisos más peligrosos de WhatsApp.
Fuente: propia con base en el escaneo de MobSF.

Análisis del archivo de manifiesto

El archivo de manifiesto describe información esencial sobre su aplicación para las herramientas de compilación de Android, el sistema operativo Android y Google Play. Además de los componentes de la aplicación incluyen todas las actividades, servicios, receptores de transmisión y proveedores de contenido.

A continuación, se presentan las diez vulnerabilidades más relevantes de los 63 encontradas a través del escaneo, de acuerdo con la severidad asignada por MobSF.

Vulnerabilidad	Severidad	Descripción
Activity (com.whatsapp.payments.receiver. IndiaUpiPayIntentReceiverActivity) is not Protected.	Alta	Se descubre que una actividad se comparte con otras aplicaciones en el dispositivo, por lo que se deja accesible para cualquier otra aplicación en el dispositivo. La presencia de intent-filter indica que la actividad se exporta explícitamente.
Activity (com.whatsapp.Conversation) is not Protected. An intent-filter exists.	Alta	Se descubre que una actividad se comparte con otras aplicaciones en el dispositivo, por lo que se deja accesible para cualquier otra aplicación en el dispositivo. La presencia de intent-filter indica que la actividad se exporta explícitamente.
Activity-Alias (com.whatsapp.SetAsProfilePhoto) is not Protected. An intent-filter exists.	Alta	Se encuentra que un Alias de actividad se comparte con otras aplicaciones en el dispositivo, por lo tanto, lo deja accesible para cualquier otra aplicación en el dispositivo. La presencia de intent-filter indica que Activity-Alias se exporta explícitamente.
Launch Mode of Activity (com.whatsapp.registration.EULA) is not standard.	Alta	Una actividad no debe tener el atributo de modo de lanzamiento establecido en "singleTask / singleInstance", ya que se convierte en la actividad raíz y es posible que otras aplicaciones lean el contenido de la intención de llamada. Por lo tanto, es necesario utilizar el atributo de modo de lanzamiento "estándar" cuando se incluye información confidencial en una intención.

Launch Mode of Activity (com.whatsapp.registration.RegisterPhone) is not standard.	Alta	Una actividad no debe tener el atributo de modo de lanzamiento establecido en "singleTask / singleInstance", ya que se convierte en la actividad raíz y es posible que otras aplicaciones lean el contenido de la intención de llamada. Por lo tanto, es necesario utilizar el atributo de modo de lanzamiento "estándar" cuando se incluye información confidencial en una intención.
Activity (com.whatsapp.registration.VerifySms) is not Protected. An intent-filter exists.	Alta	Se descubre que una actividad se comparte con otras aplicaciones en el dispositivo, por lo que se deja accesible para cualquier otra aplicación en el dispositivo. La presencia de intent-filter indica que la actividad se exporta explícitamente.
Activity-Alias (com.whatsapp.VerifySmsDeepLink) is not Protected. An intent-filter exists.	Alta	Se encuentra que un Alias de actividad se comparte con otras aplicaciones en el dispositivo, por lo tanto, lo deja accesible para cualquier otra aplicación en el dispositivo. La presencia de intent-filter indica que Activity-Alias se exporta explícitamente.
Launch Mode of Activity (com.whatsapp.registration.VerifyTwoFactorAuth) is not standard.	Alta	Una actividad no debe tener el atributo de modo de lanzamiento establecido en "singleTask / singleInstance", ya que se convierte en la actividad raíz y es posible que otras aplicaciones lean el contenido de la intención de llamada. Por lo tanto, es necesario utilizar el atributo de modo de lanzamiento "estándar" cuando se incluye información confidencial en una intención.

Launch Mode of Activity (com.whatsapp.registration.RegisterName) is not standard.	Alta	Una actividad no debe tener el atributo de modo de lanzamiento establecido en "singleTask / singleInstance", ya que se convierte en la actividad raíz y es posible que otras aplicaciones lean el contenido de la intención de llamada. Por lo tanto, es necesario utilizar el atributo de modo de lanzamiento "estándar" cuando se incluye información confidencial en una intención.
TaskAffinity is set for Activity (com.whatsapp.voipcalling.VoipActivityV2)	Alta	Si taskAffinity está configurado, entonces otra aplicación podría leer los Intents enviados a Actividades que pertenecen a otra tarea. Utilice siempre la configuración predeterminada manteniendo la afinidad como el nombre del paquete para evitar que otra aplicación lea información confidencial dentro de Intents enviados o recibidos.

Tabla 23. Top 10 vulnerabilidades con severidad alta de WhatsApp.
Fuente: propia con base en el escaneo de MobSF.

Análisis de código

Se presentan a continuación, los cinco problemas de código más relevantes de los diez encontrados a través del escaneo, de acuerdo con la severidad asignada por MobSF.

Problema	Severidad	Estándares
La aplicación utiliza un generador inseguro de números aleatorios.	Alta	CVSS V2: 7.5 (high) CWE: CWE-330 Use of Insufficiently Random Values OWASP Top 10: M5: Insufficient Cryptography OWASP MASVS: MSTG-CRYPTO-6
MD5 es un hash débil conocido por tener colisiones de hash.	Alta	CVSS V2: 7.4 (high) CWE: CWE-327 Use of a Broken or Risky Cryptographic Algorithm OWASP Top 10: M5: Insufficient Cryptography

		OWASP MASVS: MSTG-CRYPTO-4
Los archivos pueden contener información confidencial codificada como nombres de usuario, contraseñas, claves, etc.	Alta	CVSS V2: 7.4 (high) CWE: CWE-312 Cleartext Storage of Sensitive Information OWASP Top 10: M9: Reverse Engineering OWASP MASVS: MSTG-STORAGE-14
La aplicación utiliza la base de datos SQLite y ejecuta una consulta SQL sin procesar. La entrada de un usuario que no es de confianza en consultas SQL sin procesar puede provocar la inyección de SQL. Además, la información confidencial debe cifrarse y escribirse en la base de datos.	Alta	CVSS V2: 5.9 (medium) CWE: CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') OWASP Top 10: M7: Client Code Quality
La aplicación crea un archivo temporal. La información confidencial nunca debe escribirse en un archivo temporal.	Alta	CVSS V2: 5.5 (medium) CWE: CWE-276 Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2

Tabla 24**.** Top 5 problemas del código de WhatsApp.
Fuente: propia con base en el escaneo de MobSF.

4.4.1.2 Instagram

Se realizó en análisis estático de la aplicación Instagram a través de MobSF con el fin de identificar las vulnerabilidades y riesgos que posee la aplicación. La versión de Instagram que se analizó fue 153.0.0.34.96, publicada el 4 de agosto de 2020, la cual tiene como requisitos mínimos:

- Sistema operativo Android 4.1 o superior
- Se recomienda un plan ilimitado de datos de Internet

Los resultados del análisis revelaron que en la aplicación no se ejecutan 87 actividades de las cuales 13 se exportan; 68 servicios, 12; 34 receptores, 15 y 11 proveedores de contenidos, 8.

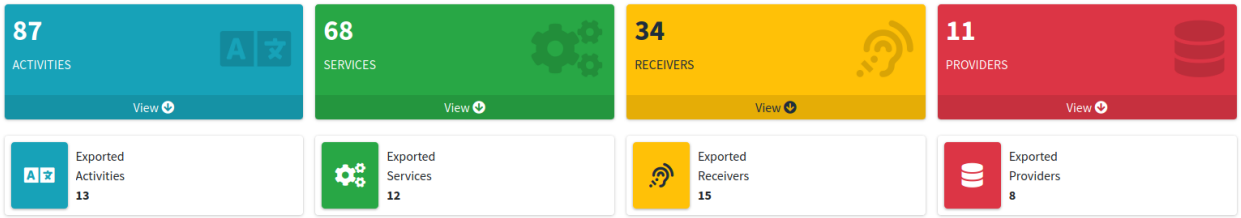


Figura 30. Posibles elementos vulnerables de Instagram.

Fuente: propia con base en el escaneo de MobSF.

Puntuación de la aplicación

La aplicación recibe una aplicación de CVSS de 5.8 y de seguridad de 5/100. Se detecta un problema con severidad media debido a que la aplicación utiliza un certificado que firma con SHA1 de RSA (se sabe que el algoritmo hash SHA1 tiene problemas de colisión y está obsoleto), pero en el archivo de manifiesto se indica que SHA256 de RSA está en uso. El reporte completo con los hallazgos del análisis realizado se adjunta en la sección de Apéndice 1.

Permisos

Se presentan en esta sección, los diez permisos más relevantes de los 56 encontrados a través del escaneo, de acuerdo con la severidad asignada por MobSF.

Permiso	Severidad	Info	Descripción
android.permission.ACCESS_FINE_LOCATION	Peligroso	(GPS) Ubicación	Acceda a fuentes de ubicación precisas, como el sistema de posicionamiento global en el teléfono, donde esté disponible. Las aplicaciones maliciosas pueden usar esto para determinar dónde se encuentra y pueden consumir energía adicional de la batería.
android.permission.ACCESS_MEDIA_LOCATION	Peligroso	acceder a cualquier ubicación geográfica	Permite que una aplicación acceda a cualquier ubicación geográfica que se conserve en la colección

			compartida del usuario.
android.permission.CAMERA	Peligroso	tomar fotos y videos	Permite que la aplicación tome fotos y videos con la cámara. Esto facilita que la aplicación recopile imágenes que la cámara está viendo en cualquier momento.
android.permission.INTERNET	Peligroso	acceso completo a Internet	Permite que una aplicación cree conexiones de red.
android.permission.READ_CONTACTS	Peligroso	leer datos de contacto	Permite que una aplicación lea todos los datos de contacto (direcciones) almacenados en tu teléfono. Las aplicaciones maliciosas pueden usar esto para enviar sus datos a otras personas.
android.permission.READ_PHONE_STATE	Peligroso	leer el estado y la identidad del teléfono	Permite que la aplicación acceda a las funciones telefónicas del dispositivo. Una aplicación con este permiso puede determinar el número de teléfono y el número de serie de este teléfono, si una llamada está activa, el número al que está conectada la llamada, etc.
android.permission.READ_PROFILE	Peligroso	leer los datos del perfil personal del usuario	Permite que una aplicación lea los datos del perfil personal del usuario.
android.permission.RECORD_AUDIO	Peligroso	grabar audio	Permite que la aplicación acceda a la ruta de grabación de audio.
android.permission.USE_CREDENTIALS	Peligroso	utilizar las credenciales de autenticación de una cuenta	Permite que una aplicación solicite tokens de autenticación.
android.permission.WRITE_EXTERNAL_STORAGE	Peligroso	leer / modificar / eliminar el	Permite que una aplicación escriba en la tarjeta SD.

		contenido de la tarjeta SD	

Tabla 25. Top 10 permisos más peligrosos de Instagram.
Fuente: propia con base en el escaneo de MobSF.

Análisis del archivo de manifiesto

Se presentan a continuación, las diez vulnerabilidades más relevantes de las 62 encontradas a través del escaneo, de acuerdo con la severidad asignada por MobSF.

Vulnerabilidad	Severidad	Descripción
Activity (com.instagram.mainactivity.LauncherActivity) is not Protected. [android:exported=true]	Alta	Se descubre que una actividad se comparte con otras aplicaciones en el dispositivo, por lo que se deja accesible para cualquier otra aplicación en el dispositivo.
Activity (com.instagram.mainactivity.MainActivity) is not Protected. [android:exported=true]	Alta	Se descubre que una actividad se comparte con otras aplicaciones en el dispositivo, por lo que se deja accesible para cualquier otra aplicación en el dispositivo.
Activity (com.instagram.url.UrlHandlerActivity) is not Protected. [android:exported=true]	Alta	Se descubre que una actividad se comparte con otras aplicaciones en el dispositivo, por lo que se deja accesible para cualquier otra aplicación en el dispositivo.
Service (com.instagram.debug.memorydump.MemoryDumpTaskService) is Protected by a permission, but the protection level of the permission should be checked. Permission: com.google.android.gms.permission.BIND_NETWORK_TASK_SERVICE [android:exported=true]	Alta	Se encuentra que un servicio se comparte con otras aplicaciones en el dispositivo, por lo tanto, lo deja accesible para cualquier otra aplicación en el dispositivo. Está protegido por un permiso que no está definido en la aplicación analizada. Si está configurado como firma, solo las aplicaciones firmadas con el mismo certificado pueden obtener el permiso.
Service (com.facebook.rti.push.service.FbnsSer	Alta	Se encuentra que un servicio se comparte con otras aplicaciones en el dispositivo, por lo

vice) is not Protected. [android:exported=true]		tanto, lo deja accesible para cualquier otra aplicación en el dispositivo.
Broadcast Receiver (com.facebook.rti.push.service.idsharing.FbnsSharingStateReceiver) is not Protected. [android:exported=true]	Alta	Se encuentra que un receptor de transmisión se comparte con otras aplicaciones en el dispositivo, por lo tanto, lo deja accesible para cualquier otra aplicación en el dispositivo.
Content Provider (com.instagram.contentprovider.users.impl.IgLoggedInUsersContentProvider) is not Protected. [android:exported=true]	Alta	Se descubre que un proveedor de contenido se comparte con otras aplicaciones en el dispositivo, por lo tanto, lo deja accesible para cualquier otra aplicación en el dispositivo.
Content Provider (com.instagram.contentprovider.latencytest.LatencyTestContentProvider) is not Protected. [android:exported=true]	Alta	Se descubre que un proveedor de contenido se comparte con otras aplicaciones en el dispositivo, por lo tanto, lo deja accesible para cualquier otra aplicación en el dispositivo.
Activity (androidx.biometric.DeviceCredentialHandlerActivity) is not Protected. [android:exported=true]	Alta	Se descubre que una actividad se comparte con otras aplicaciones en el dispositivo, por lo que se deja accesible para cualquier otra aplicación en el dispositivo.
Service (com.instagram.notifications.push.fcm.GetFCMTokenAndRegisterWithServerGCMService) is Protected by a permission, but the protection level of the permission should be checked. Permission: com.google.android.gms.permission.BIND_NETWORK_TASK_SERVICE [android:exported=true]	Alta	Se encuentra que un servicio se comparte con otras aplicaciones en el dispositivo, por lo tanto, lo deja accesible para cualquier otra aplicación en el dispositivo. Si está configurado como firma, solo las aplicaciones firmadas con el mismo certificado pueden obtener el permiso.

Tabla 26. Top 10 vulnerabilidades con severidad alta de Instagram.
Fuente: propia con base en el escaneo de MobSF.

Análisis de código

Se presentan a continuación, los cinco problemas de código más relevantes de los 11 encontrados a través del escaneo, de acuerdo con la severidad asignada por MobSF.

Problema	Severidad	Estándares
La aplicación utiliza un generador inseguro de números aleatorios.	Alta	CVSS V2: 7.5 (high) CWE: CWE-330 Use of Insufficiently Random Values OWASP Top 10: M5: Insufficient Cryptography OWASP MASVS: MSTG-CRYPTO-6
MD5 es un hash débil conocido por tener colisiones de hash.	Alta	CVSS V2: 7.4 (high) CWE: CWE-327 Use of a Broken or Risky Cryptographic Algorithm OWASP Top 10: M5: Insufficient Cryptography OWASP MASVS: MSTG-CRYPTO-4
SHA-1 es un hash débil que se sabe que tiene colisiones de hash.	Alta	CVSS V2: 5.9 (medium) CWE: CWE-327 Use of a Broken or Risky Cryptographic Algorithm OWASP Top 10: M5: Insufficient Cryptography OWASP MASVS: MSTG-CRYPTO-4
La aplicación utiliza la base de datos SQLite y ejecuta una consulta SQL sin procesar. La entrada de un usuario que no es de confianza en consultas SQL sin procesar puede provocar la inyección de SQL. Además, la información confidencial debe cifrarse y escribirse en la base de datos.	Alta	CVSS V2: 5.9 (medium) CWE: CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') OWASP Top 10: M7: Client Code Quality
La aplicación crea un archivo temporal. La información confidencial nunca debe escribirse en un archivo temporal.	Alta	CVSS V2: 5.5 (medium) CWE: CWE-276 Incorrect Default Permissions OWASP Top 10: M2: Insecure Data Storage OWASP MASVS: MSTG-STORAGE-2

Tabla 27**.** Top 5 problemas del código de Instagram.

Fuente: propia con base en el escaneo de MobSF

Se realizó un análisis estático de la aplicación WhatsApp a través de MobSF con el fin de identificar las vulnerabilidades y riesgos que posee la aplicación. La versión de WhatsApp que se analizó fue 2.20.196.17, publicada el 4 de agosto de 2020, la cual tiene como requisitos mínimos;

- Sistema operativo Android 4.0.3 o superior
- Se recomienda un plan ilimitado de datos de Internet
- Las tablets no son compatibles

Los resultados del análisis revelaron que en la aplicación se ejecutan 222 actividades de las cuales 21 se exportan; 40 servicios, 7; 29 receptores, 8 y 7 proveedores de contenidos, 2.

4.1.2 Fortify on Demand Scan

4.4.2.1 WhatsApp

Se realizó el análisis estático de la aplicación WhatsApp a través de Fortify on Demand con el fin de identificar las vulnerabilidades y riesgos que posee la aplicación. Se utilizó la versión de prueba, ya que Fortify on Demand es un programa con licencia comercial. La versión de WhatsApp que se analizó fue 2.20.196.17, publicada el 4 de agosto de 2020, la cual tiene como requisitos mínimos:

- Sistema operativo Android 4.0.3 o superior
- Se recomienda un plan ilimitado de datos de Internet
- Las tablets no son compatibles

Los resultados del análisis revelaron que en la aplicación posee dos vulnerabilidades de con criticidad media y 12 de baja.

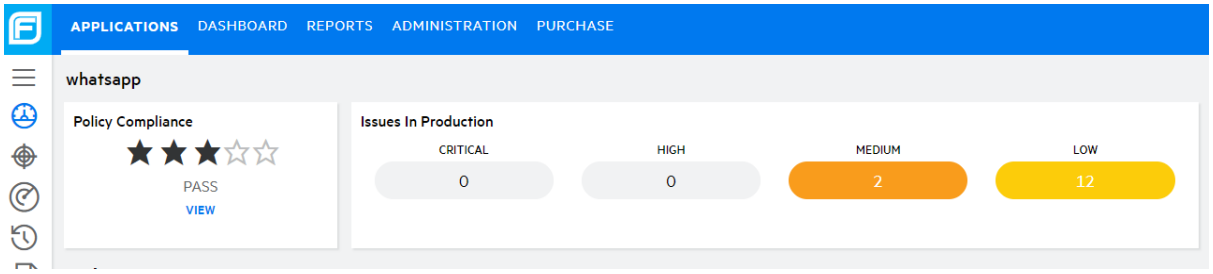


Figura 31. Vulnerabilidades de WhatsApp.

Fuente: propia con base en el escaneo de Fortify on Demand.

A continuación, se explicarán los problemas o vulnerabilidades encontrados en la aplicación de criticidad media:

Fortify on Demand encontró dos vulnerabilidades de nivel medio.

- Mala práctica de Android: falta el proveedor de seguridad actualizado de Google Play Services
- Cifrado débil: certificado débil de firma de código

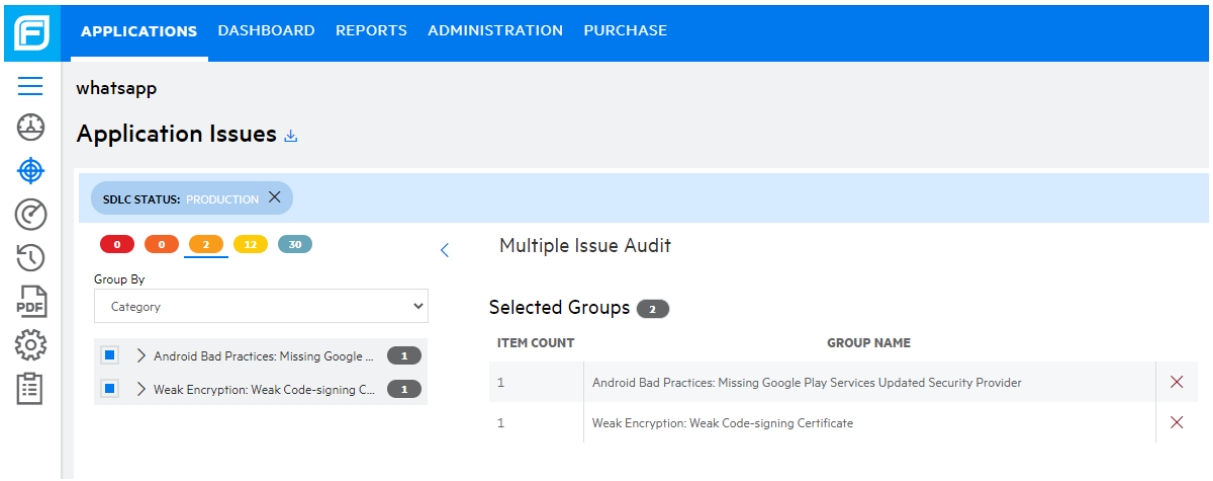


Figura 32. Vulnerabilidades de nivel Medio de WhatsApp.
Fuente: propia con base en el escaneo de Fortify on Demand.

El resumen y la explicación de la vulnerabilidad de “Mala práctica de Android” y según Fortify es el siguiente:

- **Resumen**

La aplicación no utiliza los servicios de Google Play para actualizar el proveedor de seguridad.

- **Explicación**

Android confía en el proveedor de seguridad para proporcionar comunicaciones de red seguras. Las bibliotecas criptográficas de dispositivos predeterminadas suelen ser versiones anteriores de OpenSSL que contienen fallas conocidas. Para superar esto, Google proporciona un mecanismo para que una aplicación parche su copia local de OpenSSL a través del cliente ProviderInstaller de Google Play Services. Se ha determinado que la aplicación no utiliza el proveedor actualizado, lo que deja a la aplicación expuesta a vulnerabilidades y debilidades conocidas más antiguas de OpenSSL.

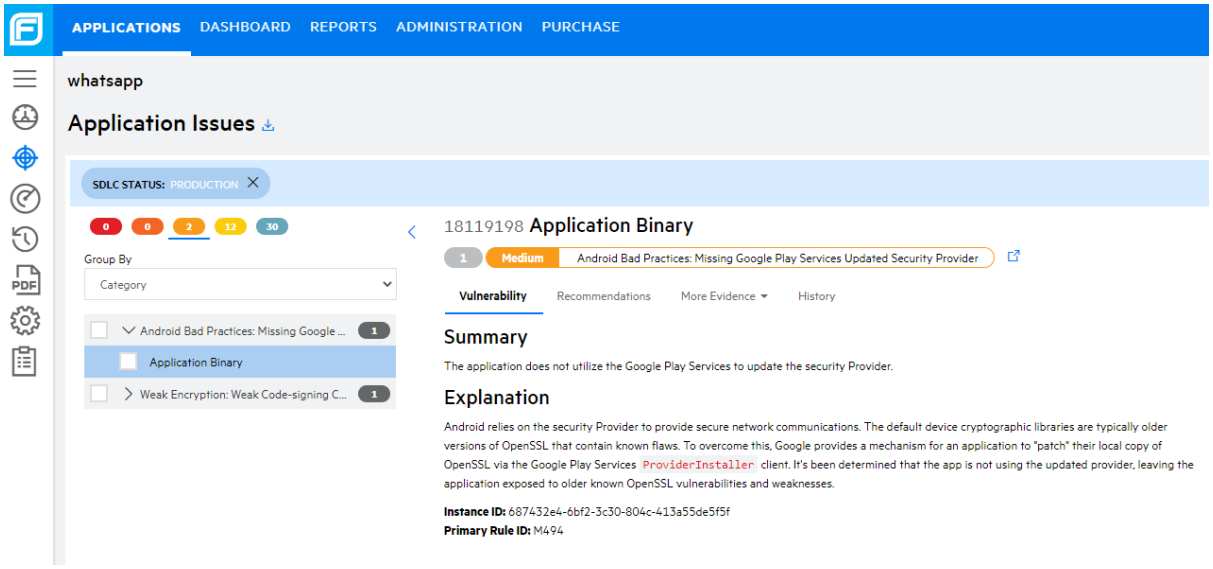


Figura 33. Mala práctica de Android vulnerabilidad.

Fuente: propia con base en el escaneo de Fortify on Demand.

El resumen y la explicación de la vulnerabilidad de “Cifrado débil” y según Fortify es el siguiente.

- **Resumen**

La aplicación está firmada con un certificado débil de firma de código .

- **Explicación**

El certificado utilizado para la firma de código de la aplicación tiene un tamaño insuficiente de clave o un algoritmo hash débil. La criptografía débil aumenta la probabilidad de que un atacante se haga pasar por el certificado original de firma de código. En el sistema operativo Android, podría llevar a que un atacante tenga una aplicación suplantada instalada sobre una aplicación legítima existente. La aplicación maliciosa podría acceder a información confidencial escrita por la aplicación legítima o engañar a los usuarios creyendo que es una versión legítima. Asegúrese de que cualquier versión de lanzamiento de la aplicación tenga un certificado suficientemente sólido de firma de código. El certificado de firma presenta los siguientes problemas:

- La longitud de la clave pública (1024 bits) es inferior a 2048 bits.
- El algoritmo de firma (dsaWithSHA1) utiliza un método hash inseguro.

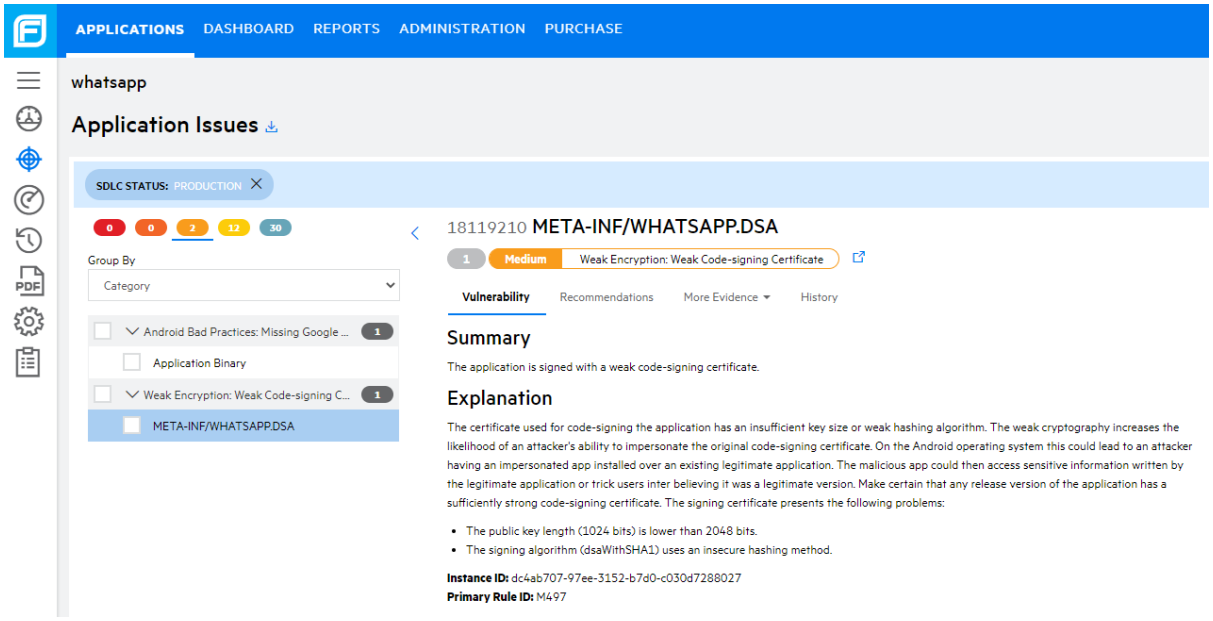


Figura 34. Cifrado débil vulnerabilidad.

Fuente: propia con base en el escaneo de Fortify on Demand.

A continuación, se mencionan las 12 vulnerabilidades de criticidad baja según Fortify on Demand.

- Transporte inseguro – 1 vulnerabilidad
- Violación de privacidad: la aplicación utiliza identificadores de dispositivos rastreables - 6 vulnerabilidades
- Inyección SQL: inyección SQL (lado del cliente) - 3 vulnerabilidades
- Hash criptográfico débil: criptografía débil - 2 vulnerabilidades

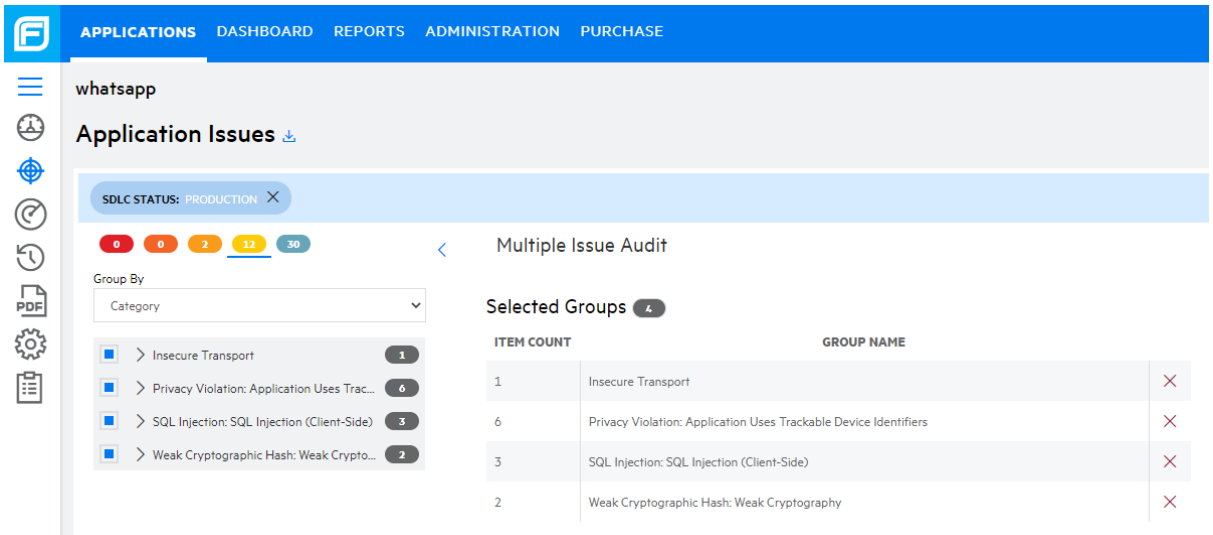


Figura 35 .Vulnerabilidades bajas de WhatsApp.

Fuente: propia con base en el escaneo de Fortify on Demand.

Debido a que se usó una versión de prueba, Fortify on Demand no reproduce reporte con los hallazgos del análisis realizado.

4.4.2.2 Instagram

Se realizó el análisis estático de la aplicación Instagram a través de Fortify on Demand con el fin de identificar las vulnerabilidades y riesgos que posee la aplicación. La versión de Instagram que se analizó fue 153.0.0.34.96, publicada el 4 de agosto de 2020, la cual tiene como requisitos mínimos

- Sistema operativo Android 4.1 o superior
- Se recomienda un plan de datos ilimitado de Internet

Los resultados del análisis revelaron que en la aplicación posee dos vulnerabilidades de con criticidad media y diez de baja.

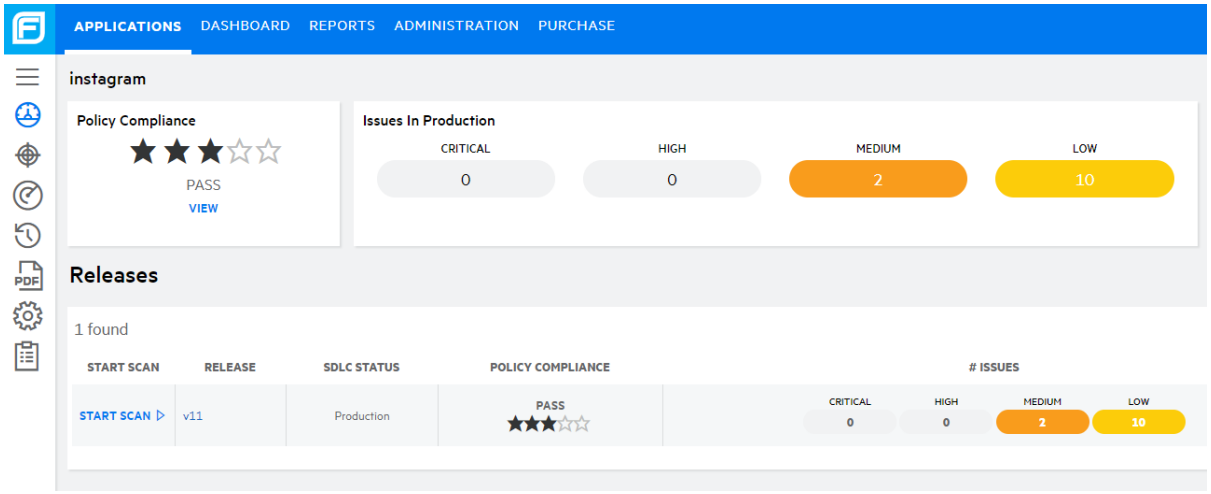


Figura 36. Vulnerabilidades de Instagram.

Fuente: propia con base en el escaneo de Fortify on Demand.

A continuación, se explicarán los problemas o vulnerabilidades encontrados en la aplicación de criticidad media:

Fortify on Demand encontró dos vulnerabilidades de nivel medio.

- Mala práctica de Android: falta el proveedor de seguridad actualizado de Google Play Services
- Cifrado débil: certificado débil de firma de código

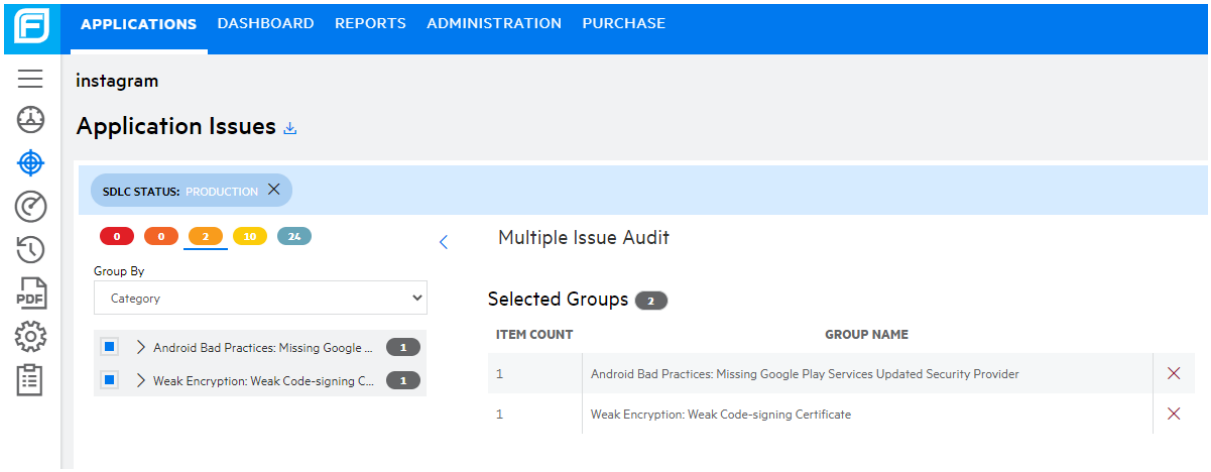


Figura 37. Vulnerabilidades de nivel Medio de Instagram.

Fuente: propia con base en el escaneo de Fortify on Demand.

El resumen y la explicación de la vulnerabilidad de “Mala práctica de Android” y según Fortify es el siguiente.

- **Resumen**

La aplicación no utiliza los servicios de Google Play para actualizar el proveedor de seguridad.

- **Explicación**

Android confía en el proveedor de seguridad para proporcionar comunicaciones de red seguras. Las bibliotecas criptográficas de dispositivos predeterminadas suelen ser versiones anteriores de OpenSSL que contienen fallas conocidas. Para superar esto, Google proporciona un mecanismo para que una aplicación parche su copia local de OpenSSL a través del cliente ProviderInstaller de Google Play Services. Se ha determinado que la aplicación no utiliza el proveedor actualizado, lo que deja a la aplicación expuesta a vulnerabilidades y debilidades conocidas más antiguas de OpenSSL.

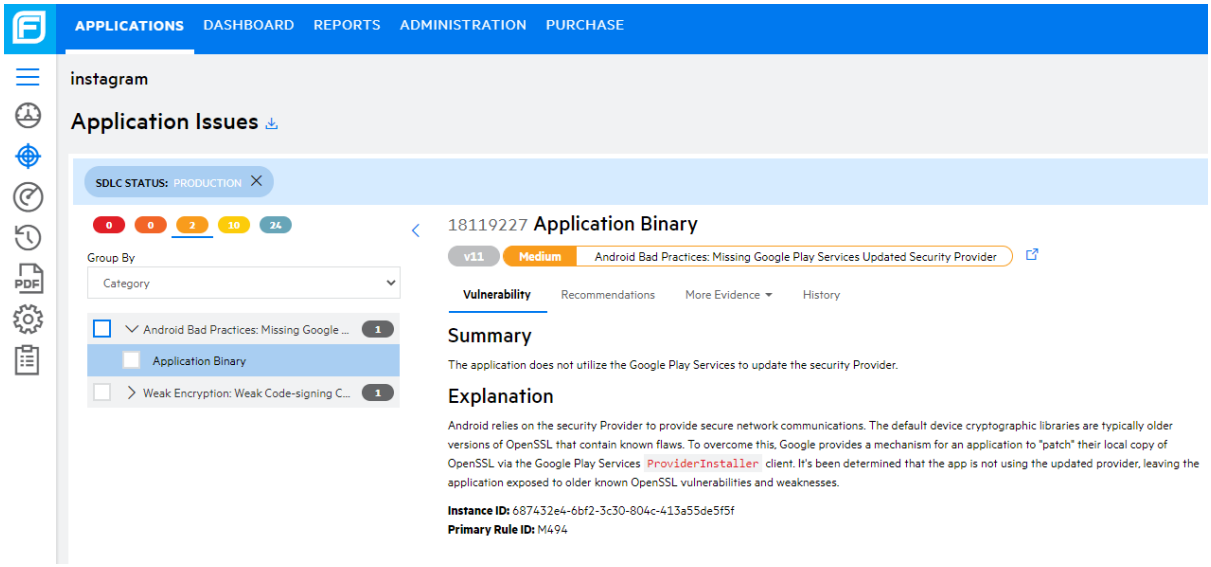


Figura 38. Mala práctica de Android vulnerabilidad.

Fuente: propia con base en el escaneo de Fortify on Demand.

El resumen y la explicación de la vulnerabilidad de “Cifrado débil” y según Fortify es el siguiente.

- **Resumen**

La aplicación está firmada con un certificado débil de firma de código.

- **Explicación**

El certificado utilizado para la firma de código de la aplicación tiene un tamaño insuficiente de clave o un algoritmo hash débil. La criptografía débil aumenta la probabilidad de que un atacante se haga pasar por el certificado original de firma de código. En el sistema operativo Android, podría llevar a que un atacante tenga una aplicación suplantada instalada sobre una aplicación legítima existente. La aplicación maliciosa podría acceder a información confidencial escrita por la aplicación legítima o engañar a los usuarios, creyendo que es una versión legítima. Asegúrese de que cualquier versión de lanzamiento de la aplicación tenga un certificado suficientemente sólido de firma de código. El certificado de firma presenta los siguientes problemas:

- La longitud de la clave pública (1024 bits) es inferior a 2048 bits.
- El algoritmo de firma (dsaWithSHA1) utiliza un método hash inseguro.

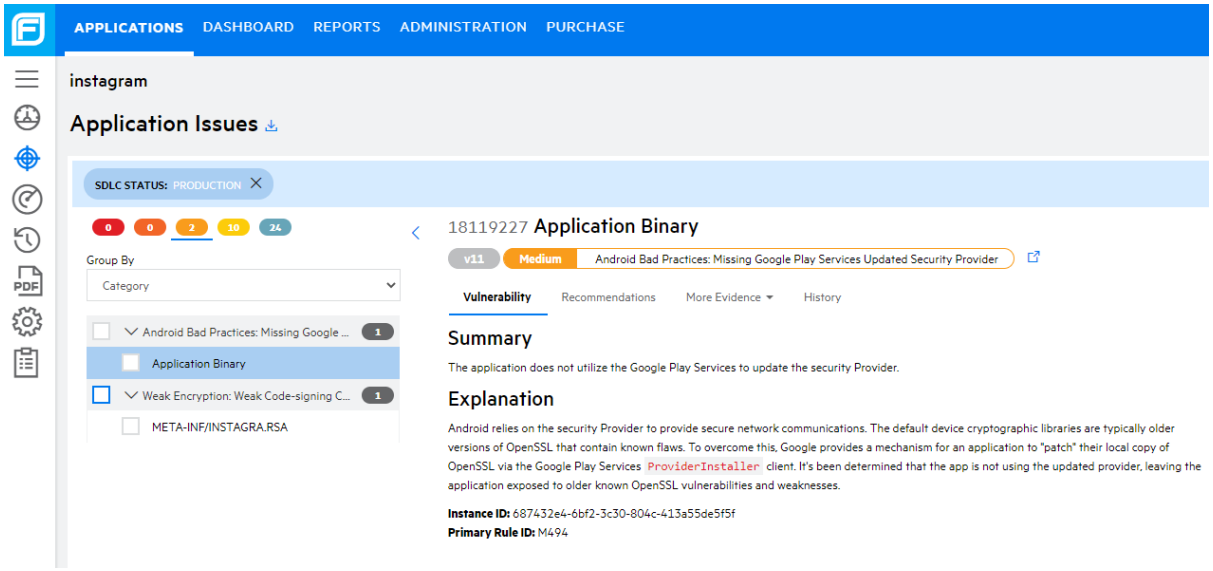


Figura 39. Cifrado débil vulnerabilidad.

Fuente: propia con base en el escaneo de Fortify on Demand.

A continuación, se mencionan las diez vulnerabilidades de criticidad baja de Instagram según Fortify on Demand.

- Transporte inseguro – 1 vulnerabilidad
- Violación de privacidad: la aplicación utiliza identificadores de dispositivos rastreables - 7 vulnerabilidades
- Hash criptográfico débil: criptografía débil - 2 vulnerabilidades

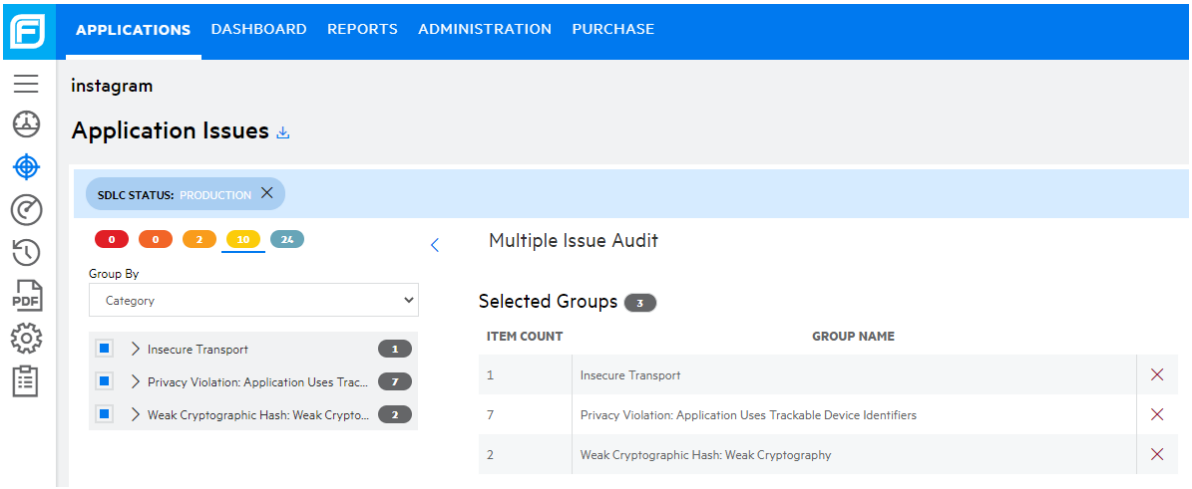


Figura 40. Vulnerabilidades bajas de WhatsApp.

Fuente: propia con base en el escaneo de Fortify on Demand.

Debido a que se usó una versión de prueba, Fortify on Demand no reproduce reporte con los hallazgos del análisis realizado.

4.5 Resultados del análisis dinámico

Se realizo un análisis de peticiones y respuestas HTTP de las aplicaciones WhatsApp e Instagram, utilizando la herramienta OWAS ZAP (Zed Attack Proxy) versión 2.9.0.

4.5.1 OWAS ZAP

4.5.1.1 WhatsApp

La aplicación utiliza conexiones seguras HTTPS en todas sus comunicaciones, por lo que debimos instalar en el dispositivo móvil Android, el certificado de confianza generado por la aplicación ZAP para poder ver las conexiones protegidas.

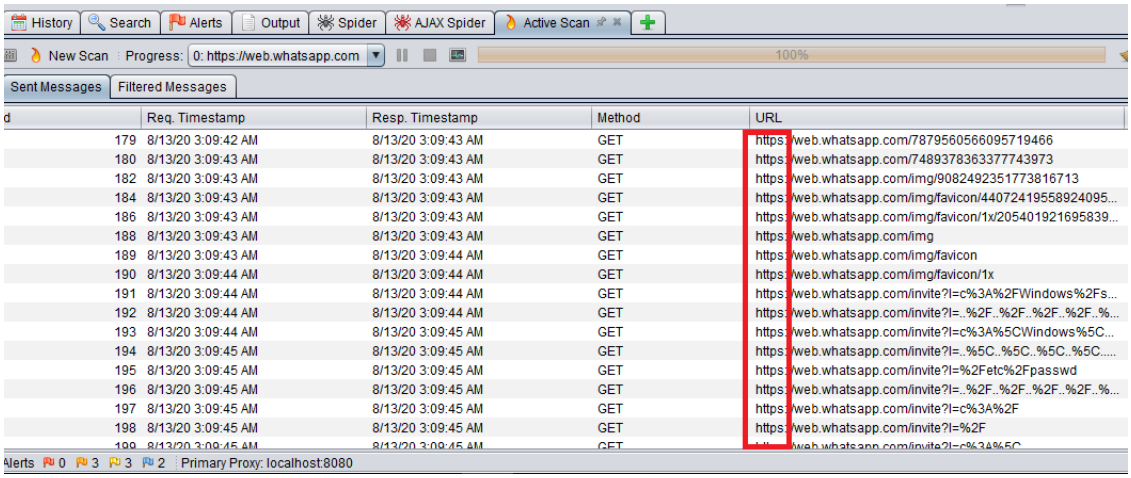


Figura 41. Trafico HTTPS generado en WhatsApp.

Fuente: propia con base en el escaneo de OWASP ZAP.

En la pestaña de *Alerts* se exponen las vulnerabilidades que se encuentran a medida que se ejecuta el análisis, en este caso se muestran siete, desplegadas por tipo de alerta y nivel de criticidad.

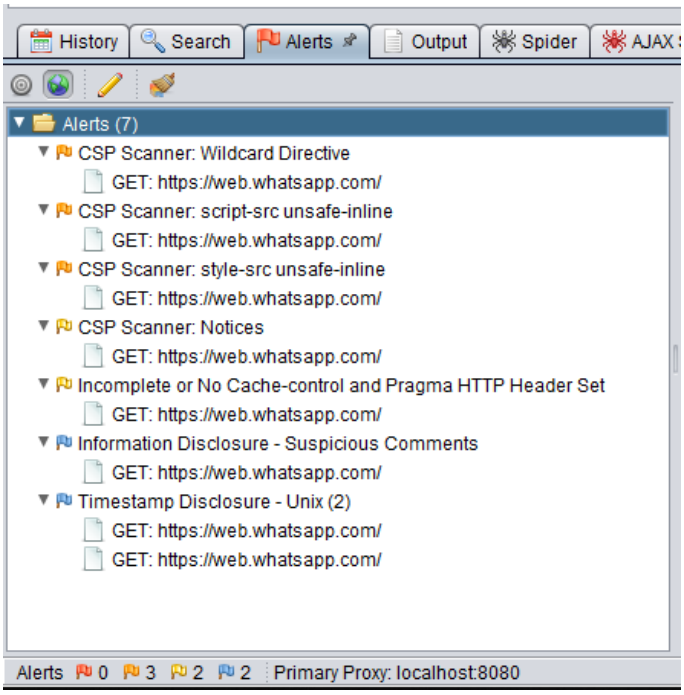


Figura 42. Vulnerabilidades generadas en WhatsApp.

Fuente: propia con base en el escaneo de OWASP ZAP.

Tres de ellas están clasificadas con riesgo medio.

- **CSP Scanner: Wildcard Directive**

Las siguientes directivas permiten fuentes comodinas (o ancestros), no están definidas o están definidas de manera demasiado amplia: img-src, ancestros de marco.

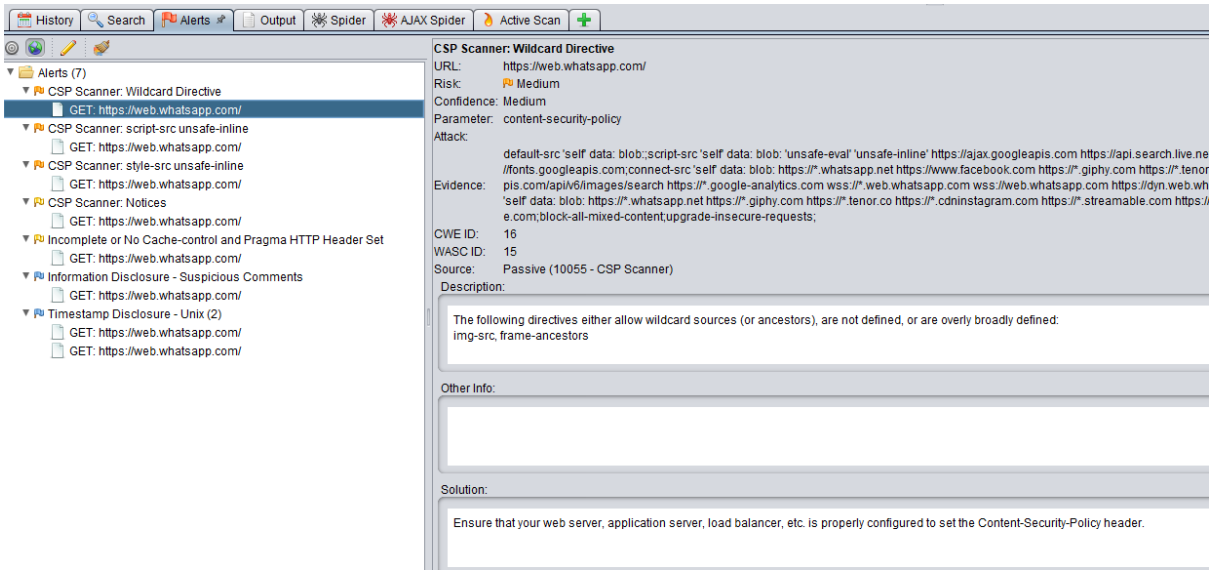


Figura 43. Alerta de Wildcard Directive.

Fuente: propia con base en el escaneo de OWASP ZAP.

- **CSP Scanner: script-src unsafe-inline**

Script-src incluye insegura en línea.

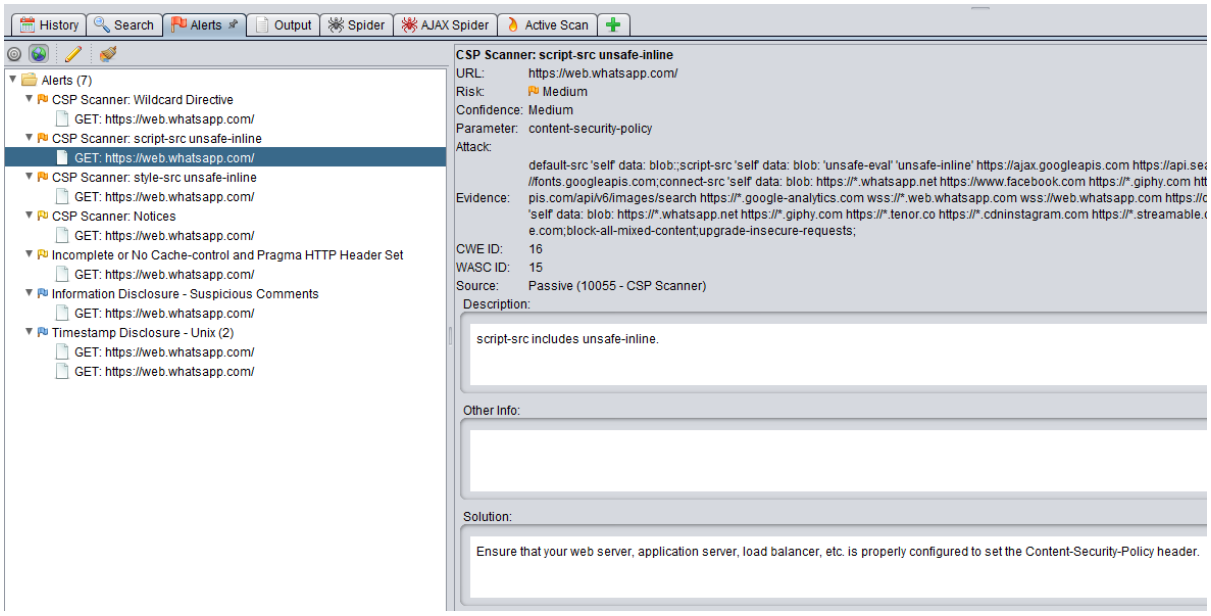


Figura 44. Alerta de script-src unsafe-inline.

Fuente: propia con base en el escaneo de OWASP ZAP.

- **CSP Scanner: style-src unsafe-inline**

Style-src incluye insegura en línea.

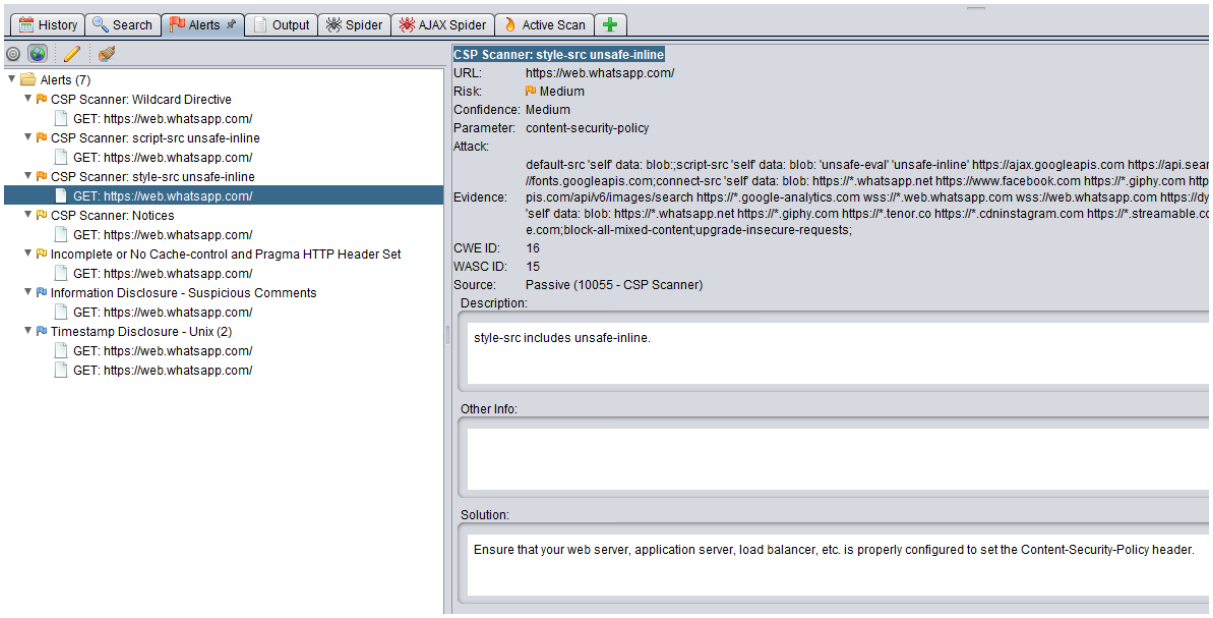


Figura 45. Alerta de style-src unsafe-inline.

Fuente: propia con base en el escaneo de OWASP ZAP.

Y dos de ellas clasificadas con riesgo bajo.

- **CSP Scanner: Notices**

Advertencias:

1:966: La directiva child-src está obsoleta a partir del nivel 3 de CSP. Los autores que deseen regular los contextos de navegación anidados y los trabajadores **deberían** usar las directivas frame-src y worker-src, respectivamente.

1:1048: La directiva block-all-mixed-content es una directiva experimental que probablemente se agregará a la especificación CSP.

1:1072: La directiva de solicitudes de actualización inseguras es una directiva experimental que probablemente se agregará a la especificación de CSP.

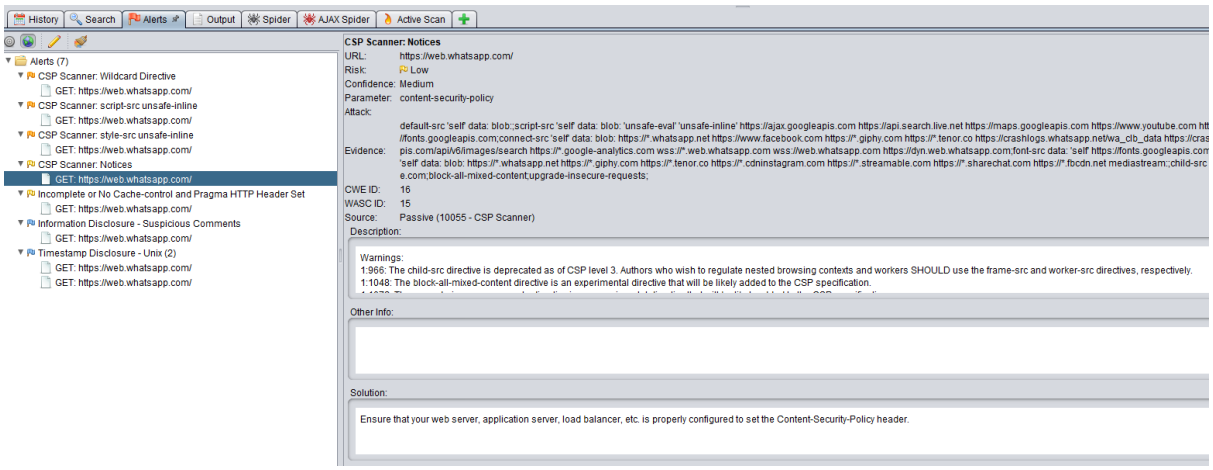


Figura 46. Alerta de Notices.

Fuente: propia con base en el escaneo de OWASP ZAP.

- **Incomplete or No Cache-control and Pragma HTTP Header Set**

El control de caché y el encabezado HTTP pragma no se han configurado correctamente o faltan, lo que permite que el navegador y los proxies almacenen contenido en caché.

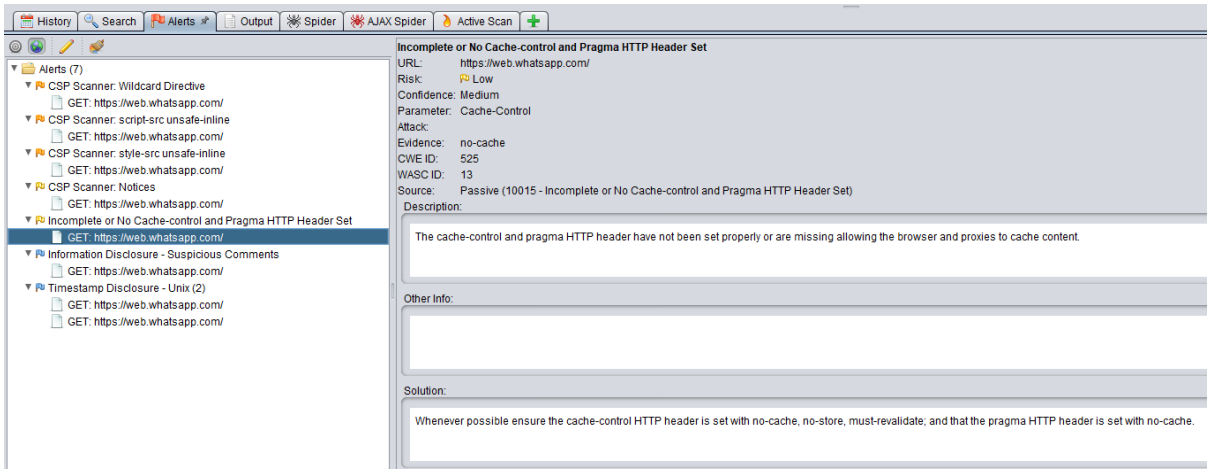


Figura 47. Alerta de Incomplete or No Cache-control and Pragma HTTP Header Set.
Fuente: propia con base en el escaneo de OWASP ZAP.

Para cada vulnerabilidad se detalla: URL afectada, tipo de riesgo, nivel de confidencialidad, parámetro y ataque realizado, CWE ID, WESC ID y descripción.

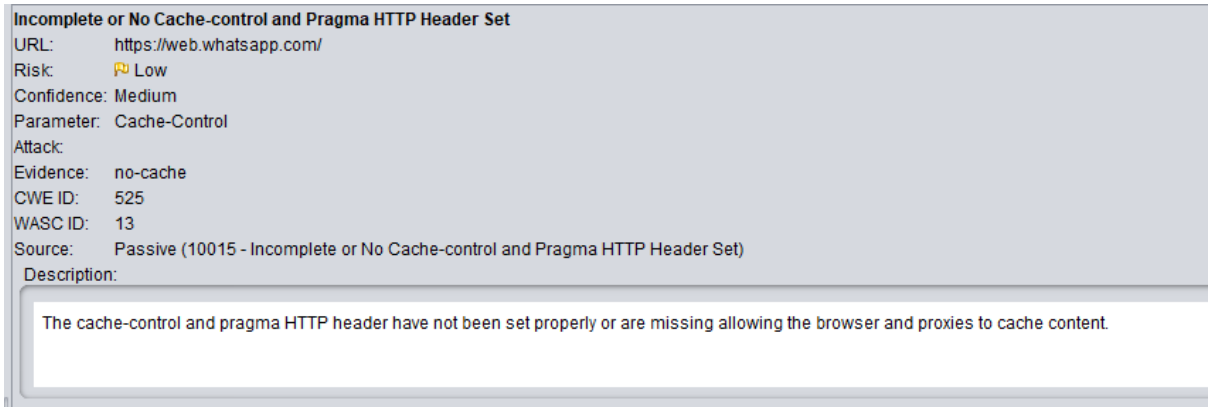


Figura 48. Detalles de vulnerabilidad.
Fuente: propia con base en el escaneo de OWASP ZAP.

En la pestaña *Spider* o araña traducido al español, se obtiene los recursos (URL) que la aplicación está utilizando.

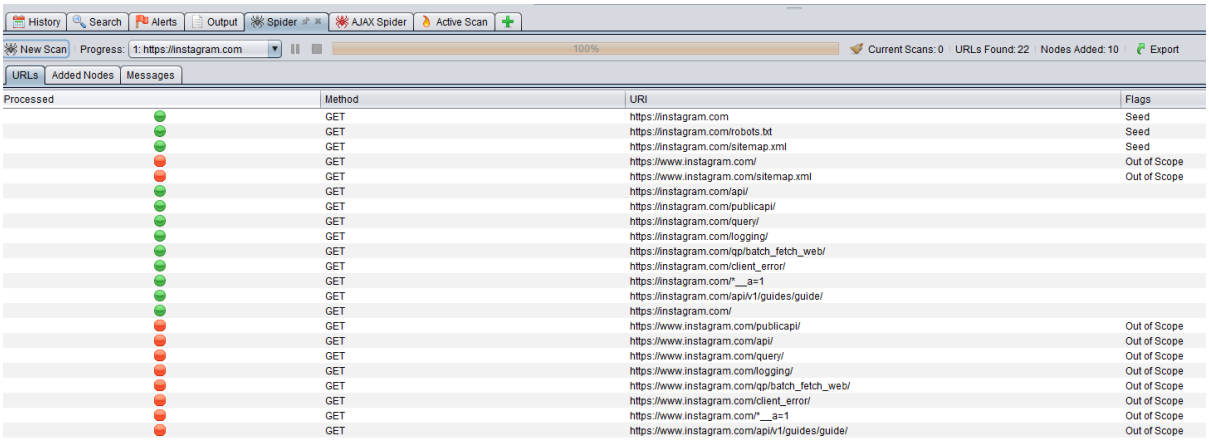


Figura 49. Detalles de la pestaña Spider de WhatsApp.

Fuente: propia con base en el escaneo de OWASP ZAP.

Para ver un mayor detalle de las vulnerabilidades, se puede generar un reporte HTML, XML, MarkDown o Json. El reporte completo con los hallazgos del análisis realizado se adjunta en la sección de Apéndice 1.

4.5.1.2 Instagram

La aplicación también utiliza conexiones seguras HTTPS en todas sus comunicaciones y se hizo uso del certificado de confianza, generado por la aplicación ZAP para poder ver las conexiones protegidas.

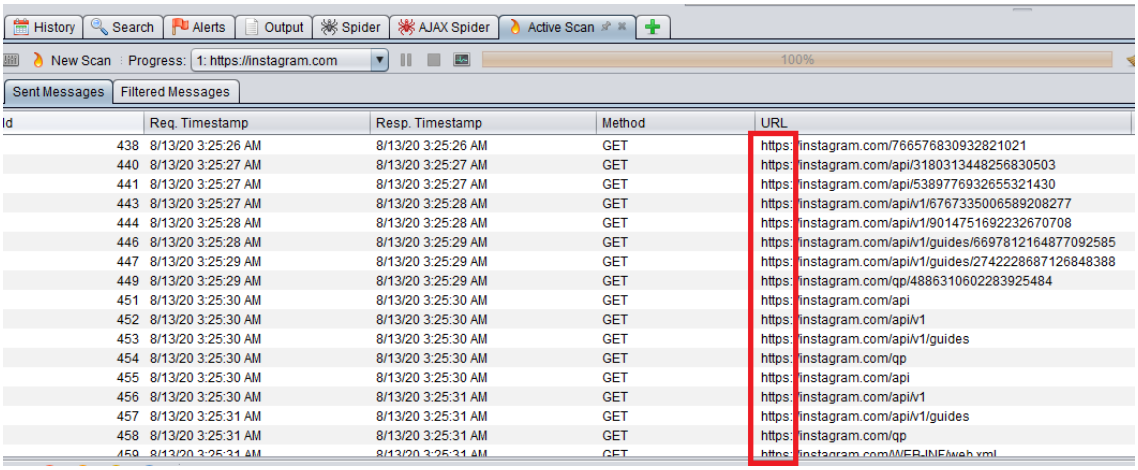


Figura 50. Trafico HTTPS generado en Instagram.

Fuente: propia con base en el escaneo de OWASP ZAP.

En la pestaña de *Alerts* se exponen las vulnerabilidades que se encuentran a medida que se ejecuta el análisis, en este caso se muestran seis, desplegadas por tipo de alerta y nivel de criticidad.

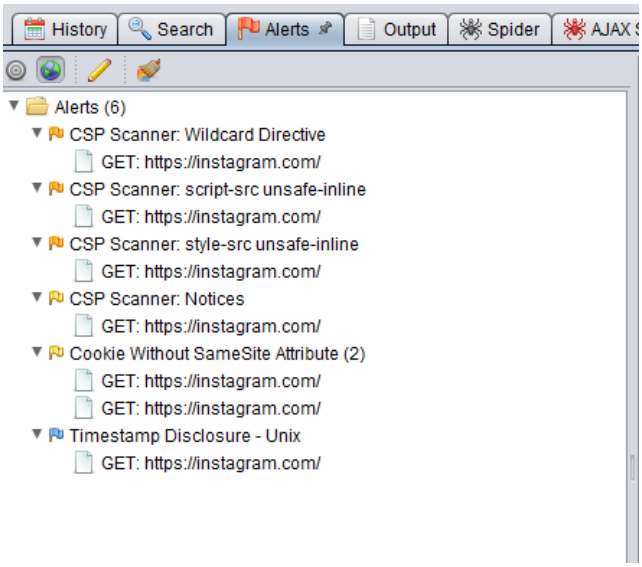


Figura 51. Vulnerabilidades generadas en Instagram.

Fuente: propia con base en el escaneo de OWASP ZAP.

Tres de ellas están clasificadas con riesgo medio.

- **CSP Scanner: Wildcard Directive**

Las siguientes directivas permiten fuentes comodín (o ancestros), no están definidas o están definidas de manera demasiado amplia: img-src, ancestros de marco.

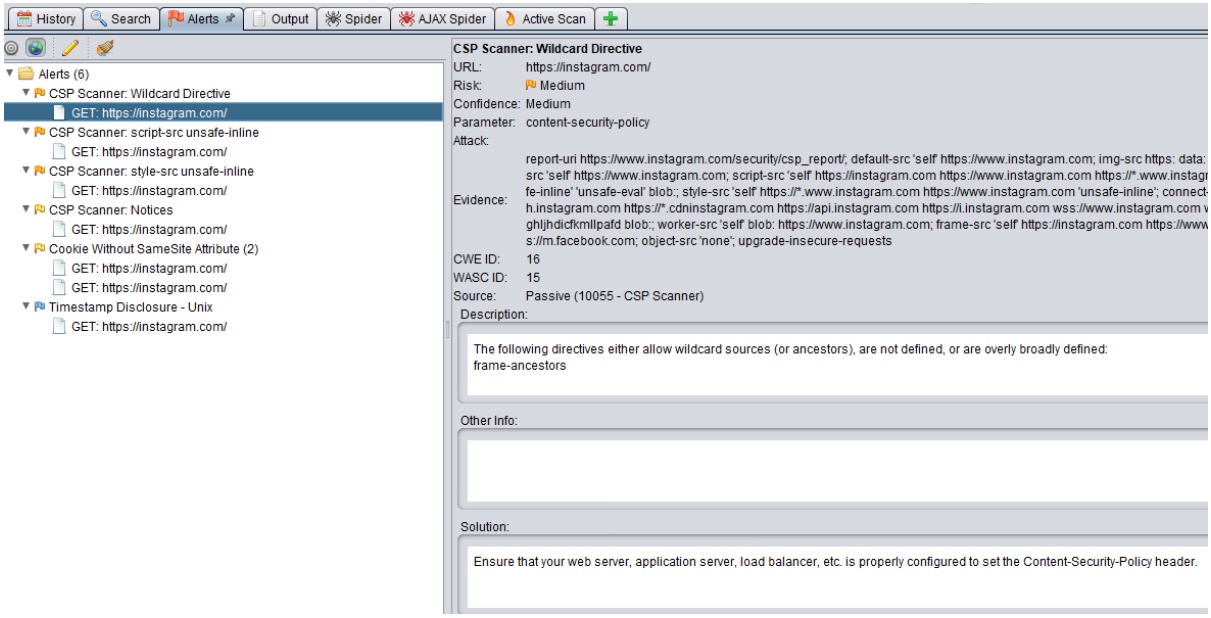


Figura 52. Alerta de Wildcard Directive.

Fuente: propia con base en el escaneo de OWASP ZAP.

- **CSP Scanner: script-src unsafe-inline**

Script-src incluye insegura en línea.

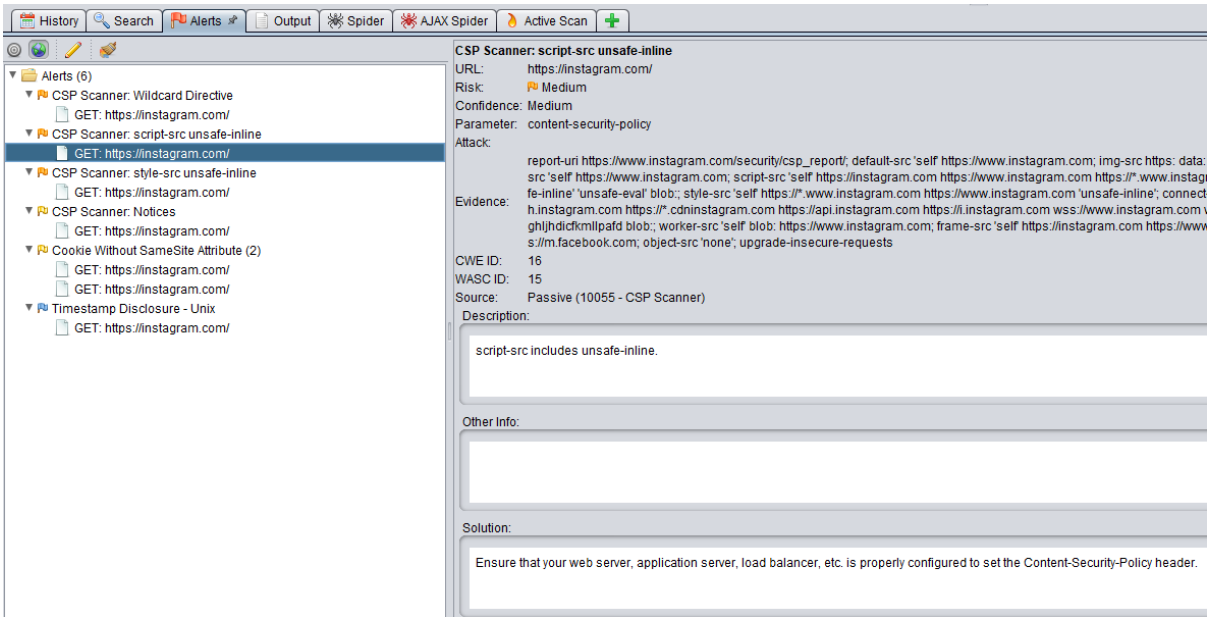


Figura 53. Alerta de script-src unsafe-inline.

Fuente: propia con base en el escaneo de OWASP ZAP.

- **CSP Scanner: style-src unsafe-inline**

Style-src incluye insegura en línea.

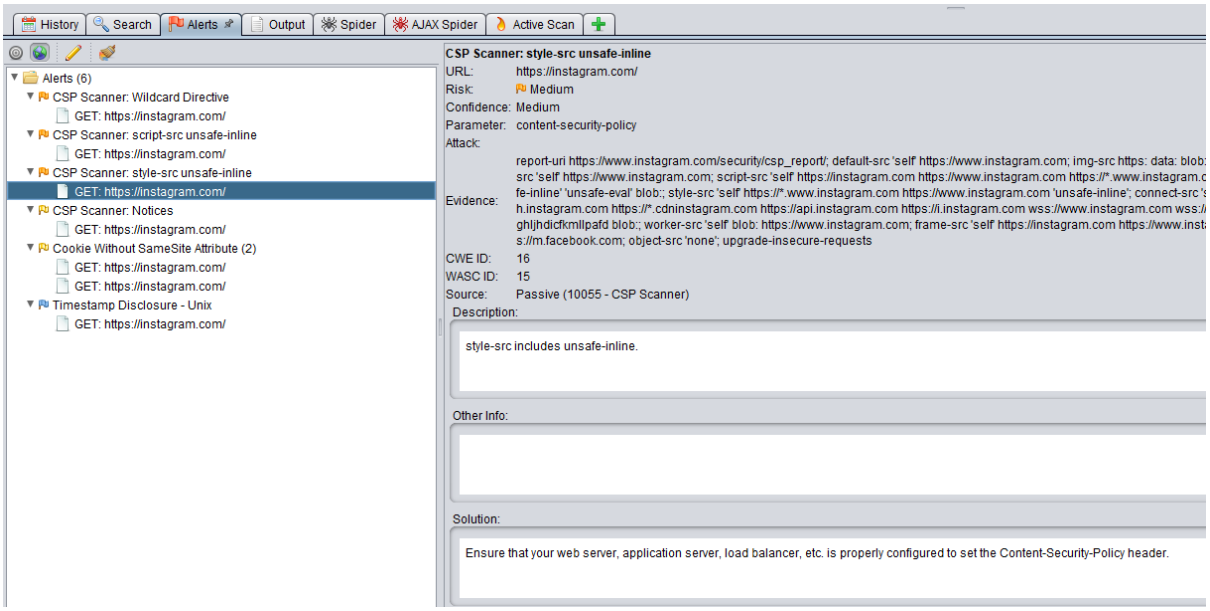


Figura 54. Alerta de style-src unsafe-inline.

Fuente: propia con base en el escaneo de OWASP ZAP.

Y dos de ellas clasificadas con riesgo bajo

- **CSP Scanner: Notices**

Advertencias:

1:254: La directiva manifest-src es una directiva experimental que probablemente se agregará a la especificación CSP.

1:1314: La directiva de solicitudes de actualización inseguras es una directiva experimental que probablemente se agregará a la especificación de CSP.

Elementos de información:

1: 1: Un borrador de la próxima versión de CSP desaprueba report-uri a favor de una nueva directiva report-to.

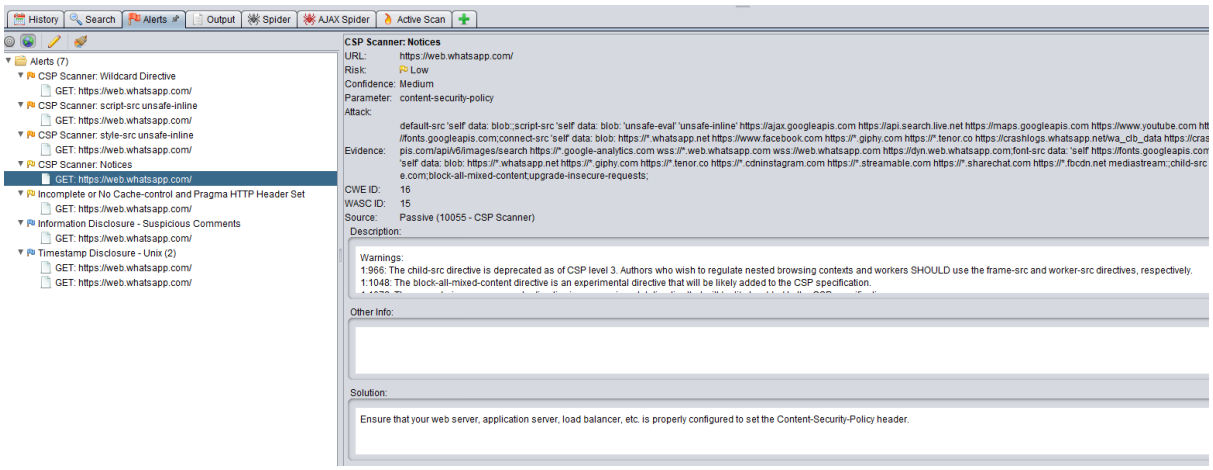


Figura 55. Alerta de Notices.

Fuente: propia con base en el escaneo de OWASP ZAP.

- **Cookie Without SameSite Attribute**

Se ha establecido una cookie sin el atributo SameSite, lo que significa que la cookie se puede enviar como resultado de una solicitud 'entre sitios'. El atributo SameSite es una medida eficaz para contrarrestar la falsificación de solicitudes entre sitios, la inclusión de scripts entre sitios y los ataques de tiempo.

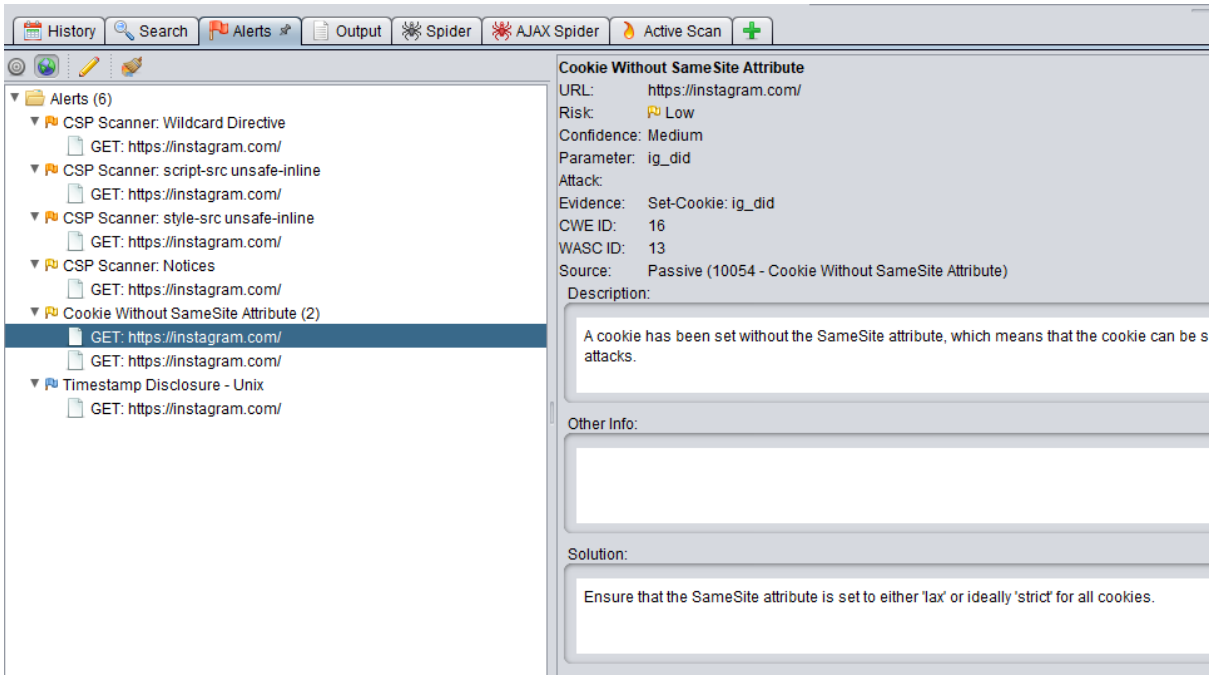


Figura 56. Alerta de Cookie Without SameSite Attribute.
Fuente: propia con base en el escaneo de OWASP ZAP.

Para cada vulnerabilidad se detalla: URL afectada, tipo de riesgo, nivel de confidencialidad, parámetro y ataque realizado, CWE ID, WESC ID y descripción.

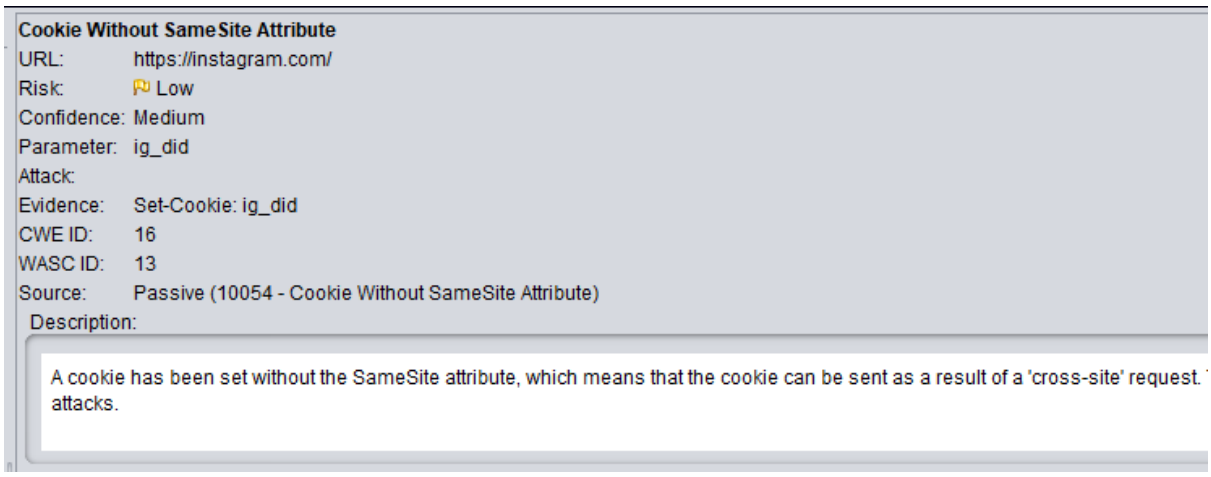
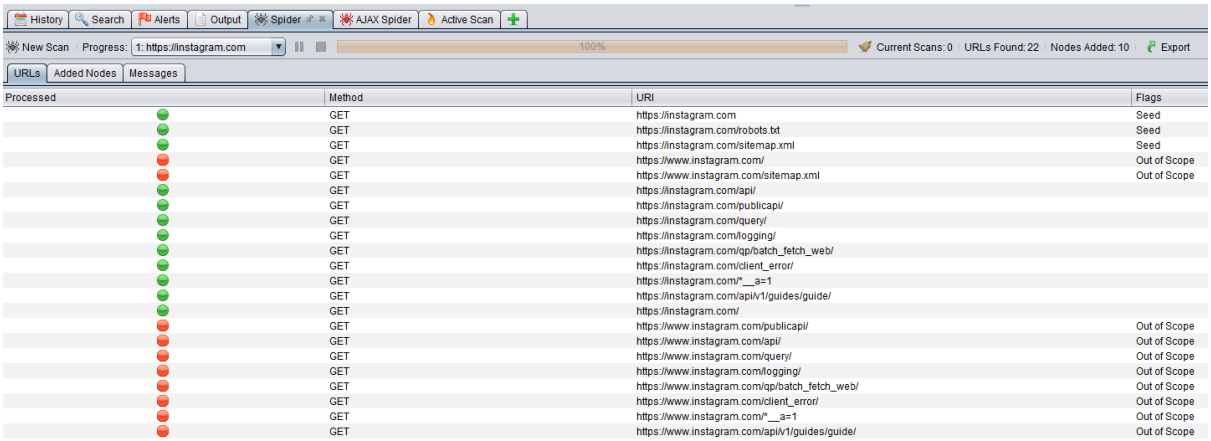


Figura 57. Detalles de vulnerabilidad.
Fuente: propia con base en el escaneo de OWASP ZAP.

En la pestaña *Spider* o araña traducido al español, se obtiene los recursos (URL) que la aplicación está utilizando.



Processed	Method	URI	Flags
	GET	https://instagram.com	Seed
	GET	https://instagram.com/robots.txt	Seed
	GET	https://instagram.com/sitemap.xml	Seed
	GET	https://www.instagram.com/	Out of Scope
	GET	https://www.instagram.com/sitemap.xml	Out of Scope
	GET	https://instagram.com/api/	
	GET	https://instagram.com/publicapi/	
	GET	https://instagram.com/query/	
	GET	https://instagram.com/logging/	
	GET	https://instagram.com/qp/batch_fetch_web/	
	GET	https://instagram.com/client_error/	
	GET	https://instagram.com/*__a=1	
	GET	https://instagram.com/api/v1/guides/guide/	
	GET	https://instagram.com/	
	GET	https://www.instagram.com/publicapi/	Out of Scope
	GET	https://www.instagram.com/api/	Out of Scope
	GET	https://www.instagram.com/query/	Out of Scope
	GET	https://www.instagram.com/logging/	Out of Scope
	GET	https://www.instagram.com/qp/batch_fetch_web/	Out of Scope
	GET	https://www.instagram.com/client_error/	Out of Scope
	GET	https://www.instagram.com/*__a=1	Out of Scope
	GET	https://www.instagram.com/api/v1/guides/guide/	Out of Scope

Figura 58. Detalles de la pestaña Spider de Instagram.
Fuente: propia con base en el escaneo de OWASP ZAP.

Para ver un mayor detalle de las vulnerabilidades, se puede generar un reporte HTML, XML, MarkDown o Json. El reporte completo con los hallazgos del análisis realizado se adjunta en la sección de Apéndice 1.

4.6 Informe de resultados

Se presentan a continuación, los hallazgos a partir de los resultados obtenidos a través de los análisis estáticos y dinámicos, tomando en consideración la guía de riesgos móviles propuesta por OWASP.

M1 – Uso inadecuado de la plataforma

Se comprueba a través de los análisis que ambas aplicaciones siguen las directrices publicadas por las plataformas y utilizan prácticas seguras de codificación y configuración en el lado del servidor de la aplicación móvil.

M2 – Almacenamiento de datos inseguros

Se evidencia a partir de los análisis que ambas aplicaciones crean archivos temporales y almacenan información confidencial dentro de ellos. La información confidencial nunca debe escribirse en un archivo temporal.

Se recomienda validar y reforzar cómo manejan el almacenamiento de información en archivos temporales y hacer uso de la memoria caché, en vez de almacenar información en archivos.

M3 – Comunicación insegura

Se comprueba a través de los análisis, que la aplicación de Instagram utiliza implementación insegura de WebView, ya que ignora los errores del certificado SSL y acepta cualquier certificado de este tipo, haciendo a esta aplicación vulnerable a los ataques MITM.

Se recomienda exigir siempre la cadena verificación de SSL, establecer una conexión segura, solo después de verificar que la identidad del servidor de endpoint utiliza certificados de confianza y no permitir certificados autofirmados.

M4 – Autenticación insegura

Se evidencia a partir de los análisis que ambas aplicaciones poseen autenticación segura, ya que ambas se cercioran de que todas las solicitudes de autenticación se realicen del lado del servidor. Esto asegurará que los datos de la aplicación solo estén disponibles después de una autenticación exitosa.

M5 – Criptografía insuficiente

Se comprueba a través de los análisis que ambas aplicaciones contienen criptografía insuficiente, haciendo uso de hash inseguros y obsoletos como MD5 y SHA-1. También, ambas utilizan un generador inseguro de números aleatorios.

Se recomienda para ambas aplicaciones seguir las directrices del NIST sobre algoritmos recomendados, aplicar estándares criptográficos que resistirán la prueba del tiempo durante al menos diez años en el futuro y hacer uso de generadores de números pseudoaleatorios (PRNGs) criptográficos que abordan este problema mencionado anteriormente, generando resultados más difíciles de predecir.

M6 – Autorización insegura

Se evidencia a partir de los análisis que ambas aplicaciones poseen autorización segura, evitando depender de roles o información de permisos que provenga del propio dispositivo móvil y verificando de forma independiente que los identificadores entrantes asociados con una solicitud que viene junto con la identificación coinciden y pertenecen a la identidad entrante.

M7 – Calidad del código en el lado del cliente

Se comprueba a través de los análisis que ambas aplicaciones no poseen una alta calidad del código al lado del cliente, ya que las dos utilizan la base de datos SQLite y ejecutan una consulta SQL sin procesar. La entrada de un usuario que no es de confianza

en consultas SQL sin procesar, puede provocar la inyección de SQL. Además, la información confidencial debe cifrarse y escribirse en la base de datos.

Se recomienda al realizar la validación de entrada, considerar todas las propiedades potencialmente relevantes, incluida la longitud, el tipo de entrada, el rango completo de valores aceptables, las entradas faltantes o adicionales, la sintaxis, la coherencia en los campos relacionados en conformidad con las reglas de la empresa.

M8 – Alteración del código

Se evidencia a partir de los análisis, que ambas aplicaciones no son vulnerables a los impactos de la modificación del código.

M9 – Ingeniería inversa

Se evidencia a partir de los análisis, que ambas aplicaciones son susceptibles a la ingeniería inversa, ya que los archivos pueden contener información confidencial codificada como nombres de usuario, contraseñas, claves, etc.

Se recomienda prioritariamente utilizar SQLCipher o una extensión similar para cifrar la base de datos de la aplicación en su dispositivo móvil. También se recomienda usar una herramienta de ofuscación durante el desarrollo de la aplicación.

M10 – Funcionalidad extraña

Se comprueba a través de los análisis, que ambas aplicaciones no poseen funcionalidad extraña o funciones ocultas de puerta trasera olvidadas por los desarrolladores.

Capítulo 5. Propuesta de metodología

Una auditoría de seguridad para aplicaciones móviles, se basa en analizar las aplicaciones, a través de escaneos de vulnerabilidad dinámicos y estáticos, con la finalidad de conocer el estado de seguridad de las aplicaciones, nivel de riesgo y posibles brechas de seguridad. Una vez realizado el análisis de vulnerabilidades y haber determinado el nivel de riesgo de las aplicaciones, se debe realizar un informe detallado que muestre los resultados, controles de seguridad, medidas de mitigación y recomendaciones realizadas por los profesionales en seguridad o encargados de la auditoría.

La metodología propuesta como parte de este proyecto de investigación está constituida por cuatro fases, para poder determinar el nivel de seguridad de las aplicaciones móviles que tienen acceso a información confidencial de la PyME y de sus clientes, sirve también, para analizar las aplicaciones más usadas por los empleados de las PyMEs (redes sociales, mensajería instantánea, etc.), con el propósito de encontrar posibles riesgos causados por

ellas y así implementar controles de seguridad para una defensa más profunda de la información.

Fases de la metodología

Se presenta a continuación, las cuatro fases de la metodología propuesta por los investigadores.

Fase 1- Planificación y preparación: en esta fase se deben de identificar y colectar información básica para realizar el análisis de la fase 2, tal y como el sistema operativo, las versiones del sistema operativo, las aplicaciones por auditar y las herramientas por utilizar para los análisis de vulnerabilidades. Al igual, se debe considerar la muestra de dispositivos por auditar.

Fase 2- Ejecución y análisis: en esta fase se ejecutan los análisis estáticos y dinámicos en las aplicaciones y dispositivos seleccionados para obtener los reportes con los hallazgos generados por las herramientas.

Fase 3- Revisión detallada: una vez obtenidos los resultados de la fase 2, se proponen posibles soluciones para corregir vulnerabilidades, políticas de seguridad y controles para mitigar los riesgos y vulnerabilidades identificadas.

Fase 4- Informe y seguimiento: en esta última fase, se realiza un informe con los resultados obtenidos de las tres fases anteriores y se presenta a la gerencia para la toma de decisiones. También, dicho informe se utilizará para darle seguimiento a los hallazgos encontrados, siguiendo siempre un proceso secuencial que permita a los “implementadores” mejorar la seguridad de sus sistemas.

5.1 Fase 1- Planificación y preparación

Es la primera fase de la metodología y consiste en definir el alcance y objetivos de la auditoría, las tecnologías que se van a analizar y las herramientas que se van a utilizar para realizar dichos análisis. Los aspectos que se debe identificar en esta primera etapa son:

- El sistema operativo del dispositivo.
- La versión del sistema operativo.
- La muestra de aplicaciones móviles que se van a auditar.
- Las herramientas para llevar a cabo los análisis de vulnerabilidades.

5.1.1 Identificar el sistema operativo y sus versiones

Sistema operativo Android

Como lo menciona Juan Garzón (2020): “Android es uno de los sistemas operativos más conocidos del mundo y el más popular en dispositivos móviles”, su facilidad para personalizar y por su amplia variedad de fabricantes, lo hacen muy atractivo para los usuarios.

Android, basado en Kernel de Linux y otros softwares de código abierto, se puede encontrar en celulares, tablets, relojes, televisores, automóviles, computadoras, gafas inteligentes, consolas, cámara de fotos y más.

Principales características

- Código abierto.
- Núcleo basado en el Kernel de Linux.
- Adaptable a muchas pantallas y resoluciones.
- Utiliza SQLite para el almacenamiento de datos.
- Ofrece diferentes formas de mensajería.
- Navegador web basado en WebKit incluido.
- Soporte de Java y muchos formatos multimedia.
- Soporte de HTML, HTML5, Adobe Flash Player, etc.
- Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software.
- Catálogo de aplicaciones gratuitas o pagas en el que pueden ser descargadas e instaladas (Google Play).
- Bluetooth.
- Multitarea real de aplicaciones.

Versiones más recientes de Android.

Nombre	Versión	Fecha de lanzamiento
Android Marshmallow	versión 6.0-6.0.1	5 de octubre de 2015.
Android Nougat	versión 7.0-7.1.2	15 de junio de 2016.
Android Oreo	versión 8.0-8.1	21 de agosto de 2017.
Android Pie	versión 9.0	6 de agosto de 2018.
Android 10	versión 10.0	3 de septiembre de 2019.

Tabla 28. Versiones más recientes de Android.
Fuente propia, basada en la información de Adslzone.

Sistema operativo iOS

Como lo describe Adslzone (2020), “iOS es un sistema operativo lanzado y utilizado por Apple. Su nombre proviene de iPhone OS. Es decir, iPhone Operative System o Sistema Operativo de iPhone”, al contrario de Android, es un sistema operativo cerrado que solo se pude utilizar en dispositivos de marca Apple.

Principales características

- Código cerrado
- Orientado específicamente para su uso mediante dispositivos móviles con pantalla táctil.
- iOS es una variante del Mac OS X
- Está basado en Unix**.**
- Capacidad para trabajar con múltiples programas a la vez y en segundo plano, lo que es conocido como la multitarea.
- Incorpora la tecnología multitouch**.**

Versiones más recientes de iOS

Nombre	Versión	Fecha de Lanzamiento
iOS 10	10.X	Junio 13, 2016
iOS 11	11.X	Junio 5, 2017
iOS 12	12.X	Junio 4, 2018
iOS 13	13.X	Junio 3, 2019

Tabla 29. Versiones más recientes de iOS.
Fuente propia, basada en la información de Apple.

5.1.2 Aplicaciones por auditar

Se recomienda tomar en consideración todas aquellas aplicaciones móviles que tengan acceso a información confidencial de la PyME o de sus clientes y las aplicaciones más utilizadas por los empleados de la organización (redes sociales, mensajería instantánea, etc.) para poder elegir el muestreo de aplicaciones por auditar en ambos sistemas operativos (iOS y Android).

Para efectos de determinar las aplicaciones más utilizadas por los empleados de la organización, se recomienda censar el uso de las mismas a través de encuesta de papel, encuestas en línea, correos o dispositivos móviles si se cuenta con el presupuesto.

5.1.3 Herramientas para análisis

Android

Lo primero que se debe hacer y cómo se menciona en el capítulo 4 del caso práctico, es conseguir el archivo .APK (Android Application Package) para analizar una aplicación Android, el cual es un archivo ejecutable de aplicaciones para Android. Este formato es una variante del formato JAR de Java y se usa para distribuir e instalar componentes empaquetados para la plataforma Android, tanto teléfonos inteligentes como tablets. Existe una variedad de herramientas diseñadas exclusivamente para la extracción de los .APK (APKPure, APK Extractor o Smart Apk Extractor, entre otros) a las cuales se puede tener acceso de forma gratuita desde la tienda de Google Play Store. Para efectos del caso práctico del trabajo de investigación, se utilizó APK Extractor y los pasos de instalación y extracción del APK se encuentran en el Apéndice 2.

Una vez obtenido el .APK, se debe seleccionar la herramienta para realizar los análisis estáticos y dinámicos que mejor se adapte a los aspectos que se desean examinar de la aplicación, por ejemplo, conexiones o APIs, permisos que requiere la aplicación, análisis de código, algoritmo de encriptación, información de certificados, etc.

iOS

Se debe conseguir el archivo .ipa para analizar una aplicación de iOS. El formato de archivos .ipa (iOS App Store Package) es el utilizado para las aplicaciones de Apple en los dispositivos iPhone y iPad. Generalmente, su utilización no requiere la compresión como los archivos RAR y ZIP y solo funciona en los dispositivos que usen iOS. Fue desarrollado por el equipo de Steve Jobs en 2007 como base de las aplicaciones de Apple y es la más común en los dispositivos iOS. Las aplicaciones descargadas de otra página web que no sea iTunes no se podrá sincronizar en el dispositivo iOS (iPhone y iPad) sin Jailbreak.

Existe una variedad de herramientas diseñadas exclusivamente para la extracción de los .ipa (DMG Extractor, Crunch o iMazing, entre otros) a las cuales se pueden obtener desde la tienda de Apple AppStore.

De igual forma tal y como se realizan con el .apk de Android, después de obtener el archivo .ipa se debe seleccionar la herramienta para realizar los análisis estáticos y dinámicos que mejor se adapte a los aspectos que se desean examinar de la aplicación.

Seguidamente, se enlistará una serie de herramientas utilizadas para realizar análisis estáticos y dinámicos en aplicaciones Android y iOS.

Nota: Las herramientas enumeradas en las tablas siguientes se presentan en orden alfabético. Los investigadores no respaldan a ninguno de los proveedores o herramientas enumerándolos en las siguientes tablas. Se ha hecho todo lo posible para proporcionar esta información con la mayor precisión posible.

Análisis Estático (SAST)		
Nombre	**Licencia**	**Descripción / Notas**
CodeSonar	Comercial	Herramienta que admite C, C ++, Java y C \ # y se asigna a las 10 principales vulnerabilidades de OWASP.
ImmuniWeb	Código abierto	Pruebas de aplicaciones móviles y backend. Cero falso-positivo SLA. Cumplimiento de PCI DSS y GDPR. Puntuaciones CVE, CWE y CVSSv3. Pautas de remediación procesables. Integración de herramientas SDLC y CI / CD. Aplicación de parches virtuales con un clic a través de WAF. Acceso 24/7 a analistas de seguridad.
Insider CLI	Código abierto	Una herramienta de prueba de seguridad de aplicaciones estáticas (SAST) de código abierto escrita en GoLang para Java Maven y Android), Kotlin (Android), Swift (iOS), .NET Full Framework, C # y Javascript (Node.js).
Kiuwan	Comercial	Las pruebas de seguridad de Kiuwan incluyen análisis de código estático y análisis de composición de software, con automatización en cualquier etapa del SDLC. Cobertura de los principales lenguajes y frameworks populares para el desarrollo móvil, con integración a nivel IDE.
Micro Focus Fortify on	Comercial	Las pruebas de seguridad incluyen

Demand		análisis de código estático y escaneo programado para aplicaciones móviles y brindan resultados precisos. Identifique las vulnerabilidades de seguridad en: cliente, servidor y red. Fortify permite un escaneo estándar que ayuda a identificar malware. Fortify admite múltiples plataformas como Google Android, Apple iOS, Microsoft Windows y Blackberry.
Mobile Security Framework (MobSF)	Código abierto	Mobile Security Framework (MobSF) es una aplicación móvil todo en uno automatizada (Android / iOS / Windows), un marco de prueba, análisis de malware y evaluación de seguridad capaz de realizar análisis estáticos y dinámicos.
SonarQube	Código abierto	Escanea el código fuente en 15 idiomas para detectar errores, vulnerabilidades y errores de código. Complementos de SonarQube IDE para Eclipse, Visual Studio e IntelliJ.
Veracode	Comercial	Android, ASP.NET, C\#, C, C++, Classic ASP, COBOL, ColdFusion/Java, Go, Groovy, iOS, Java, JavaScript, Perl, PhoneGap/Cordova, PHP, Python, React Native, RPG, Ruby on Rails, Scala, Titanium, TypeScript, VB.NET, Visual Basic 6, Xamarin
WhiteHat Security	Comercial	Es una plataforma de seguridad basada en la nube. Es compatible con las plataformas Android y iOS. La plataforma Sentinel proporciona información e informes detallados para conocer el estado del proyecto.

		Pruebas automatizadas de aplicaciones móviles estáticas y dinámicas. Las pruebas se realizan en el dispositivo real mediante la instalación de la aplicación móvil, no utiliza ningún emulador para las pruebas. Ofrece una descripción clara y concisa de las vulnerabilidades de seguridad y proporciona una solución.

Tabla 30. Herramientas para realizar análisis estáticos en Android y iOS.
Fuente: propia.

Análisis Dinámico (DAST)		
Nombre	**Licencia**	**Descripción / Notas**
Drozer	Código abierto	Drozer es compatible con emuladores y dispositivos Android reales para pruebas de seguridad. Ejecuta código habilitado para Java en el propio dispositivo. Brinda soluciones en todas las áreas de la ciberseguridad. El soporte de Drozer se puede ampliar para encontrar y explotar debilidades ocultas. Solo es compatible con la plataforma Android.
ImmuniWeb	Código abierto	Pruebas de aplicaciones móviles y backend. Cero falso-positivo SLA. Cumplimiento de PCI DSS y GDPR. Puntuaciones CVE, CWE y CVSSv3. Pautas de remediación procesables. Integración de herramientas SDLC y CI / CD. Aplicación de parches virtuales con un clic a través de WAF. Acceso 24/7 a analistas de seguridad.

Inspeckage	Código abierto	Inspeckage (Android Package Inspector) es un framework de análisis para aplicaciones de Android que funciona como un módulo de Xposed. Permite ver en tiempo real los eventos que ocurren en el teléfono.
Mobile Security Framework (MobSF)	Código abierto	Mobile Security Framework (MobSF) es una aplicación móvil todo en uno automatizada (Android / iOS / Windows), un marco de prueba, análisis de malware y evaluación de seguridad capaz de realizar análisis estáticos y dinámicos.
OWASP Zed Attack Proxy	Código abierto	La herramienta de prueba de seguridad de código abierto más popular del mundo. ZAP es mantenido activamente por cientos de voluntarios internacionales. Es muy fácil de instalar. ZAP está disponible en 20 idiomas diferentes.
Veracode	Comercial	Android, ASP.NET, C\#, C, C++, Classic ASP, COBOL, ColdFusion/Java, Go, Groovy, iOS, Java, JavaScript, Perl, PhoneGap/Cordova, PHP, Python, React Native, RPG, Ruby on Rails, Scala, Titanium, TypeScript, VB.NET, Visual Basic 6, Xamarin
WhiteHat Security	Comercial	Es una plataforma de seguridad basada en la nube. Es compatible con las plataformas Android y iOS. La plataforma Sentinel proporciona información e informes detallados para conocer el estado del proyecto.

		Pruebas automatizadas de aplicaciones móviles estáticas y dinámicas. Las pruebas se realizan en el dispositivo real mediante la instalación de la aplicación móvil, no utiliza ningún emulador para las pruebas. Ofrece una descripción clara y concisa de las vulnerabilidades de seguridad y proporciona una solución.

Tabla 31. Herramientas para realizar análisis dinámicos en Android y iOS.
Fuente: propia.

5.2 Fase 2- Ejecución y análisis

Se ejecutan durante esta fase, los análisis tanto estáticos como dinámicos, con el propósito de obtener uno sobre las vulnerabilidades de las aplicaciones seleccionadas en la fase 1.

Aspectos importantes que se deben considerar para esta etapa:

- Es de gran importancia tener en cuenta que no todas las herramientas de análisis estáticos y dinámicos pueden ser utilizadas en ambas tecnologías (Android y iOS).
- Otro aspecto importante por considerar para esta fase, es que se debe contar con un ambiente de virtualización o ambiente prueba, en el cual se instalarán y configurarán las herramientas para realizar los análisis.
    - Para crear dicho ambiente de prueba se debe considerar qué sistema de virtualización se utilizará, entre los más reconocidos se encuentra: VMware, Citrix y Virtual Box de Oracle. Para efectos del caso práctico se utilizó la herramienta Virtual Box, puesto que cumplía con las funcionalidades necesarias, facilidad de uso y además por ser un software gratuito.
- Se debe contar con un conocimiento básico de los sistemas operativos por utilizar en el ambiente de prueba, para poder realizar la instalación de las herramientas de escaneo a mediante ejecutables (Windows) o línea de comandos (Linux).
- Herramientas de tipo comercial o pagadas, que brindan pruebas gratis por 30 días o menos, no proporcionan la opción de exportar los reportes, sino un resumen acerca de las vulnerabilidades detectadas a grandes rasgos por medio de su interfaz web.

El último paso de esta fase consiste en realizar una revisión preliminar de los resultados de los análisis, con el objetivo de obtener la información necesaria para tomar la decisión de cómo proceder en la siguiente fase.

5.3 Fase 3- Revisión detallada

Es importante en esta etapa y para las personas que implementan la metodología, identificar las vulnerabilidades más críticas existentes en las aplicaciones, para así poder proponer posibles soluciones para corregir las debilidades de seguridad de las aplicaciones, políticas de seguridad y controles para mitigar los riesgos.

5.3.1 Posibles soluciones y controles para mitigar los riesgos y vulnerabilidades identificadas.

Se presenta a continuación, el top 10 de riesgos en seguridad más críticos en aplicaciones móviles, según OWASP versión 2016 y sus correspondientes causas y su posible control para mitigar el riesgo.

M1 – Uso inadecuado de la plataforma

Causas:

- La violación de las directrices publicadas. Todas las plataformas tienen directrices de desarrollo para la seguridad (Android, iOS, Windows Phone). Si una aplicación contradice las mejores prácticas recomendadas por el fabricante, estará expuesta a este riesgo. Por ejemplo, existen directrices sobre cómo usar el iOS Keychain o cómo proteger los servicios exportados en Android. Las aplicaciones que no sigan estas pautas experimentarán este riesgo.
- Violación de convención o práctica común. No todas las mejores prácticas están codificadas en la guía del fabricante. En algunos casos, existen mejores prácticas de facto que son comunes en las aplicaciones móviles.
- Mal uso involuntario. Algunas aplicaciones tienen la intención de hacer lo correcto, pero en realidad se equivocan en una parte de la implementación. Esto podría ser un error simple, como configurar el indicador incorrecto en una llamada a la API o podría ser un malentendido sobre cómo funcionan las protecciones.

Controles

- Se deben utilizar prácticas seguras de codificación y configuración en el lado del servidor de la aplicación móvil.

Dificultad de ataque: fácil.

M2 – Almacenamiento de datos inseguros

Causas

Esto se ve más en procesos internos indocumentados o poco documentados, como:

- La forma en que el sistema operativo almacena en caché los datos, imágenes, pulsaciones de teclas, registros y búferes;
- La forma en que el marco de desarrollo almacena en caché los datos, imágenes, pulsaciones de teclas, registros y búferes;
- La forma o la cantidad de anuncios de datos, analíticos, sociales o marcos de habilitación almacenan en caché los datos, imágenes, pulsaciones de teclas, registros y búferes.

Controles

Validar y reforzar cómo manejan los siguientes tipos de funciones:

- Almacenamiento en caché de URL (tanto solicitud como respuesta).
- Almacenamiento en caché de pulsaciones de teclado.
- Copiar / Pegar almacenamiento en caché del búfer.
- Fondo de la aplicación.
- Datos intermedios.
- Inicio sesión.
- Almacenamiento de datos HTML5.
- Objetos de cookies del navegador.
- Datos analíticos enviados a terceros.

Dificultad de ataque: fácil.

M3 – Comunicación insegura

Causas:

- Este riesgo cubre todos los aspectos de la obtención de datos del punto A al punto B, pero haciéndolo de forma insegura. Abarca comunicaciones de móvil a móvil, comunicaciones de aplicación a servidor o comunicaciones de móvil a otra cosa.
- Este riesgo incluye todas las tecnologías de comunicación que podría utilizar un dispositivo móvil: TCP / IP, WiFi, Bluetooth / Bluetooth-LE, NFC, audio, infrarrojos, GSM, 3G, SMS, etc.
- Las características destacadas incluyen empaquetar algún tipo de datos sensibles y transmitirlos dentro o fuera del dispositivo.
- Si los datos se almacenan localmente en el mismo dispositivo, eso son datos inseguros.

- Si los datos se pueden cambiar mientras están en tránsito sin que el cambio sea detectable (por ejemplo, a través de un ataque man-in-the-middle), ese es un buen ejemplo de este riesgo.
- Si se pueden exponer, aprender o derivar datos confidenciales, observando las comunicaciones a medida que ocurren (es decir, escuchando a escondidas) o grabando la conversación a medida que ocurre y atacándola más tarde (ataque fuera de línea), ese también es un problema de comunicación inseguro.
- No configurar y validar correctamente una conexión TLS (por ejemplo, verificación de certificados, cifrados débiles, otros problemas de configuración de TLS) están todos aquí en una comunicación insegura.

Controles:

- Validar que la capa de red es segura y no susceptible de eavesdropping.
- Aplique SSL / TLS para transportar canales que la aplicación móvil utilizará para transmitir información confidencial, tokens de sesión u otros datos confidenciales a una API de backend o servicio web.
- Para entidades externas, como empresas de análisis de terceros, redes sociales, etc., hacer el uso de sus versiones SSL / TLS cuando una aplicación ejecuta una rutina a través del navegador / webkit.
- Utilizar altos estándares de cifrado de la industria, con longitudes de clave adecuadas.
- Utilizar certificados firmados por un proveedor de CA de confianza.
- No permitir certificados autofirmados y considere la posibilidad de fijar certificados para aplicaciones conscientes de la seguridad.
- Exigir siempre la cadena verificación de SSL.
- Establecer una conexión segura solo después de verificar que identidad del servidor de endpoint utiliza certificados de confianza.
- No enviar datos confidenciales a través de canales alternativos (por ejemplo, SMS, MMS o notificaciones).
- Aplicar una capa separada de cifrado a los datos confidenciales antes de que se entreguen al canal SSL.

Dificultad de ataque: fácil.

M4 – Autenticación insegura

Causas:

- Si la aplicación móvil puede ejecutar de forma anónima una solicitud de servicio de la API de backend sin proporcionar un token de acceso, esta aplicación sufre una autenticación insegura;

- Si la aplicación móvil almacena contraseñas o secretos compartidos localmente en el dispositivo, lo más probable es que sufra una autenticación insegura;
- Si la aplicación móvil utiliza una política de contraseña débil para simplificar el ingreso de una contraseña, sufre de autenticación insegura o
- Si la aplicación móvil utiliza una función como TouchID, sufre de autenticación insegura.

Controles:

- Si está transfiriendo una aplicación web a su equivalente móvil, los requisitos de autenticación de las aplicaciones móviles deben coincidir con los del componente de la aplicación web. Por lo tanto, no debería ser posible autenticarse con menos factores de autenticación que el navegador web;
- La autenticación de un usuario localmente puede conducir a vulnerabilidades de bypass del lado del cliente. Si la aplicación almacena datos localmente, la rutina de autenticación se puede omitir en dispositivos con jailbreak mediante la manipulación en tiempo de ejecución o la modificación del binario. Si existe un requisito empresarial convincente para la autenticación fuera de línea, consulte M10 para obtener orientación adicional sobre cómo prevenir ataques binarios contra la aplicación móvil;
- Siempre que sea posible, asegúrese de que todas las solicitudes de autenticación se realicen en el lado del servidor. Una vez realizada la autenticación, los datos de la aplicación se cargarán en el dispositivo móvil. Esto asegurará que los datos de la aplicación solo estén disponibles después de una autenticación exitosa;
- Si se requiere el almacenamiento de datos del lado del cliente, los datos deberán cifrarse con una clave de cifrado que se obtenga de forma segura a partir de las credenciales de inicio de sesión del usuario. Esto garantizará que solo se pueda acceder a los datos de la aplicación almacenados después de ingresar con éxito las credenciales correctas. Existen riesgos adicionales de que los datos se descifren mediante ataques binarios. Consulte M9 para obtener orientación adicional sobre cómo prevenir ataques binarios que conducen al robo de datos locales;
- La funcionalidad de autenticación persistente (recuérdame) implementada en aplicaciones móviles nunca debe almacenar la contraseña de un usuario en el dispositivo;
- Idealmente, las aplicaciones móviles deberían utilizar un token de autenticación específico del dispositivo que el usuario pueda revocar dentro de la aplicación móvil. Esto garantizará que la aplicación pueda mitigar el acceso no autorizado de un dispositivo robado / perdido;

- No utilice valores falsos para autenticar a un usuario. Esto incluye identificadores de dispositivos o ubicación geográfica;
- La autenticación persistente dentro de las aplicaciones móviles debe implementarse como opt-in y no estar habilitada de forma predeterminada;
- Si es posible, no permita que los usuarios proporcionen números PIN de cuatro dígitos para las contraseñas de autenticación.

Dificultad de ataque: fácil.

M5 – Criptografía insuficiente

Causas

- La aplicación móvil puede usar un proceso detrás del cifrado / descifrado que es fundamentalmente defectuoso y puede ser aprovechado por el adversario para descifrar datos confidenciales.
- La aplicación móvil puede implementar o aprovechar un algoritmo de cifrado / descifrado que es de naturaleza débil y que el adversario puede descifrar directamente.
- Procesos de gestión de claves deficientes
    - Incluir las claves en el mismo directorio legible por el atacante que el contenido cifrado.
    - Poner las claves a disposición del atacante.
    - Evite el uso de claves codificadas dentro de su binario.
    - Las claves se pueden interceptar mediante ataques binarios.
- Uso de algoritmos inseguros y obsoletos.
    - RC2
    - MD4
    - MD5
    - SHA1
- Creación y uso de protocolos de cifrado personalizados.

Controles

- Evite el almacenamiento de datos sensibles en un dispositivo móvil siempre que sea posible.
- Aplicar estándares criptográficos que resistirán la prueba del tiempo durante al menos diez años en el futuro y
- Siga las pautas del NIST sobre algoritmos recomendados (consulte las referencias externas).

Dificultad de ataque: fácil.

M6 – Autorización insegura

Causas:

- Presencia de vulnerabilidades de referencia directa a objetos inseguros (IDOR), si ve una vulnerabilidad de referencia directa a objetos inseguros (IDOR), lo más probable es que el código no esté realizando una verificación de autorización válida.
- Endpoints ocultos: por lo general, los desarrolladores no realizan verificaciones de autorización en la funcionalidad oculta del backend, ya que asumen que la funcionalidad oculta solo la verá alguien en el rol correcto.
- Transmisiones de permisos o roles de usuario: si la aplicación móvil transmite los roles o permisos del usuario a un sistema backend como parte de una solicitud, está sufriendo una autorización insegura.

Controles

- Verifique las funciones y los permisos del usuario autenticado, utilizando solo la información contenida en los sistemas back-end. Evite depender de roles o información de permisos que provenga del propio dispositivo móvil.
- El código de backend debe verificar de forma independiente que los identificadores entrantes asociados con una solicitud (operandos de una operación solicitada) que vienen junto con la identificación coinciden y pertenecen a la identidad entrante.

Dificultad de ataque: fácil.

M7 – Calidad del código en el lado del cliente

Causas:

- La característica clave de este riesgo es que el código se ejecuta en el dispositivo móvil y el código debe cambiarse de una manera bastante localizada. Arreglar la mayoría de los riesgos requiere cambios en el código, pero en el caso de la calidad del código, el riesgo proviene de usar la API incorrecta, usar una API de manera insegura, usar construcciones de lenguaje inseguras o algún otro problema a nivel de código. Es importante destacar que este no es un código que se ejecuta en el servidor. Este es un riesgo que captura código incorrecto que se ejecuta en el propio dispositivo móvil.

Controles

- Mantener patrones de codificación consistentes con los que todos en la organización estén de acuerdo;
- Escriba código que sea fácil de leer y esté bien documentado;
- Cuando utilice búferes, siempre valide que las longitudes de los datos de búfer entrantes no excedan la longitud del búfer de destino;

- Mediante la automatización, identifique desbordamientos de búfer y fugas de memoria mediante el uso de herramientas de análisis estático de terceros y
- Priorice la resolución de desbordamientos de búfer y pérdidas de memoria sobre otros problemas de "calidad del código".

Dificultad de ataque: difícil.

M8 – Alteración del código

Causas

- Técnicamente, todo el código móvil es vulnerable a la manipulación del código. El código móvil se ejecuta en un entorno que no está bajo el control de la organización que lo produce.
- Aunque el código móvil es inherentemente vulnerable, es importante preguntarse si vale la pena detectarlo y tratar de evitar la modificación no autorizada del código. Las aplicaciones escritas para ciertos verticales comerciales (juegos, por ejemplo) son mucho más vulnerables a los impactos de la modificación del código que otras (hospitalidad, por ejemplo).

Controles

- La aplicación móvil debe poder detectar en el tiempo de ejecución que se ha agregado o cambiado el código de lo que sabe sobre su integridad en el momento de la compilación.
- Detección de root de Android,
- Dispositivo Android: Busque claves de prueba
- Verificar certificados OTA
- Compruebe si hay varias apk rooted conocidas
- Compruebe si hay binarios SU
- Detección de jailbreak de iOS

Dificultad de ataque: fácil.

M9 – Ingeniería inversa

Causas:

- Generalmente, la mayoría de las aplicaciones son susceptibles a la ingeniería inversa debido a la naturaleza inherente del código. La mayoría de los lenguajes que se utilizan para escribir aplicaciones en la actualidad son ricos en metadatos que ayudan enormemente al programador a depurar la aplicación. Esta misma capacidad también ayuda a un atacante a comprender cómo funciona la aplicación. Se dice que una aplicación es susceptible de ingeniería inversa si un atacante puede hacer cualquiera de las siguientes cosas:

- Comprender claramente el contenido de la tabla de cadenas de un binario
- Realice con precisión análisis multifuncional
- Obtenga una recreación razonablemente precisa del código fuente a partir del binario.

Controles

- Una herramienta de ofuscación.
- Un buen ofuscador tendrá las siguientes habilidades:
  - Limite los métodos / segmentos de código para ofuscar;
  - Ajustar el grado de-ofuscación para equilibrar el impacto en el rendimiento;
  - Resiste la de-ofuscación de herramientas como IDA Pro y Hopper;
  - Ofuscar tablas de cadenas y métodos

Dificultad de ataque: fácil.

M10 – Funcionalidad extraña

Causas

- A menudo, los desarrolladores incluyen funciones ocultas de puerta trasera u otros controles de seguridad de desarrollo internos que no están destinados a ser lanzados a un entorno de producción.
- La característica definitoria de este riesgo es dejar la funcionalidad habilitada en la aplicación que no estaba destinada a ser lanzada.

Controles:

- Realizar una revisión manual del código seguro utilizando SME más conocedores de este código.
- Examine los ajustes de configuración de la aplicación para descubrir cualquier interruptor oculto;
- Verifique que todo el código de prueba no esté incluido en la compilación de producción final de la aplicación;
- Examinar todos los puntos finales de API a los que accede la aplicación móvil para verificar que estos puntos finales estén bien documentados y disponibles públicamente;
- Examinar todas las declaraciones de registro para asegurarse de que no se escriba en los registros nada demasiado descriptivo sobre el backend.

Dificultad de ataque: fácil.

5.3.2 Políticas de seguridad

Una política debe ser específica para cada empresa puesto que debe basarse en los requerimientos de la PyME, su perfil de riesgo y su situación. En esta sección de la fase 3

se abordarán las políticas que se pueden aplicar para proteger los dispositivos móviles de ataques y amenazas. Desde técnicas básicas de bloqueo de pantalla hasta la encriptación de información en los dispositivos móviles.

Las políticas que se van a plantear en esta sección estarán basadas en los principales principios de seguridad, los cuales resumen los requisitos que deben cumplirse con el fin de proporcionar un nivel aceptable de seguridad:

- **Confidencialidad:** los datos sólo deben ser leídos por sus usuarios previstos y autorizados.
- **Integridad:** los datos sólo deben ser modificados por sus usuarios previstos y autorizados.
- **Disponibilidad:** usuarios autorizados siempre deben ser capaces de acceder a los datos que tienen acceso.

5.3.2.1 Política Screen-lock (bloqueo de pantalla)

Esta política consiste en implementar configuraciones que permitan bloquear la pantalla del dispositivo móvil, por medio de PIN, patrón de puntos, contraseña o biometría. Esto proporciona un componente de autenticación que ayuda a aumentar la seguridad y privacidad del usuario. Además, ayuda a evitar que un tercero pueda tener acceso directo a los contenidos del dispositivo. Otro aspecto que se debe tomar en cuenta a la hora de configurar el bloqueo de pantalla es el nivel mínimo de longitud del PIN o la contraseña, para que el mismo sea resistente a ataques de fuerza bruta.

Una política de bloqueo de pantalla, por lo tanto, no debe ser demasiado simple o fácil de predecir. Una política adecuada de bloqueo de pantalla para dispositivos móviles debe incluir lo siguiente:

- Los usuarios deben tener un PIN o contraseña activada en el dispositivo.
- Especificar una longitud mínima, número de letras, dígitos, símbolos, etc.
- Si se realizan “X” intentos de inicio de sesión fallidos, limpiar el dispositivo.
- Se debe configurar que después de cierto tiempo de inactividad del dispositivo se realice el bloqueo de pantalla.

5.3.2.2 Bloqueo remoto, tracking y borrado de la información

Los dispositivos móviles que se utilizan en una PyME, por lo general, tienen más probabilidad de interactuar con datos sensibles que los de uso personal. La propiedad intelectual y la privacidad de los datos está en riesgo si un dispositivo se pierde o es robado.

Una manera de implementar dos de los principios de seguridad antes mencionados (confidencialidad e integridad) para proteger los dispositivos móviles y los datos de la PyME o de sus clientes, es por medio de:

- **Bloqueo remoto:** tiene como fin prohibir el acceso directo al dispositivo.

- **Borrado remoto:** esta política de limpieza intenta asegurar que los datos no estén en riesgo cuando un dispositivo móvil se pierda o sea robado.
- **Geolocalización:** realiza seguimiento de los dispositivos móviles.

Esto permitirá que un usuario o un administrador pueda activar de forma remota el bloqueo del dispositivo, conocer la ubicación o iniciar un borrado que elimine todo el contenido del dispositivo. Además, se debe tomar en cuenta que aunque un dispositivo esté bloqueado, los datos todavía están disponibles y en riesgo, por lo que el borrado es una opción más segura y casi indispensable para resguardar la información de una PyME.

5.3.2.3 Encriptación o cifrado de datos

Es recomendable implementar una técnica de cifrado de datos para proteger la información digital y garantizar la confidencialidad. El cifrado de los datos en un dispositivo móvil es la encriptación de los datos almacenamos a través de un algoritmo informático y se recomienda para prevenir que sean extraídos de forma legible. Esto garantiza su confidencialidad en los dispositivos móviles, aunque no protege contra la extracción, ya que aunque un atacante logre extraer los datos almacenados, no podrá leerlos a menos que logre descifrarlos.

Los dispositivos Android y iOS más recientes ya vienen con la función de encriptar los datos de los dispositivos móviles por defecto, pero en muchos dispositivos que no son de última generación, la posibilidad de cifrar o encriptar su contenido es una opción configurable desde el apartado de seguridad. La encriptación o cifrado por sí sola no resuelve los problemas de seguridad, pero no deja de ser una pieza muy importante para proteger nuestra información.

5.3.2.4 Control de aplicaciones

Las funciones que se describieron en las secciones anteriores son ejemplos recurrentes de los mecanismos de protección que se utilizan contra las amenazas externas. Pero no protegen contra amenazas internas, como las aplicaciones maliciosas que se podrían instalar negligentemente por parte del usuario y finalmente, podrían robar los datos utilizando métodos legítimos. Estas aplicaciones podrían haberse instalado sin el conocimiento del usuario, utilizando correos electrónicos de phishing, ingeniería social, malware, ransonware u otros. A nivel de PyMEs, además de la revisión de aplicaciones, es recomendable proporcionar una lista negra o lista blanca de aplicaciones. El concepto de una lista negra se describe como una lista de prohibidos, mientras que una lista blanca se refiere a una de permitidos. Esto con el fin de orientar a los usuarios sobre las aplicaciones que deben o no instalar en sus dispositivos móviles.

5.2.3 Políticas de seguridad para BYOD

Hoy es muy común que los miembros de una organización hagan uso de sus dispositivos móviles personales para realizar sus labores en la organización donde brindan sus servicios y lo que era un dispositivo personal, se convierte en parte de la red empresarial de las pequeñas y medianas empresas. Seguidamente, se detallan tres de las áreas que se deben considerar para definir las políticas de BYOD de una PyME.

5.2.3.1 Definición de política de uso y gestión.

Una política BYOD es un conjunto de reglas que gobiernan los aspectos relacionados con el uso de dispositivos personales para acceder y utilizar recursos de las pequeñas y medianas empresas.

Detalle de plataformas, sistemas operativos y dispositivos aceptados

Dentro de una política de BYOD es necesario establecer qué tipo de plataformas, sistemas operativos y dispositivos móviles formarán parte de la modalidad BYOD, con base en los requerimientos o funciones que realizan las PyMEs. Estas especificaciones deben revisarse periódicamente.

También será necesario definir las condiciones y plazos de actualización de los dispositivos, así como la obligación de la actualización. Además, será necesario especificar en la política las consecuencias de no seguir las normas establecidas.

Controles de seguridad y gestión

Se debe especificar aspectos como los requerimientos para el acceso a la red en la parte más técnica de la política, las configuraciones de seguridad recomendadas y obligatorias, los controles de seguridad y gestión que se aplicarán sobre el dispositivo y los datos, (incluyendo el posible reseteo o borrado del dispositivo), la monitorización y auditorías que se efectuará sobre el dispositivo, las aplicaciones permitidas y no permitidas, las posibles restricciones en el acceso a datos empresariales, etc.

5.2.3.2 Uso de aplicaciones Mobile Device Management (MDM).

Como parte de la metodología, se recomienda a las pequeñas y medianas empresas (PyME), la implementación de un Mobile Device Management (MDM), el cual es un software que permite controlar, asegurar y reforzar políticas en dispositivos móviles. Algunas de las ventajas y funcionalidades de un MDM son las siguientes:

- Permiten realizar un seguimiento de los dispositivos móviles, usuarios y datos.
- Permiten elaborar informes del uso de los dispositivos móviles.
- Gestión de la información relacionada con el uso del dispositivo móvil.

- Control de las aplicaciones que pueden ser utilizadas.
- Facilitan la administración de los dispositivos.
- Separación de datos organizacionales y personales mediante contenedores.
- Protección antimalware, anti-phishing y antispam, usando antivirus u otras tecnologías.
- Seguridad en caso de pérdida o robo de dispositivos mediante el rastreo GPS o el bloqueo o borrado remoto.

Además, es importante mencionar cinco campos en los que se puede actuar desde una aplicación MDM en cuanto a la seguridad:

- Gestión de aplicaciones
- Gestión de políticas
- Gestión de seguridad
- Gestión de inventario

A nivel de seguridad, estos programas permiten asegurar, monitorear y gestionar diversas funciones de forma remota desde un servidor centralizado en el que se definen las políticas o actuaciones a realizar sobre los dispositivos móviles.

5.2.3.3 Concientización del usuario

Un área critica dentro de la metodología es aplicar políticas de seguridad para concientizar y educar a los trabajadores dentro de las pequeñas y medianas empresas, con el fin de evitar ataques de ingeniería social, malware, phishing u otros, que puedan poner en riesgo la información.

La comunicación es una pieza fundamental de la metodología. De nada sirve establecer las mejores políticas, herramientas de seguridad y gestión, si los usuarios no son conscientes de ello.

Los usuarios deben conocer también las consecuencias del incumplimiento de las políticas. Una vez informados los detalles, los usuarios deberán firmar la aceptación de la política.

5.2.3.4 Evaluación y seguimiento

Una vez establecidas e implementadas las políticas, se debe revisar periódicamente la validez de las políticas definidas para asegurar que se siguen adaptando a los objetivos y requerimientos de las PyMEs.

5.4 Fase 4 - Informe y seguimiento

Informe

El objetivo de esta última fase de la metodología es revelar los resultados obtenidos. En esta fase se elabora el informe detallado de la auditoría, el cual debe contener la información que fue determinada en el plan inicial de la metodología o fase 1, los criterios acordados, el tiempo de duración, la identificación del auditor, las conclusiones detalladas con las políticas y mitigaciones propuestas y la declaración de confidencialidad.

Seguimiento

Después de la auditoría, se debe establecer plazos para la implementación y verificación de resultados de las acciones correctivas a las cuales, señalar los aspectos positivos que se presentaron durante la auditoría y darle seguimiento para velar que las acciones correctivas y políticas se implementen.

Esta metodología debe revisarse periódicamente (cada 6-12 meses) para asegurar que se adapta en todo momento a la empresa y sus necesidades de negocio.

Capítulo 6. Conclusiones y recomendaciones

Se presentan en este capítulo, las conclusiones obtenidas de acuerdo con los objetivos planteados en el Capítulo 1, además, se brindan las recomendaciones que puedan ayudar a implementar auditorías de seguridad para aplicaciones móviles en PyMEs con políticas de BYOD:

6.1 Conclusiones

Conclusiones del objetivo 1: "Conocer las herramientas disponibles para realizar auditorías en aplicaciones móviles". Este objetivo fue alcanzado y se concluye que:

- Es de suma importancia antes de elegir las herramientas que se van a utilizar para realizar los análisis, tomar en consideración el tipo de dispositivo móvil que se va a auditar (tablet, celular, etc.), el sistema operativo del dispositivo móvil (Android, iOS, etc.), si se cuenta con el presupuesto para adquirir licencias y los resultados que esperan obtener de dichas herramientas.
- Hoy existe una gran variedad de herramientas gratuitas y libres que ayudan a facilitar la tarea de detectar vulnerabilidades y riesgos a través de análisis estáticos y dinámicos de aplicaciones móviles, las cuales se pueden tomar en cuenta en caso de no contar con el presupuesto para adquirir licencias y poder incluirlas en el proceso de auditorías de las PyMEs que cuentan con políticas de BYOD.

- Una vez que se han seleccionado las herramientas y se sabe cuáles dispositivos y sistemas operativos se van a auditar, es necesario realizar pruebas para determinar si las herramientas seleccionadas son las adecuadas para las auditorías de seguridad que desea realizar las PyMEs.

Conclusiones del objetivo 2: "Identificar riesgos y vulnerabilidades actuales de las aplicaciones móviles". Este objetivo fue alcanzado y se concluye que:

- Por medio del análisis estático y dinámico realizado en el caso práctico, se logró identificar una serie de riesgos y vulnerabilidades presentes en dos de las aplicaciones más utilizadas en los dispositivos móviles y a su vez en las PyMEs.
- Identificar los riesgos y vulnerabilidades en las aplicaciones móviles permite que las PyMEs tengan la capacidad de reconocer cuáles son los dispositivos más vulnerables y las áreas de mejora, para así implementar nuevos controles y políticas de seguridad en los dispositivos móviles de BYOD.
- La implementación de controles de gestión de riesgo reduce notablemente los incidentes de seguridad y protegen el activo más importante para las PyMEs, su información.

Conclusiones del objetivo 3: "Investigar los frameworks de seguridad para aplicaciones móviles". Este objetivo fue alcanzado y se concluye que:

- Tras realizar las investigaciones de los frameworks de seguridad de aplicaciones móviles, descubrimos que en el top 10 de OWASP y en el top 25 de CWE, son los más relevantes debido a las altas normas de seguridad que aplican a la hora de desarrollar aplicaciones móviles.
- OWASP proporciona lineamientos más amplios para la seguridad de aplicaciones móviles, por lo que se decidió elegir dicha metodología como referencia para la realización de auditorías de seguridad en PyMEs con políticas de BYOD.
- Proteger las aplicaciones móviles contra los diez riesgos/vulnerabilidades presentadas por el Framework OWASP MASVS, logra un elevado nivel de seguridad y confianza en las PyMEs que cuentan con políticas de BYOD.

Conclusiones del objetivo 4: "Desarrollar una prueba práctica para identificar las vulnerabilidades y riesgos en una aplicación existente". Este objetivo fue alcanzado y se concluye que:

- Para realizar los análisis estáticos y dinámicos de las aplicaciones móviles, es necesario que las PyMEs cuenten con un ambiente donde puedan instalar las herramientas para realizar las auditorías de seguridad. Para efectos del desarrollo

de la prueba práctica de esta investigación, se utilizó Virtual Box de Oracle y se instaló y configuró la herramienta MobSF, OWASP ZAP y Genymotion para poder realizar los análisis.

- Se debe de contar con personal capacitado en el área de ciberseguridad y que además tenga conocimiento de ambientes virtuales, auditorías de seguridad y manejo de sistemas operativos como Windows y Linux.
- Al realizar el análisis de vulnerabilidades de dos de las aplicaciones más utilizadas a nivel mundial, se determina que ambas aplicaciones a pesar de que provienen de una compañía de gran prestigio (Facebook), cuentan con un gran número de vulnerabilidades que pueden poner en riesgo la información de las PyMEs.

Conclusiones del objetivo 5: "Establecer controles para reducir el impacto de los riesgos identificados en aplicaciones móviles". Este objetivo fue alcanzado y se concluye que:

- Una vez identificados cuáles son los principales riesgos y vulnerabilidades presentes en las aplicaciones, es posible recomendar controles que puedan ser implementados en las PyMEs para reducir su impacto.
- Es importante recalcar que implementar los controles o políticas de seguridad a los riesgos encontrados, no eliminará el riesgo por completo; no obstante, reduce la posibilidad de que el riesgo se materialice.
- Los controles o políticas de seguridad que se implementan en las PyMEs para las aplicaciones móviles se deben evaluar cada vez que se liberan nuevas versiones de las mismas, debido a que las nuevas versiones pueden introducir nuevos riesgos, haciendo que los controles actuales ya no sean efectivos.

Conclusiones del objetivo 6: "Recomendar una metodología que se adapte como base para realizar auditorías de seguridad en aplicaciones móviles para las PyMEs con políticas de BYOD". Este objetivo fue alcanzado y se concluye que:

- Con base en los resultados obtenidos de los análisis, se recomienda la implementación de MASVS y MSTG de OWASP en el proceso de auditorías, ya que se adhiere a las mejores prácticas de seguridad de aplicaciones móviles.
- Como parte de la metodología, se recomienda a las pequeñas y medianas empresas (PyME), la implementación de un Mobile Device Management (MDM), el cual es un software que permite controlar, asegurar y reforzar políticas en dispositivos móviles.

6.2 Recomendaciones

Se presentan a continuación, algunas recomendaciones que pueden implementar las PyMEs a la hora de auditar las aplicaciones móviles:

Aunque muchos usuarios ya son conscientes acerca de las amenazas de seguridad debido al uso de las aplicaciones, es necesaria una mayor concientización sobre los riesgos derivados de la descarga de aplicaciones en dispositivos móviles. Entre las recomendaciones, se destacan:

1. Seguir las reglas de seguridad establecidas por los responsables de Tecnologías de Información de la empresa, en cuanto a normas de seguridad en el uso del dispositivo móvil corporativo o bien, buscar el asesoramiento especializado de un experto.
2. Usar siempre la tienda de aplicaciones oficial del dispositivo para descargar las aplicaciones.
3. Revisar qué permisos solicitan las apps al instalarse y que estos sean los apropiados para la función que va a desempeñar.
4. Configurar los niveles de privacidad que la aplicación permite.
5. Revisar la configuración de geolocalización y verificar si es necesario o no que esté siempre activada.
6. En aplicaciones de redes sociales, no abrir enlaces que provengan de usuarios desconocidos, especialmente cuando estos van en forma de enlaces cortos.
7. No compartir contraseñas ni información sensible a través de apps.
8. Mantener actualizado el sistema operativo.
9. Mantenerse informado acerca de las últimas amenazas existentes.
10. Tener en cuenta que lo que se comparte por una red social queda permanentemente compartido.

Capítulo 7. Reflexiones finales

La posibilidad en la era digital, de interconectarse, conversar, opinar e interactuar, es más fácil de lo que se cree, basta con tener a la mano un dispositivo digital como una tablet, un teléfono móvil o una computadora, herramientas a las que se puede adquirir fácilmente y las cuales se han convertido en una necesidad para desarrollar actividades laborales, académicas e incluso familiares.

Se sabe que hay un cambio en la forma cómo el cliente final está adquiriendo y contratado productos y servicios, esto ha conllevado que las empresas hagan grandes esfuerzos por llegar al siguiente nivel, la era de la movilidad digital. Es muy frecuente ver como miles de empresas, hoy ya generan valor a su negocio, ofertando sus servicios a través de aplicaciones móviles y es en este punto en el que vemos que del otro lado del negocio, el del cibercrimen, aprovecha para apuntar a su siguiente objetivo, las apps. Se nota recientemente que ya los hackers están volteando a mirar y fijar su interés en la vulneración de las aplicaciones móviles, que hoy por hoy, están más expuestas que los tradicionales servicios web.

Ante este panorama, surge el reto de proteger la información, pues usar las tecnologías, como se mencionó anteriormente, ya es del día a día de cada persona, por eso, es muy común hacer compras en línea, compartir datos, leer, escuchar música, conectarse a una red desde un lugar público y un sinfín de actividades que parecen inofensivas de realizar en el mundo digital, pero que puede ser vulnerable y de fácil acceso para los cibercriminales.

Es de gran importancia la adopción de un framework de ciberseguridad para apps, como lo puede ser el referido “OWASP Mobile Security top 10”, un marco de control preventivo que ayuda a los desarrolladores y en especial al equipo de ciberseguridad, en la evaluación de un top 10 de amenazas más importantes de las aplicaciones, para así reducir el riesgo de impacto o de materialización de incidente de seguridad del aplicativo, así también como establecer en la organización técnicas o buenas prácticas para el desarrollo seguro en aplicaciones móviles, los cuales se traducirán de cara al cliente en una mejor calidad el servicio y de cara al negocio en el crecimiento de la oferta de productos o servicios seguros y confiables.

Capítulo 8. Trabajos a futuro

se detallan a continuación, tres trabajos a futuro que pueden surgir a través de la investigación realizada en esta tesis.

- **Implementar el trabajo investigación en una PyMEs de Costa Rica:** debido a que la metodología no pudo implementarse como parte de esta tesis, queda como trabajo a futuro implementar la auditoría de dispositivos móviles en una PyME y así poder probar si es efectiva.
- **Análisis de aplicaciones móviles para la plataforma iOS:** tomando como base lo analizado en esta tesis y principalmente, las vulnerabilidades y taxonomía de malas prácticas, se puede realizar un trabajo similar para analizar la seguridad de

aplicaciones móviles para la plataforma iOS de Apple. Dentro de las ventajas en el desarrollo de una investigación así está, por ejemplo, el hecho de que al ser una plataforma estandarizada, casi no existe fragmentación, lo cual permite estudiar más rápido posibles vulnerabilidades que pudieran tener las aplicaciones de iOS.

- **Ampliar el alcance de la auditoría a niveles L2 y R de MASVS:** ya que el nivel elegido para realizar la auditoría de acuerdo con MASVS fue L1, basado en las aplicaciones elegidas para el estudio, podría ser de gran utilidad ampliar el alcance de la auditoría a niveles L2 y R para otras aplicaciones, que por su nivel de tratamiento de datos sensibles o realización de operación, así lo requieran.

Glosario

A

- **Android Application Package (APK):** son archivos ejecutables para Android
- **Aplicaciones (APP/App):** una aplicación es un programa informático diseñado como una herramienta para realizar operaciones o funciones específicas.
- **Application Programming Interface (APIs):** no es más que un conjunto de reglas que describen como una aplicación puede interactuar con otra.

B

- **Black-Box:** prueba de caja negra, se define como una técnica de prueba en la que la funcionalidad de la aplicación se prueba sin mirar la estructura del código interno.

C

- **Common Vulnerability Exposure (CVE):** es una entrada en la lista de vulnerabilidades, es decir, es una instancia específica de una debilidad en un producto o sistema.
- **Common Weakness Enumeration (CWE):** es una entrada en la base de datos correspondiente a una debilidad y no está relacionada con un producto o sistema.
- **Common Vulnerability Scoring System (CVSS):** sistema de puntaje diseñado para proveer un método abierto y estándar que permite estimar el impacto derivado de vulnerabilidades
- **Cross-Origin Resource Sharing (CROS):** define una forma para que las aplicaciones web cliente que se cargan en un dominio interactúen con recursos en un dominio diferente.

D

- **DevSecOps:** no es solo un método que se basa en la integración de la seguridad en el proceso DevOps
- **Dynamic Application Security Testing (DAST):** es una metodología de prueba de seguridad de caja negra en la que una aplicación se prueba desde el exterior.

F

- **Framework:** es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos que puede servir de base para la organización y desarrollo de software.
- **Fuzzing:** es una técnica de prueba de software automatizada que implica proporcionar datos no válidos, inesperados o aleatorios como entradas a un programa de computadora.

H

- **HyperText Markup Language (HTML):** es un lenguaje de marcado de hipertexto, este lenguaje se escribe en su totalidad con elementos que están constituidos por etiquetas, contenido y atributos

I

- **Instituto Nacional de Ciberseguridad de España (INCIBE):** 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.

K

- **Kernel:** es la parte central de un sistema operativo y es el que se encarga de realizar toda la comunicación segura entre el software y el hardware

M

- **Malware (“mal**icious soft**ware”):** es cualquier software diseñado intencionalmente para causar daños a una computadora, servidor, cliente o red informática
- **Mobile Application Security Verification Standard (MASVS):** un esfuerzo comunitario para establecer un marco de requisitos de seguridad necesarios para diseñar, desarrollar y probar aplicaciones móviles seguras en iOS y Android.
- **Mobile Security Framework (MobSF):** es un entorno completo de análisis que incluye funcionalidad para la realización de pruebas tanto estáticas como dinámicas.
- **Mobile Security Testing Guide (MSTG):** es un manual completo para pruebas de seguridad de aplicaciones móviles e ingeniería inversa.

O

- **Open Web Application Security Project OWASP:** es una comunidad en línea que produce artículos, metodologías, documentación, herramientas y tecnologías disponibles gratuitamente en el campo de la seguridad de aplicaciones web
- **Operative System (OS):** es un conjunto de programas de bajo nivel que permite la abstracción de las peculiaridades del hardware específico del teléfono móvil y provee servicios a las aplicaciones móviles, que se ejecutan sobre él

P

- **Pequeña y mediana empresa (PyMEs):** se trata de la empresa mercantil, industrial o de

otro tipo que tiene un número reducido de trabajadores y que registra ingresos moderados:

S

- **Secure Sockets Layer (SSL):**es una tecnología de seguridad estándar para establecer un enlace cifrado entre un servidor y un cliente
- **Smartphones:** es un teléfono celular con pantalla táctil y un robusto sistema operativo con el que los usuarios pueden conectarse a Internet, instalar aplicaciones y llevar a cabo muchas de las actividades que podrían realizar en una computadora.
- **Software Development Life Cycle (SDLC):** es un proceso utilizado por la industria del software para diseñar, desarrollar y probar software de alta calidad.
- **Static Application Security Testing (SAST):** es un conjunto de tecnologías diseñadas para analizar el código fuente de la aplicación, el código de bytes y los binarios para las condiciones de codificación y diseño que son indicativas de vulnerabilidades de seguridad.
- **Structured Query Language (SQL):** es un tipo de lenguaje de programación que ayuda a solucionar problemas específicos o relacionados con la definición, manipulación e integridad de la información representada por los datos que se almacenan en las bases de datos.

T

- **Tampering:** se refiere a la modificación desautorizada a los datos o al software instalado en un sistema, incluyendo borrado de archivos
- **Transport Layer Security (TLS):** o seguridad en capas de transporte: el protocolo criptográfico que garantiza las comunicaciones en Internet.

W

- **Web app:** es un software de aplicación que se ejecuta en un servidor web,

---

X

- **Xtensible Markup Language (XML):** es un lenguaje de marcado que define un conjunto de reglas para codificar documentos en un formato que es legible por humanos y legible por máquina.

Referencias bibliográficas

- A2Secure (2019), Herramientas de prueba de seguridad de aplicaciones (AST). A2Secure. https://www.a2secure.com/blog/herramientas-de-prueba-de-seguridad-de-aplicaciones-ast/

- Andaluciasmart (2016). Vectores de ataque. Andaluciaesdigital. https://www.andaluciaesdigital.es/documents/20182/345997/Informe.-+Vectores+de+ataque+dirigidos+a+la+Smart+City.pdf/a0c1786c-6bcd-4b7a-b1a0-fffd7b9f7b9b

- Blanco, C., Lasheras, J., Valencia-García, R., Fernández-Medina, E., Toval, A., & Piattini, M. (2007). Ontologías de seguridad: revisión sistemática y comparativa. Researchgate. https://www.researchgate.net/publication/229083904_Ontologias_de_seguridad_revision_sistematica_y_comparativa

- Cyber Noobing (2019). The OWASP Top 10 – A Brief Overview. Cyber Noobing. https://cybernoobing.com/2019/07/28/the-owasp-top-10-brief-overview-10-6/

- Denise Giusto (2016). Cómo analizar archivos APK con MobSF. Welivesecurity. https://www.welivesecurity.com/la-es/2016/12/19/analizar-apk-con-mobsf/

- Denise Giusto (2017). Cómo analizar apps de Android con Inspeckage. Welivesecurity. https://www.welivesecurity.com/la-es/2017/10/20/analizar-apps-android-inspeckage/

- Fabian S. (2013). Qué es Windows Phone. Microsoft. https://answers.microsoft.com/es-es/mobiledevices/forum/all/qu%C3%A9-es-windows-phone/583642b0-9025-433c-a089-c9feb37eabeb

- Fernando Saavedra (2019). OWASP Top Ten Mobile Risks. Audea https://www.audea.com/owasp-top-ten-mobile-risks/

- Google Developers. (2020). Run apps on the Android Emulator. Google. https://developer.android.com/studio/run/emulator

- Guillermo Westreicher (2016). PyME – Pequeña y mediana empresa. Economipedia. https://economipedia.com/definiciones/PyME.html

- INCIBE. (2017). Instituto Nacional de Ciberseguridad de España. Amenaza vs Vulnerabilidad, ¿sabes en qué se diferencian? INCIB. https://www.incibe.es/protege-tu-empresa/blog/amenaza-vs-vulnerabilidad-sabes-se-diferencian.

- Juan Garzón (2020). Android 11 a Android 1.5: Cada versión de Android y sus novedades. Cnet. https://www.cnet.com/es/imagenes/android-11-novedades-actualizacion-android-historia-google/\

- LanceTalent (2019). Los 3 Tipos de Aplicaciones Móviles: Ventajas e Inconvenientes. LanceTalent. https://www.lancetalent.com/blog/tipos-de-aplicaciones-moviles-ventajas-inconvenientes/

- Matthias Kadenbach (2018). OWASP Top 10 compared to SANS CWE 25. Templarbit. https://www.templarbit.com/blog/2018/02/08/owasp-top-10-vs-sans-cwe-25/

- Milan Lavín. (2013). ¿Qué es BYOD?, ventajas e inconvenientes. Computerhoy. https://computerhoy.com/noticias/moviles/que-es-byod-ventajas-e-inconvenientes-7250

- Newzoo. (2020). Informe global del mercado móvil de Newzoo. Newzoo. https://newzoo.com/insights/articles/newzoo-global-mobile-market-report-samsung-is-leading-smartphone-brand-with-859-million-active-devices/

- Ostec (2019). Primeros pasos para realizar un Análisis de Vulnerabilidad en redes corporativas. Ostec. https://ostec.blog/es/generico/primeros-pasos-para-realizar-un-analisis-de-vulnerabilidad-en-redes-corporativas

- OWASP (2019). OWASP Mobile Application Security Verification Standard. OWASP. https://github.com/OWASP/owasp-masvs

- OWASP (2020). OWASP Foundation. Code Injection. OWASP. https://owasp.org/www-community/attacks/Code_Injection#

- OWASP (2020). OWASP Mobile Security Testing Guide. OWASP. https://github.com/OWASP/owasp-mstg/

- Redeszone (2020). Proyecto OWASP: Descubre qué es y para qué sirve. Redeszone. https://www.redeszone.net/tutoriales/seguridad/que-es-owasp-seguridad-aplicaciones/

- Roberto Adeva (2020). Qué es Android: todo sobre el sistema operativo de Google. Adslzone. https://www.adslzone.net/reportajes/software/que-es-android/

- Rocío García (2020). Qué es iOS. Adslzone https://www.adslzone.net/reportajes/software/que-es-ios/

- Srinivas Kedarisetty (2017). Application Security Standards. CastSoftware. https://www.castsoftware.com/blog/application-security-standards#:~:text=Application%20security%20standards%20are%20established,vulnerabilities%20in%20complex%20software%20systems.

- T. R. Gruber et al., A translation approach to portable ontology specifications, Knowledge acquisition, vol. 5, no. 2, pp. 199-221, 1993.

- TecnologíaFácil (2016). Qué es Android. TecnologíaFácil. https://tecnologia-facil.com/que-es/que-es-android/

- TrustRadius (2020). Static Application Security Testing (SAST) Tools. TrustRadius. https://www.trustradius.com/dynamic-application-security-testing-dast

- TrustRadius (2020). Static Application Security Testing (SAST) Tools. TrustRadius. https://www.trustradius.com/static-application-security-testing-sast

- Yi Min Shum (2020). Situación Global Mobile 2020. Yiminshum. https://yiminshum.com/mobile-movil-app-2020/

- Yuban NL. VirtualBox. Xataka. https://www.xataka.com/basics/virtualbox-que-como-usarlo-para-crear-maquina-virtual-windows-u-otro-sistema-operativo

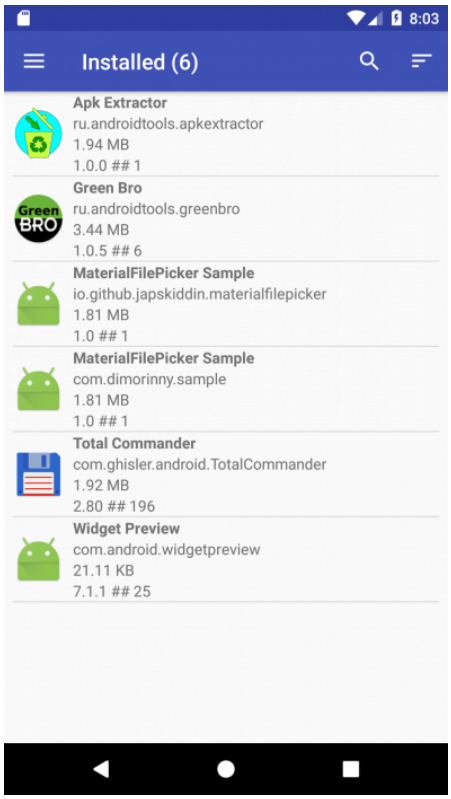
Apéndices

Apéndice 1 Resultados de los análisis del caso práctico.

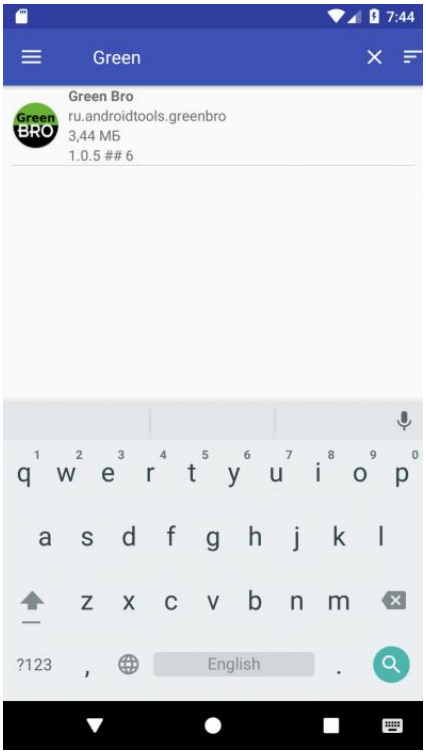
Los resultados de los análisis de vulnerabilidades pueden ser encontrados también en: https://drive.google.com/file/d/1y5Dl_y-wch3-hWU6_Qbl-TPNZRCiaYn3/view?usp=sharing

Apéndice 2 -Como utilizar de APK Extractor

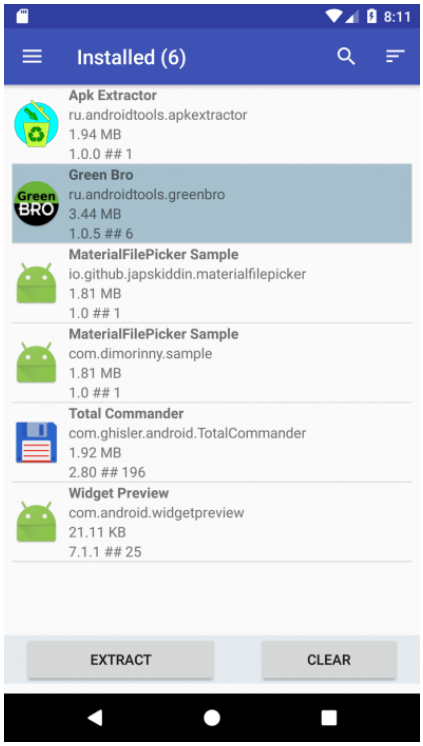
1. Cuando inicie la aplicación, verá una lista de todas las aplicaciones que están instaladas en su dispositivo, incluido el sistema. El cambio entre las aplicaciones instaladas y del sistema puede hacerse a través del menú lateral.



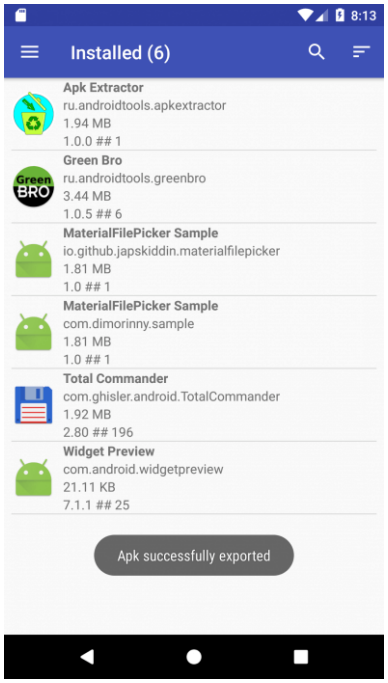
1. Si conoce el nombre de la aplicación que necesita, puede escribir su nombre en la barra de búsqueda, haga clic en el icono de la lupa.



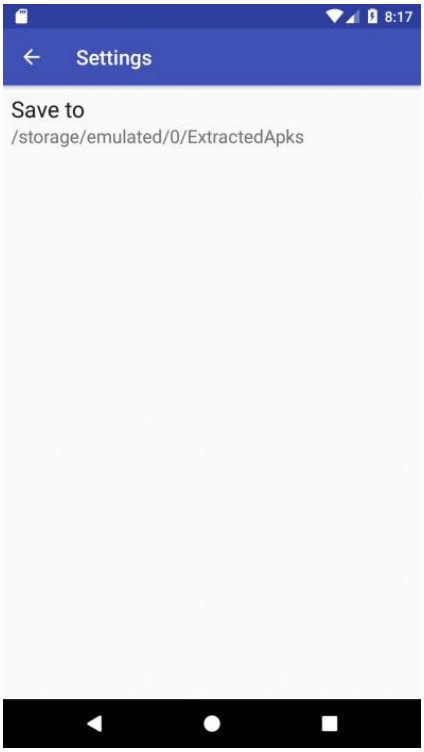
2. Para extraer el APK, haga clic en la aplicación deseada de la lista. Habrá dos botones "Extraer" y "Borrar". El botón "Borrar" eliminará la selección de la lista.



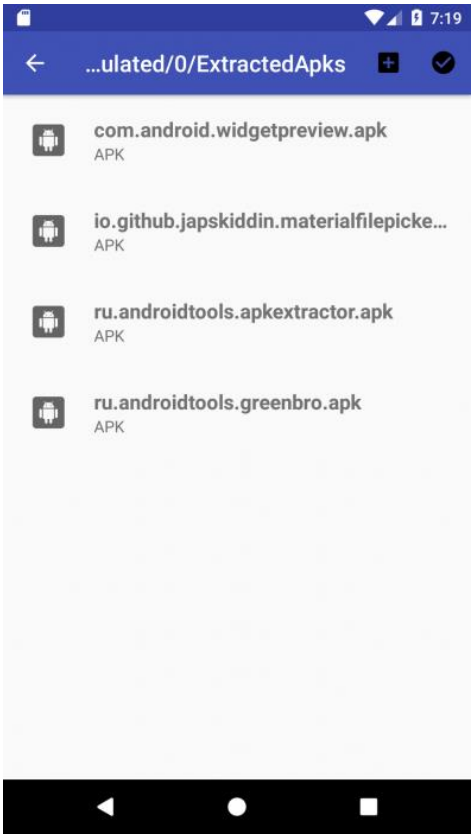
3. Cuando haga clic en "Extraer", se iniciará el proceso de extracción de APK, después de lo cual verá un mensaje emergente sobre un resultado positivo o negativo.



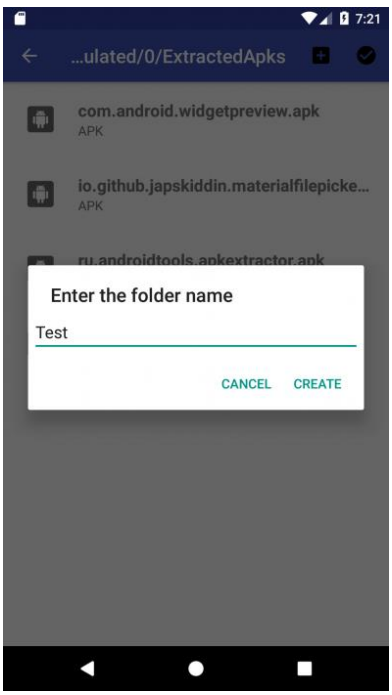
4. Por defecto, los APK se almacenan en un almacenamiento externo en la carpeta ExtractedApks. Para cambiar la ruta, debe abrir la "Configuración" en el menú lateral.

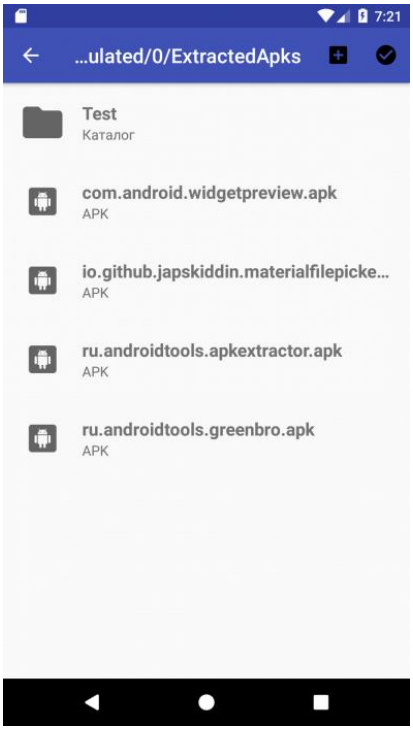


5. Después de hacer clic en "Guardar en", se abrirá una ventana con la selección de la carpeta para guardar.

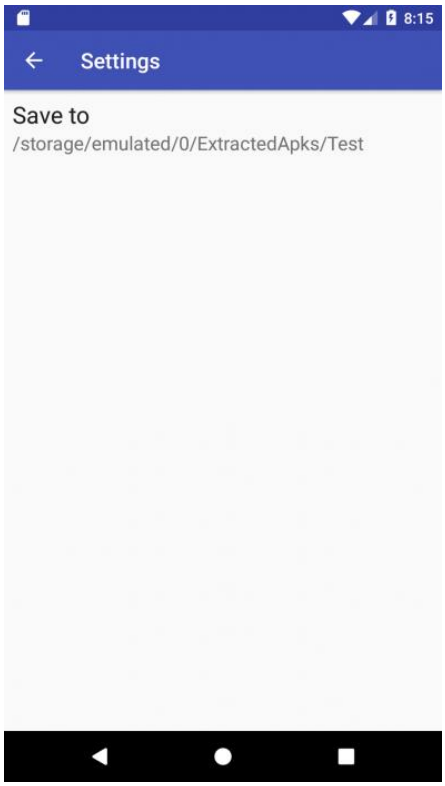


6. Aquí puede seleccionar una carpeta existente y hacer clic en la marca de verificación para guardar la ruta, o crear una nueva carpeta nueva haciendo clic en el signo más.





7. Ahora, al hacer clic en la marca de verificación, volverá a la aplicación y la nueva ruta se guardará en la configuración.



Apéndice 3- Configurando MobSF en Ubuntu 18.04

Tome en cuenta que esta es una guía para configurar MobSF para el análisis estático. Para el análisis dinámico, puede consultar la página MobSF en github.

https://github.com/MobSF/Mobile-Security-Framework-MobSF

Requerimientos:

- Git
- Python 3.6+
- JDK 8+

Paso a Paso:

1. Después de instalar un Ubuntu 18.04 nuevo, instale git.

   Comando: *sudo apt-get install git*

```
ask@ubuntu:~$ sudo apt-get install git
[sudo] password for ask:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 130 not upgraded.
Need to get 4,733 kB of archives.
After this operation, 33.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

2. Hacer apt update

   Comando: *sudo apt update*

```
ask@ubuntu:~$ sudo apt update
[sudo] password for ask:
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
130 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

3. Instalar python3-pip

   Comando: *sudo apt install python3-pip*

```
ask@ubuntu:~$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  build-essential dh-python dpkg-dev fakeroot g++ g++-7 gcc gcc-7
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl
  libasan4 libatomic1 libc-dev-bin libc6-dev libcilkrts5 libdpkg-perl
  libexpat1 libexpat1-dev libfakeroot libgcc-7-dev libitm1 liblsan0 libmpx2
  libpython3-dev libpython3.6 libpython3.6-dev libpython3.6-minimal
  libpython3.6-stdlib libquadmath0 libstdc++-7-dev libtsan0 libubsan0
  linux-libc-dev make manpages-dev python-pip-whl python3-dev
  python3-distutils python3-lib2to3 python3-setuptools python3-wheel python3.6
  python3.6-dev python3.6-minimal
Suggested packages:
  debian-keyring g++-multilib g++-7-multilib gcc-7-doc libstdc++6-7-dbg
  gcc-multilib autoconf automake libtool flex bison gcc-doc gcc-7-multilib
  gcc-7-locales libgcc1-dbg libgomp1-dbg libitm1-dbg libatomic1-dbg
  libasan4-dbg liblsan0-dbg libtsan0-dbg libubsan0-dbg libcilkrts5-dbg
  libmpx2-dbg libquadmath0-dbg glibc-doc bzr libstdc++-7-doc make-doc
  python-setuptools-doc python3.6-venv python3.6-doc binfmt-support
```

4. Instalar jdk. Verifique la versión después de la instalación para volver a confirmarla.

Comando: *sudo apt install default-jre*

```
ask@ubuntu:~$ sudo apt install default-jre
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java default-jre-headless fonts-dejavu-extra java-common
  libatk-wrapper-java libatk-wrapper-java-jni libgif7 openjdk-11-jre
  openjdk-11-jre-headless
Suggested packages:
  fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
  | fonts-wqy-zenhei
The following NEW packages will be installed:
  ca-certificates-java default-jre default-jre-headless fonts-dejavu-extra
  java-common libatk-wrapper-java libatk-wrapper-java-jni libgif7
  openjdk-11-jre openjdk-11-jre-headless
0 upgraded, 10 newly installed, 0 to remove and 123 not upgraded.
Need to get 39.5 MB of archives.
After this operation, 178 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

```
ask@ubuntu:~$ java -version
openjdk version "11.0.4" 2019-07-16
OpenJDK Runtime Environment (build 11.0.4+11-post-Ubuntu-1ubuntu218.04.3)
OpenJDK 64-Bit Server VM (build 11.0.4+11-post-Ubuntu-1ubuntu218.04.3, mixed mod
e, sharing)
```

5. Instalar el paquete python.

Comando: *sudo apt install python3-venv python3-pip python3-dev build-essential \ libffi-dev libssl-dev libxml2-dev libxslt1-dev libjpeg8-dev zlib1g-dev*

```
ask@ubuntu:~$ sudo apt install python3-venv python3-pip python3-dev build-essent
ial libffi-dev libssl-dev libxml2-dev libxslt1-dev libjpeg8-dev zlib1g-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.4ubuntu1).
build-essential set to manually installed.
python3-dev is already the newest version (3.6.7-1~18.04).
python3-dev set to manually installed.
python3-pip is already the newest version (9.0.1-2.3~ubuntu1.18.04.1).
The following additional packages will be installed:
  gir1.2-harfbuzz-0.0 icu-devtools libglib2.0-dev libglib2.0-dev-bin
  libgraphite2-dev libharfbuzz-dev libharfbuzz-gobject0 libicu-dev
  libicu-le-hb-dev libicu-le-hb0 libiculx60 libjpeg-turbo8-dev libpcre16-3
  libpcre3-dev libpcre32-3 libpcrecpp0v5 pkg-config python3.6-venv
```

6. Clone MobSF

   Comando: *git clone https://github.com/MobSF/Mobile-Security-Framework-MobSF.git*

```
ask@ubuntu:~$ git clone https://github.com/MobSF/Mobile-Security-Framework-MobSF
.git
Cloning into 'Mobile-Security-Framework-MobSF'...
remote: Enumerating objects: 142, done.
remote: Counting objects: 100% (142/142), done.
remote: Compressing objects: 100% (110/110), done.
remote: Total 13999 (delta 50), reused 101 (delta 31), pack-reused 13857
Receiving objects: 100% (13999/13999), 348.54 MiB | 1.04 MiB/s, done.
Resolving deltas: 100% (6207/6207), done.
Checking out files: 100% (363/363), done.
```

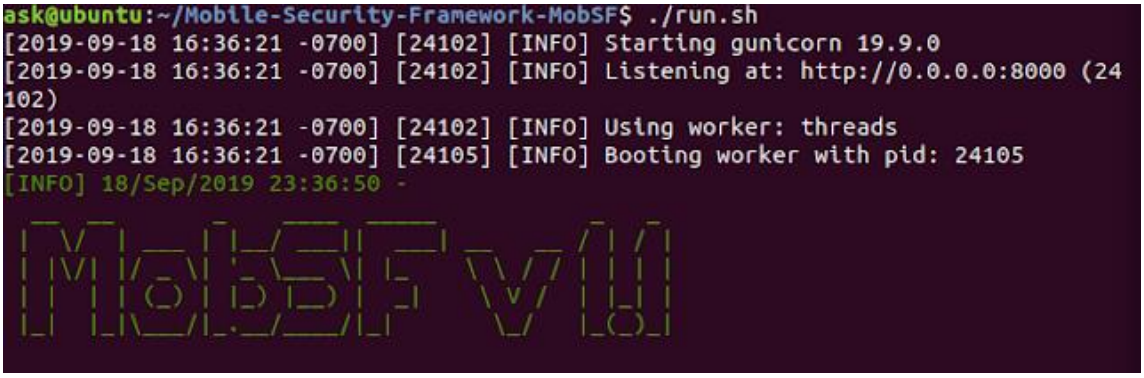
7. Corra el comando setup.sh. Espere a que se complete la instalación. Tomará un poco de tiempo.

   Comando: *./setup.sh*

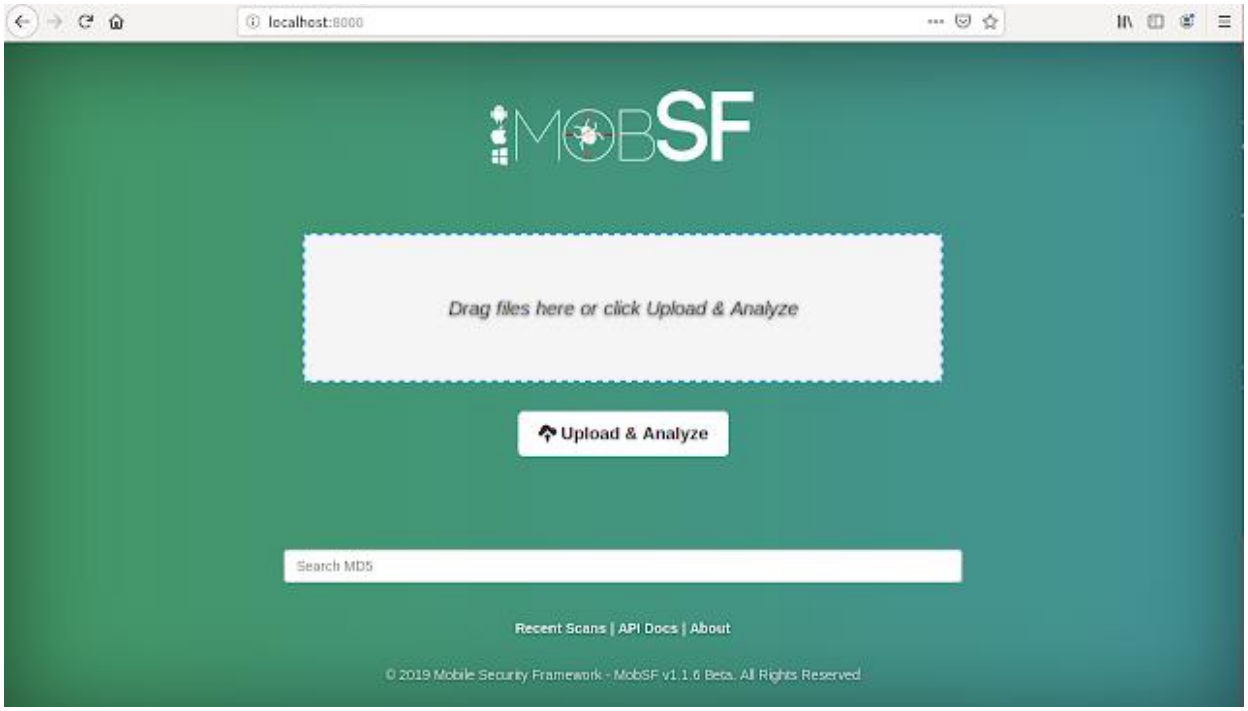
```
ask@ubuntu:~$ cd Mobile-Security-Framework-MobSF/
ask@ubuntu:~/Mobile-Security-Framework-MobSF$ ./setup.sh
[INSTALL] Found Python3
pip 9.0.1 from /usr/lib/python3/dist-packages (python 3.6)
[INSTALL] Found pip
Collecting pip
  Downloading https://files.pythonhosted.org/packages/30/db/9e38760b32e3e7f40cce
46dd5fb107b8c73840df38f0046d8e6514e675a1/pip-19.2.3-py2.py3-none-any.whl (1.4MB)
    100% |                                | 1.4MB 628kB/s
Installing collected packages: pip
Successfully installed pip-19.2.3
[INSTALL] Using venv
[INSTALL] Installing APKiD requirements - yara-python
Collecting wheel
```

8. Corra MobSF

   Comando: *./run.sh*



9. Navegue a http: // localhost: 8000 y habrá instalado con éxito MobSF.



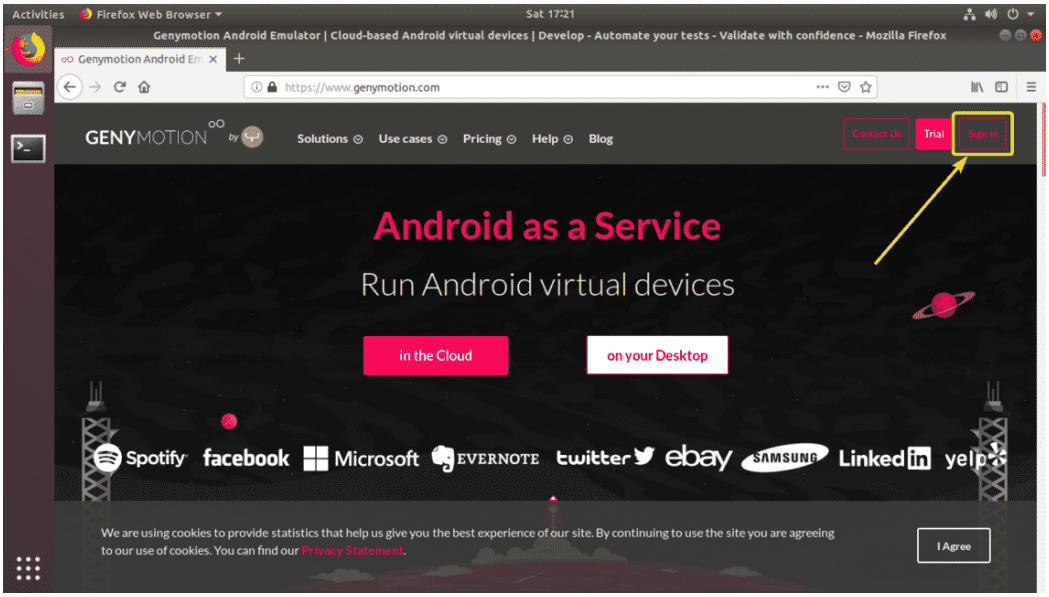
Apéndice 4 – Genymotion

Descarga e instalación de Genymotion

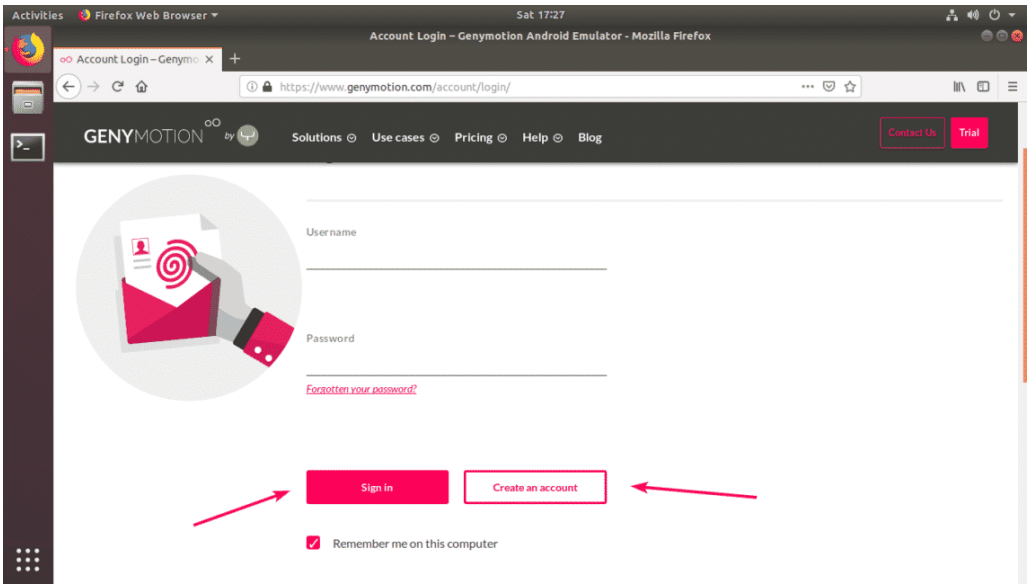
Descarga de Genymotion

Genymotion no está disponible en el repositorio oficial de paquetes de Ubuntu 18.04 LTS. Pero puedes descargar Genymotion fácilmente desde el sitio web oficial de Genymotion e instalarlo en tu máquina Ubuntu 18.04 LTS.

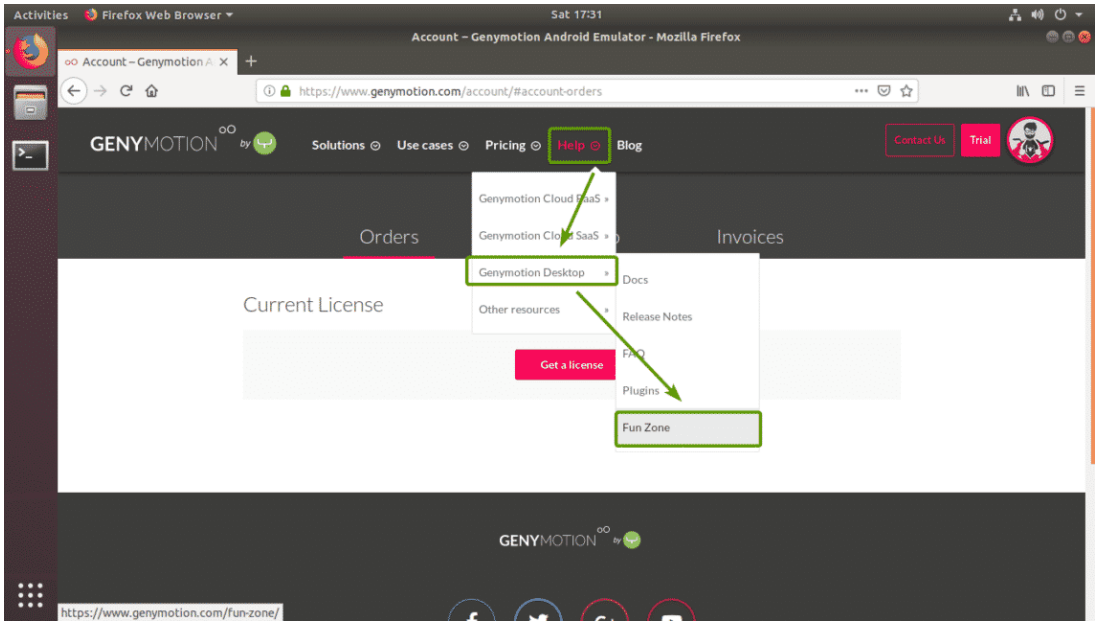
1. Primero, vaya al sitio web oficial de Genymotion en https://www.genymotion.com desde su navegador web favorito y haga clic en Iniciar sesión.



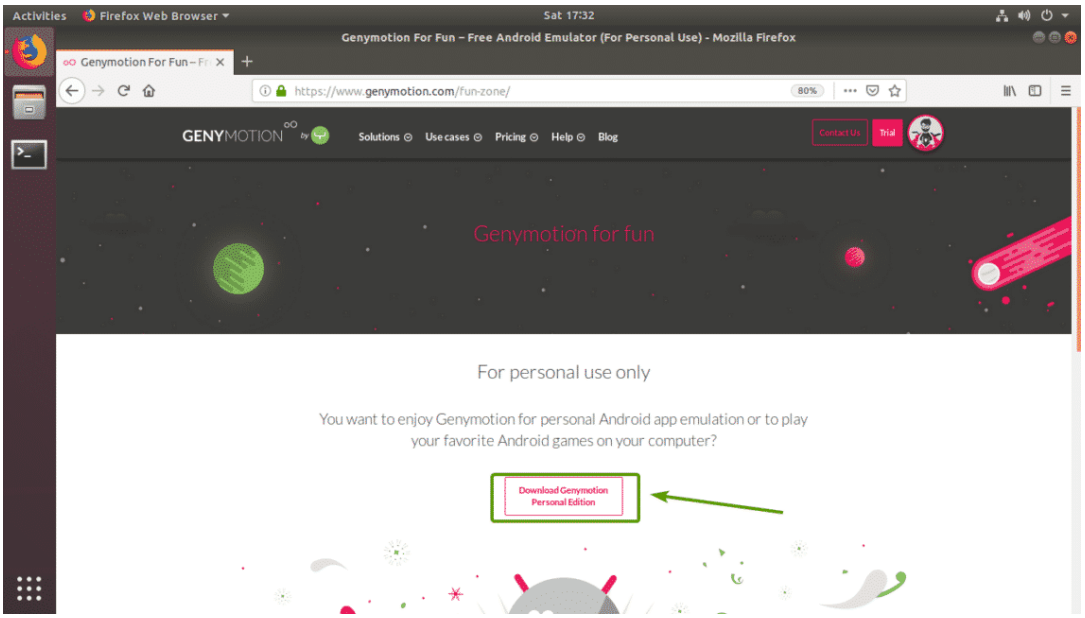
2. Si tiene una cuenta Genymotion, simplemente inicie sesión con su cuenta. Si no tiene una, simplemente haga clic en Crear una cuenta, cree una nueva cuenta de Genymotion e inicie sesión.



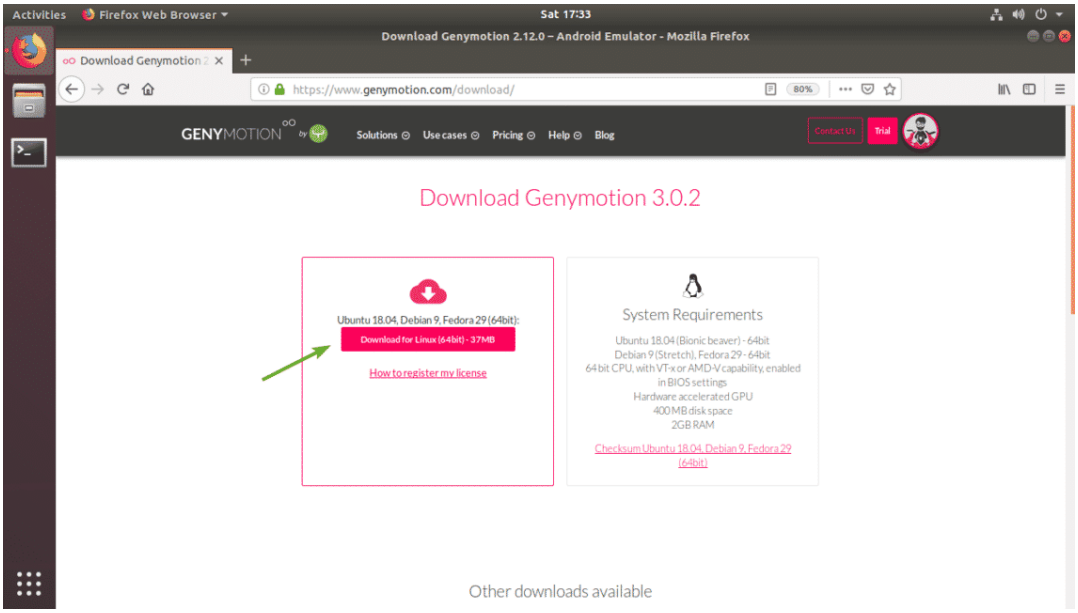
3. Una vez que haya iniciado sesión, vaya a Ayuda> Genymotion Desktop> Fun Zone como se indica en la captura de pantalla a continuación.



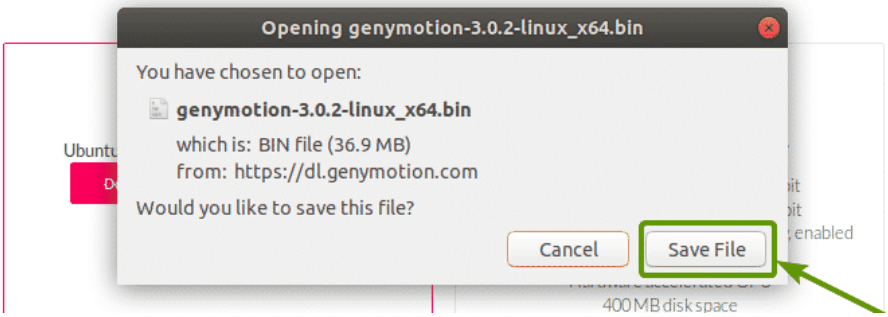
4. Ahora, haga clic en el botón Descargar Genymotion Personal Edition como se indica en la captura de pantalla a continuación.



5. Ahora, haga clic en el botón Descargar para Linux (64 bits) como se indica en la captura de pantalla a continuación.



6. Su navegador debería pedirle que guarde el archivo de instalación de Genymotion. Haga clic en Guardar archivo para guardarlo.

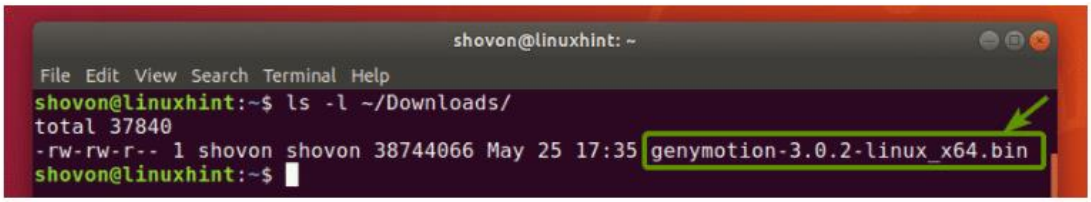


7. Su navegador debería comenzar a descargar el instalador de Genymotion.

Instalación de Genymotion

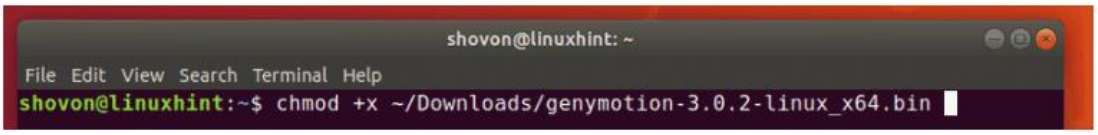
1. Una vez descargado el instalador de Genymotion, debería poder encontrarlo en el directorio ~ / Downloads /.

```
$ ls -l ~/Downloads/
```



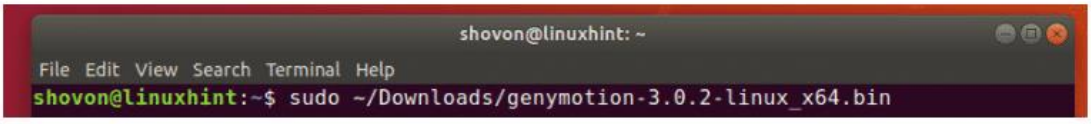
2. Ahora, haga que el instalador sea ejecutable con el siguiente comando:

```
$ chmod +x ~/Downloads/genymotion-3.0.2-linux_x64.bin
```

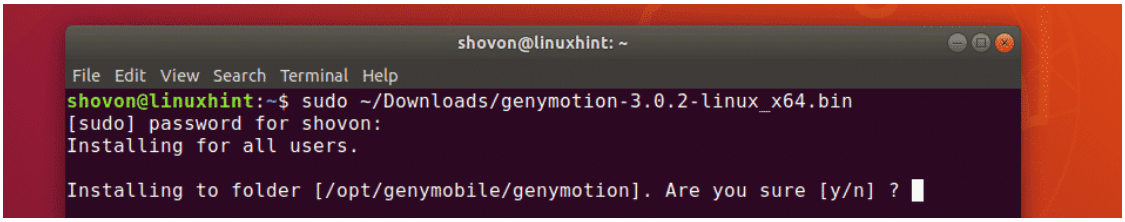


3. Ahora, ejecute el instalador de Genymotion con el siguiente comando

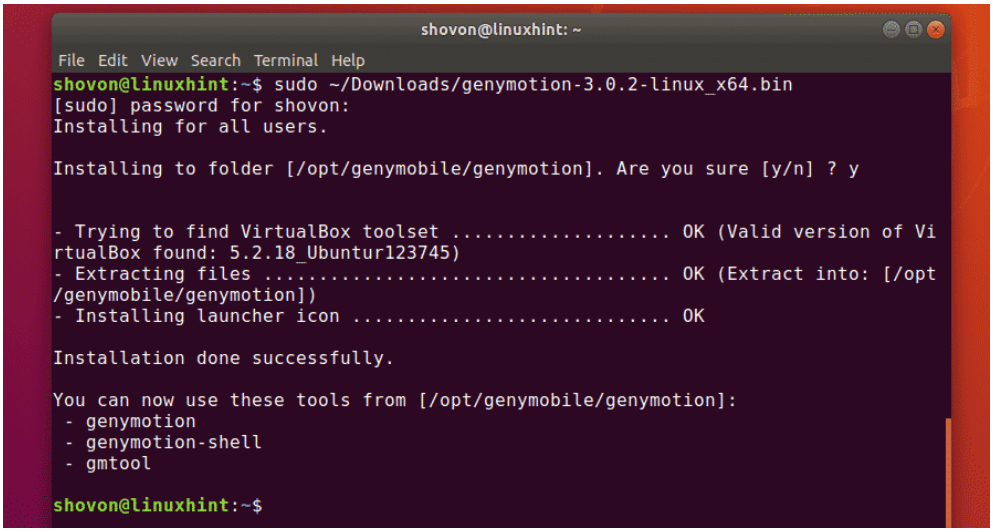
```
$ sudo ~/Downloads/genymotion-3.0.2-linux_x64.bin
```



4. Ahora, presione “y” y luego presione <Enter> para continuar.



5. Debe instalarse Genymotion. \



Apéndice 5 - OWASP ZAP Proxy

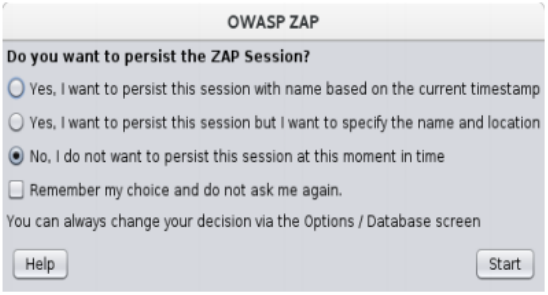
Instalación y configuración de OWASP ZAP

Instalación

1. Lo primero que debe hacer es instalar ZAP en el sistema en el que desea realizar el pentesting.
2. Descargue el instalador apropiado de la ubicación de descarga de ZAP en https://www.zaproxy.org/download/ y ejecute el instalador
   Nota: Tenga en cuenta que ZAP requiere Java 8+ para funcionar. El instalador de Mac OS / X incluye una versión apropiada de Java, pero debe instalar Java 8+ por separado para Windows, Linux. Las versiones de Docker no requieren instalar Java.
3. Una vez que se complete la instalación, inicie ZAP y lea los términos de la licencia.
4. Haga clic en Aceptar si acepte los términos y ZAP terminará de instalarse, luego ZAP se iniciará automáticamente.

Persistencia de una sesión

1. Cuando inicie ZAP por primera vez, se le preguntará si desea mantener la sesión de ZAP. Por defecto, Las sesiones ZAP siempre se graban en el disco en una base de datos HSQLDB con un nombre predeterminado y ubicación. Si no persiste la sesión, esos archivos se eliminan cuando sale de ZAP.
2. Si opta por mantener una sesión, la información de la sesión se guardará en la base de datos local. Para que pueda acceder a él más tarde, y podrá proporcionar nombres y ubicaciones personalizados para guardar los archivos.



Por ahora, seleccione No, no quiero persistir en esta sesión en este momento, luego haga clic en Iniciar. Las sesiones de ZAP no se mantendrán por ahora.

Interfaz de Usuario de ZAP Escritorio

La interfaz de usuario de ZAP Desktop se compone de los siguientes elementos:

1. **Barra de menú:** proporciona acceso a muchas de las herramientas automáticas y manuales.

2. **Barra de menú:** proporciona acceso a muchas de las herramientas automáticas y manuales.
3. **Ventana de árbol:** muestra el árbol de Sitios y el árbol de Scripts.
4. **Ventana del área de trabajo:** muestra solicitudes, respuestas y scripts y le permite editarlos.
5. **Ventana de información:** muestra detalles de las herramientas automáticas y manuales.
6. **Pie de página:** muestra un resumen de las alertas encontradas y el estado de las herramientas automatizadas.

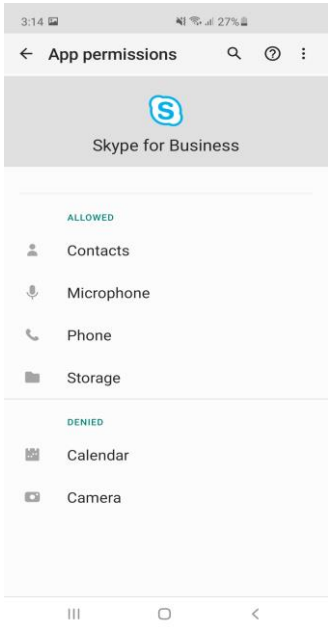
Apéndice 6– Otras aplicaciones analizadas

Recopilación de información de las aplicaciones a auditar.

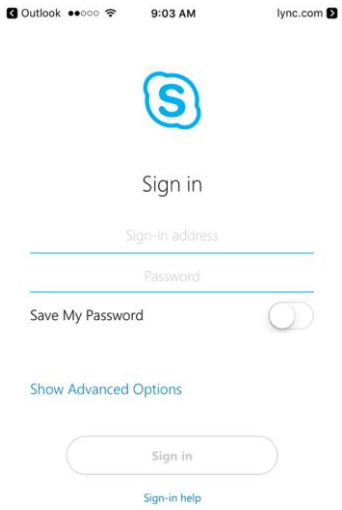
- Skype for Business se utiliza para realizar principalmente llamadas de voz y videoconferencia. Ofrece llamadas de videoconferencia que permite admitir hasta 250 personas.
- Microsoft Teams se utiliza como una plataforma de colaboración integral, que ofrece funciones de chat, voz y llamadas.

Skype for Business

- **Funcionalidad:** ofrece una gran experiencia de conferencia en línea con las características de audio y vídeo, pantalla compartida y facilidad de uso.
- **Permisos requeridos para la instalación:** contactos, micrófono, llamadas, calendario, cámara, almacenamiento.



- **Autenticación:** autenticación basada en certificados, autenticación multifactor, autenticación moderna, administración de aplicaciones móviles (a través de Intune).

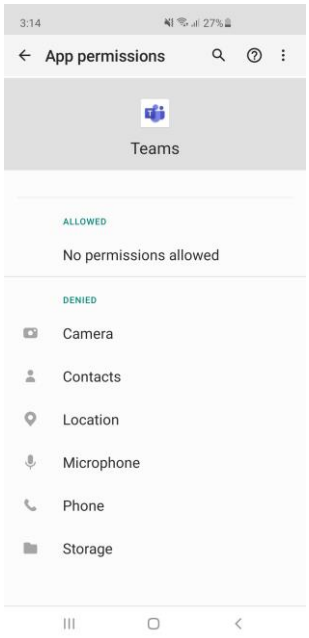


- **Componentes hardware:** micrófono, cámara, almacenamiento, bluetooth.

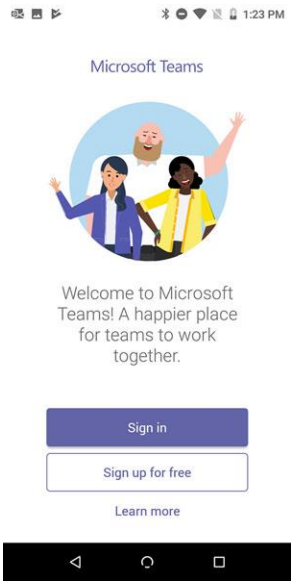
- **Conexiones de red**: datos móviles / Wi-Fi

Microsoft Teams:

- **Funcionalidad:** Microsoft Teams es su centro de trabajo en equipo, que reúne todo lo que un equipo necesita: conversaciones de chat y subprocesos, reuniones y videoconferencias, llamadas, colaboración de contenido con el poder de las aplicaciones de Microsoft 365.
- **Permisos requeridos para la instalación:** Contactos, Micrófono, Llamadas, Calendario, Cámara, Almacenamiento.



- **Autenticación:** Single Sign On, autenticación moderna.



- **Componentes hardware:** micrófono, cámara, GPS, almacenamiento.
- **Conexiones de red**: datos móviles / Wi-Fi

Adjunto resultados de ambos escaneos estáticos.

Apéndice 7 – Encuesta realizada a PYMES sobre el interés de aplicar la metodología.