UNIVERSIDAD ESTATAL DE BOLÍVAR

FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

CARRERA DE SOFTWARE

TRABAJO DE INTEGRACIÓN CURRICULAR
PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SOFTWARE

FORMA: PROYECTO DE INVESTIGACIÓN

TEMA:

HACKING ÉTICO A DISPOSITIVOS MÓVILES ANDROID POR PLAYLOAD, AÑO 2024

AUTOR:

KLEBER ALBERTO REMACHE PÉREZ

DIRECTOR:

ING. DANILO GEOVANNY BARRENO NARANJO

GUARANDA – ECUADOR

2024

TEMA DEL PROYECTO DE INVESTIGACIÓN

Hacking Ético a dispositivos móviles Android por Playload, año 2024.

AGRADECIMIENTO

Quiero agradecer a Dios por estar a mi lado en cada adversidad, brindándome la sabiduría y compasión durante las decisiones que he tomado a lo largo de mi vida. A mi mamá y papá por ser mi inspiración en la vida, enseñándome a ser perseverante y sin importar que tan difícil se vea la situación jamás me rinda y siga a delante con la frente en alto para así lograr cumplir todas mis metas y con esto demostrarles que todo su sacrificio durante estos años no fue en vano.

A mis hermanas por estar junto a mi brindándome su compañía y alegrías a mi vida las cuales siempre mantendré en mi corazón como uno de los tesoros más invaluables de todo el mundo entero.

Quiero expresar mis más sinceros agradecimientos a mis queridos abuelitos por sus consejos, valores y virtudes que me brindaron desde pequeño y que estas estarán vivas en mi persona y en cada paso que dé en mi vida. A mis tíos y primos por darme su apoyo incondicional y por compartir cada alegría junto a mí, su mera presencia es un regalo invaluable que atesorare siempre.

Finalmente expresar mi mayor gratitud a todos mis profesores, que en mi impartieron todo su valioso conocimiento, orientación y responsabilidad a través de cada clase para así otorgarnos todas las herramientas para afrontar la vida, encaminándonos para ser unos excelentes profesionales.

DEDICATORIA

Quiero dedicar la realización y finalización de esta tesis a mi mamá y papá, mediante su apoyo incondicional me dieron todas las virtudes y valores necesarias para hacer de mí una persona de bien, justa, responsable, alegre, compasiva, honesta y perseverante. Características que siempre serán reflejadas en mi persona sin importar el momento, lugar o circunstancia en la que me encuentre.

Finalmente dedicar este éxito a mis amigos pues su compañía y apoyo me impulsaron a seguir siempre sonriendo, brindar ayuda a quien más los necesite sin importar si aquella persona era de buen o mal corazón. Por cada salida juntos, por todos los momentos agradables de mi vida que convivimos, por cada abrazo que me brindaron, por su amistad sincera, lealtad y por su compañía en situaciones difíciles. Siempre estaré eternamente agradecido con cada uno de usted por el resto de mi vida y estaré ahí para brindarle mi ayuda cuando más lo necesiten.

CERTIFICADO DE VALIDACIÓN

UNIVERSIDAD ESTATAL DE BOLIVAR

FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

CERTIFICADO DE VALIDACIÓN

Ing. Danilo Geovanny Barreno Naranjo, Ing. Rodrigo Humberto Del Pozo Durango e Ing. Carlos Enrique Taco Padilla, en su orden Director y Pares Académicos del Trabajo de Integración Curricular "HACKING ÉTICO A DISPOSITIVOS MÓVILES ANDROID POR PLAYLOAD, AÑO 2024" desarrollado por el señor Kleber Alberto Remache Pérez.

CERTIFICAN

Que, luego de revisado el Trabajo de Integración Curricular en su totalidad, cumple con las exigencias académicas de la carrera SOFTWARE, por lo tanto, autorizamos su presentación y defensa.

Guaranda, 4 de abril del 2025

Ing. Danilo Barreno Naranjo
Director

Ing. Rodrigo Del Pozo Durango
Par Académico

Ing. Carlos Taco Padilla
Par Académico

DERECHOS DE AUTOR

Yo Remache Pérez Kleber Alberto portador de la Cédula de Identidad No 1250616040 en calidad de autor y titular de los derechos morales y patrimoniales del Trabajo de Titulación: HACKING ÉTICO A DISPOSITIVOS MÓVILES ANDROID POR PLAYLOAD, AÑO 2024, modalidad Proyecto de Investigación, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN, concedemos a favor de la Universidad Estatal de Bolívar, una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Conservamos a mi favor todos los derechos de autor sobre la obra, establecidos en la normativa citada.

Así mismo, autorizo a la Universidad Estatal de Bolívar, para que realice la digitalización y publicación de este trabajo de titulación en el Repositorio Digital, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

El autor declara que la obra objeto de la presente autorización es original en su forma de expresión y no infringe el derecho de autor de terceros, asumiendo la responsabilidad por cualquier reclamación que pudiera presentarse por esta causa y liberando a la Universidad de toda responsabilidad.

Kleber Alberto Remache Pérez

C.I: 1250616040

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

En la actualidad la tecnología ha permitido a los seres humanos desarrollar numerosas herramientas o equipos para sobrellevar su día a día de manera más fácil, este cambio principalmente se lo ha logrado evidenciar a través de los dispositivos móviles ya que estos se han convertido en una extensión más de su cuerpo siendo utilizados en diversas actividades en el ámbito académico, laboral o de entretenimiento.

El uso del Hacking Ético aplicando la herramienta playload nos permite conocer las vulnerabilidades que pueden ser aprovechadas por los ciberdelincuentes dentro del dispositivo móvil de la víctima, donde el atacante ejecuta cargas maliciosas para así logar sustraer la información sensible mediante la herramienta playload.

En el presente trabajo investigativo se optó por utilizar la investigación descriptiva como pilar principal, logrando así documentar los instrumentos utilizados en la creación, desarrollo y despliegue del playload, así como las brechas de seguridad detectadas a lo largo del proceso. Teniendo en cuenta la posible utilización de ingeniería social que en muchas ocasiones es la responsable de manipular el comportamiento de los usuarios debido a que un atacante de ingeniería social se hace pasar por alguien simpático, digno de confianza o con autoridad y engaña a la víctima para que confíe en él para después poder manipularla para que revele información privada (Bodnar, 2020).

La finalidad principal de este proyecto investigativo, se basa en describir detalladamente las vulnerabilidades que pueden sufrir los dispositivos móviles Android, para así lograr clasificarlas y finalmente explicar el proceso de Hacking Ético en dispositivos móviles con sistema operativo Android con la herramienta playload.

RESUMEN

Este proyecto de investigación explora las vulnerabilidades en dispositivos móviles Android mediante Hacking Ético con la aplicación de la herramienta playload, ataques diseñados para explotar fallos de seguridad. Los antecedentes destacan la predominancia de Android, con un 76% del mercado global, y su exposición a amenazas, agravada por la fragmentación de versiones y el desconocimiento de los usuarios, lo que facilita ataques como los identificados por investigaciones previas sobre malware y permisos excesivos en aplicaciones. El principal objetivo de este proyecto, es describir diversas vulnerabilidades a través de un entorno controlado, empleando sub-herramientas como MSFVenom para generar playload APK y Metasploit Framework para simular intrusiones. Mediante la implementación de una investigación descriptiva con un enfoque cualitativo y cuantitativo, se documentan brechas de seguridad explotadas mediante ingeniería social o la ejecución de APK maliciosos, revelando cómo los ciberdelincuentes sustraen datos sensibles, como contactos o geolocalización. Los resultados, obtenidos de pruebas en varios dispositivos móviles Android, clasifican vulnerabilidades según su probabilidad de explotación (baja, media, alta) y su impacto (menor, moderado, catastrófico), identificando comandos críticos como "dump_sms" o "geolocate" con alto riesgo. Estos hallazgos evidencian la facilidad de acceso no autorizado y subrayan la necesidad de actualizar sistemas y concienciar a los usuarios. En conclusión, el Hacking Ético con playload permite comprender las debilidades de Android, ofreciendo un marco para mitigar riesgos en un sistema que, a pesar de su popularidad, enfrenta retos significativos frente a agentes maliciosos, especialmente en contextos donde solo el 40% de los usuarios actualiza regularmente sus dispositivos.

Palabras Clave: Hacking Ético**,** Ingeniería Social**,** Android**,** Ciberdelincuentes.

ABSTRACT

This research project explores in Android mobile devices through Ethical Hacking and the use of playload malicious loads designed to exploit security flaws. Background research highlights Android's dominance, holding a 76% global market share, and its exposure to threats, worsened by version fragmentation and user unawareness, which enable attacks such as those previously identified involving malware and excessive app permissions. The main objective is to describe these vulnerabilities in a controlled environment, utilizing tools like MSFVenom to create APK playload and Metasploit Framework to simulate intrusions. By implementing a descriptive research with a qualitative and quantitative approach, the study documents security gaps exploited via social engineering or malicious APK execution, revealing how cybercriminals extract sensitive data, such as contacts or geolocation. The results, obtained from tests on several Android mobile devices, rank vulnerabilities according to their likelihood of exploitation (low, medium, high) and impact (minor, moderate, catastrophic), identifying high-risk commands like "dump_sms" and "geolocate." These findings underscore the ease of unauthorized access and emphasize the need for system updates and user awareness. In conclusion, Ethical Hacking with playload provides insights into Android's weaknesses, offering a framework to mitigate risks in a widely popular system that faces significant challenges from malicious actors, particularly when only 40% of users regularly update their devices.

CAPÍTULO I

FORMULACIÓN GENERAL DEL PROYECTO

1.1 Descripción del Problema

El progreso tecnológico ha facilitado la creación y mejora de herramientas que antes eran complejas. Actualmente, los dispositivos móviles con sistema operativo Android lideran el mercado, albergando datos personales de millones de usuarios, lo que los convierte en blancos prioritarios para ciberdelincuentes. Estos buscan extraer información sensible o monitorear actividades cotidianas para comercializarla ilegalmente en mercados clandestinos.

Los playload, definidos como códigos maliciosos inyectados en el sistema de la víctima, permiten a los atacantes ejecutar tareas específicas tras explotar vulnerabilidades, causando el daño previsto (Cilleruelo, 2022).

Según la empresa de Tech Crunch (kaspersky, 2020), de los seis mil millones de dispositivos móviles activos globalmente, el 87% de los Android presenta vulnerabilidades críticas, y el 95% es susceptible a ataques vía mensajes de texto, evidenciando su exposición.

Esta situación se agrava por el desconocimiento de los usuarios, quienes, atraídos por ofertas de contenido "gratis", instalan aplicaciones maliciosas sin percibir el riesgo de robo de datos o vigilancia remota. Esto se facilita mediante ingeniería social, técnica que manipula psicológicamente a las víctimas para que revelen información privada.

El atacante, haciéndose pasar por una figura confiable, engaña al usuario y explota su confianza para acceder a datos como credenciales (Bodnar, 2020). En consecuencia, la combinación de fallos técnicos y factores humanos amplifica la amenaza a la privacidad en estos dispositivos.

1.2 Formulación del Problema

¿Qué información aporta el uso de playload en la aplicación de hacking ético durante la identificación de las vulnerabilidades en dispositivos con sistema Android?

1.3 Preguntas de Investigación

¿Cuáles son los riesgos específicos que introduce la instalación de aplicaciones maliciosas en dispositivos Android?

¿Qué información sensible puede ser sustraída del dispositivo móvil vulnerado?

¿Qué pasos técnicos se deben seguir para documentar el proceso de Hacking Ético con playload en dispositivos Android?

¿Cómo contribuye el uso de la herramienta playload en el hacking ético a la identificación de vulnerabilidades en dispositivos móviles con sistema operativo Android?

1.4 **Justificación**

Los dispositivos móviles que emplean sistema operativo Android, que son alrededor del 74% en el mercado global en 2024 (counterpointresearch, 2024), son esenciales, pero vulnerables a ciberataques debido a fallas como CVE-2024-43093 y CVE-2024-50302, reportadas en marzo de 2025 (source.android, 2024), que permiten accesos no autorizados y exposición de datos sensibles. Solo el 40% de usuarios actualiza pronto (Business of Apps, 2024), frente al 85% de iOS, y el 39.3% no usa seguridad móvil (AV-Comparatives, 2024), lo que, junto a investigaciones como la de playload en Hardware (Vina, 2024), justifica analizar estas amenazas.

El presente proyecto investigativo empleará al Hacking Ético en un entorno controlado, usando MSFVenom para crear playload APK, Metasploit Framework para sesiones remotas, y Android Debug Bridge (ADB) con Android SDK para interactuar con dispositivos Android recientes (ej. Android 14 o 15). Estas herramientas, aplicadas éticamente, simularán ataques reales, permitiendo analizar comandos ejecutados por ciberdelincuentes y documentar vulnerabilidades, garantizando reproducibilidad y seguridad al evitar daños fuera del experimento.

Se espera identificar vulnerabilidades críticas como ejecución remota y robo de datos (contactos, historial de llamadas, registro de mensajes recibidos y geolocalización), clasificándolas con un Sistema Común de Puntuación de Vulnerabilidades (CVSS), para medir su severidad, y documentar comandos como “dump_sms” en Metasploit. Esto sensibilizará a usuarios, considerando que el 40% no actualiza y el 39.3% carece de protección, ofreciendo estrategias prácticas y datos para fortalecer la seguridad de Android.

Este proyecto fomenta buenas prácticas entre usuarios, aporta un marco analítico para estudios futuros, como los de chipsets (Klischies et al., 2024), y podría influir en políticas de ciberseguridad ante la baja actualización del 40% en Android frente al 85% de iOS. Además, capacitará a profesionales y abordará una brecha significativa en un sistema que domina el 74% del mercado.

1.5 Objetivos: General y Específicos

1.5.1 General

Describir mediante el Hacking Ético, las vulnerabilidades que pueden sufrir los dispositivos móviles Android utilizando la herramienta playload.

1.5.2 Específicos

- Describir las consecuencias de las descargas de aplicaciones en los dispositivos móviles Android.
- Categorizar las vulnerabilidades que sufren los dispositivos móviles, según su probabilidad e impacto.
- Documentar el proceso de Hacking Ético en dispositivos móviles Android mediante la herramienta playload.

1.6 Idea a Defender

El uso del Hacking Ético utilizando la herramienta playload nos permite conocer las vulnerabilidades que pueden ser aprovechadas por los ciberdelincuentes dentro del dispositivo móvil de la víctima.

1.7 Variables

1.7.1 Variable Dependiente:

- Vulnerabilidades que pueden ser aprovechadas en los dispositivos móviles Android.

Tabla 1. Matriz de Operacionalización de Variable Dependiente

Variable	Definición Conceptual	Definición Operacional	Dimensiones	Indicadores	Ítems
Vulnerabilidades que pueden ser aprovechadas en los dispositivos móviles Android.	Fallos de seguridad en el sistema operativo Android, explotados con la instalación de aplicaciones maliciosas.	Se medirán mediante una guía de observación para registrar la ejecución de los comandos y una lista de cotejo para clasificar las vulnerabilidades.	• Ejecución de los comandos ejecutados. • Severidad de las vulnerabilidades	• Nivel de acceso otorgado por permisos. • Tipo de datos sensibles. • Cantidad de datos sensibles comprometidos	• Registro de los comandos para vulnerar la información de los usuarios. • Análisis de la información obtenida.

Elaborado por: *Remache Kleber, 2025.*

Resultados:

- Se identificaron 6 vulnerabilidades con un nivel de impacto catastrófico para la información personal contenida dentro del dispositivo móvil de la víctima.
- Los comandos ejecutados durante el secuestro de información sensible de la víctima utilizan la escala de permisos de la aplicación para generar un acceso a los datos en el equipo.

1.7.2 **Variable Independiente:**

- El Hacking Ético a los dispositivos móviles Android por la herramienta de playload.

Tabla 2. Matriz de Operacionalización de Variable Independiente

Variable	Definición Conceptual	Definición Operacional	Dimensiones	Indicadores	Ítems
El Hacking Ético a los dispositivos móviles Android con la herramienta playload.	Técnica utilizada para realizar una evaluación de ciberseguridad referente a las vulnerabilidades de los dispositivos móviles con sistema operativo Android.	Se realizo a través de un entorno de pruebas en Kali Linux y la instalación en los dispositivos móviles Android simulando ataques.	• Configuración del playload. • Herramientas de software	• Comandos ejecutados • Funcionalidad del comando. • Dispositivos vulnerados. • Función de las herramientas utilizadas.	• ¿Qué tipo de playload se utilizó en la creación del entorno? • ¿Cuántos dispositivos móviles con sistema operativo Android se utilizaron?

Elaborado por: *Remache Kleber, 2025.*

Resultados:

- Se utilizaron 2 dispositivos móviles de marcas reconocidas que funcionaban con el sistema operativo Android para la ejecución del playload en el entorno de pruebas.
- Se creo un playload basado en el tipo de conexión reverse_tcp, permitiendo un control en el dispositivo vulnerado.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes

La creciente dependencia de los dispositivos móviles Android ha intensificado el interés por su seguridad, dado que su dominio del 76% del mercado global (Bilić, 2020) los convierte en un blanco atractivo para ciberdelincuentes. En este contexto, diversas investigaciones han explorado las vulnerabilidades de este sistema operativo y las herramientas empleadas para explotarlas, sentando las bases para el presente estudio.

De acuerdo con la autora Vargas Lady (Santana, 2023), en su trabajo de investigación **"**ANÁLISIS DE VULNERABILIDADES CRÍTICAS DEL SISTEMA OPERATIVO MÓVIL ANDROID MEDIANTE PENTESTING**"** examinó cómo las descargas de aplicaciones desde la Play Store exponen a los usuarios a riesgos significativos. A través de una metodología exploratoria, identificó que la falta de conocimiento sobre permisos y origen de las apps facilita la instalación de software malicioso, un hallazgo que subraya la relevancia de describir las consecuencias de estas prácticas, alineándose con el primer objetivo específico de esta investigación. Por su parte, las empresas Tech Crunch y Ericsson (kaspersky, 2020), en su estudio "Amenazas de seguridad móvil dirigidas a dispositivos Android", estimaron que más de seis mil millones de usuarios enfrentan riesgos debido al aumento del malware móvil. Utilizando un enfoque cualitativo, señalaron que el 87% de los dispositivos Android presentan vulnerabilidades críticas, muchas explotables mediante mensajes de texto o aplicaciones maliciosas, lo que refuerza la necesidad de categorizar dichas brechas, como propone el segundo objetivo específico

En un análisis más técnico, López Sempere (Sempere, 2021), en "Identificación y detección de vulnerabilidades", destacó el uso de playload como Meterpreter en

ataques remotos. Mediante una metodología descriptiva, demostró cómo estas herramientas permiten a los atacantes ejecutar comandos avanzados y evadir sistemas de seguridad, operando directamente en la memoria de los dispositivos comprometidos. Este enfoque técnico es fundamental para documentar el proceso de Hacking Ético con playload, objetivo específico tercero de esta tesis. Asimismo, Ciberpyme (Ciberpyme, 2023), en su artículo "Vulnerabilidades críticas en móviles Android", identificó fallos de severidad alta que facilitan la escalada de privilegios y la ejecución remota de código, enfatizando la urgencia de comprender cómo los ciberdelincuentes aprovechan estas debilidades en Android.

A su vez, el panorama regional aporta un contexto relevante. (Onofa, 2022) reportó que Ecuador, ubicado en el puesto 119 del Índice Global de Ciberseguridad, enfrenta crecientes amenazas de malware, como el ransomware BlackCat que afectó al municipio de Quito en abril de 2022. Aunque este caso no se centra en playload ni Android específicamente, ilustra la vulnerabilidad de la región a ciberataques, justificando la pertinencia local de estudiar herramientas como las empleadas en esta investigación.

En síntesis, estos antecedentes evidencian que las vulnerabilidades en Android, potenciadas por el desconocimiento de los usuarios y el uso de playload, son un problema global y local bien documentado. Sin embargo, persiste la necesidad de un análisis descriptivo que detalle el proceso de explotación mediante Hacking Ético y clasifique sus impactos, brecha que este proyecto busca abordar al integrar herramientas como MSFVenom y Metasploit Framework en un entorno controlado.

2.2 Científico

Desde una perspectiva macro, Android lidera el mercado móvil global con un 76% de participación, según (Bilić, 2020), lo que amplifica su exposición a amenazas a gran escala. Esta predominancia se ve comprometida por una fragmentación persistente: aunque muchas versiones coexisten, un porcentaje significativo de dispositivos opera con sistemas obsoletos, incrementando su vulnerabilidad a malware, que representa el 99% de las detecciones en móviles.

A nivel meso, la región latinoamericana, incluyendo Ecuador, enfrenta retos específicos. (Onofa, 2022), señala que países como este, en el puesto 119 del Índice Global de Ciberseguridad, son blancos recurrentes de ataques como el ransomware BlackCat, evidenciando una infraestructura digital frágil que agrava las debilidades de Android.

En el ámbito micro, vulnerabilidades específicas como CVE-2023-21096 y CVSS 9.8 afectan versiones recientes (Android 12, 12.1, 13), permitiendo ejecución remota de código sin interacción del usuario (Lopez, 2023). Estas fallas, combinadas con parches de 2023 para riesgos de gravedad media, subrayan la necesidad de actualizaciones constantes.

Sin embargo, solo el 40% de los usuarios las implementa oportunamente (Business of Apps, 2024), dejando a la mayoría expuesta a explotación remota o escalada de privilegios. Este análisis multiescalar revela que la seguridad de Android depende tanto de factores estructurales como de comportamientos individuales, creando un escenario favorable para agentes maliciosos.

2.3 **Conceptual**

Los playload se pueden clasificar de dos modos: según el sentido de la conexión entre la máquina del atacante y la de la víctima y de acuerdo a su función.

Según el sentido de la conexión, existen dos tipos de playload:

- **Bind (directo):** En el que la máquina del hacker se conecta a la de la víctima.
- **Reverse (inverso):** En el que la máquina de la víctima se conecta a la del hacker.

Comúnmente, es mucho más fácil utilizar playload de tipo reverse, ya que no cuentan con obstáculos como firewalls y, además, la víctima no tiene que tener ningún puerto abierto para que funcionen (Cilleruelo, 2022).

Con la ayuda de los comandos Meterpreter, es un playload que se utiliza para ejecutar tareas maliciosas en el ordenador de la víctima de un ciberataque. Es un tipo

de playload muy poderoso y difícil de detectar, ya que opera en niveles muy bajos de la máquina del usuario. En consecuencia, le da al atacante una amplia gama de funciones para ejecutar en el ordenador comprometido; por otra parte, la víctima podría ni siquiera enterarse de que alguien está usando su ordenador en segundo plano (Cilleruelo, 2022).

Y la utilización de MSFVenom, que es una herramienta dentro del framework Metasploit que se utiliza para generar un playload o cargas útiles. Un playload es un fragmento de código que realiza acciones específicas una vez que se ha explotado una vulnerabilidad en un sistema objetivo. MSFVenom es especialmente conocido por su capacidad para crear playload personalizados que pueden utilizarse en ataques de penetración y pruebas de seguridad (petrecere, 2023).

2.4 **Legal**

El Hacking Ético en dispositivos móviles en Ecuador no está prohibido, pero debe realizarse con el consentimiento de la persona o entidad empresarial. Caso contrario en cualquier acción realizada sin autorización puede resultar en severas consecuencias legales, incluyendo penas privativas de libertad según nos menciona el Código Orgánico Penal y la ley Orgánica de Protección de Datos Personales.

Código Orgánico Integral Penal

Artículo 232: Este artículo establece sanciones para el delito de "Ataque a la integridad de sistemas informáticos". Se pena con prisión de tres a cinco años a quien destruya, dañe, borre, altere o cause mal funcionamiento en sistemas informáticos sin autorización (Zúñiga Ledy, 2014)

Artículo 233: Relacionado con el acceso no autorizado a sistemas informáticos, este artículo también contempla penas similares para quienes accedan a estos sistemas sin el consentimiento del propietario (Zúñiga Ledy, 2014).

Artículo 234: Penaliza la revelación de secretos y la obtención de datos personales sin autorización, lo que puede ser relevante si un hacker ético accede a información sensible sin el debido consentimiento (Zúñiga Ledy, 2014).

Ley Orgánica de Protección de Datos Personales

Artículo 1: Esta ley protege la información personal y establece que cualquier tratamiento de datos debe realizarse con el consentimiento explícito del titular. La violación de esta norma puede acarrear sanciones administrativas y penales (Barrezueta, 2021).

2.1. Georreferencial

En este proyecto no se aplica la georreferenciación, debido a que no se utilizará la técnica de asignación de coordenadas geográficas a entidades para su localización precisa en un sistema de información geográfica.

CAPITULO III

METODOLOGÍA

3.1 Tipo de Investigación

Investigación Descriptiva

Para la realización de este proyecto se empleó el tipo de investigación descriptiva con la finalidad de poder documentar el uso del Hacking Ético a dispositivos móviles que cuenta con sistema operativo Android y las posibles vulnerabilidades que permiten el acceso a los ciberataques con la herramienta playload, centrándonos en la recolección de información y la utilización de las herramientas de software para así poder conocer su procedimiento de uso.

3.2 Enfoque de la investigación

Cualitativo

Para este proyecto de investigación se utilizará un método cualitativo que permitirá realizar una descripción exhaustiva de los mecanismos utilizados, así como las brechas de seguridad encontradas a lo largo del proceso durante la configuración del entorno de pruebas para el dispositivo móviles Android, proporcionando así una compresión más profunda y contextualizada de las posibles amenazas que enfrentan los dispositivos móviles con sistema operativo Android.

Donde se explicará de manera general las probabilidades (baja, media y alta) para la ejecución de un comando por parte del ciberdelincuente hacia el dispositivo vulnerado y el nivel de impacto (menor, moderado y catastrófico) que tendrá la información

obtenida de la víctima, basándonos en el grado privacidad que puede vulnerar cada comando en el sistema Android del dispositivo.

Cuantitativo

Se integró un método cuantitativo para registrar variables numéricas, como las tasas de éxito (Si/No) de comandos específicos (dump_calllog, dump_sms, dump_contacts). La muestra incluyó 50 dispositivos Android de diferentes modelos, permitiendo comparar vulnerabilidades y enriquecer la comprensión de los impactos en dichos comandos.

3.3 Métodos de Investigación

Mixta Inductivo – Deductivo

Para esta investigación se utilizó una metodología mixta, la cual estará compuesta por una parte inductiva y una deductiva, permitiendo así poder describir detalladamente las herramientas utilizadas en la creación y ejecución de la herramienta del playload, así como conocer las vulnerabilidades que son aprovechadas por los ciberdelincuentes.

Con la utilización de un análisis para el método inductivo se podrán identificar los posibles patrones y las características comunes en los dispositivos móviles Android. Mientras, que con el método deductivo se podrán describir las funciones de cada comando que nos permite utilizar la herramienta playload, para así obtener una comprensión más profunda y sistemática de las brechas de seguridad o por el caso contrario si al momento de la instalación del APK en el dispositivo móvil fueron detectadas como virus por la seguridad del sistema.

3.4 Técnicas e Instrumentos de Recopilación de Datos

- La observación directa (Guía de observación).
- Revisión documental
- Herramientas de software
- Lista de Cotejo

3.5 **Universo, Población y Muestra**

UNIVERSO

El universo de estudio abarca los dispositivos móviles equipados con sistema operativo Android, considerando su amplia prevalencia y la fragmentación de modelos que incrementa su susceptibilidad a ciberataques. Este universo incluye dispositivos de diversos fabricantes y versiones, reflejando la diversidad del ecosistema Android y su relevancia en el contexto de la ciberseguridad.

POBLACIÓN

El conjunto total de dispositivos Android disponibles en el mercado, abarcando una amplia gama de versiones, desde las más antiguas hasta las más recientes, y representando diferentes fabricantes. Esta población es relevante para el estudio, dado que la fragmentación de versiones y la falta de actualizaciones regulares (solo el 40% de los usuarios actualiza con frecuencia) son factores clave que aumentan la vulnerabilidad.

MUESTRA

La muestra seleccionada consiste en 50 dispositivos Android, representando diversos modelos de fabricantes para garantizar la variabilidad en los resultados.

Es importante destacar que no se involucran sujetos humanos en la investigación, manteniendo un enfoque controlado y ético, con pruebas realizadas en un entorno aislado para evitar riesgos y garantizar la consistencia en los resultados, lo que permite una comprensión detallada de cómo estos 50 dispositivos móviles Android reflejan las vulnerabilidades del sistema operativo en diferentes configuraciones.

3.6 **Procesamiento de la Información**

Para el procesamiento de la información durante la realización y finalización del proyecto investigativo se llevó a cabo una observación directa, lo que permitirá documentar a través de una guía de observación todos los procedimientos y el funcionamiento de la herramienta de playload en el dispositivo móvil y que información puede ser vulnerada del usuario.

También se realizó una revisión documental basándonos en artículos, libros, foros, páginas web, etc. Para así lograr obtener una mayor comprensión de todos los alcances y vulnerabilidades que se puedan obtener durante la realización de este proyecto investigativo.

La utilización de las herramientas de software fue crucial para acceder dentro del dispositivo porque con su ayuda se logró la creación, desarrollo, despliegue del playload y con esto se analizaron las consecuencias que puede sufrir el dispositivo móvil ante un ataque que pongan en riesgo la información personal de los usuarios.

Finalmente, se realizó una lista de cotejo para conocer la escala del nivel de riesgo total en la cual se encuentra el dispositivo vulnerado y como esto puede perjudicar a la víctima y en el peor de los casos se incrementó el número de víctimas debido a que los contactos del usuario fueron comprometidos.

CAPITULO IV

RESULTADOS Y DISCUSIÓN

4.1 Análisis, Interpretación y Discusión de Resultados

4.1.1 Generación del Entorno de Pruebas

Para comenzar la creación de nuestro entorno de pruebas, se debe conocer la dirección IP del ordenador que actuará como atacante (ciberdelincuente). Dentro del sistema operativo Kali Linux, esta dirección se la puede conocer mediante el manejo de la terminal en conjunto a la ejecución de los comandos:

- Hostname -I
- ifconfig

1.- Una vez conocida nuestra dirección IP, colocamos el siguiente comando en la terminal para la creación de nuestro APK que contendrá la herramienta de playload y su finalidad será brindarnos acceso al dispositivo de la víctima.

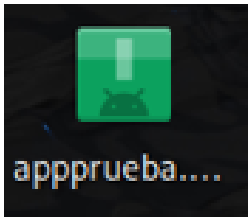
Figura 1. Creación del APK.

Elaborado por: *Remache Kleber, 2025*

Para esto se específica el sentido reverse_tcp que nos permitirá tener acceso al dispositivo vulnerado una vez que el usuario inicialice la aplicación y la información sea enviada a través de la conexión wifi a la que se encuentre conectada el dispositivo. Posteriormente especificamos nuestra dirección IP (atacante), seleccionamos el puerto de nuestro equipo el cual debe encontrarse habilitado únicamente para nuestro APK, por el cual los datos de la víctima llegarán una vez vulnerado el equipo y finalmente

colocamos la ruta donde nuestro APK se creará y le asignaremos un nombre en concreto que pueda llamar la atención del usuario.

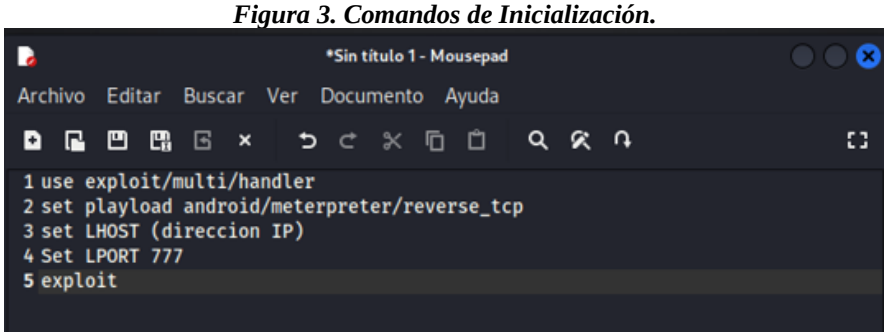
Figura 2. APK creado en el sistema Kali Linux.



Elaborado por: *Remache Kleber, 2025*

2.- En este paso crearemos un archivo de texto el cual ejecutara los siguientes comandos que inicializaran nuestro playload dentro del dispositivo infectado, pero este se encontrará en estado de inactividad hasta que la víctima active nuestro playload:

Figura 3. Comandos de Inicialización.



Elaborado por: *Remache Kleber, 2025*

Se inicializa el módulo exploit/multi/handler dentro del equipo del atacante para controlar las conexiones que permitirán la recepción de la información de los equipos vulnerados. Posteriormente se definen los parámetros correspondientes a nuestra aplicación maliciosa mediante el código "set playload android/meterpreter/reverse_tcp", para después indicar la dirección IP de la máquina atacante acompañada del número de puerto por el cual se recibirá la información personal de la víctima que se encuentra dentro del dispositivo móvil. Finalmente se ejecuta se ejecuta el comando "exploit" cuyo propósito será el de mantener en sesión de espera la conexión entre el atacante y la víctima.

3.- Ingresamos a nuestro escritorio a través de la terminal de Kali Linux por medio del comando **"cd Escritorio"** una vez dentro digitamos **"ls"** esto nos permitirá visualizar

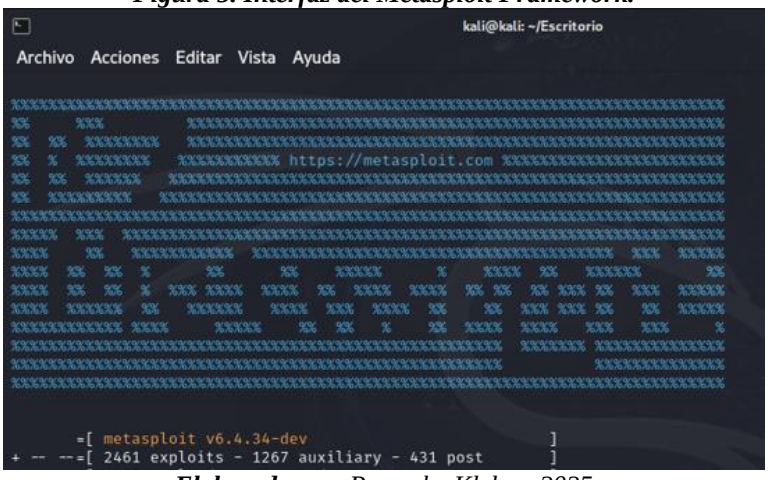
nuestro APK creado, para después darle la orden de ejecutar el archivo de texto que contendrá los comandos **"msfconsole -r comandos"** mencionados anteriormente.

Figura 4. Primeros pasos de ejecución.

Elaborado por: *Remache Kleber, 2025*

4.- Una vez ejecutados nuestros comandos se inicializará nuestro APK conteniendo la herramienta playload y posteriormente se visualizará en nuestra pantalla el icono de la herramienta de software de Metasploit Framework.

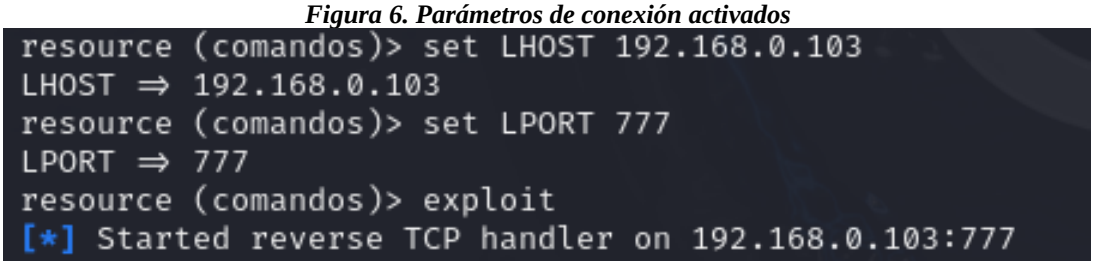
Figura 5. Interfaz del Metasploit Framework.



Elaborado por: *Remache Kleber, 2025*

5.- En este paso nuestro playload comenzara a ejecutarse y verificara los datos que ingresamos referentes a la dirección IP y el número de puerto del equipo atacante. En caso de que no se reconozca el puerto asignado nos saldrá un error de que ese número de puerto está siendo ocupado por aplicación. Finalmente visualizaremos a través de la terminal la dirección IP del equipo atacante acompañada la inicialización del método reverse dando a entender que nuestro playload se encuentra en modo de espera.

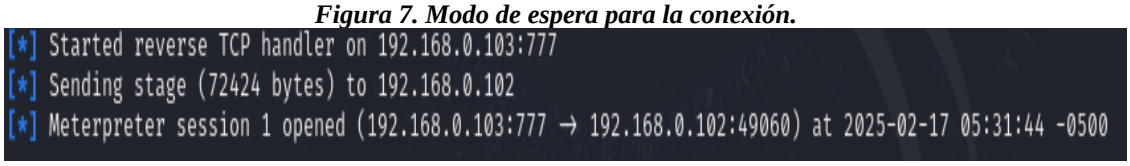
Figura 6. Parámetros de conexión activados



Elaborado por: *Remache Kleber, 2025*

6.- En este paso se encontrará activo el modo de espera, donde el atacante podrá mantener la conexión con el dispositivo vulnerado. Cuando la víctima ejecute nuestro APK la herramienta playload nos creará una sesión Meterpreter para este dispositivo y nos permitirá conocer la dirección IP, fecha y hora en la que se encuentra el dispositivo móvil vulnerado.

Figura 7. Modo de espera para la conexión.



Elaborado por: *Remache Kleber, 2025*

7.- Una vez establecida la conexión con el equipo vulnerado ingresamos en la terminal el comando **"help"** para así lograr visualizar el listado de comandos útiles a ejecutar dentro del entorno del atacante.

Tabla 3. Lista de Comandos Básicos

Comandos Básicos	
Comandos	**Descripción**
background	Pasa a segundo plano la sesión actual.
bg	Alias para el fondo.
bgkill	Eliminar un script Meterpreter en segundo plano.
bglist	Lista los scripts en ejecución en segundo plano.
bgrun	Ejecuta un script de Meterpreter como un hilo en segundo plano.
channel	Muestra información o controla los canales activos.
close	Cerrar un canal.
detach	Desconectar la sesión de Meterpreter (para http/https).
disable_unicode_encoding	Desactiva la codificación de cadenas Unicode.
enable_unicode_encoding	Activa la codificación de cadenas Unicode.
exit	Terminar la sesión de Meterpreter.
get_timeouts	Obtener los valores de tiempo de espera de la sesión actual.
guid	Obtener el GUID de la sesión.
help	Menú de ayuda.
info	Muestra información sobre un módulo de Post.
irb	Abrir un Shell Ruby interactivo en la sesión actual.
load	Cargar una o más extensiones de Meterpreter.
machine_id	Obtener el MSF ID de la maquina adjunta a la sesión.
pry	Abrir el depurador Pry en la sesión actual.
quit	Terminar la sesión de Meterpreter.
read	Lee datos de un canal.
resource	Ejecutar los comandos almacenados en un archivo.
run	Ejecuta un script Meterpreter o un módulo Post
secure	Renegociar el cifrado de paquetes TLV en la sesión.
sessions	Cambiar rápidamente a otra sesión.
set_timeouts	Establecer los valores actuales de tiempo de espera de la sesión.
sleep	Forzar a Meterpreter a quedarse en silencio, luego restablecer la sesión.
transport	Gestionar los mecanismos de transporte
Use	Alias obsoleto para “carga”.
Uuid	Obtener el UUID de la sesión actual.
Write	Escribe los datos en un canal.

Elaborado por: *Remache Kleber, 2025.*

Tabla 4. Listado de Comandos para la red.

Stdapi: Comandos de red	
Comandos	**Descripción**
ifconfig	Interfaces de visualización de la dirección IP del equipo.
Ipconfig	Interfaces de visualización.
Portfwd	Reenviar un puerto local a un servicio remoto
route	Ver y modificar la tabla de enrutamiento.

Elaborado por: *Remache Kleber, 2025*

Tabla 5. Listado de comandos para la visualización de archivos.

Stdapi: Comandos del sistema de archivos	
Comandos	**Descripción**
Cat	Leer el contenido de un archivo en la pantalla.
Cd	Cambiar directorio.
Checksum	Recuperar la suma de comprobación de un archivo.
Cp	Copiar origen a destino.
Del	Eliminar el archivo especifico.
Dir	Listar archivos (alias para ls).
Download	Descargar un archivo o directorio.
Edit	Editar un archivo.
Getlwd	Imprimir directorio de trabajo local (alias para lpwd).
getwd	Imprimir directorio de trabajo.
lcat	Leer el contenido de un archivo local en la pantalla.
lcd	Cambiar el directorio de trabajo local.
ldir	Lista de archivos locales (alias para lls).
lls	Lista de archivos locales.
lmkdir	Crear un nuevo directorio en la máquina local.
lpwd	Imprimir directorio de trabajo local.
ls	Lista de archivos.
mkdir	Crear directorio
mv	Mover el origen al destino.
pwd	Imprimir directorio de trabajo.
rm	Eliminar el archivo especificado.
rmdir	Eliminar directorio.
search	Buscar archivos.
upload	Subir un archivo o directorio

Elaborado por: *Remache Kleber, 2025*

Tabla 6. Listado de comandos para la interfaz de usuario.

Stdapi: Comandos de la interfaz de usuario.	
Comandos	**Descripción**
Screenshare	Vea el escritorio del usuario remoto en tiempo real.
screenshot	Captura de pantalla del escritorio interactivo.

Elaborado por: *Remache Kleber, 2025*

Tabla 7. Listado de comandos para el acceso al sistema.

Stdapi: Comandos del sistema	
Comandos	**Descripción**
execute	Ejecutar un comando.
getenv	Obtener uno o más valores de variables de entorno.
getpid	Obtener el identificador del proceso actual.
getuid	Obtener el usuario con el que se está ejecutando el servidor.
localtime	Muestra la fecha y la hora local del sistema de destino.
pgrep	Filtrar procesos por nombre.
ps	Lista de procesos en ejecución.
shell	Entrar en un terminal (Shell) de comandos del sistema.
sysinfo	Obtener información sobre el sistema remoto, como el sistema operativo.

Elaborado por: *Remache Kleber, 2025*

Tabla 8. Listado de comandos para el acceso de la cámara web.

Stdapi: Comandos de la cámara web	
Comandos	**Descripción**
record_mic	Grabar audio desde el micrófono por defecto durante X segundos.
webcam_chat	Iniciar un chat de video.
webcam_list	Lista de cámaras web.
webcam_snap	Tomar una instantánea de la cámara web especificada.
webcam_stream	Reproducir una transmisión de video desde la cámara web especificada.

Elaborado por: *Remache Kleber, 2025*

Tabla 9. Listado de comandos para la reproducción de audio del equipo.

Stdapi: Comandos de salida de audio	
Comandos	**Descripción**
play	Reproducir un archivo de audio de forma de onda (.wav) en el sistema de destino.

Elaborado por: *Remache Kleber, 2025*

Tabla 10. Listado de comandos básicos a ejecutar en el dispositivo móvil.

Comandos de Android	
Comandos	**Descripción**
activity_start	Iniciar una actividad Android a partir de una cadena Uri.
check_root	Comprueba si el dispositivo esta rooteado.
dump_calllog	Obtener registro de llamadas.
dump_contacts	Obtener lista de contactos.
dump_sms	Obtener mensajes sms.
geolocate	Obtener la latitud y longitud actuales mediante geolocalización.
hide_app_icon	Ocultar el icono de la aplicación del iniciador.
interval_collect	Gestionar las capacidades de recopilación de intervalos.
send_sms	Enviar SMS desde la sesión de destino.
set_audio_mode	Ajustar el modo de timbre.
sqlite_query	Consulta de una base de datos SQLite desde el almacenamiento.
wakelock	Activar/desactivar el bloqueo de pantalla.
wlan_geolocate	Obtener la latitud y longitud actuales utilizando la información de WLAN.

Elaborado por: *Remache Kleber, 2025*

Tabla 11. Listado de comandos para controladores de aplicaciones.

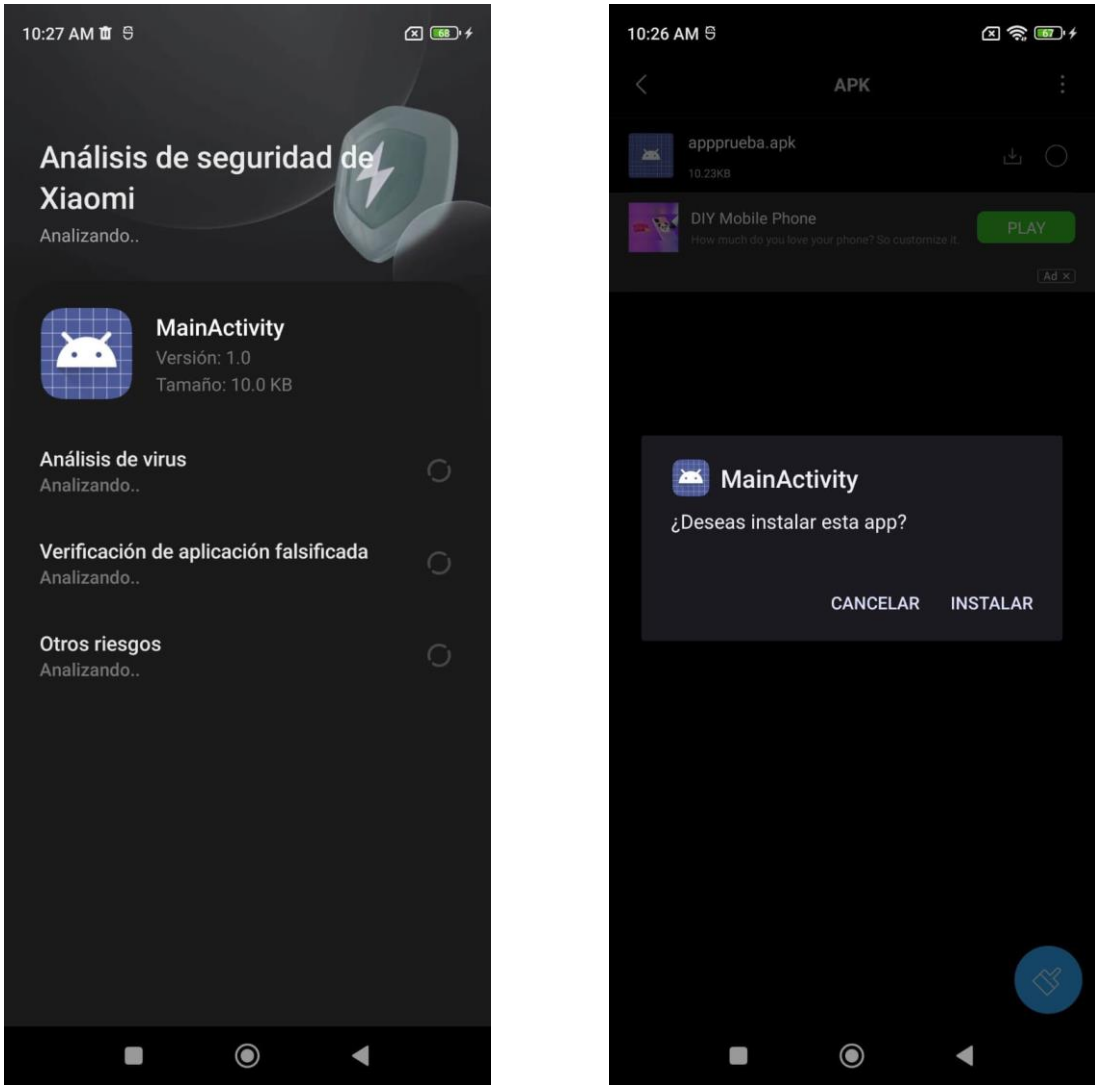
Comandos del controlador de aplicaciones	
Comandos	**Descripción**
app_install	Solicitud para instalar el archivo APK.
app_list	Lista de aplicaciones instaladas en el dispositivo.
app_run	Iniciar actividad principal para el nombre del paquete.
app_uninstall	Solicitud para desinstalación de la aplicación.

Elaborado por: *Remache Kleber, 2025*

4.1.2 **Instalación en el Dispositivo Móvil**

La instalación de nuestro archivo APK en el dispositivo móvil con sistema operativo Android de la víctima, se realiza con normalidad pasando por todos los filtros de seguridad con los que cuenta el dispositivo. Para que así el usuario no sospeche que la aplicación contiene códigos maliciosos que puede ser capaces de vulnerar la privacidad del propietario del equipo y que la información sea sustraída por un ciberdelincuente.

Figura 8. Visualización de la instalación en el dispositivo móvil #1.

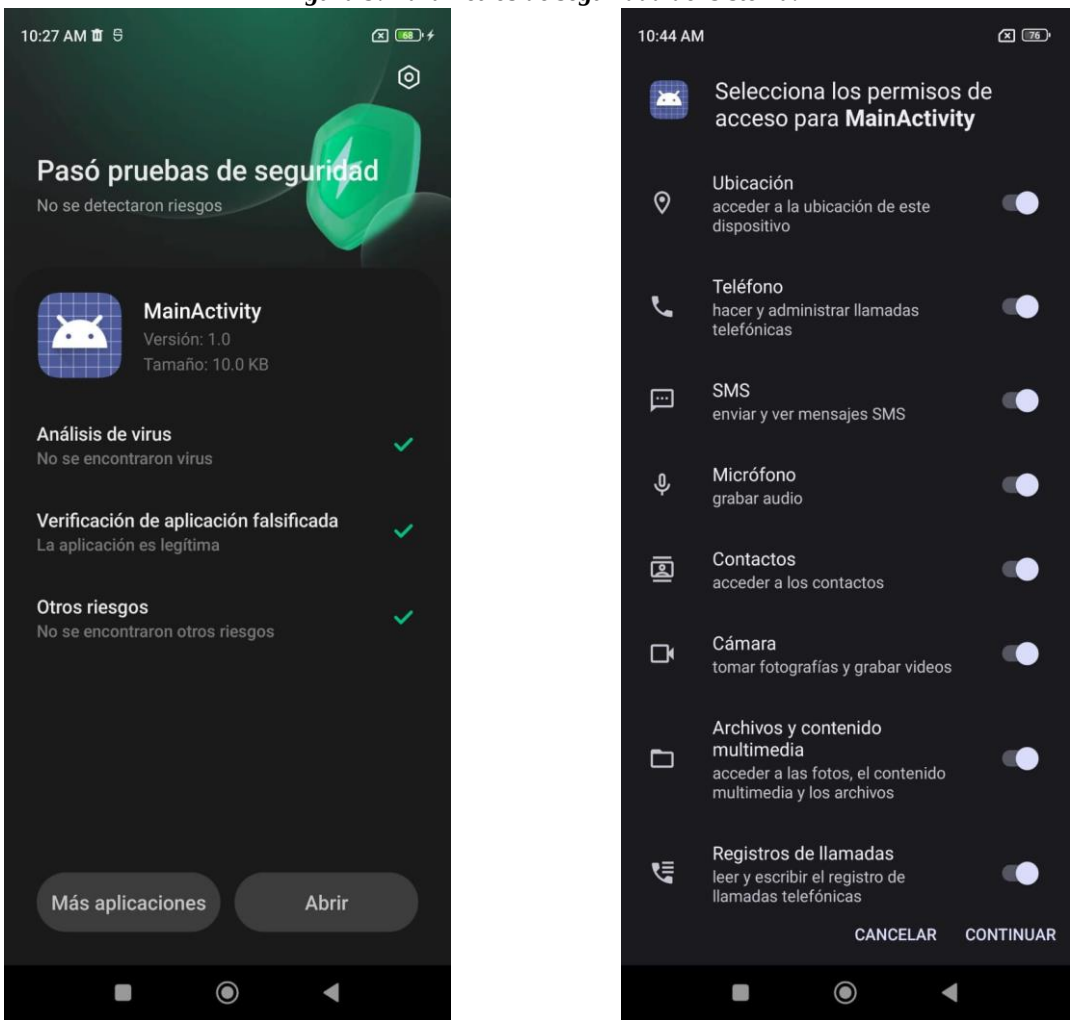


Elaborado por: *Remache Kleber, 2025*

Una vez que el usuario ejecuta la aplicación, se le pedirá que acepte todos los permisos necesarios para el funcionamiento de la misma, como se puede observar no se diferencia del funcionamiento de cualquier aplicación descargada desde la tienda oficial de aplicaciones del dispositivo móvil Android (Play Store).

Y después de ejecutar la aplicación al ciberdelincuente le aparecerá la información del dispositivo móvil vulnerado en su ordenador, proporciónale así la hora y fecha en que se ejecutó por primera vez la aplicación en el dispositivo de la víctima.

Figura 9. Parámetros de seguridad del sistema.



Elaborado por: *Remache Kleber, 2025*

4.2 **Análisis para la clasificación de Vulnerabilidades**

Tabla 12. Análisis de las vulnerabilidades para los comandos ejecutados en el entorno de prueba.

Tipos de Comandos	Probabilidad	Impacto
activity_start	Baja	Menor
check_root	Media	Menor
dump_calllog	Alta	Catastrófico
dump_contacts	Alta	Catastrófico
dump_sms	Alta	Catastrófico
geolocate	Alta	Catastrófico
hide_app_icon	Baja	Catastrófico
interval_collect	Media	Moderado
send_sms	Media	Catastrófico
set_audio_mode	Baja	Menor
Wakelock	Media	Menor
wlan_geolocate	Alta	Catastrófico
app_list	Baja	Menor
app_run	Baja	Menor
app_uninstall	Baja	Menor

Elaborado por: *Remache Kleber, 2025*

4.3 Interpretación de los comandos para causar vulnerabilidades

La ejecución de los comandos mencionados anteriormente dentro de la herramienta playload en un dispositivo móvil con sistema operativo Android permitió realizar una interpretación a cada comando de forma individual basándonos en la probabilidad (media y alta) que un comando sea utilizado por un ciberdelincuente dentro del dispositivo de la víctima y el nivel de impacto (moderado y catastrófico) que tendrá el uso de dicho comando afectando así la información sensible alojada en el dispositivo móvil.

La probabilidad para el uso del comando **dump_calllog** por parte de un ciberdelincuente a un dispositivo vulnerado es alta debido a que se necesita conocer qué tipo de información presenta la víctima y por esto su nivel de impacto es catastrófico, pues así el ciberdelincuente obtendrá todo su historial de llamadas en un archivo de texto dentro del ordenador empleado para los ataques.

Para el comando **dump_contacts** la probabilidad de ser ejecutado en el dispositivo de la víctima es alta debido a que el ciberdelincuente empieza a recolectar información sensible de la víctima y el impacto que generará el uso de este comando es catastrófico no solo para la víctima sino también para todos los contactos agregados en el dispositivo pues le permite al ciberdelincuente obtener los nombres y números telefónicos de otros usuarios, esta información se podrá visualizar por medio de un archivo de texto en el equipo empleado para los ataques.

La probabilidad de que un ciberdelincuente ejecute el comando **dump_sms** es alta debido a que este comando le permitirá obtener la información de la app de mensajería que se encuentra por defecto en el dispositivo y su impacto será catastrófico porque así el atacante podrá visualizar todo los mensajes enviados y recibidos de la víctima muchas veces estos mensajes contendrán información demasiado sensible de una o varias personas.

La ejecución del comando **geolocate** por parte del ciberdelincuente en el dispositivo de la víctima es alta debido a que se necesitara conocer de qué lugar es procedente el propietario del dispositivo y su impacto es catastrófico pues se podrá visualizar la ubicación exacta de la víctima a través de coordenadas geográficas (latitud y altitud).

El uso del comando **send_sms** por parte del ciberdelincuente cuenta con una probabilidad media debido a que se estará arriesgando a ser descubierto por parte de la víctima y que se llegue a la conclusión de que el dispositivo se encuentra vulnerado, el impacto en caso de que la víctima no se dé cuenta será catastrófico por que el ciberdelincuente podrá hacerse pasar por el propietario del dispositivo ante sus contactos y empezar a enviar mensajes a estos con la finalidad de ampliar su repertorio de víctimas.

El funcionamiento del comando **wlan_geolocate** es similar a uno presentado anteriormente abarcando una probabilidad alta de que sea ejecutado con un impacto catastrófico debido a que se visualizara la ubicación de la víctima por medio de su red inalámbrica (WLAN) para así obtener la latitud y altitud que se encuentra el propietario del dispositivo.

4.4 Discusión de los resultados obtenidos

Los siguientes resultados obtenidos durante el proceso de creación del entorno de pruebas y la instalación de la aplicación que permitirá ejecutar la herramienta playload en el dispositivo móvil con sistema operativo Android de la víctima a través del uso del hacking ético, fueron recolectados por medio del instrumento de observación directa aplicando una ficha de observación, para así poder documentar el comportamiento de los comandos maliciosos dentro del dispositivo vulnerado y así lograr sustraer la información personal del usuario.

La revisión documental permitió conocer el amplio panorama que presenta nuestra investigación y las posibles extensiones que pueden ser implementadas a futuro o ser

combinadas con diferentes métodos de Hacking Ético para evaluar los distintos parámetros del sistema operativo Android dentro de los dispositivos móviles.

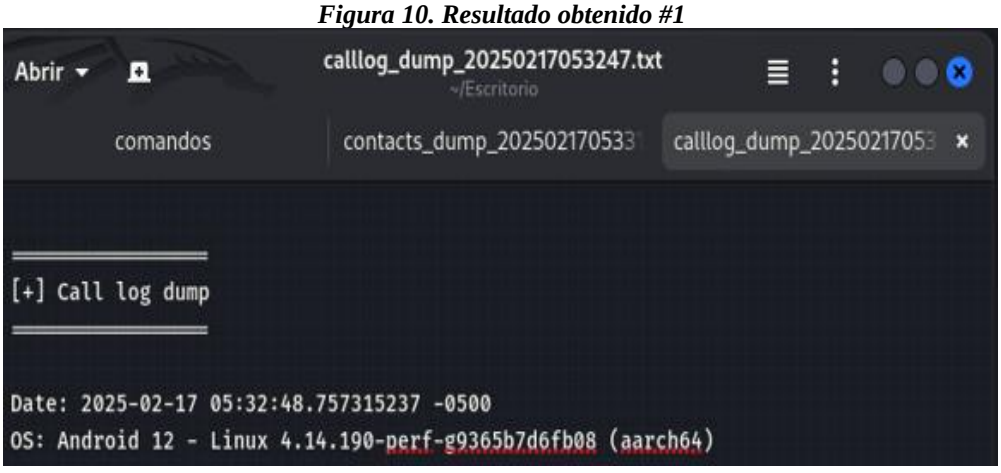
Las herramientas de software nos proporcionaron una gran ayuda durante el proceso de creación del entorno de pruebas, la utilización de MSFVenom nos permitió generar un playload enfocado al sistema operativo Android, mientras que la utilización de Metasploit Framework presento las vulnerabilidades que puede sufrir el dispositivo una ingresado los comandos maliciosos y la información que puede ser sustraída de este, la ejecución de Android Debug Bridge (ADB) fue esencial pues esta es la encargada de mantener conectados los dispositivos del atacante y la víctima.

Finalmente, el uso de Android SDK permitió que la aplicación aprobara los filtros de seguridad del sistema operativo correspondiente al dispositivo móvil y por consecuente habilito los permisos necesarios para el correcto funcionamiento de la aplicación.

DISPOSITIVO #1

Resultado obtenido de la ejecución del comando dump_calllog por parte del atacante hacia el dispositivo de la víctima.

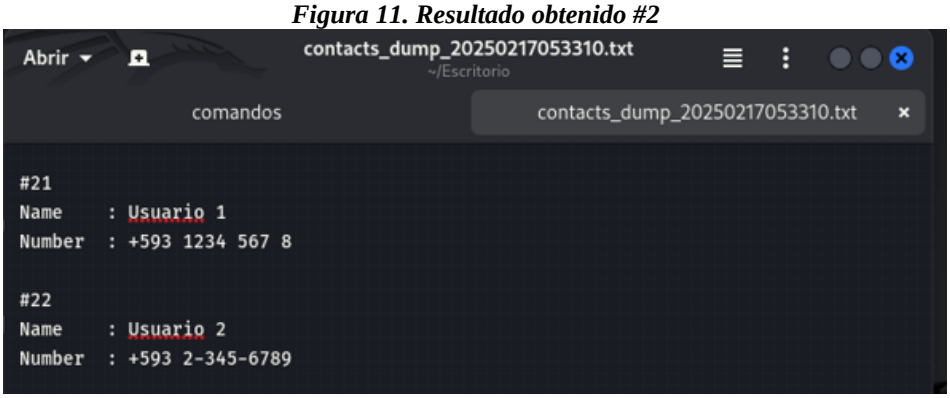
Figura 10. Resultado obtenido #1



Elaborado por: *Remache Kleber, 2025*

Resultado obtenido de la ejecución del comando dump_contacts por parte del atacante hacia el dispositivo de la víctima.

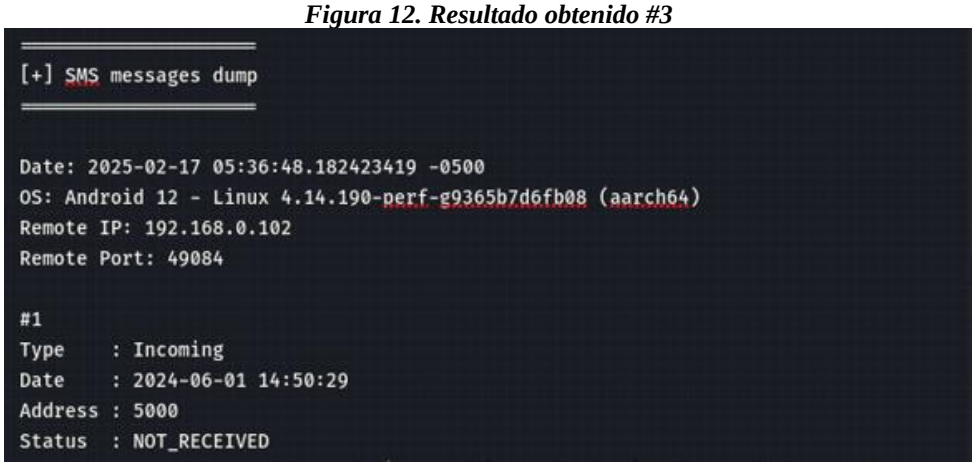
Figura 11. Resultado obtenido #2



Elaborado por: *Remache Kleber, 2025*

Resultado obtenido de la ejecución del comando dump_sms por parte del atacante hacia el dispositivo de la víctima.

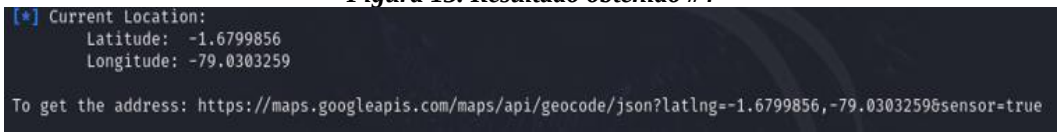
Figura 12. Resultado obtenido #3



Elaborado por: *Remache Kleber, 2025*

Resultado obtenido de la ejecución del comando geolocate y wlan_geolocate por parte del atacante hacia el dispositivo de la víctima.

Figura 13. Resultado obtenido #4



Elaborado por: *Remache Kleber, 2025*

Resultado obtenido de la ejecución del comando send_sms por parte del atacante hacia el dispositivo de la víctima.

Figura 14. Resultado obtenido #5

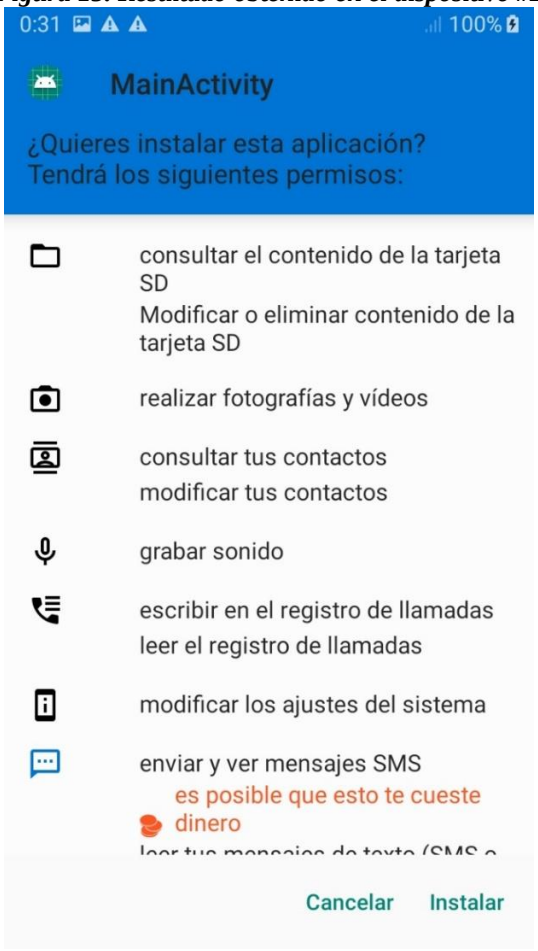
```
msf6 exploit(multi/handler) > [-] You must enter both a destination address -d and the SMS text body -t
[-] Unknown command: [-]. Run the help command for more details.
```

Elaborado por: *Remache Kleber, 2025*

DISPOSITIVO #2

Visualización de la instalación en el dispositivo móvil #2

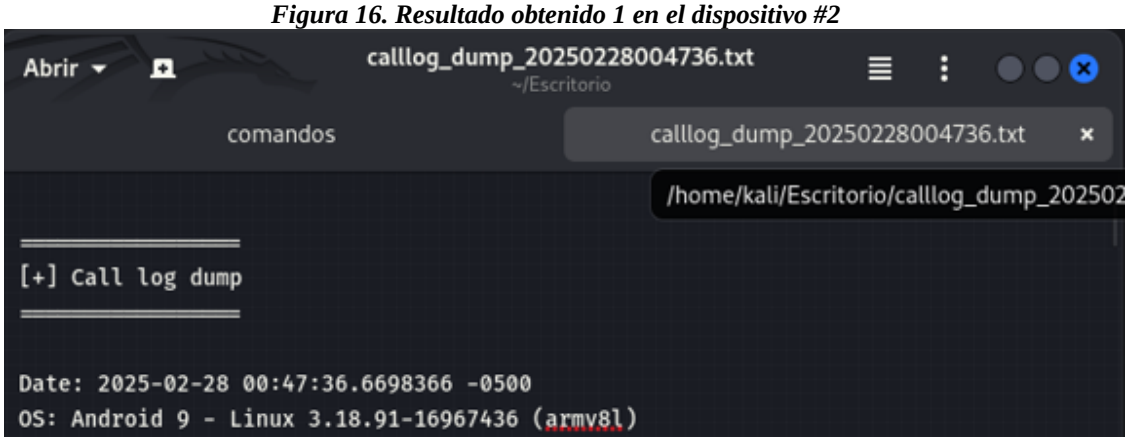
Figura 15. Resultado obtenido en el dispositivo #2



Elaborado por: *Remache Kleber, 2025*

Resultado obtenido de la ejecución del comando dump_calllog por parte del atacante hacia el dispositivo de la víctima.

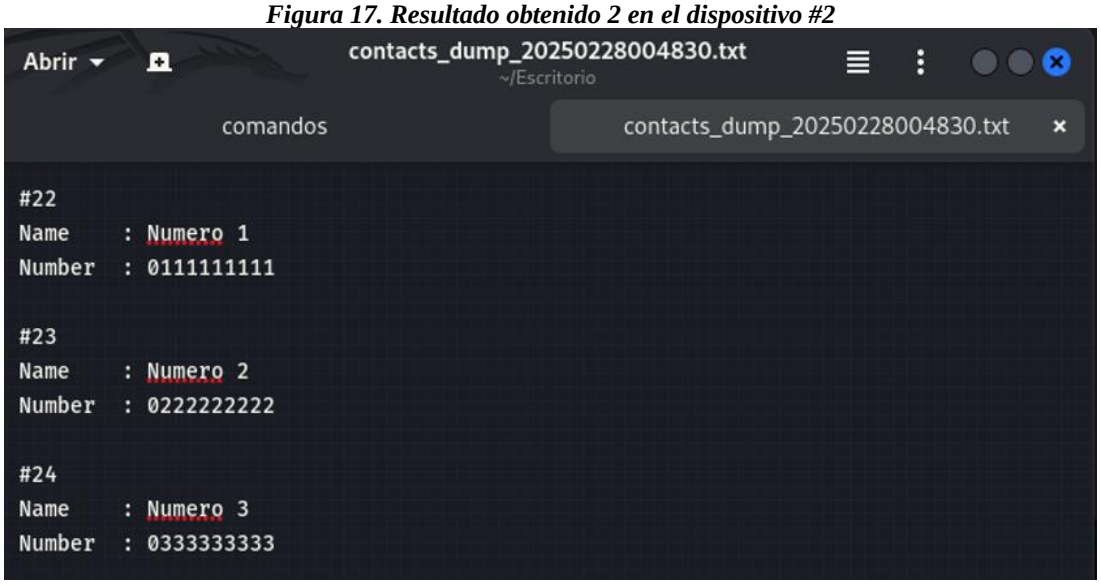
Figura 16. Resultado obtenido 1 en el dispositivo #2



Elaborado por: *Remache Kleber, 2025*

Resultado obtenido de la ejecución del comando dump_contacts por parte del atacante hacia el dispositivo de la víctima.

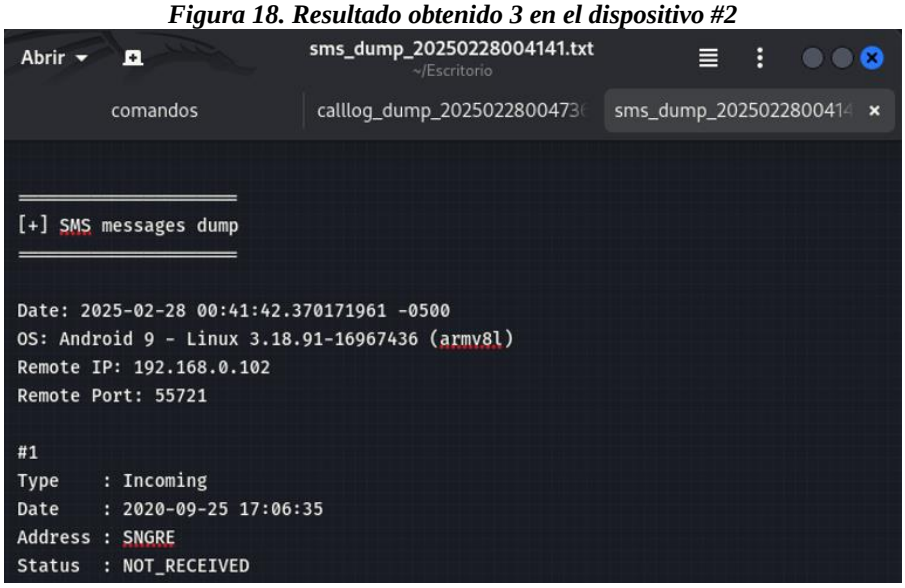
Figura 17. Resultado obtenido 2 en el dispositivo #2



Elaborado por: *Remache Kleber, 2025*

Resultado obtenido de la ejecución del comando dump_sms por parte del atacante hacia el dispositivo de la víctima.

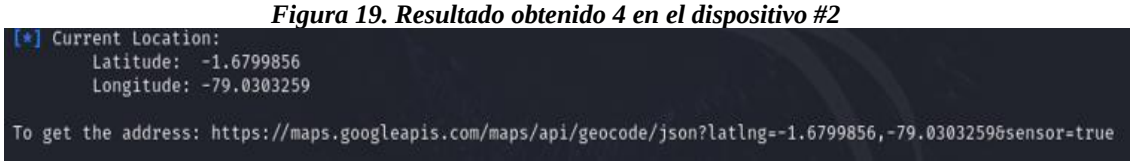
Figura 18. Resultado obtenido 3 en el dispositivo #2



Elaborado por: *Remache Kleber, 2025*

Resultado obtenido de la ejecución del comando geolocate y wlan_geolocate por parte del atacante hacia el dispositivo de la víctima.

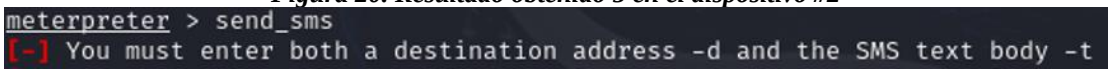
Figura 19. Resultado obtenido 4 en el dispositivo #2



Elaborado por: *Remache Kleber, 2025*

Resultado obtenido de la ejecución del comando send_sms por parte del atacante hacia el dispositivo de la víctima.

Figura 20. Resultado obtenido 5 en el dispositivo #2



Elaborado por: *Remache Kleber, 2025*

CAPITULO V

PROPUESTA

Para la realización del siguiente trabajo de investigación, no es indispensable la presencia de una carta de aceptación debido a que se la aplicará a dispositivos móviles Android en conjunto a la utilización de herramientas de software, una guía de observación y una lista de cotejo, para así poder registrar todos los resultados necesarios durante la finalización de este proyecto investigativo.

Tabla 13. Tasa de éxito gama media para la ejecución del comando: dump_calllog

MODELOS	DISPOSITIVOS	TASA DE ÉXITO
Media	Samsung Galaxy A52	Si
	Samsung Galaxy A25	Si
	Xiaomi Redmi Note 11	No
	Xiaomi Redmi 12	Si
	Xiaomi Redmi note 13 Pro Plus	Si
	POCO X3 Pro	No
	Samsung Galaxy A26	No
	Xiaomi Redmi Note 9	Si
	Poco X5	Si
	Xiaomi Redmi Note 10	No
	Xiaomi Redmi Note 10 T	Si
	Poco F4	Si
	Xiaomi Redmi Note 8 Pro	Si
	Samsung Galaxy A56	Si
	Xiaomi Redmi Note 9S	Si
	Samsung Galaxy A36	No
	Xiaomi Redmi Note 8	Si
	Samsung Galaxy A25	Si
	POCO M4 Pro	No
	Xiaomi Redmi Note 9S	Si
Total		20

Elaborado por: *Remache Kleber, 2025*

RESULTADOS OBTENIDOS:

Total de datos: 20

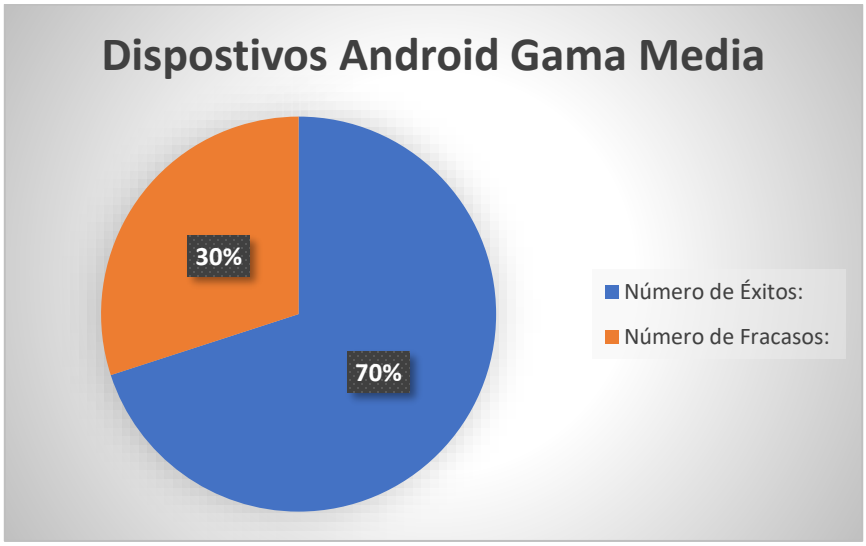
Número de Éxitos: 14

Porcentaje de éxitos: (14/20) * 100 = 70%

Número de Fracasos: 6

Porcentaje de fracasos: (6/20) * 100 = 30%

Figura 21Dispostivos Android Gama Media: comando dump_calllog



Elaborado por: *Remache Kleber, 2025*

ANÁLISIS DE RESULTADOS

El análisis del comando dump_calllog en dispositivos de gama media revela una tasa de éxito del 70%, lo que indica que, en la mayoría de los casos, el playload logra extraer el registro de llamadas de los dispositivos evaluados. Sin embargo, el 30% de fallos sugiere la existencia de restricciones en ciertos modelos, posiblemente debido a políticas de seguridad reforzadas, versiones de Android con mayor control sobre el acceso a datos sensibles o la implementación de mecanismos de protección a nivel de software por parte de los fabricantes.

Tabla 14. Tasa de éxito gama baja para la ejecución del comando: dump_calllog

MODELOS	DISPOSITIVOS	TASA DE ÉXITO
Baja	Samsung Galaxy J2 Core	Si
	POCO M5	Si
	Samsung Galaxy 3	No
	Xiaomi Redmi 7	Si
	Samsung Galaxy M13	Si
	Samsung Galaxy J7 prime	No
	Xiaomi Redmi 12C	No
	Samsung Galaxy J2 prime	Si
	POCO C31	No
	Samsung Galaxy J7 Neo	Si
	Xiaomi Redmi 9A	No
	Xiaomi Redmi Note 6	Si
	Samsung Galaxy J3	No
	POCO M2	No
	Xiaomi Redmi 7	Si
	Samsung Galaxy J5	Si
	Xiaomi Redmi 8	No
	POCO M3	No
	Xiaomi Redmi 9C	Si
	POCO X3 PRO	Si
	Xiaomi Mi 9 Lite	No
	Samsung GalaxyA03	No
	POCO X2	Si
	Xiaomi Redmi 7A	Si
	POCO F3	Si
	Samsung Galaxy A10	Si
	Xiaomi Redmi 8A	No
	Xiaomi Redmi K20	No
	POCO C3	No
	Samsung Galaxy M11	No
Total		30

Elaborado por: *Remache Kleber, 2025*

RESULTADOS OBTENIDOS:

Total de datos: 30

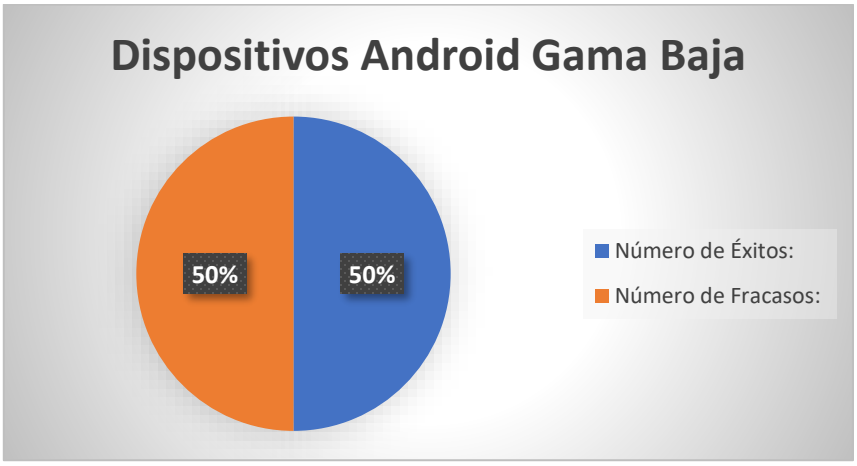
Número de Éxitos: 15

Porcentaje de éxitos: (15/30) * 100 = 50%

Número de Fracasos: 15

Porcentaje de fracasos: (15/30) * 100 = 50%

Figura 22. Dispositivos Android Gama Baja: comando dump_calllog



Elaborado por: *Remache Kleber, 2025*

ANÁLISIS DE RESULTADOS

El análisis del comportamiento del comando dump_calllog en dispositivos de gama baja revela una tasa de éxito del 50%, lo que indica que la efectividad del playload es limitada en este segmento. En conclusión, el análisis en dispositivos de gama media, la tasa de éxito alcanzó un 70%, lo que indica que en la mayoría de los casos fue posible extraer el registro de llamadas, aunque un 30% de fallos señala la existencia de restricciones, posiblemente relacionadas con políticas de seguridad más estrictas o actualizaciones de Android que limitan el acceso a datos sensibles. En dispositivos de gama baja, la tasa de éxito descendió al 50%, reflejando una efectividad reducida en comparación con los dispositivos de gama media.

Tabla 15. Tasa de éxito gama media para la ejecución del comando: dump_contacts

MODELOS	DISPOSITIVOS	TASA DE ÉXITO
Media	Samsung Galaxy A52	Si
	Samsung Galaxy A25	Si
	Xiaomi Redmi Note 11	No
	Xiaomi Redmi 12	No
	Xiaomi Redmi note 13 Pro Plus	Si
	POCO X3 Pro	No
	Samsung Galaxy A26	No
	Xiaomi Redmi Note 9	Si
	Poco X5	Si
	Xiaomi Redmi Note 10	Si
	Xiaomi Redmi Note 10 T	No
	Poco F4	Si
	Xiaomi Redmi Note 8 Pro	No
	Samsung Galaxy A56	No
	Xiaomi Redmi Note 9S	Si
	Samsung Galaxy A36	Si
	Xiaomi Redmi Note 8	Si
	Samsung Galaxy A25	No
	POCO M4 Pro	Si
	Xiaomi Redmi Note 9S	Si
Total		20

Elaborado por: *Remache Kleber, 2025*

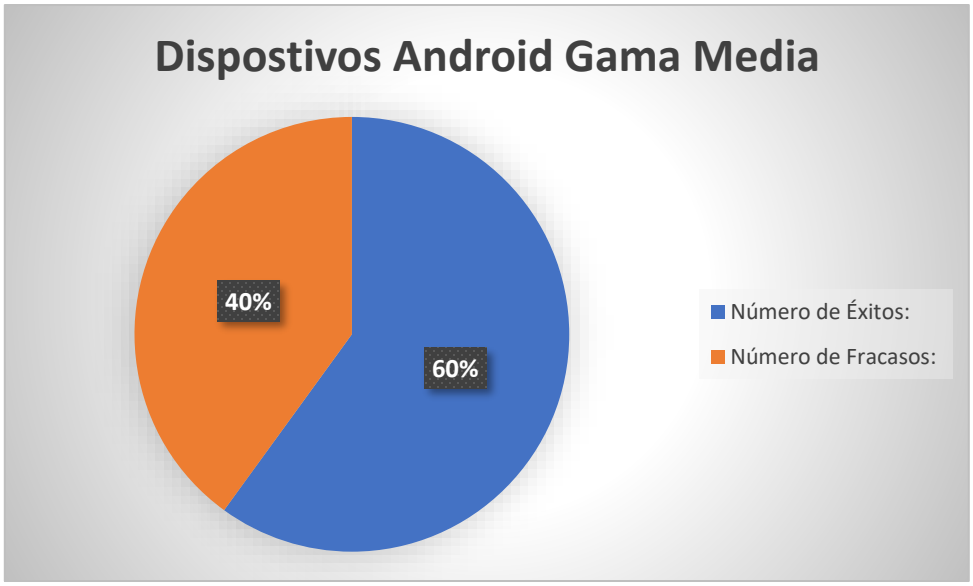
RESULTADOS OBTENIDOS:

Total de datos: 20

Número de Éxitos: 12
Porcentaje de Éxitos: (12/20) ×100=60%

Número de Fracasos: 8
Porcentaje de Fracasos: (8/20) ×100=40%

Figura 23. Dispositivos Android Gama Media: comando dump_contacts



Elaborado por: *Remache Kleber, 2025*

ANÁLISIS DE RESULTADOS

El análisis revela que la ejecución del comando dump_contacts en dispositivos de gama media obtuvo una tasa de éxito del 60%, lo que indica que en la mayoría de los casos fue posible acceder a la información de los contactos almacenados en el dispositivo. Sin embargo, un 40% de los intentos resultaron fallidos, lo que evidencia que una proporción significativa de dispositivos presenta mecanismos que impiden o dificultan la ejecución del comando.

Tabla 16.Tasa de éxito gama baja para la ejecución del comando: dump_contacts

MODELOS	DISPOSITIVOS	TASA DE ÉXITO
Baja	Samsung Galaxy J2 Core	Si
	POCO M5	Si
	Samsung Galaxy 3	No
	Xiaomi Redmi 7	No
	Samsung Galaxy M13	Si
	Samsung Galaxy J7 prime	Si
	Xiaomi Redmi 12C	No
	Samsung Galaxy J2 prime	Si
	POCO C31	No
	Samsung Galaxy J7 Neo	Si
	Xiaomi Redmi 9A	No
	Xiaomi Redmi Note 6	Si
	Samsung Galaxy J3	No
	POCO M2	No
	Xiaomi Redmi Note 7	Si
	Samsung Galaxy J5	No
	Xiaomi Redmi 8	Si
	POCO M3	No
	Xiaomi Redmi 9C	No
	POCO X3 PRO	Si
	Xiaomi Mi 9 Lite	No
	Samsung GalaxyA03	No
	POCO X2	Si
	Xiaomi Redmi 7A	Si
	POCO F3	Si
	Samsung Galaxy A10	Si
	Xiaomi Redmi 8A	No
	Xiaomi Redmi K20	No
	POCO C3	Si
	Samsung Galaxy M11	Si
Total		30

Elaborado por: *Remache Kleber, 2025*

RESULTADOS OBTENIDOS:

Total de datos: 30

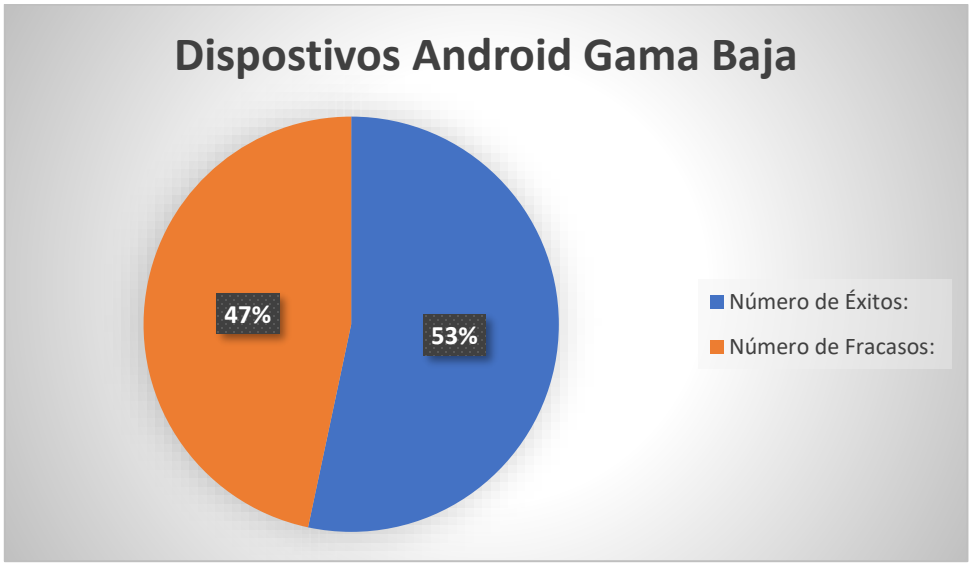
Número de Éxitos: 16

Porcentaje de Éxitos: (16/30) ×100=53.33%

Número de Fracasos: 14

Porcentaje de Fracasos: (14/30) ×100=46.67%

Figura 24. Dispositivos Android Gama Baja: comando dump_contacts



Elaborado por: *Remache Kleber, 2025*

ANÁLISIS DE RESULTADOS

El análisis de la ejecución del comando dump_contacts en dispositivos de gama baja indica una tasa de éxito del 53.33%, lo que sugiere que poco más de la mitad de los dispositivos permiten extraer la información de contactos, mientras que un 46.67% impiden su ejecución.

En conclusión, el análisis de la ejecución del comando dump_contacts en dispositivos de gama media y baja muestra que, aunque en ambos casos se logró acceder a la

información de los contactos en más de la mitad de los dispositivos, existen barreras de seguridad notables en ambos segmentos. En dispositivos de gama media, la tasa de éxito fue del 60%, con un 40% de fallos, lo que refleja la presencia de mecanismos de protección que dificultan la ejecución del comando. En dispositivos de gama baja, la tasa de éxito disminuyó a un 53.33%, con un 46.67% de fallos, lo que indica que las restricciones de seguridad en este segmento son aún más significativas.

Tabla 17.Tasa de éxito gama media para la ejecución del comando: dump_sms

Modelos	Dispositivos	Tasa de éxito
Media	Samsung Galaxy A52	Si
	Samsung Galaxy A25	Si
	Xiaomi Redmi Note 11	Si
	Xiaomi Redmi 12	Si
	Xiaomi Redmi note 13 Pro Plus	Si
	POCO X3 Pro	Si
	Samsung Galaxy A26	Si
	Xiaomi Redmi Note 9	Si
	Poco X5	Si
	Xiaomi Redmi Note 10	Si
	Xiaomi Redmi Note 10 T	Si
	Poco F4	Si
	Xiaomi Redmi Note 8 Pro	Si
	Samsung Galaxy A56	No
	Xiaomi Redmi Note 9S	Si
	Samsung Galaxy A36	Si
	Xiaomi Redmi Note 8	Si
	Samsung Galaxy A25	No
	POCO M4 Pro	Si
	Xiaomi Redmi Note 9S	Si
Total		20

Elaborado por: *Remache Kleber, 2025*

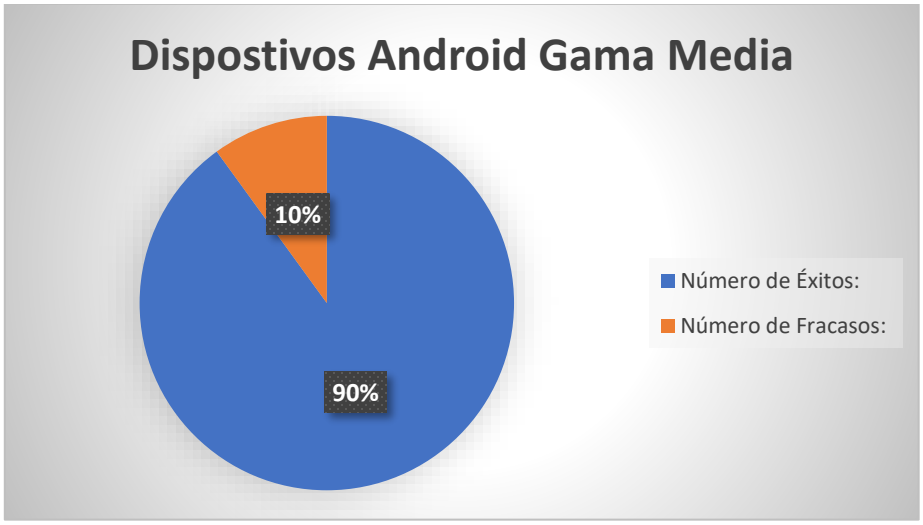
RESULTADOS OBTENIDOS:

Total de datos: 20

Número de Éxitos: 18
Porcentaje de Éxitos: (18/20) ×100=90%

Número de Fracasos: 2
Porcentaje de Fracasos: (2/20) ×100=10%

Figura 25. Dispositivos Android Gama Media: comando dump_sms



Elaborado por: *Remache Kleber, 2025*

ANÁLISIS DE RESULTADOS

Los resultados obtenidos muestran que el comando dump_sms tiene una alta tasa de éxito del 90%, lo que indica que en la mayoría de los dispositivos de gama media se logró extraer los mensajes de texto. Solo el 10% de los dispositivos fallaron en la ejecución de este comando.

Tabla 18.Tasa de éxito gama baja para la ejecución del comando: dump_sms

Modelos	Dispositivos	Tasa de éxito
Baja	Samsung Galaxy J2 Core	Si
	POCO M5	No
	Samsung Galaxy 3	No
	Xiaomi Redmi 7	Si
	Samsung Galaxy M13	No
	Samsung Galaxy J7 prime	Si
	Xiaomi Redmi 12C	No
	Samsung Galaxy J2 prime	Si
	POCO C31	Si
	Samsung Galaxy J7 Neo	Si
	Xiaomi Redmi 9A	No
	Xiaomi Redmi Note 6	Si
	Samsung Galaxy J3	No
	POCO M2	Si
	Xiaomi Redmi Note 7	Si
	Samsung Galaxy J5	No
	Xiaomi Redmi 8	No
	POCO M3	Si
	Xiaomi Redmi 9C	Si
	POCO X3 PRO	No
	Xiaomi Mi 9 Lite	No
	Samsung GalaxyA03	Si
	POCO X2	Si
	Xiaomi Redmi 7A	Si
	POCO F3	Si
	Samsung Galaxy A10	No
	Xiaomi Redmi 8A	No
	Xiaomi Redmi K20	Si
	POCO C3	No
	Samsung Galaxy M11	Si
Total		30

Elaborado por: *Remache Kleber, 2025*

RESULTADOS OBTENIDOS:

Total de datos: 30

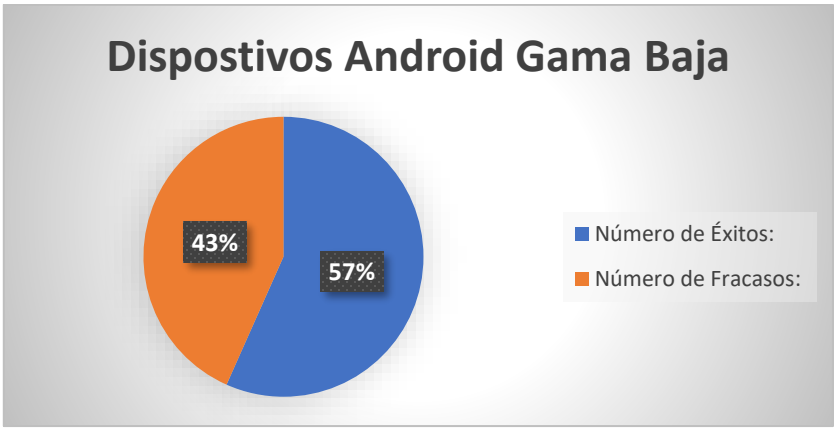
Número de Éxitos: 17

Porcentaje de Éxitos: (17/30) ×100=56.67%

Número de Fracasos: 13

Porcentaje de Fracasos: (13/30) ×100=43.33%

Figura 26.Dispositivos Android Gama baja: comando dump_sms



Elaborado por: *Remache Kleber, 2025*

ANÁLISIS DE RESULTADOS

Los resultados indican que el comando dump_sms tiene una tasa de éxito del 56.67% en dispositivos de gama baja, lo que significa que, en más de la mitad de los casos, el comando logró extraer los mensajes de texto. Sin embargo, el 43.33% de los dispositivos fallaron en la ejecución.

En conclusión, el análisis de la ejecución del comando dump_sms muestra una notable diferencia en su efectividad entre dispositivos de gama media y baja. En dispositivos de gama media, la tasa de éxito alcanza un 90%, lo que indica que el playload tiene una alta capacidad para extraer los mensajes de texto, con solo un 10% de fallos. En contraste, en dispositivos de gama baja, la tasa de éxito disminuye significativamente a un 56.67%, reflejando que, aunque en más de la mitad de los casos se logró extraer los mensajes, el 43.33% de los dispositivos presentaron fallos, lo que sugiere mayores barreras de seguridad y restricciones.

6 CONCLUSIONES

- Esta investigación logró describir, mediante Hacking Ético y el uso de playload, las vulnerabilidades inherentes a dispositivos móviles Android, cumpliendo el objetivo general.
- Las pruebas en un entorno controlado, utilizando dispositivos móviles que función con sistema operativo Android como "víctimas", demostraron la facilidad con que agentes maliciosos pueden sustraer datos sensibles como contactos, registro de mensajes, historial de llamadas y geolocalización al explotar fallos de seguridad. Este proceso evidenció cómo la falta de conciencia del usuario, combinada con técnicas de ingeniería social, amplifica el riesgo al instalar un APK maliciosos, alineándose con el objetivo específico de detallar las consecuencias de dichas descargas.
- La sistematización de los resultados y se clasificó en tabla de clasificación, cumpliendo así el segundo objetivo específico al categorizar vulnerabilidades según su probabilidad de explotación (baja, media, alta) y su impacto (menor, moderado, catastrófico). Los Comandos como “dump_sms”, “dump_calllog”, “dump_contacts” “send_sms”, “geolocate” y “wlan_geolocate” demostraron un potencial catastrófico, permitiendo al atacante no solo comprometer al usuario inicial, sino también expandir su alcance a través de contactos vulnerados.
- Este proceso permitió la documentación exhaustiva, desde la generación de la herramienta playload con MSFVenom hasta su ejecución vía Metasploit Framework, cumpliendo al tercer objetivo específico. Estos hallazgos resaltan la criticidad de proteger la privacidad en Android, un sistema expuesto por su fragmentación y baja tasa de actualización (solo 40% de usuarios). En síntesis, el Hacking Ético revela las debilidades explotables, proporcionando una base para estrategias de mitigación y sensibilización frente a amenazas cibernéticas.

7 RECOMENDACIONES

- Dominar el manejo de Kali Linux, comprendiendo la funcionalidad de los comandos en la terminal para evitar configuraciones erróneas que comprometan el sistema. Se aconseja mantener actualizadas las herramientas de software, como MSFVenom y Metasploit, para prevenir fallos operativos durante las pruebas.
- Antes de iniciar el entorno de pruebas, validar la dirección IP y el puerto del equipo atacante, asegurándose de que este último esté libre; de lo contrario, asignar uno alternativo para garantizar la conectividad. Finalmente concluir adecuadamente cada sesión de prueba, cerrando procesos activos que puedan sobrecargar recursos y ralentizar el sistema operativo del equipo de prueba.
- Examinar los permisos solicitados por las aplicaciones antes de su instalación en dispositivos Android, priorizando la seguridad de los datos sensibles frente a descargas no verificadas. Activando la autenticación de dos factores en aplicaciones críticas y durante la navegación, fortaleciendo la protección contra accesos no autorizados en dispositivos móviles.
- Comparar los riesgos de instalar aplicaciones externas frente a las obtenidas desde Google Play Store, considerando que muchas fuentes no oficiales albergan malware o exploits detectados previamente en la tienda oficial. Desconfiando supuestas ofertas de contenido gratuito, ya que la ingeniería social es una táctica frecuente para inducir la instalación de APK maliciosos y ampliar el alcance de los atacantes.

8 BIBLIOGRAFÍA

Android Debug Bridge (adb) | Android Studio. (s. f.). Android Developers. Recuperado 26 de octubre de 2024, de https://developer.android.com/tools/adb?hl=es-419

Barrezueta, H. D. P. (2021). *DIRECTOR DEL REGISTRO OFICIAL*.

Android Debug Bridge (adb) | Android Studio. (s. f.). Android Developers. Recuperado 26 de octubre de 2024, de https://developer.android.com/tools/adb?hl=es-419

AV-Comparatives. (2024, diciembre 25). IT Security Survey 2025. *AV-Comparatives*. https://www.av-comparatives.org/surveys/it-security-survey-2025/

Barrezueta, H. D. P. (2021). *DIRECTOR DEL REGISTRO OFICIAL*. https://www.finanzaspopulares.gob.ec/wp-content/uploads/2021/07/ley_organica_de_proteccion_de_datos_personales.pdf

Bilić, D. (2020, enero 15). *Cómo funciona el malware dirigido a dispositivos móviles*. https://www.welivesecurity.com/la-es/2020/01/15/como-funciona-el-malware-dirigido-a-dispositivos-moviles/

Bodnar, D. (2020, octubre 29). *Ingeniería social y cómo protegerse*. Ingeniería social y cómo protegerse. https://www.avast.com/es-es/c-social-engineering

Business of Apps. (2024). *Android Version Adoption Rates (2025)*. Business of Apps. https://www.businessofapps.com/data/android-version-adoption-rates/

Ciberpyme. (2023, agosto 14). Vulnerabilidades críticas en móviles Android. *Revista de Ciberseguridad y Seguridad de la Información para Empresas y*

Organismos Públicos. https://www.ciberseguridadpyme.es/actualidad/vulnerabilidades-criticas-en-moviles-android/

Cilleruelo, C. (2022, octubre 5). *Tipos de payload | KeepCoding Bootcamps*. https://keepcoding.io/blog/tipos-de-payload/

counterpointresearch. (2024, Q4). *Global Smartphone Sales Share by Operating System*. https://www.counterpointresearch.com/insights/global-smartphone-os-market-share/

kaspersky. (2017, noviembre 30). *Amenazas para la seguridad móvil | Problemas de seguridad de Android*. /. https://latam.kaspersky.com/resource-center/threats/mobile

Klischies, D., Mackensen, P., & Moonsamy, V. (2024). *Vulnerability, Where Art Thou? An Investigation of Vulnerability Management in Android Smartphone Chipsets*. https://doi.org/10.14722/ndss.2025.241161

Lopez, V. (2023, julio 21). *Vulnerabilidades en Android 2023*. S2 Grupo. https://s2grupo.es/vulnerabilidades-en-android-2023/

Onofa, M. (2022, junio 30). Ataques cibernéticos amenazan seguridad en Ecuador. *Diálogo Américas*. https://dialogo-americas.com/es/articles/ataques-ciberneticos-amenazan-seguridad-en-ecuador/

petrecere. (2023, septiembre 8). Guía completa sobre MSFVENOM. *RONBHACK*. https://ronbhack.com/guia-completa-sobre-msfvenom/

¿Qué es Kali Linux? (2023, agosto 23). IONOS Digital Guide. https://www.ionos.com/es-us/digitalguide/servidores/configuracion/kali-linux/

Revilla, J. (2024, febrero 20). *Las grandes amenazas que se ciernen sobre los dispositivos móviles en este año*. Escudo Digital. https://www.escudodigital.com/ciberseguridad/grandes-amenazas-se-ciernen-sobre-dispositivos-moviles-en-2024_58125_102.html

Rizaldos, H. (2018, octubre 22). *Qué es Metasploit | OpenWebinars*. OpenWebinars.net. https://openwebinars.net/blog/que-es-metasploit/

Sanchez, D. (2020, mayo 13). *Sesion 4 MSF Venom Encoders | PDF*. Scribd. https://es.scribd.com/presentation/664276788/SESION-4-MSF-VENOM-ENCODERS

Santana, L. L. V. (2023). *INGENIERÍA DE TENOLOGÍAS DE LA INFORMACIÓN*.

Sempere, A. (2021). Universitat Politècnica de València. *Ingeniería del agua*, *18*(1), ix. https://doi.org/10.4995/ia.2014.3293

source.android. (2024, noviembre 4). *Boletín de seguridad de Android de noviembre de 2024*. Android Open Source Project. https://source.android.com/docs/security/bulletin/2024-11-01?hl=es-419

Vaati, E. (2020, julio 2). *Qué es Android SDK y cómo empezar a usarlo | Envato Tuts+*. Code Envato Tuts+. https://code.tutsplus.com/es/the-android-sdk-tutorial--cms-34623t

Vina, A. (2024, abril 10). *What is MediaPipe? A Guide for Beginners*. Roboflow Blog. https://blog.roboflow.com/what-is-mediapipe/

Android Debug Bridge (adb) | Android Studio. (s. f.). Android Developers. Recuperado 26 de octubre de 2024, de https://developer.android.com/tools/adb?hl=es-419

AV-Comparatives. (2024, diciembre 25). IT Security Survey 2025. *AV-Comparatives*. https://www.av-comparatives.org/surveys/it-security-survey-2025/

Barrezueta, H. D. P. (2021). *DIRECTOR DEL REGISTRO OFICIAL*. https://www.finanzaspopulares.gob.ec/wp-content/uploads/2021/07/ley_organica_de_proteccion_de_datos_personales.pdf

Bilić, D. (2020, enero 15). *Cómo funciona el malware dirigido a dispositivos móviles*. https://www.welivesecurity.com/la-es/2020/01/15/como-funciona-el-malware-dirigido-a-dispositivos-moviles/

Bodnar, D. (2020, octubre 29). *Ingeniería social y cómo protegerse*. Ingeniería social y cómo protegerse. https://www.avast.com/es-es/c-social-engineering

Business of Apps. (2024). *Android Version Adoption Rates (2025)*. Business of Apps. https://www.businessofapps.com/data/android-version-adoption-rates/

Ciberpyme. (2023, agosto 14). Vulnerabilidades críticas en móviles Android. *Revista de Ciberseguridad y Seguridad de la Información para Empresas y Organismos Públicos.* https://www.ciberseguridadpyme.es/actualidad/vulnerabilidades-criticas-en-moviles-android/

Cilleruelo, C. (2022, octubre 5). *Tipos de payload | KeepCoding Bootcamps*. https://keepcoding.io/blog/tipos-de-payload/

counterpointresearch. (2024, Q4). *Global Smartphone Sales Share by Operating System*. https://www.counterpointresearch.com/insights/global-smartphone-os-market-share/

kaspersky. (2017, noviembre 30). *Amenazas para la seguridad móvil | Problemas de seguridad de Android*. /. https://latam.kaspersky.com/resource-center/threats/mobile

Klischies, D., Mackensen, P., & Moonsamy, V. (2024). *Vulnerability, Where Art Thou? An Investigation of Vulnerability Management in Android Smartphone Chipsets*. https://doi.org/10.14722/ndss.2025.241161

Lopez, V. (2023, julio 21). *Vulnerabilidades en Android 2023*. S2 Grupo. https://s2grupo.es/vulnerabilidades-en-android-2023/

Onofa, M. (2022, junio 30). Ataques cibernéticos amenazan seguridad en Ecuador. *Diálogo Américas*. https://dialogo-americas.com/es/articles/ataques-ciberneticos-amenazan-seguridad-en-ecuador/

petrecere. (2023, septiembre 8). Guía completa sobre MSFVENOM. *RONBHACK*. https://ronbhack.com/guia-completa-sobre-msfvenom/

¿Qué es Kali Linux? (2023, agosto 23). IONOS Digital Guide. https://www.ionos.com/es-us/digitalguide/servidores/configuracion/kali-linux/

Revilla, J. (2024, febrero 20). *Las grandes amenazas que se ciernen sobre los dispositivos móviles en este año*. Escudo Digital. https://www.escudodigital.com/ciberseguridad/grandes-amenazas-se-ciernen-sobre-dispositivos-moviles-en-2024_58125_102.html

Rizaldos, H. (2018, octubre 22). *Qué es Metasploit | OpenWebinars*. OpenWebinars.net. https://openwebinars.net/blog/que-es-metasploit/

Sanchez, D. (2020, mayo 13). *Sesion 4 MSF Venom Encoders | PDF*. Scribd. https://es.scribd.com/presentation/664276788/SESION-4-MSF-VENOM-ENCODERS

Santana, L. L. V. (2023). *INGENIERÍA DE TENOLOGÍAS DE LA INFORMACIÓN*.

Sempere, A. (2021). Universitat Politècnica de València. *Ingeniería del agua, 18*(1), ix. https://doi.org/10.4995/ia.2014.3293

source.android. (2024, noviembre 4). *Boletín de seguridad de Android de noviembre de 2024*. Android Open Source Project. https://source.android.com/docs/security/bulletin/2024-11-01?hl=es-419

Vaati, E. (2020, julio 2). *Qué es Android SDK y cómo empezar a usarlo | Envato Tuts+*. Code Envato Tuts+. https://code.tutsplus.com/es/the-android-sdk-tutorial--cms-34623t

Vina, A. (2024, abril 10). *What is MediaPipe? A Guide for Beginners*. Roboflow Blog. https://blog.roboflow.com/what-is-mediapipe/

Caballero, A. (2022). Kali Linux. Curso Pràctico. Madrid: Ra-Ma Editorial.

Cilleruelo, C. (2022, octubre 5). *Tipos de payload | KeepCoding Bootcamps*. https://keepcoding.io/blog/tipos-de-payload/

Cilleruelo, C. (2022, octubre 5). *Tipos de payload | KeepCoding Bootcamps*. https://keepcoding.io/blog/tipos-de-payload/

Cilleruelo, C. (2022, octubre 5). *Tipos de payload | KeepCoding Bootcamps*. https://keepcoding.io/blog/tipos-de-payload/

Ciberpyme. (2023, agosto 14). Vulnerabilidades críticas en móviles Android. *Revista de Ciberseguridad y Seguridad de la Información para Empresas y Organismos Públicos.* https://www.ciberseguridadpyme.es/actualidad/vulnerabilidades-criticas-en-moviles-android/

Gonzalèz, P. (2012). Metasploit para Pentesters. Madrid: Ra-Ma Editorial.

kaspersky. (2020, noviembre 30). *Amenazas para la seguridad móvil | Problemas de seguridad de Android*. /. https://latam.kaspersky.com/resource-center/threats/mobile

kaspersky. (2020, noviembre 30). *Amenazas para la seguridad móvil | Problemas de seguridad de Android*. /. https://latam.kaspersky.com/resource-center/threats/mobile

Android Debug Bridge (adb) | Android Studio. (s. f.). Android Developers. Recuperado 26 de octubre de 2024, de https://developer.android.com/tools/adb?hl=es-419

AV-Comparatives. (2024, diciembre 25). IT Security Survey 2025. *AV-Comparatives*. https://www.av-comparatives.org/surveys/it-security-survey-2025/

Barrezueta, H. D. P. (2021). *DIRECTOR DEL REGISTRO OFICIAL*. https://www.finanzaspopulares.gob.ec/wp-content/uploads/2021/07/ley_organica_de_proteccion_de_datos_personales.pdf

Bilić, D. (2020, enero 15). *Cómo funciona el malware dirigido a dispositivos móviles*. https://www.welivesecurity.com/la-es/2020/01/15/como-funciona-el-malware-dirigido-a-dispositivos-moviles/

Bodnar, D. (2020, octubre 29). *Ingeniería social y cómo protegerse*. Ingeniería social y cómo protegerse. https://www.avast.com/es-es/c-social-engineering

Business of Apps. (2024). *Android Version Adoption Rates (2025)*. Business of Apps. https://www.businessofapps.com/data/android-version-adoption-rates/

Ciberpyme. (2023, agosto 14). Vulnerabilidades críticas en móviles Android. *Revista de Ciberseguridad y Seguridad de la Información para Empresas y Organismos Públicos.* https://www.ciberseguridadpyme.es/actualidad/vulnerabilidades-criticas-en-moviles-android/

Cilleruelo, C. (2022, octubre 5). *Tipos de payload | KeepCoding Bootcamps*. https://keepcoding.io/blog/tipos-de-payload/

counterpointresearch. (2024, Q4). *Global Smartphone Sales Share by Operating System*. https://www.counterpointresearch.com/insights/global-smartphone-os-market-share/

kaspersky. (2017, noviembre 30). *Amenazas para la seguridad móvil | Problemas de seguridad de Android*. /. https://latam.kaspersky.com/resource-center/threats/mobile

Klischies, D., Mackensen, P., & Moonsamy, V. (2024). *Vulnerability, Where Art Thou? An Investigation of Vulnerability Management in Android Smartphone Chipsets*. https://doi.org/10.14722/ndss.2025.241161

Lopez, V. (2023, julio 21). *Vulnerabilidades en Android 2023*. S2 Grupo. https://s2grupo.es/vulnerabilidades-en-android-2023/

Onofa, M. (2022, junio 30). Ataques cibernéticos amenazan seguridad en Ecuador. *Diálogo Américas*. https://dialogo-americas.com/es/articles/ataques-ciberneticos-amenazan-seguridad-en-ecuador/

petrecere. (2023, septiembre 8). Guía completa sobre MSFVENOM. *RONBHACK*. https://ronbhack.com/guia-completa-sobre-msfvenom/

¿Qué es Kali Linux? (2023, agosto 23). IONOS Digital Guide. https://www.ionos.com/es-us/digitalguide/servidores/configuracion/kali-linux/

Revilla, J. (2024, febrero 20). *Las grandes amenazas que se ciernen sobre los dispositivos móviles en este año*. Escudo Digital. https://www.escudodigital.com/ciberseguridad/grandes-amenazas-se-ciernen-sobre-dispositivos-moviles-en-2024_58125_102.html

Rizaldos, H. (2018, octubre 22). *Qué es Metasploit | OpenWebinars*. OpenWebinars.net. https://openwebinars.net/blog/que-es-metasploit/

Sanchez, D. (2020, mayo 13). *Sesion 4 MSF Venom Encoders | PDF*. Scribd. https://es.scribd.com/presentation/664276788/SESION-4-MSF-VENOM-ENCODERS

Santana, L. L. V. (2023). *INGENIERÍA DE TENOLOGÍAS DE LA INFORMACIÓN*.

Sempere, A. (2021). Universitat Politècnica de València. *Ingeniería del agua*, *18*(1), ix. https://doi.org/10.4995/ia.2014.3293

source.android. (2024, noviembre 4). *Boletín de seguridad de Android de noviembre de 2024*. Android Open Source Project. https://source.android.com/docs/security/bulletin/2024-11-01?hl=es-419

Vaati, E. (2020, julio 2). *Qué es Android SDK y cómo empezar a usarlo | Envato Tuts+*. Code Envato Tuts+. https://code.tutsplus.com/es/the-android-sdk-tutorial--cms-34623t

Vina, A. (2024, abril 10). *What is MediaPipe? A Guide for Beginners*. Roboflow Blog. https://blog.roboflow.com/what-is-mediapipe/

Android Debug Bridge (adb) | Android Studio. (s. f.). Android Developers. Recuperado 26 de octubre de 2024, de https://developer.android.com/tools/adb?hl=es-419

AV-Comparatives. (2024, diciembre 25). IT Security Survey 2025. *AV-Comparatives*. https://www.av-comparatives.org/surveys/it-security-survey-2025/

Barrezueta, H. D. P. (2021). *DIRECTOR DEL REGISTRO OFICIAL*. https://www.finanzaspopulares.gob.ec/wp-content/uploads/2021/07/ley_organica_de_proteccion_de_datos_personales.pdf

Bilić, D. (2020, enero 15). *Cómo funciona el malware dirigido a dispositivos móviles*. https://www.welivesecurity.com/la-es/2020/01/15/como-funciona-el-malware-dirigido-a-dispositivos-moviles/

Bodnar, D. (2020, octubre 29). *Ingeniería social y cómo protegerse*. Ingeniería social y cómo protegerse. https://www.avast.com/es-es/c-social-engineering

Business of Apps. (2024). *Android Version Adoption Rates (2025)*. Business of Apps. https://www.businessofapps.com/data/android-version-adoption-rates/

Ciberpyme. (2023, agosto 14). Vulnerabilidades críticas en móviles Android. *Revista de Ciberseguridad y Seguridad de la Información para Empresas y Organismos Públicos.* https://www.ciberseguridadpyme.es/actualidad/vulnerabilidades-criticas-en-moviles-android/

Cilleruelo, C. (2022, octubre 5). *Tipos de payload | KeepCoding Bootcamps*. https://keepcoding.io/blog/tipos-de-payload/

counterpointresearch. (2024, Q4). *Global Smartphone Sales Share by Operating System*. https://www.counterpointresearch.com/insights/global-smartphone-os-market-share/

kaspersky. (2017, noviembre 30). *Amenazas para la seguridad móvil | Problemas de seguridad de Android*. /. https://latam.kaspersky.com/resource-center/threats/mobile

Klischies, D., Mackensen, P., & Moonsamy, V. (2024). *Vulnerability, Where Art Thou? An Investigation of Vulnerability Management in Android Smartphone Chipsets*. https://doi.org/10.14722/ndss.2025.241161

Lopez, V. (2023, julio 21). *Vulnerabilidades en Android 2023*. S2 Grupo. https://s2grupo.es/vulnerabilidades-en-android-2023/

Onofa, M. (2022, junio 30). Ataques cibernéticos amenazan seguridad en Ecuador. *Diálogo Américas*. https://dialogo-americas.com/es/articles/ataques-ciberneticos-amenazan-seguridad-en-ecuador/

petrecere. (2023, septiembre 8). Guía completa sobre MSFVENOM. *RONBHACK*. https://ronbhack.com/guia-completa-sobre-msfvenom/

¿Qué es Kali Linux? (2023, agosto 23). IONOS Digital Guide. https://www.ionos.com/es-us/digitalguide/servidores/configuracion/kali-linux/

Revilla, J. (2024, febrero 20). *Las grandes amenazas que se ciernen sobre los dispositivos móviles en este año*. Escudo Digital. https://www.escudodigital.com/ciberseguridad/grandes-amenazas-se-ciernen-sobre-dispositivos-moviles-en-2024_58125_102.html

Rizaldos, H. (2018, octubre 22). *Qué es Metasploit | OpenWebinars*. OpenWebinars.net. https://openwebinars.net/blog/que-es-metasploit/

Sanchez, D. (2020, mayo 13). *Sesion 4 MSF Venom Encoders | PDF*. Scribd. https://es.scribd.com/presentation/664276788/SESION-4-MSF-VENOM-ENCODERS

Santana, L. L. V. (2023). *INGENIERÍA DE TENOLOGÍAS DE LA INFORMACIÓN*.

Sempere, A. (2021). Universitat Politècnica de València. *Ingeniería del agua*, *18*(1), ix. https://doi.org/10.4995/ia.2014.3293

source.android. (2024, noviembre 4). *Boletín de seguridad de Android de noviembre de 2024*. Android Open Source Project. https://source.android.com/docs/security/bulletin/2024-11-01?hl=es-419

Vaati, E. (2020, julio 2). *Qué es Android SDK y cómo empezar a usarlo | Envato Tuts+*. Code Envato Tuts+. https://code.tutsplus.com/es/the-android-sdk-tutorial--cms-34623t

Vina, A. (2024, abril 10). *What is MediaPipe? A Guide for Beginners*. Roboflow Blog. https://blog.roboflow.com/what-is-mediapipe/

Onofa, M. (2022, junio 30). Ataques cibernéticos amenazan seguridad en Ecuador. *Diálogo Américas*. https://dialogo-americas.com/es/articles/ataques-ciberneticos-amenazan-seguridad-en-ecuador/

petrecere. (2023, septiembre 8). Guía completa sobre MSFVENOM. *RONBHACK*. https://ronbhack.com/guia-completa-sobre-msfvenom/

¿Qué es Kali Linux? (2023, agosto 23). IONOS Digital Guide. https://www.ionos.com/es-us/digitalguide/servidores/configuracion/kali-linux/

Revilla, J. (2024, febrero 20). *Las grandes amenazas que se ciernen sobre los dispositivos móviles en este año*. Escudo Digital. https://www.escudodigital.com/ciberseguridad/grandes-amenazas-se-ciernen-sobre-dispositivos-moviles-en-2024_58125_102.html

Rizaldos, H. (2018, octubre 22). *Qué es Metasploit | OpenWebinars*. OpenWebinars.net. https://openwebinars.net/blog/que-es-metasploit/

Sanchez, D. (2020, mayo 13). *Sesion 4 MSF Venom Encoders | PDF*. Scribd. https://es.scribd.com/presentation/664276788/SESION-4-MSF-VENOM-ENCODERS

Santana, L. L. V. (2023). *INGENIERÍA DE TENOLOGÍAS DE LA INFORMACIÓN*.

Sempere, A. (2021). Universitat Politècnica de València. *Ingeniería del agua*, *18*(1), ix. https://doi.org/10.4995/ia.2014.3293

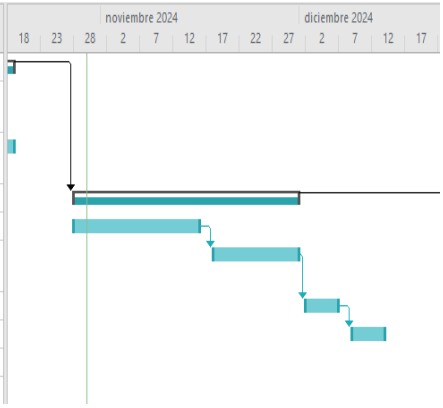
Vaati, E. (2020, julio 2). *Qué es Android SDK y cómo empezar a usarlo | Envato Tuts+*. Code Envato Tuts+. https://code.tutsplus.com/es/the-android-sdk-tutorial--cms-34623t

Zúñiga Ledy, G. C. (10 de Febrero de 2014). *Organization of American States.* Obtenido de https://www.oas.org/juridico/PDFs/mesicic5_ecu_ane_con_judi_c%C3%B3d_org_int_pen.pdf

ANEXOS

9.1 Cronograma (Gantt)

Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
	Recolección de información	**25 días?**	**lun 16/9/24**	**vie 18/10/24**		
	Recopilación de la información sobre los playload	11 días	lun 16/9/24	lun 30/9/24		
	Identificación de herramientas de software	14 días	mar 1/10/24	vie 18/10/24	2	
	Realización de pruebas	**26 días?**	**lun 28/10/24**	**sáb 30/11/24**	1	
	Preparación del entorno de prueb	15 días	lun 28/10/24	vie 15/11/24		
	Implementación de cargas útiles utilizando herramientas	11 días	lun 18/11/24	sáb 30/11/24	5	
	Recomendaciones	5 días	lun 2/12/24	vie 6/12/24	6	
	Conclusión	5 días	lun 9/12/24	vie 13/12/24	7	
	Finalización de la investigación	**10 días**	**lun 6/1/25**	**vie 17/1/25**	4	
	Vulnerabilidades encontradas	5 días	lun 6/1/25	vie 10/1/25		



Elaborado por: *Remache Kleber, 2025*

9.2 Presupuesto Ejecutado

Categoría 1: Equipos		
Rubros	**Cantidad**	**Precio**
Laptop (HP)	1	$670.00
Dispositivo móvil	1	$270.00
Router	1	$25.00
	Total 1:	$965.00
Categoría 2: Herramientas de Software		
Kali Linux (laptop)	1	$0.00
MSFVenom (laptop)	1	$0.00
Metasploit Framework (laptop)	1	$0.00
Android Debug Bridge (ADB)	1	$0.00
Android SDK	1	$0.00
	Total 2:	$0.00
Categoría 3: Servicios Básicos		
Servicios	**Precio**	
Servicio de Internet	$35.50,00	
Servicio de Electricidad	$10.10,00	

Total 3:	$45.60,00
Total 1 + Total 3:	$1.010,60

Elaborado por: *Remache Kleber, 2025*

9.2.1 Descripción de las herramientas de software

- **Kali Linux:** Es un sistema operativo que se utiliza principalmente para proteger y optimizar ordenadores y redes al igual que para descifrar contraseñas. Dado que estas características también pueden utilizarse con fines ilegales, la distribución no carece de polémica (¿Qué es Kali Linux?, 2023).
- **MSFVenom:** Se trata de una herramienta que tiene la capacidad de generar playload para diversos sistemas operativos (Sanchez, 2020).
- **Metasploit Framework:** Es un proyecto de código abierto que nos ayuda a investigar las vulnerabilidades de seguridad para poder empezar a evitar riesgos de seguridad y documentar qué vulnerabilidades podemos encontrar en nuestros proyectos (Rizaldos, 2018).
- **Android Debug Bridge (ADB):** Es una herramienta de línea de comandos versátil que te permite comunicarte con un dispositivo facilitando una variedad de acciones en dispositivos, como instalar y depurar apps (Android Debug Bridge (adb) | Android Studio, s. f.).
- **Android SDK:** Es un paquete de desarrollo de software desarrollado por Google y permite crear aplicaciones Android sin necesidad de ser un experto para poder usarlo (Vaati, 2020).

9.3 **Carta de aceptación de la organización donde se aplicó el trabajo de integración curricular.**

Para la realización de este trabajo de investigación no se requiere de una carta de aceptación por el motivo que esta será aplicada a dispositivos móviles que cuente con un sistema operativo Android en un entorno controlado para así evitar posibles consecuencias que lleguen a afectar a muchos usuarios. En conjunto a la utilización de herramientas de software gratuitas para así poder realizar la creación, el desarrollo y despliegue del playload.

9.4 Instrumentos de recopilación de datos

MODELO DE GUÍA DE OBSERVACIÓN

Proyecto:

Dispositivo empleado:

Fecha de la Observación: (_____/_____ /____)

Herramienta usada:

PREPARACIÓN DEL ENTORNO DE PRUEBAS		
ASPECTOS A OBSERVAR	**RESULTADO**	**OBSERVACIONES**
IMPLEMENTACIÓN DE COMANDOS EN EL DISPOSITIVO MÓVIL		
COMANDOS A EJECUTAR	**RESULTADO**	**OBSERVACIONES**
INFORMACIÓN DEL USUARIO		
RESULTADO FINAL	**RESULTADO**	**OBSERVACIONES**

Elaborado por: *Remache Kleber, 2025*

COMENTARIOS/SUGERENCIAS:

……
……
……
……
……

LISTA DE COTEJO

ASPECTOS EVALUADOS	DISPOSITIVO #1		DISPOSITIVO #2	
	SI	NO	SI	NO

Elaborado por: *Remache Kleber, 2025*

Escala de Nivel de Impacto	Descripción del Impacto	Dispositivos	
		Dispositivo #1	Dispositivo #2
1 – 4 Impacto Bajo			
4 – 7 Impacto Medio			
7-10 Impacto Alto			

Elaborado por: *Remache Kleber, 2025*

COMENTARIOS:

……

……

……

……

……

9.5 **Otros que considere relevantes para sustentar su proyecto.**

GUÍA DE OBSERVACIÓN DISPOSITIVO #1 GAMA MEDIA

Proyecto: Hacking Ético a dispositivos móviles Android por Playload, año 2024.

Dispositivo empleado: Xiaomi Redmi Note 9S

Fecha de la Observación: (_28_/__10___ /__2024__**)**

Herramienta usada: Playload

PREPARACIÓN DEL ENTORNO DE PRUEBAS		
ASPECTOS A OBSERVAR	**RESULTADO**	**OBSERVACIONES**
Herramientas de software utilizadas	Exitoso	Todas las herramientas de software fueron empleadas durante la creación del entorno.
Utilización de método reverse	Exitoso	Fue aplicado exitosamente para recibir la información de la víctima.
Verificación de la dirección IP actual del ordenador.	Exitoso	Se obtuvo la IP exitosamente del equipo.
Verificación del número de puerto a cuál se recibirá la información.	Exitoso	El puerto del dispositivo estuvo disponible.
Creación del APK para realizar el ataque a un dispositivo.	Exitoso	El APK fue creado con éxito cumpliendo así su objetivo planteado.
Acceso al dispositivo infectado.	Exitoso	El dispositivo fue vulnerado.
Sección iniciada por la víctima.	Exitoso	La victima inicializó la aplicación con éxito.
IMPLEMENTACIÓN DE COMANDOS EN EL DISPOSITIVO MÓVIL		
COMANDOS A EJECUTAR	**RESULTADO**	**OBSERVACIONES**

dump_calllog	Exitoso	Fue vulnerado el registro de llamada de la víctima con éxito.
dump_contacts	Exitoso	La aplicación que contiene la información de los contactos de la víctima fue vulnerada.
dump_sms	Exitoso	La aplicación de mensajería del dispositivo fue vulnerada mostrando así los mensajes que le llegaron a la víctima.
geolocate	Exitoso	Se obtuvo de manera exitosa la localización en tiempo real de la víctima.
send_sms	Fallido	No se logró enviar mensajes SMS.
wlan_geolocate	Exitoso	Se obtuvo de manera exitosa la localización en tiempo real de la víctima.
INFORMACIÓN DEL USUARIO		
RESULTADO FINAL	**RESULTADO**	**OBSERVACIONES**
Información sustraída de la víctima.	Obtenida	La información personal de la víctima fue sustraída de manera exitosa vulnerado así la seguridad del dispositivo móvil y sus aplicaciones.

Elaborado por: *Remache Kleber, 2025*

COMENTARIOS/SUGERENCIAS:

El atacante (ciberdelincuente) no pudo enviar mensajes SMS desde el dispositivo de la víctima debido a la ausencia de una tarjeta SIM y debido a esto la ejecución del

comando send_sms fue tomado como desconocido por nuestra aplicación de la herramienta playload.

Debido a la presencia de cualquier alteración en la latencia de la señal wifi por parte del atacante o la victima la sesión creada para la comunicación de los dispositivos puede cerrarse de manera inesperada, con esto se pudo constatar que el atacante necesita que la víctima ejecute de nuevo la aplicación en su dispositivo móvil Android.

GUÍA DE OBSERVACIÓN DISPOSITIVO #2 GAMA BAJA

Proyecto: Hacking Ético a dispositivos móviles Android por Playload, año 2024.

Dispositivo empleado: Samsung J7 Neo

Fecha de la Observación: (_28_/__10___ /__2024__**)**

Herramienta usada: Playload

PREPARACIÓN DEL ENTORNO DE PRUEBAS		
ASPECTOS A OBSERVAR	**RESULTADO**	**OBSERVACIONES**
Herramientas de software utilizadas	Exitoso	Todas las herramientas de software fueron empleadas durante la creación del entorno.
Utilización de método reverse	Exitoso	Fue aplicado exitosamente para recibir la información de la víctima.
Verificación de la dirección IP actual del ordenador.	Exitoso	Se obtuvo la IP exitosamente del equipo.
Verificación del número de puerto a cuál se recibirá la información.	Exitoso	El puerto del dispositivo estuvo disponible.
Creación del APK para realizar el ataque a un dispositivo.	Exitoso	El APK fue creado con éxito cumpliendo así su objetivo planteado.

Acceso al dispositivo infectado.	Exitoso	El dispositivo fue vulnerado.
Sección iniciada por la víctima.	Exitoso	La victima inicializó la aplicación con éxito.
IMPLEMENTACIÓN DE COMANDOS EN EL DISPOSITIVO MÓVIL		
COMANDOS A OBSERVAR	**RESULTADO**	**OBSERVACIONES**
dump_calllog	Exitoso	Fue vulnerada la información del registro de llamada de la víctima con éxito, la cual se encontraba contenida en su dispositivo.
dump_contacts	Exitoso	La información que contiene la aplicación de los contactos de la víctima fue obtenida.
dump_sms	Exitoso	La aplicación de mensajería del dispositivo fue vulnerada y se visualizó el historial de los mensajes que fueron recibidos por la víctima.
geolocate	Exitoso	Se obtuvo de manera exitosa la localización en tiempo real de la víctima.
send_sms	Fallido	El ciberdelincuente no puedo enviar mensajes SMS a los contactos de la víctima suplantando su identidad.
wlan_geolocate	Exitoso	Se obtuvo de manera exitosa la localización en tiempo real de la víctima.
INFORMACIÓN DEL USUARIO		

RESULTADO FINAL	RESULTADO	OBSERVACIONES
Información sustraída de la víctima.	Obtenida	La información personal de la víctima fue sustraída de manera exitosa vulnerado así la seguridad del dispositivo móvil y sus aplicaciones.

Elaborado por: *Remache Kleber, 2025*

COMENTARIOS/SUGERENCIAS:

No se lograron enviar mensajes SMS desde el dispositivo de la víctima debido a la ausencia de una tarjeta SIM y por esto la ejecución del comando send_sms fue tomado como desconocido por nuestra aplicación de la herramienta playload. No se presentó ninguna complicación al momento de la instalación del APK ni se visualizó alguna especie de reporte de seguridad que indique archivos maliciosos o dañinos.

LISTA DE COTEJO

ASPECTOS EVALUADOS	DISPOSITIVO #1 GAMA MEDIA		DISPOSITIVO #2 GAMA BAJA	
	SI	NO	SI	NO
Seguridad del dispositivo vulnerada.	✓		✓	
Ejecución de comandos maliciosos.	✓		✓	
Dirección IP obtenida del dispositivo móvil de la víctima.	✓		✓	
Acceso al dispositivo de la víctima.	✓		✓	
Correcto funcionamiento en diversas versiones del sistema operativo Android.	✓		✓	
Lista de mensajes del usuario vulnerada.	✓		✓	
Información personal de los contactos del usuario vulnerada.	✓		✓	
Registro de llamadas del usuario obtenido.	✓		✓	
Geolocalización obtenida en tiempo real del usuario por parte del atacante.	✓		✓	
Envió de mensajes por parte del atacante a los contactos de la víctima.		X		X

Elaborado por: *Remache Kleber, 2025*

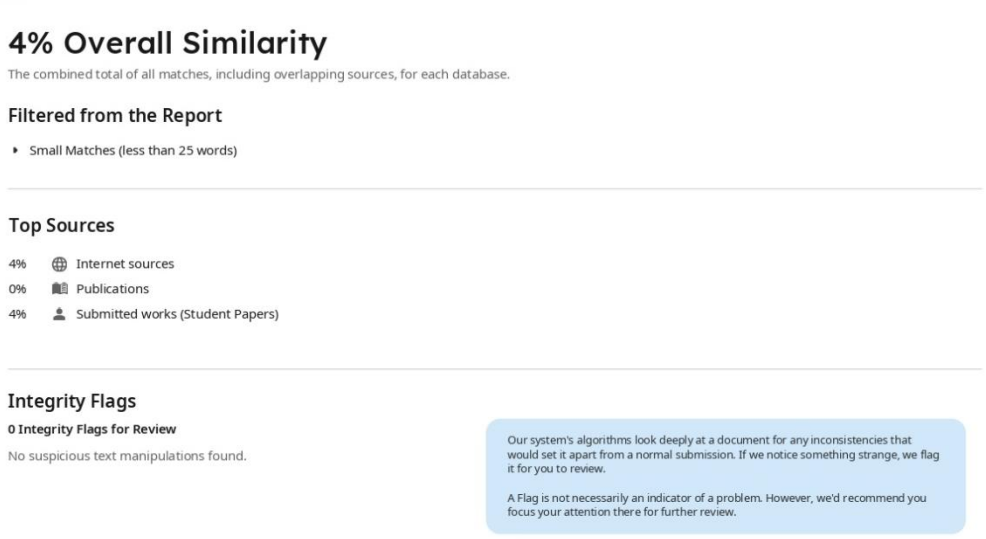
Escala de Nivel de Impacto	Descripción del Impacto	Número de Dispositivos	
		Dispositivo #1 Gama Media	Dispositivo #2 Gama Baja
1 – 4 Impacto Bajo	La información personal de la víctima no pudo ser vulnerada debido la seguridad del dispositivo.		
4 – 7 Impacto Medio	La información personal de la víctima fue medianamente vulnerada.		
7-10 Impacto Alto	La información personal de la víctima fue totalmente vulnerada logrando superar la seguridad del dispositivo.	9	9

Elaborado por: *Remache Kleber, 2025*

COMENTARIOS:

Ambos dispositivos móviles que funcionaban con sistema operativo Android utilizados durante la creación y ejecución de los comandos maliciosos, no realizaron ninguna especie de análisis de seguridad durante la instalación del APK que contenía nuestro playload, por lo tanto, esto nos demuestra que sin importar el modelo del dispositivo y la marca que lo fabrique, su información puede ser vulnerada por la herramienta playload en conjunto con la utilización del Hacking perjudicando así al usuario y a los contactos registrados en el dispositivo.

10 Certificado Antiplagio



4% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Small Matches (less than 25 words)

Top Sources

4% Internet sources
0% Publications
4% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

11 Link del repositorio digital de biblioteca donde fue subido el proyecto