Actividad 2.5

Informe de Vulnerabilidades

Nombre: Diego Bastián Rojo Peralta

Asignatura: Programación Android

Docente: Oscar Monardez

Fecha: 23-10-2025

Herramienta Utilizada: MobSF Live

Índice:

1. Introducción.

En el presente informe presenta los resultados del análisis de seguridad realizado a la aplicación Android desarrolla a través de las clases, la cual contempla la parte más importante que es la implementación del funcionamiento de un mapa en tiempo real, como parte de la actividad 2.5 de la asignatura Programación Android. El objetivo principal de la actividad/evaluación es identificar vulnerabilidades relevantes mediante herramientas de análisis estático, clasificarlas por su severidad y proponer medidas de mitigación/eliminación que permitan el fortalecimiento la seguridad de la aplicación.

2. Metodología Aplicada.

El análisis fue realizado mediante el uso de la herramienta de código abierto MobSF Live, una herramienta de análisis estático que permite la evaluación de aplicaciones móviles directamente desde el navegador, sin necesidad de virtualización ni instalación local.

En la herramienta, se cargó el archivo APK de la aplicación en la plataforma (app-debug.apk), y se revisaron los informes generados, los cuales incluyen detalles sobre permisos peligrosos, configuraciones inseguras, firma de certificados, exposición de componentes y otros riesgos relevantes.

Para clasificar y contextualizar los hallazgos que se evidenciaron a través del análisis entregado por MobSF Live, se utilizó como referencia el estándar OWASP Mobile Application Security Project, una iniciativa internacional que busca mejorar la seguridad de las aplicaciones móviles mediante guías, estándares y herramientas de evaluación. Este informe los hallazgos fueron categorizados según los criterios del MASVS, lo que permitió identificar vulnerabilidades reales que afectan la confidencialidad, integridad y disponibilidad de la aplicación.

Aunque, ¿qué es el criterio MASVS?:

- MASVS (Mobile Application Security Verification Standard): Es un estándar que define los requisitos mínimos de seguridad que debe cumplir una aplicación móvil. Sirve como marco de referencia para desarrolladores, testers y auditores, permitiendo verificar si una app protege adecuadamente los datos del usuario, la lógica interna y la comunicación con servicios externos.

Este marco define los requisitos mínimos de seguridad que debe cumplir una aplicación móvil y permite validar si las vulnerabilidades encontradas representan riesgos reales para la confidencialidad, integridad y disponibilidad de los datos del usuario.

3. Resumen de los Hallazgos.

El análisis estático realizado con MobSF Live permitió identificar un conjunto de vulnerabilidades que afectan distintos aspectos de la seguridad de la aplicación Android. Estas vulnerabilidades fueron clasificadas según su severidad, considerando el impacto potencial sobre la confidencialidad, integridad y disponibilidad de los datos del usuario, así como el riesgo de explotación por terceros.

Para evidenciar de manera resumida, se hará a través de la presente tabla de resumen con la cantidad de hallazgos entregados por MobSF:

Severidad	**Cantidad**
Critico	1
Alto	3
Medio	3
Baja	1
Total	8

Los hallazgos encontrados y mostrados por MobSF, abarcan desde el uso de “Permisos Peligrosos” en conjunto con las configuraciones inseguras por el archivo de la aplicación denominado “AndroidManifest.xml”, hasta la firma por las certificaciones de depuración y la exposición de componentes internos sin protección.

Y, por último, el análisis arrojo la detección del uso de permisos de uso común de abuso por programas malignos o mejor conocido como Malware, lo que refuerza la necesidad de aplicar las medidas correctivas y/o fortalecer la seguridad general del proyecto.

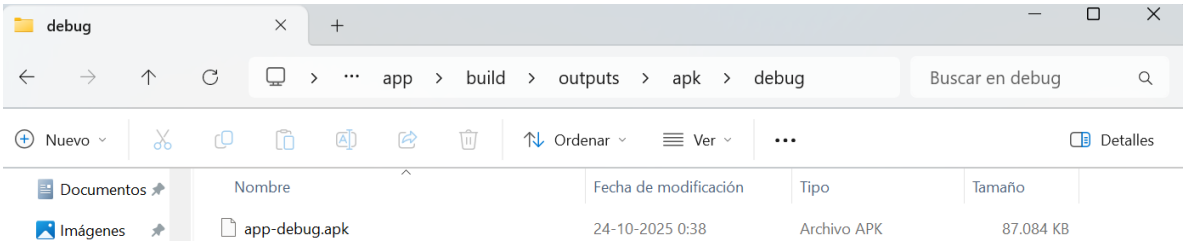
4. Vulnerabilidades Detectadas.

En este apartado se detallarán de manera profunda, las vulnerabilidades identificas mediante el análisis estático realizado por MobSF Live. Cada hallazgo incluye una descripción técnica, nivel de severidad, impacto potencial sobre la seguridad de la misma aplicación, una recomendación de remedición la cual esta basada por las buenas prácticas y estándares como OWASP MASVS y, por último, una evidencia visual del hallazgo.

4.1 – Aplicación Firmada con certificado de Depuración.

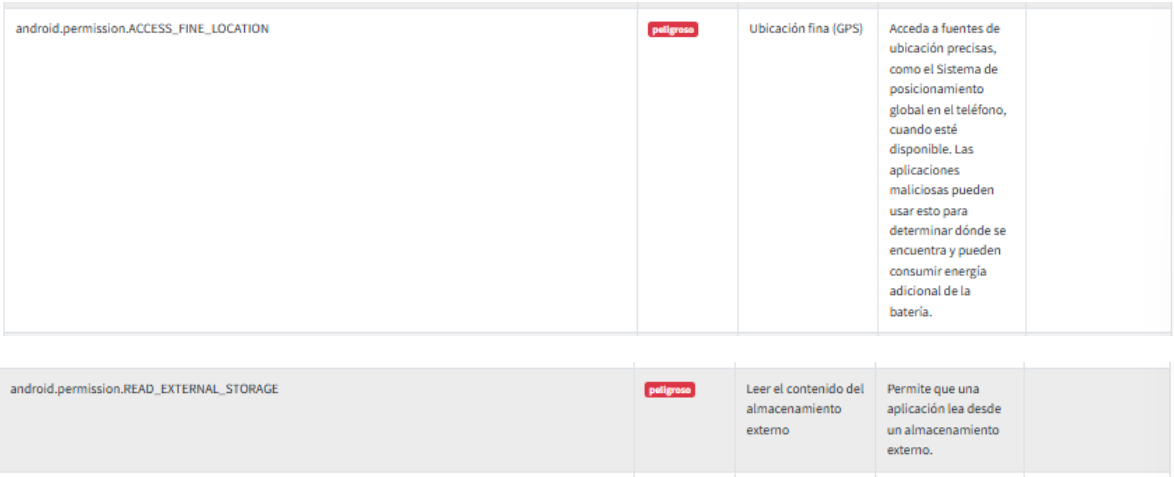
- Nivel de Severidad: Alta.
- Descripción Técnica: La aplicación esta con la firma de Certificado de Depuración (debug). La cual permite que sea instalada y modificada con demasiada facilidad, lo cual es un riesgo de seguridad para el entorno de producción.
- Impacto sobre la seguridad: Gracias a lo anterior esto permitiría la manipulación del código, y sobre todo el acceso no autorizado a funciones internas.
- Recomendación de remedición: Firmar la aplicación con un certificado de producción antes de una distribución pública.
- Evidencia:

- Evidencia de la firma al depurar y hacer apk:



4.2 – Uso de Permisos Peligrosos.

- Nivel de Severidad: Alta/Critica.
- Descripción Técnica: La aplicación solicita permisos considerados peligrosos por Android, los cuales son “READ_EXTERNAL_STORAGE” y “ACCESS_FINE_LOCATION”. Se consideran peligrosos ya que esos permiten acceder a archivos personales del usuario y su ubicación precisa.
- Impacto sobre la seguridad: Impacto directo sobre el riesgo de exposición de datos sensibles y el rastreo de ubicación sin consentimiento.
- Recomendación de remediación: Solicitar los permisos cuando solamente sea estrictamente necesarios, además de la justificación del porque del uso ante el usuario, aplicación de cifrados y validación de datos.
- Evidencia:



- Fragmento del código identificado:

```
<!-- Permisos necesarios -->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

4.3 – Configuraciones inseguras en el manifiesto.

- Nivel de Severidad: Alta/Media.
- Descripción Técnica: El análisis detecto configuraciones inseguras en el archivo “AndroidManifest.xml”, en el cual incluye los siguientes puntos: “Android:debuggable=”true” = lo cual permite depuraciones incluso en producción. Y “Android:allowBackup=”true” = el cual permite respaldar y restaurar datos de la app.
- Impacto sobre la seguridad: El impacto que tiene no es menor ya que da paso a la posibilidad de la manipulación de la app en ejecución y fuga de datos sensibles del usuario.
- Recomendación de remediación: Establecer los puntos mencionados con un “false” en versiones de producción. Además de aplicar un cifrado a los datos de almacenamiento local.
- Evidencia:

2	Depuración habilitada para la aplicación [android:debuggable=true]	Alto	La depuración se habilitó en la aplicación, lo que facilita a los ingenieros inversos conectar un depurador a ella. Esto permite volcar un seguimiento de pila y acceder a las clases auxiliares de depuración.	
3	Se puede hacer una copia de seguridad de los datos de la aplicación[android:allowBackup=true]	advertencia	Esta marca permite que cualquier persona haga una copia de seguridad de los datos de su aplicación a través de adb. Permite a los usuarios que han habilitado la depuración USB copiar datos de la aplicación fuera del dispositivo.	

- Fragmento identificado en el código:

```
<application
    android:allowBackup="true"
```

```
android:debuggable="true"
```

4.4 – Actividades expuestas sin protección.

- Nivel de Severidad: Baja.
- Descripción Técnica: Las actividades tipo “AudioActivity”, “DescargaActivity” y “MapsActivity”, se encuentran marcadas como exportadas (exported=”true”) sin una restricción de acceso.
- Impacto sobre la seguridad: Lo anterior denota en que cualquiera otra aplicación instalada en el dispositivo, podría llegar a invocar estas actividades, de esta manera, generando riesgos de ejecución no autorizada.
- Recomendación de remediación: Poner a exported=”false” y/o proteger las actividades con los permisos definidos en el manifiesto.
- Evidencia:

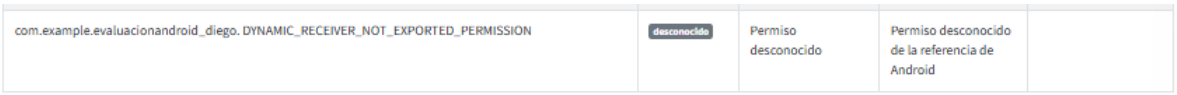
3	Se puede hacer una copia de seguridad de los datos de la aplicación[android:allowBackup=true]	advertencia	Esta marca permite que cualquier persona haga una copia de seguridad de los datos de su aplicación a través de adb. Permite a los usuarios que han habilitado la depuración USB copiar datos de la aplicación fuera del dispositivo.	
4	**Actividad** (com.example.evaluacionandroid_diego. AudioActivity) no está protegido. [android: exportado = verdadero]	advertencia	Se encuentra que una actividad se comparte con otras aplicaciones en el dispositivo, por lo que se puede acceder a ella para cualquier otra aplicación en el dispositivo.	
5	**Actividad** (com.example.evaluacionandroid_diego. VideoActivity) no está protegido. [android: exportado = verdadero]	advertencia	Se encuentra que una actividad se comparte con otras aplicaciones en el dispositivo, por lo que se puede acceder a ella para cualquier otra aplicación en el dispositivo.	
6	**Actividad** (com.example.evaluacionandroid_diego. DescargaActivity) no está protegido. [android: exportado = verdadero]	advertencia	Se encuentra que una actividad se comparte con otras aplicaciones en el dispositivo, por lo que se puede acceder a ella para cualquier otra aplicación en el dispositivo.	
7	**Actividad** (com.example.evaluacionandroid_diego. MapaActivity) no está protegido. [android: exportado = verdadero]	advertencia	Se encuentra que una actividad se comparte con otras aplicaciones en el dispositivo, por lo que se puede acceder a ella para cualquier otra aplicación en el dispositivo.	

- Fragmento identificado en el código:

```
<!-- Actividades multimedia -->
<activity android:name=".AudioActivity" android:exported="true" />
<activity android:name=".VideoActivity" android:exported="true" />
<activity android:name=".DescargaActivity" android:exported="true" />
<activity android:name=".MapaActivity" android:exported="true" />
```

4.5 – Permiso definido sin Referencia.

- Nivel de Severidad: Media.
- Descripción Técnica: El análisis arrojo un permiso personalizado (DYNAMIC_RECEIVE_NOT_EXPORTED_PERMISSION) sin documentación ni referencia clara.
- Impacto sobre la seguridad: Lo anterior puede denotar una mala configuración o una exposición innecesaria de componentes internos.
- Recomendación de remediación: Documentar el propósito del permiso o eliminarlo si no es necesario. Además de asegurar que los componentes estén correctamente restringidos.
- Evidencia:



4.6 – Uso de Permisos comúnmente abusados por Malware.

- Nivel de Severidad: Media.
- Descripción Técnica: La aplicación solicita varios permisos que están entre los mas abusados por malware conocidos, según el análisis de MobSF Live. Los cuales se incluyen: ACCESS_FINE_LOCATION, ACCESS_COARSE_LOCATION, INTERNET, ACCESS_NETWORK_STATE, READ_EXTERNAL_STORAGE.
- Impacto sobre la seguridad: SI bien los permisos pueden ser legítimos, la presencia simultanea aumenta con creces el riesgo de malinterpretación por los sistemas de seguridad o de explotación si la app es comprometida.
- Recomendación de remediación: Se recomienda la justificación de cada permiso, limitar su uso a lo estrictamente necesario, y aplicar controles de acceso y validación en el código.

- Evidencia:

PERMISO	ESTADO	INFORMACIÓN	DESCRIPCIÓN	ASIGNACIONES DE CÓDIGOS
android.permission.ACCESS_COARSE_LOCATION	peligroso	Ubicación aproximada (basada en red)	Acceda a fuentes de ubicación aproximadas, como la base de datos de la red móvil, para determinar una ubicación aproximada del teléfono, cuando esté disponible. Las aplicaciones maliciosas pueden usar esto para determinar aproximadamente dónde se encuentra.	
android.permission.ACCESS_FINE_LOCATION	peligroso	Ubicación fina (GPS)	Acceda a fuentes de ubicación precisas, como el Sistema de posicionamiento global en el teléfono, cuando esté disponible. Las aplicaciones maliciosas pueden usar esto para determinar dónde se encuentra y pueden consumir energía adicional de la batería.	
android.permission.ACCESS_NETWORK_STATE	normal	Ver el estado de la red	Permite que una aplicación vea el estado de todas las redes.	
android.permiso.INTERNET	normal	acceso completo a Internet	Permite que una aplicación cree sockets de red.	
android.permission.READ_EXTERNAL_STORAGE	peligroso	Leer el contenido del almacenamiento externo	Permite que una aplicación lea desde un almacenamiento externo.	

PERMISOS ABUSADOS

Principales permisos de malware 5/25

android.permission.ACCESS_FINE_LOCATION, android.permission.ACCESS_COARSE_LOCATION, android.permission.INTERNET, android.permission.ACCESS_NETWORK_STATE, android.permission.READ_EXTERNAL_STORAGE

Otros permisos comunes 0/44

Los permisos de malware son los permisos principales de los que se abusa ampliamente por parte del malware conocido.
Otros permisos comunes son permisos de los que comúnmente abusa el malware conocido.

- Fragmento identificado en el código:

```
<!-- Permisos necesarios -->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

5. Conclusión.

Después de varias horas revisando informes, configuraciones y permisos, puedo decir que esta actividad me permitió entender con más claridad cómo se manifiestan las vulnerabilidades en una aplicación Android. Aunque el proceso fue agotador, entre interpretar los resultados de MobSF Live, redactar los hallazgos y buscar referencias como OWASP MASVS, siento que valió la pena porque ahora tengo una base más sólida para evaluar riesgos reales en apps móviles.

Me di cuenta de que cosas que parecen pequeñas, como dejar el activado o firmar con un certificado de depuración, pueden abrir puertas a ataques serios. También entendí que no se trata solo de detectar errores, sino de justificar cada decisión técnica, documentar con evidencia y aplicar buenas prácticas que realmente protejan al usuario.

Reconozco que aún hay mucho por aprender, pero esta actividad me ayudó a conectar teoría con práctica. No fue fácil, pero me esforcé por entregar un informe completo, estructurado y alineado con los estándares internacionales. Y aunque estoy cansado, me quedo con la satisfacción de haber cumplido con los objetivos de la asignatura y haber mejorado mi capacidad para analizar y documentar la seguridad en aplicaciones móviles.

6. Bibliografía.

6. Referencias

A continuación, se presentan las fuentes utilizadas para fundamentar el análisis de seguridad, la clasificación de vulnerabilidades y las recomendaciones de remediación aplicadas en este informe:

Estándares y guías internacionales:

- OWASP Foundation. (2023). *OWASP Mobile Application Security Project*. Recuperado de https://owasp.org/www-project-mobile-security/
- OWASP Foundation. (2023). *Mobile Application Security Verification Standard (MASVS)*. Recuperado de https://github.com/OWASP/owasp-masvs

Herramientas utilizadas:

- MobSF Live. (2023). *Mobile Security Framework – Live Edition*. Recuperado de https://mobsf.live
- MobSF Documentation. (2023). *Understanding MobSF Reports*. Recuperado de https://mobsf.github.io/docs

Documentación oficial de Android:

- Android Developers. (2023). *App Signing*. Recuperado de https://developer.android.com/studio/publish/app-signing
- Android Developers. (2023). *Permissions Overview*. Recuperado de https://developer.android.com/guide/topics/permissions/overview
- Android Developers. (2023). *Security Best Practices*. Recuperado de https://developer.android.com/topic/security/best-practices