VNIVERSIDAD NACIONAL
AVTNOMA DE
MEXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Posgrado en Ciencia e Ingeniería de la Computación

"Propuesta de metodología forense digital para la recolección de datos no volátiles en dispositivos móviles con sistema operativo de código abierto"

T E S I S
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA (COMPUTACIÓN)

PRESENTA:
Ángel Hernández Segura

Dr. Enrique Daltabuit Godas
Posgrado en Ciencia e Ingeniería de la Computación.

M.C. Gerardo Huerta Lozada
Banco de México.

MÉXICO, D. F. Septiembre 2013

Universidad Nacional Autónoma de México
UNAM
Biblioteca Central
Dirección General de Bibliotecas de la UNAM

UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A mis Padres,

por su apoyo y su guía incondicional.

A mis hermanos,
por impulsarme y crecer juntos.

A Lilian,
por su amor y por nuestros grandes planes.

Al Dr. Enrique Daltabuit,
por mostrarme el mundo de la seguridad informática.

Al M.C. Gerardo Huerta,
por su guía y sus invaluables consejos.

A mis Profesores,
por ser una parte fundamental de mi formación.

Índice

Lista de Figuras.

Lista de Tablas.

Resumen.

En este trabajo se propone el diseño para llevar a cabo la adquisición de datos no volátiles en dispositivos móviles con S.O. de código abierto. Se implementa dicho diseño y se llevan a cabo pruebas en diferentes modelos de teléfonos móviles con S.O. Android.

Para llevar a cabo la propuesta de diseño se estudian las características de los dispositivos móviles "inteligentes", las principales guías de análisis forense, las herramientas actuales de adquisición digital forense y sus limitaciones y las características de seguridad y protección de datos que implementa Android.

Como resultado se obtiene un modelo de adquisición física de datos no volátiles aplicable en dispositivos Android, se llevan a cabo adquisiciones prácticas, se muestran los datos obtenidos y el tiempo requerido.

Como conclusión, se muestra que es viable llevar a cabo dichas adquisiciones mostrando que se pueden mejorar los métodos y las herramientas actuales de adquisición digital forense para dispositivos móviles.

Capítulo 1.

1.1 Introducción

El análisis forense se refiere a la correcta aplicación de técnicas científicas con la finalidad de identificar, recolectar, almacenar y presentar pruebas en un proceso legal. Con esto en mente, el presente trabajo pretende analizar algunos de los problemas a los que se enfrentan los investigadores digitales forenses, por otra parte, muestra las capacidades y beneficios que tienen los comúnmente llamados "teléfonos inteligentes" e intenta enfatizar la importancia de aplicar técnicas digitales forenses en dispositivos móviles.

Actualmente, los "teléfonos inteligentes" han remplazado a las computadoras de escritorio en muchas actividades como búsqueda rápida de información, lectura de correos, contacto con amigos a través de redes sociales, banca electrónica, etc. Pero además han agregado nuevas actividades como localización y posicionamiento, búsqueda de rutas, envío y recepción de mensajes cortos, etc. Incluso, se comienzan a vislumbrar nuevas actividades como pago a través de teléfonos inteligentes. Todo esto representa una gran cantidad de información personal y sensible almacenada en estos dispositivos.

Durante el capítulo 3 se presentan tres casos en los cuales se evidencia cómo una correcta aplicación de los procesos de identificación, recolección, almacenamiento y presentación pueden ayudar a identificar responsables y futuras medidas de mitigación.

Finalmente, éste trabajo se centra en el proceso de recolección de la información no volátil en dispositivos móviles con sistema operativo de código abierto. Hace un análisis del estado actual del arte y propone una metodología que pretende ser una guía muy específica a seguir durante el proceso de recolección de datos en teléfonos con las características ya descritas.

1.1 Objetivo

El objetivo de este trabajo es diseñar un procedimiento que complemente los procedimientos actuales de adquisición de datos no volátiles para dispositivos móviles con S.O. de código abierto. El procedimiento aportado ayudará a ampliar la cantidad de dispositivos a los que se les puede extraer una copia de los datos de usuario.

Como principales características el procedimiento a diseñar deberá ser capaz de extraer los datos de usuario de un dispositivo móvil con S.O. de código abierto aun cuando dicho dispositivo se encuentre bloqueado con contraseña, pin, patrón o algún otro control de acceso.

Con la finalidad de cumplir con su objetivo, este trabajo analiza las características de los dispositivos móviles, las principales guías internacionales de metodología forense, las características y sistemas de protección de datos del S.O. Android así como las herramientas actuales de extracción de datos y sus limitantes.

1.2 Alcance

Para fines prácticos este trabajo se limitará a estudiar dispositivos con S.O. Android ya que cumple con la característica de ser un S.O. de código abierto y uno de los más populares actualmente.

De igual forma, nos limitaremos a trabajar con dispositivos que permitan cargar y ejecutar versiones modificadas de su S.O.

La búsqueda de vulnerabilidades para permitir la carga y ejecución de algún S.O. en dispositivos que no lo permiten por configuración de fábrica no es parte del alcance de este trabajo.

Este trabajo pretende aportar un procedimiento general para la adquisición de permisos de administración y posteriormente para la obtención de copias de los datos del usuario. No es parte del alcance de este trabajo demostrar que el procedimiento propuesto funciona en todos los dispositivos Android disponibles en el mercado. Para fines prácticos nos enfocaremos en un sub-conjunto de dispositivos Android y validaremos la funcionalidad del procedimiento en estos.

El estudio de los controles para mitigar los ataques que puedan surgir a raíz del procedimiento propuesto se encuentran fuera del alcance de este trabajo.

1.3 Contribuciones.

El presente trabajo pretende mejorar las herramientas y los métodos actuales de adquisición forense, presentados en la sección 4.3.

Dichas mejoras ayudarán a las agencias públicas y privadas de investigación y de procuración de justicia descritas en la sección 5.2 a complementar sus métodos y mejorar sus resultados.

Capítulo 2. Breve descripción de los teléfonos móviles inteligentes y su impacto.

Un teléfono móvil inteligente se puede definir, de forma muy general, como un teléfono móvil con capacidades de cómputo mejoradas, diferentes dispositivos de comunicación como Bluetooth y WiFi, redes celulares, GPRS, USB, capacidad aumentada de almacenamiento y posibilidad de agregar aplicaciones.

En 1993 IBM presentó el que se considera el primer teléfono inteligente, sus funciones incluían: teléfono, localizador, envío/recepción de fax, correo electrónico, calendario, libreta de direcciones, block de notas e incluso una pantalla táctil, sistema operativo basado en DOS con una interfaz gráfica. Su precio rondaba los $1,000 USD. Se consideraba muy costoso, grande y pesado. (Microsoft Research, 1993). Desde entonces hasta el 2007, cuando Apple presentó su primer iPhone, él que muchos consideran una revolución en la telefonía móvil, otras compañías como Palm y Blackberry trabajaron en dispositivos que poco a poco nos acercaron a los teléfonos inteligentes como los conocemos ahora. En 2007 también aparece el sistema operativo Android, en ese momento sus principales competidores eran iOS de Apple, Windows Mobile de Microsoft, Blackberry de RIM y Symbian que surgió de la alianza entre varios fabricantes. A pesar de haber ingresado a un mercado altamente competitivo, al momento de escribir este trabajo, Android posee la mayoría del mercado ~75% (IDC, 2012).

Aún con estos números a favor, el futuro de Android es incierto, el número de ataques que han surgido hacia la plataforma y la falta de control de su mercado de aplicaciones lo han colocado en el centro de atención por parte de muchos analistas de seguridad (Enck, Jan/Feb 2009). A mi parecer, la adopción ha sido muy rápida y la diversificación de aplicaciones muy grande, de tal forma que podemos realizar compras, hacer transferencias y en general almacenar información sensible. Siempre ha sido una constante que donde hay dinero, información sensible y pocos controles aparecerá alguien tratando de sacar provecho.

2.1 Arquitectura física.

En cuanto a la arquitectura física podríamos diferenciar a los teléfonos móviles inteligentes y a los tradicionales de la siguiente manera: Un teléfono móvil tradicional cuenta con un solo procesador llamado *procesador de banda base* (BP por sus siglas en ingles), usando este procesador se ejecuta la pila de protocolos GSM, la interfaz del usuario y las aplicaciones agregadas (calendario, agenda, alarma, etc). En cambio un teléfono inteligente, tiene 2 procesadores, el primero es el *procesador de Banda Base* dedicado exclusivamente a la ejecución de los protocolos GSM y otro que puede tener 2 o más núcleos con el que se ejecuta el sistema operativo, interfaz del usuario, aplicaciones,

etc. A este segundo procesador se le conoce como *procesador de aplicaciones (*AP por sus siglas en inglés). (Welte, 2010)

En los dispositivos actuales ambos procesadores comparten la memoria RAM y la memoria Flash asignando porciones fijas a las que cada procesador puede acceder, de esta forma se evitan conflictos por los recursos de memoria. El diagrama lógico de los componentes se muestra en la figura 1.

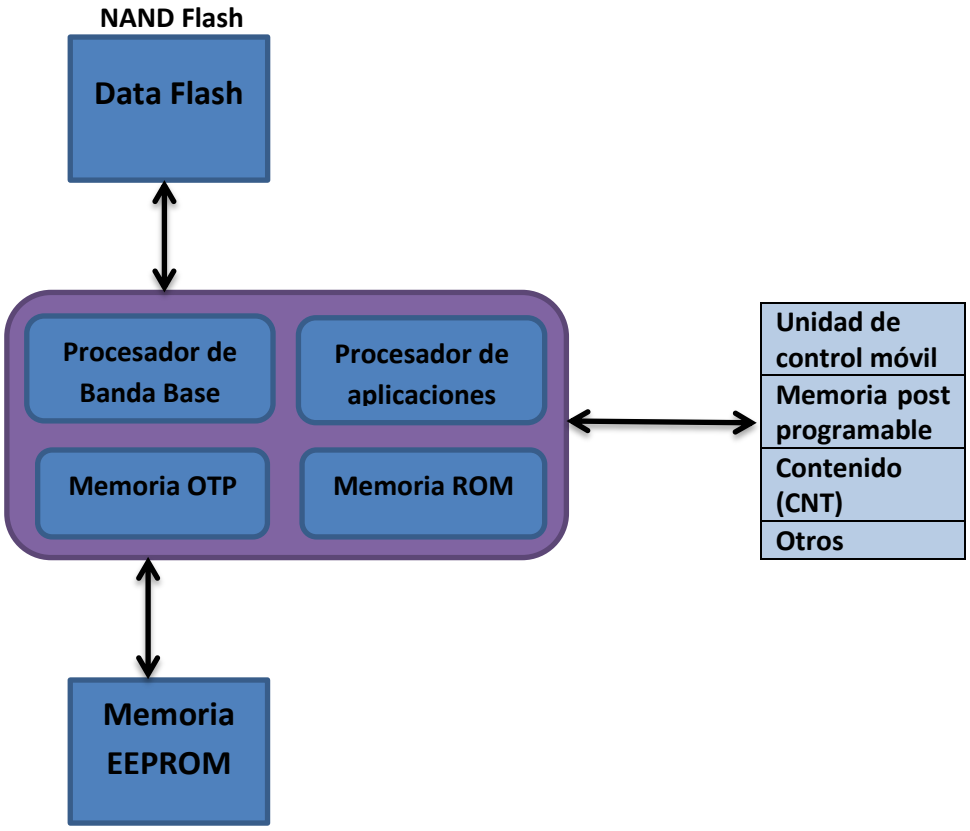


Fig. 1 Componentes típicos de hardware para un teléfono móvil inteligente.

Se puede observar el procesador de banda base y el procesador de aplicaciones encapsulados junto con el "OTP" y la "ROM" en una misma unidad. Sus funciones son:

- Procesador de banda base: Se encarga de ejecutar las funciones relacionadas con el modelo de protocolos GSM
- Procesador de Aplicaciones: Se encarga de ejecutar las funciones relacionadas con el sistema operativo y las aplicaciones que se ejecutan desde éste.
- Memoria OTP (Memoria programable una sola vez): Son unidades de memoria que se programan desde fábrica y a las que un usuario no tiene acceso.

Esta unidad de procesadores y memorias de lectura se comunica con unidades de memoria externa:

- "Data Flash": Es una memoria de tipo NAND de mucho interés para nosotros porque almacena datos no volátiles. Históricamente, se utilizaban memorias tipo NOR para este tipo de datos, sin embargo, para aprovechar su menor costo, mayor densidad y mayor velocidad de escritura/borrado se utilizan memorias tipo NAND en dispositivos móviles, tarjetas de memoria flash, reproductores MP3, dispositivos de memoria USB y discos de estado sólido. (Micron, 2006) En ésta unidad se encuentran los datos almacenados de las aplicaciones.
- Flash principal: De igual forma que la memoria "Data Flash" ésta unidad almacena datos no volátiles, sin embargo, generalmente es del tipo NOR. En ella se almacenan el sistema operativo y por omisión las aplicaciones, a partir de la versión 2.2 de Android se introduce la posibilidad de almacenar aplicaciones en unidad de memoria externa comúnmente llamada tarjeta SD. (Android Developers)

Dentro de la Flash principal se pueden reconocer diferentes unidades que almacenan los siguientes datos:

- MCU (Unidad de Control Móvil): Almacena los principales programas que se ejecutan en el teléfono.
- PPM (Memoria Post Programable): Almacena los paquetes de lenguajes que se pueden encontrar en el teléfono.
- CNT (Contenido): Las imágenes, sonidos y video por defecto que se encuentran en el teléfono.

Finalmente, la EEPROM que es una memoria programable contiene configuración específica al teléfono como la calibración de los giroscopios, de la batería, sensores de luz, etc.

Del modelo de arquitectura física podemos reconocer que la información no volátil de las aplicaciones se encuentra en la memoria Flash, independientemente de sí es NAND o NOR. Por lo que debemos ahora conocer las llamadas y los métodos que nos proporciona el sistema operativo para interactuar con estas unidades de memoria.

2.2 Arquitectura del S.O.

La historia de los sistemas operativos móviles ha estado en constante cambio y es difícil predecir qué sistema dominará el mercado en los siguientes años, sin embargo, por los anuncios que los fabricantes y los desarrolladores han hecho en los últimos meses del 2012 es claro que la oferta de sistemas operativos de código abierto para dispositivos móviles aumentará.

Estas son algunos de los anuncios más importantes:

- Noviembre, 2011. Ubuntu anuncia sus planes para ofrecer una versión de su S.O. para dispositivos móviles. (McHugh, 2011)
- Enero, 2012. Samsung anuncia en conjunto con Intel que están trabajando en su S.O. llamada "Tizen", diseñado para móviles y basado en Linux. (Boxall, 2012)
- Enero, 2012. Mozilla anuncia que se encuentra trabajando en su sistema operativo Firefox OS para dispositivos móviles, basado en HTML5 y con el apoyo de diversos fabricantes y operadores. (Scott, 2012)
- Enero, 2013. Mozilla anuncia los primeros dispositivos móviles con Firefox OS. (Ashe, 2013)
- Enero, 2013. Samsung anuncia que los primeros dispositivos móviles con S.O. llamado "Tizen", basado en Linux llegarán al mercado en 2013. (Noyes, 2013)
- Enero, 2013. Ubuntu anuncia que los primeros dispositivos móviles con sistema operativo Ubuntu llegarán en Octubre 2013. (Méndez, 2013)

El creciente interés por ingresar al mercado de S.O. para dispositivos móviles responde no sólo a los teléfonos inteligentes. Poder ofrecer un sistema operativo de bajo consumo de energía, eficiente en cuanto a uso de recursos y popular significa la posibilidad de ingresar también a otros dispositivos cómo televisores, sistemas de posicionamiento global, automóviles y electrodomésticos.

Por eso es importante no menospreciar los anuncios que hacen los fabricantes y operadores en este respecto y aunque no es posible predecir con exactitud cuáles son los sistemas operativos que dominarán el mercado, por el momento, parece estar plagado de sistemas basados en Linux.

Por esta razón y porque más adelante aprovecharemos el carácter abierto que nos ofrece, en esta sección nos referiremos específicamente a los detalles del sistema operativo Android.

Android fue creado por la compañía Android Inc., adquirida por Google y liberada como un proyecto de código abierto con el nombre de "Android Open Source Project" en 2007. El proyecto se puede obtener de un repositorio central y modificarse bajo las condiciones de la licencia llamada "Apache 2.0" (Android Project)

Las aplicaciones en Android están programadas en Java y el sistema operativo se encarga de ejecutarlas en un ambiente controlado. Dalvik es el nombre de la máquina virtual que Android provee con la finalidad de ejecutar las aplicaciones de usuario. Existen también aplicaciones propias del sistema que no se ejecutan en la máquina virtual Dalvik, las cuales mencionaremos más adelante.

Finalmente, es importante mencionar que las actualizaciones al sistema operativo se realizan en pocos meses por lo que detalles y artículos dependientes de una sola versión pueden perder relevancia muy fácilmente. En estos casos la documentación, el código fuente y blogs especializados son la mejor fuente de información actualizada.

2.2.1 Estructura de Android

El diagrama de la figura 2 muestra los principales componentes del sistema operativo Android.

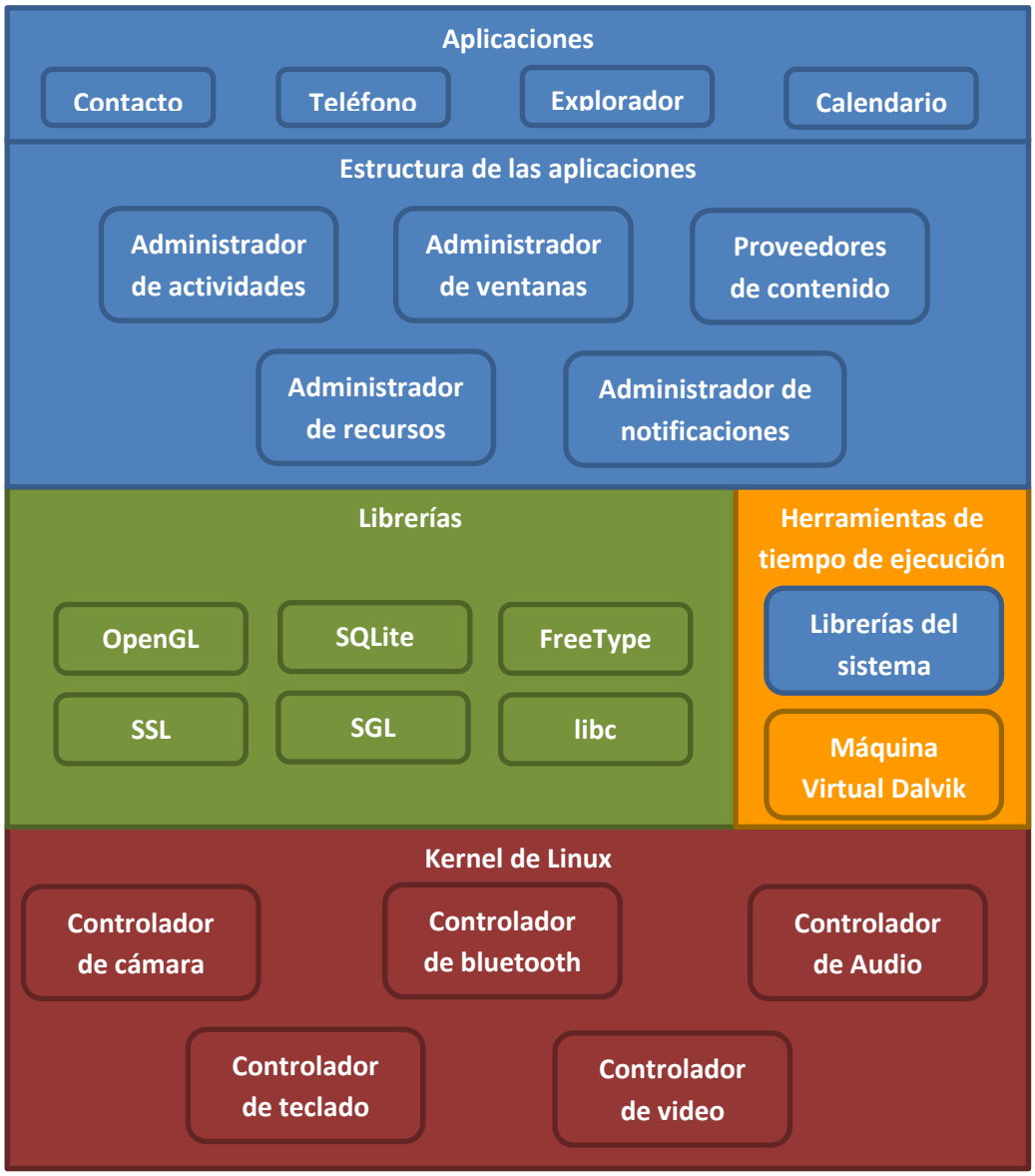


Fig. 2 Principales componentes del S.O. Android.

El sistema se puede dividir en 5 estructuras:

- El Kernel y las herramientas de bajo nivel.
- Las librerías del sistema.
- Las herramientas de tiempo de ejecución.
- El marco o estructura para las aplicaciones.

- Y las aplicaciones.

El conjunto de estructuras del sistema operativo muestra un diseño enfocado hacia la mejora en el rendimiento de los dispositivos móviles, el lenguaje en que están programados los módulos y su relación con la velocidad de ejecución, la forma en que funcionan las aplicaciones y los diferentes estados en los que se pueden encontrar, el kernel de Linux y las optimizaciones que se realizaron para el ahorro de energía en dispositivos móviles, etc. Sin embargo, el tema que nos ocupa es la adquisición y almacenamiento de la información en las aplicaciones por lo que las siguientes secciones más que enfocarse en las interacciones del hardware con el sistema operativo o en aplicaciones específicas se enfocará en el marco de las aplicaciones, las librerías y las herramientas de tiempo de ejecución. Específicamente en el archivo "AndroidManifest.xml", los proveedores de contenido, la base de datos SQLite y la máquina virtual Dalvik. La importancia de estudiar cada una de éstas se explicará en cada caso.

2.2.1.1 Archivo "AndroidManifest.xml"

Toda aplicación en Android debe contener su propia copia del archivo "AndroidManifest.xml" en la raíz de su directorio, este archivo provee al sistema operativo con información necesaria para ejecutar la aplicación. Algunas de las funciones del archivo "AndroidManifest.xml" son:

- Declara qué permisos necesita la aplicación para interactuar con otras aplicaciones y su información. Cuando una aplicación se instala en Android el sistema operativo da a conocer la solicitud de recursos que la aplicación realiza. Para ser instalada, el usuario debe aceptar dichos permisos concediendo así acceso de la aplicación a los recursos solicitados.
- Declara qué permisos deben tener otras aplicaciones en su archivo "AndroidManifest.xml" para interactuar con la aplicación en cuestión y con su información.

De esta forma el desarrollador de una aplicación declara de manera explícita los recursos, los datos y las funciones a las que debe tener acceso su aplicación para funcionar de forma correcta y también específica cuáles de sus recursos y datos desea proteger de otras aplicaciones. (Android)

Cada permiso se identifica con una etiqueta única. Se debe declarar el uso de dichos recursos utilizando la etiqueta *<uses-permission>* como se muestra en la figura 3.

```
1 <manifest . . . >
2     <permission android:name="com.ejemplo.project.DIRECCION" . . . />
3     <uses-permission android:name=" com.ejemplo.project.DIRECCION" />
4     . . .
5     <application . . .>
6        <activity android:name="com.ejemplo.project.REGISTRO"
7                  android:permission="com.ejemplo.project.DIRECCION"
8                  . . . >
9           . . .
10        </activity>
11    </application>
12 </manifest>
```

Fig. 3 Etiqueta <uses-permission> del archivo AndroidManifest.xml

En la línea 2 a través de la etiqueta *<permission>* se define un permiso para el dato "DIRECCION" y en la línea 3 a través de la etiqueta *<uses-permission>* se solicita acceso a dicho dato a pesar de que ha sido dentro de la misma aplicación que se definió.

De la misma forma una aplicación puede solicitar información y recursos del sistema utilizando permisos pre-definidos por Android, algunos de ellos se muestran en la figura 4.

```
1 android.permission.CALL_EMERGENCY_NUMBERS
2 android.permission.READ_OWNER_DATA
3 android.permission.SET_WALLPAPER
4 android.permission.DEVICE_POWER
```

Fig. 4 Permisos predefinidos en Android para intercambio de información.

Los primeros 2 solicitan acceso a datos del dispositivo, el tercero a propiedades del sistema operativo y el último a interacción con el hardware.

Las propiedades y el uso del archivo "AndroidManifest.xml" son muy importantes para nuestro tema ya que permiten o limitan la interacción con los datos almacenados por aplicación en los dispositivos Android.

La estructura completa del archivo "AndroidManifest.xml" se muestra en el Anexo A.

2.2.1.2 Proveedores de Contenido

Los proveedores de contenido o por su nombre en inglés "Content Providers" son procedimientos que administran el acceso a los datos almacenados por las aplicaciones. Cada aplicación que desea compartir datos con otras debe implementar su propio proveedor de contenidos y definir los datos que desea compartir y cada aplicación que desea acceder a datos almacenados por otra aplicación debe implementar un cliente de proveedor de contenidos. La finalidad es contar con un método estándar y consistente de intercambio de datos y comunicación entre procesos.

La comunicación hacia el proveedor de contenidos se lleva a cabo de la siguiente manera: Una aplicación A que desea acceder a los datos de una aplicación B se comunica con el "Content Resolver" de la aplicación B, éste se encarga de localizar el proveedor de contenido adecuado y facilita las funciones de creación, consulta, actualización y eliminación de datos dentro del proveedor de contenido para la aplicación B.

La aplicación que desea acceder a los datos almacenados en un proveedor de contenidos específico debe solicitar los permisos correspondientes en el archivo "AndroidManifest.xml" y el usuario debe aceptarlos al momento de instalar dicha aplicación.

Un proveedor de contenidos, entendido como un método programado dentro de una aplicación específica, debe definir los datos que desea compartir así como los permisos con los que debe contar la aplicación que deseé consultarlos, de esta forma se supone que el usuario conoce los datos a los que una aplicación tratará de ingresar. (Android)

Android de manera nativa incluye proveedores de contenido que administran datos como imágenes, alarmas, videos, contactos, calendario e información personal, estos proveedores de datos están disponibles para cualquier aplicación que solicite los permisos necesarios. Una lista completa de los proveedores de contenido existentes de manera nativa en Android se puede encontrar en la documentación del paquete Android.provider (Android)

En caso de que una aplicación no defina proveedores de contenido y por lo tanto no defina datos a compartir con otras aplicaciones, no se puede utilizar este método para acceder a los datos almacenados por dicha aplicación.

Como se puede apreciar, este método es muy conveniente para extraer datos de aplicaciones utilizando funciones propias de Android, sin embargo, se encuentra limitado por lo menos en 2 aspectos:

- Un investigador forense debe ser capaz de instalar la aplicación que se encargue de extraer los datos almacenados en el dispositivo.
- Se asume que, por lo menos, las aplicaciones que almacenan datos de interés para la investigación forense definan un proveedor de contenidos (compartiendo sus datos).

2.2.1.3 Base de datos SQLite

SQLite es un manejador de base de datos de código libre que soporta características propias de los manejadores de bases de datos relacionales como la sintaxis SQL, transacciones y scripts, además de estar optimizada para uso reducido de memoria en tiempo de ejecución. (SQLite)

SQLite se encuentra disponible en los dispositivos con sistema operativo Android y se utiliza por las aplicaciones para almacenar datos que posteriormente serán compartidos utilizando un proveedor de contenidos en cuyo caso, serán almacenadas en la carpeta: *DATA/data/Nombre_Aplicacion/databases/ARCHIVO.sql*

En caso de que una aplicación no defina un proveedor de contenidos pero reciba datos de otra aplicación se utiliza el paquete genérico android.databases para manejar dichos datos en lugar de android.database.sqlite

El manejador de base de datos se encuentra por omisión instalado en la carpeta *tools/* del sistema de archivos de Android, se puede usar esta herramienta para explorar los datos almacenados en las bases de datos de las aplicaciones obteniendo acceso a un intérprete de comandos del sistema Android. Más adelante se explicará cómo acceder a este intérprete de comandos. (Android)

Ejecutar instrucciones SQL de manera adecuada será muy importante para extraer la información que se encuentre almacenada en los archivos de SQLite para cada aplicación.

2.2.1.4 Máquina Virtual Dalvik

Dalvik es la máquina virtual que Android utiliza para interpretar archivos .class, escritos y compilados en Java, e interpretarlos cómo archivos .dex permitiendo así una ejecución mucho más eficiente en dispositivos móviles.

La necesidad de implementar una máquina virtual específicamente para Android nace a partir de la diversidad de dispositivos que deben ser capaces de ejecutar Android. Al no tener control sobre las especificaciones físicas de cada dispositivo móvil, Android agrega una aplicación que le permite interpretar código escrito en Java, eliminar redundancias y ejecutarlo de manera más eficiente. Esto permite disminuir el consumo de energía por parte del procesador, ofrecer una experiencia de uso más confortable y permitir al programador generar su código una sola vez y ejecutarlo en una diversidad de sistemas que implementen Android. ((Google), 2008)

Los requisitos mínimos para que un sistema implemente Android se muestran en la tabla 1.

Característica	Requisito Mínimo
Memoria	128 MB RAM, 256 MB Flash Externa
Almacenamiento Externo	Mini SD o Micro SD
Pantalla Principal	16 bits de color o más
Botones de navegación	Navegación de 5 vías con botones de encendido/apagado, cámara y volumen
Cámara	2 MP con sensor CMOS
USB	Interfaz estándar de mini USB
Bluetooth	Versión 1.2 o 2.0

Tabla 1 Requisitos mínimos de Android. “The Dalvik virtual machine architecture” (Ehringer, 2010)

La mayoría de los dispositivos disponibles en el mercado exceden las características mínimas necesarias para implementar Android, por lo que el rango de dispositivos y equipos que tienen la capacidad de funcionar empleando dicho sistema operativo es muy grande. Por lo que se hace más evidente la necesidad de una capa de abstracción que permita a la misma aplicación ejecutarse en diferentes dispositivos sin necesidad de ser rediseñada ni recompilada.

La forma de operar de la máquina virtual Dalvik en un sistema Android puede explicarse de la siguiente manera: Cada aplicación en Android se ejecuta en su propio proceso, con su propia máquina virtual Dalvik. Dalvik está preparado para crear varias instancias de sí mismo en un dispositivo y ejecutarse de forma eficiente. Dalvik ejecuta archivos .dex (Ejecutable Dalvik) optimizados para un bajo consumo de memoria y obtenidos a partir de los archivos .class (compilados de Java) de los desarrolladores y convertidos en .dex utilizando la herramienta "dx tool".

La herramienta "dx tool" utiliza los archivos .class obtenidos del compilador de Java. En Java cada archivo fuente que contiene una clase después de compilarse genera un archivo .class, por lo que un archivo con 5 clases generan 5 archivos .class, cada uno interpretado por la máquina virtual de Java. En Android, la herramienta "dx tool" utiliza los diferentes archivos .class y genera un solo archivo .dex, eliminando redundancias en variables, métodos y uso de librerías, este archivo .dex será interpretado por la máquina virtual Dalvik utilizando menos memoria que los diferentes .class

Google publicó datos sobre la optimización de espacio en memoria que se obtiene después de utilizar la herramienta "dx tool", dichos datos se muestran en la tabla 2.

Código	Archivo JAR sin comprimir (bytes)	Archivo JAR compreso (bytes)	Archivo dex sin comprimir (bytes)
Librerías comunes del sistema	21,445,300 (100%)	10,662,048 (50%)	10,311,972 (48%)
Aplicación de Explorador WEB	470,312 (100%)	232,065 (49%)	209,248 (44%)
Aplicación de reloj/alarma	119,200 (100%)	61,658 (52%)	53,020 (44%)

Tabla 2 Datos de optimización por uso de la herramienta "dx tool". "Dalvik Internal" (Bornstein)

Como se mencionó previamente, en Android cada aplicación se ejecuta en su propio proceso y dentro de su propia máquina virtual Dalvik. A cada proceso se le asigna un identificador de usuario con recursos limitados y acceso a sus propios datos, por lo que una aplicación no puede interferir con la ejecución de otra ni con sus datos. La única interfaz de comunicación de datos entre procesos es el "Content Resolver" que permite a una aplicación solicitar datos de otra siempre y cuando el archivo "AndroidManifest.xml" de la aplicación solicitante así lo especifique y los permisos de la aplicación destino lo permitan.

A pesar de que la máquina virtual Dalvik por sí misma no es la encargada de proteger la seguridad de los datos de las aplicaciones, si se encarga de proveer un ambiente aislado para cada aplicación permitiendo que Android a través del uso del "AndroidManifest.xml", administración y control de acceso para usuarios pueda separar los recursos y protegerlos.

2.2.2 Sistema de archivos de Android.

Los sistemas de archivos son diferentes formas de organizar y de mantener la información en dispositivos de almacenamiento con la finalidad de almacenar, extraer y modificar datos. Entre las funciones de los sistemas de archivos se encuentran: mantener la integridad, almacenar y mantener meta-datos, organizar los datos almacenados y establecer y hacer cumplir convenciones para nombres de archivos. Existen diferentes sistemas de archivos, generalmente relacionados con los sistemas operativos, para Windows tradicionalmente se usan: FAT, FAT32 y NTFS; para Linux: EXT3 y EXT4 y para OS-X: HFS+

Entender los diferentes sistemas de archivos es importante para los investigadores forenses y para el diseño de sus herramientas porque permite interpretar los datos encontrados en un medio de almacenamiento y ayuda a identificar y reconstruir archivos específicos, en algunas ocasiones, a pesar de que éstos hayan sido eliminados desde el sistema operativo. En el capítulo 4 se platicará de los métodos de adquisición física y lógica, sus repercusiones en cuanto a la calidad y cantidad de datos extraídos de un medio de almacenamiento y su relación con la importancia de conocer los sistemas de archivos.

2.2.2.1 YAFFS y Android.

YAFFS (En inglés, Yet Another Flash File System), es el primer sistema de archivos específicamente diseñado para memorias FLASH tipo NAND. Cuenta con técnicas de corrección de errores y de verificación de integridad. Entre las características de YAFFS están (Darren Quick, Mohammed Alzaabi, 2011):

- *Journaling:* Un sistema de archivos centrado en bitácoras que registra los cambios en el sistema y provee protección de daños producidos por perdida de energía en el dispositivo. Esto se traduce en mayor consumo de *RAM* para mantener las bitácoras, sin embargo, su consumo de memoria es menor comparado con otros sistemas de archivos que también implementan técnicas de *Journaling*.
- Recolección de basura. Cuenta con un sistema que se ejecuta automáticamente cuando el espacio libre del dispositivo se encuentra muy bajo. Los dispositivos de almacenamiento tipo NAND tienen la limitación de contar con una vida útil corta y un limitado número de veces que se puede borrar un bloque de datos, por lo tanto, el procedimiento que emplea YAFFS para optimizar la vida del dispositivo consiste en "marcar" sub-segmentos de bloques como "sucios", cuando se tiene una serie de

"sub-segmentos" continuos sucios, es decir, un segmento completamente "sucio", se lleva a cabo el borrado del segmento completo. Éste procedimiento pretende disminuir el número de operaciones de borrado que se aplican en el dispositivo y por lo tanto aumentar la vida útil de éste.

- Muy flexible, muchos de sus parámetros son configurables y se puede adaptar para trabajar con diferentes geometrías, dispositivos flash, diversas opciones de corrección de errores, etc.

Android ha utilizado YAFFS en todos sus dispositivos hasta la versión Android 2.3 llamada "Gingerbread" y diseñada para el sistema de archivos EXT4 (Extended File System 4), el primer dispositivo oficial Android con EXT4 fue el Google Nexus S anunciado en diciembre del 2010. (Tim Bray, 2010)

El cambio se debe a que YAFFS permite realizar una sola operación lectura/escritura a la vez, lo que provoca que las actividades que requieren interacción con el dispositivo de almacenamiento tengan que esperar a que no exista otra aplicación haciendo uso de la memoria. Este problema aumenta cuando se agregan procesadores de aplicación de más de un núcleo. Es por eso que hasta antes de "Gingerbread" y del Google Nexus S, los teléfonos Android contaban con un solo núcleo para su procesador de aplicaciones.

2.2.2.2 EXT4 y Android

EXT4 a diferencia de YAFFS no fue especialmente diseñado para medios de almacenamiento NAND, sino para Discos mecánicos, por lo que su funcionamiento es muy diferente. Algunos de los detalles del funcionamiento de EXT4 son (Defreez, 2012):

- EXT4 es una mejora de EXT3 que a su vez fue una mejora de EXT2. El funcionamiento de éstos es muy similar.
- La unidad más pequeña de almacenamiento es un bloque, típicamente EXT2 y EXT3 utilizan bloques de 4KB, sin embargo, EXT4 utiliza bloques de 64KB. Los datos se leen y se escriben en múltiplos de bloques en lugar de múltiplos de bits o de bytes.
- EXT2, EXT3 y EXT4 dividen el dispositivo de almacenamiento en grupos de bloques, el tamaño de los grupos se define cuando se crea el sistema de archivos. Cada grupo de bloques contiene una copia del "súper-bloque". El "súper-bloque" a su vez contiene metadatos acerca del sistema de archivos. Por lo que existen múltiples copias de metadatos del sistema de archivos almacenados en el medio. Estas copias se utilizan, para recuperar datos en caso de que el "súper-bloque" se dañe.
- Los datos de un mismo archivo por lo general se almacenan en el mismo bloque o en bloques cercanos con la finalidad de disminuir el número de movimientos que debe realizar la cabeza lectora del disco donde se encuentra almacenado.
- Cada grupo de bloques contiene una "tabla descriptora de grupo" que almacena metadatos acerca de la organización del grupo de bloques. Por cada grupo de bloques

existe también una tabla de “inodos” que almacena información acerca de los archivos almacenados en el bloque.

- Cada archivo almacenado en el sistema de archivos tiene metadatos que en conjunto se llaman “inodos”. Todos los “inodos” para los archivos almacenados en un mismo grupo de bloques se encuentran en la “tabla de inodos”. Los datos que se pueden encontrar en un “inodo” son: Identificador del usuario que es dueño del archivo (UID), los permisos de lectura, escritura y ejecución para el dueño del archivo, miembros del grupo y otros y apuntadores hacia el espacio en el dispositivo de almacenamiento donde se encuentra el archivo.
- Cuando un archivo se elimina, se borran los apuntadores del “inodo” hacia los datos, se desasocia el “inodo” y los bloques asociados a los datos se liberan, sin embargo, los datos permanecen en el medio hasta que alguna operación de escritura los sobre-escriba. Esta forma de operar es la que hace posible la recuperación de archivos eliminados en EXT4.

A pesar de las grandes diferencias entre ambos sistemas de archivos, Android provee una interfaz que permite a los programadores, hasta cierto punto, ignorar el tipo de sistema de archivos y sus detalles y enfocarse más en el diseño e implementación. Esta interfaz se llama Sistema de Archivos Virtual (VFS por sus siglas en inglés). VFS permite modificar con cierta facilidad el sistema de archivos de los medios de almacenamiento sin necesidad de hacer grandes cambios en el resto del sistema. En cambio para los científicos forenses, este tipo de modificaciones son muy importantes y es necesario conocer el sistema de archivos en uso y los detalles de éste ya que de esto dependerá la cantidad y la calidad de la información que se puede extraer de una adquisición forense.

2.3 Listado de la información personal que almacena un dispositivo móvil inteligente.

La cantidad y la variedad de datos personales que puede contener un dispositivo móvil inteligente es muy amplia y varía dependiendo de las preferencias de cada usuario, las aplicaciones instaladas, el sistema operativo y el uso que se le dé a cada dispositivo. Sin embargo, para el sistema operativo Android, Google provee una interfaz de programación de aplicaciones (API por sus siglas en inglés) que cuenta con un mínimo de proveedores de contenido de los que se puede obtener información personal.

Es importante recordar que para hacer uso de estos proveedores de contenido una aplicación debe especificar su intensión de uso en su archivo AndroidManifest.xml

En las tablas 3, 4, 5 y 6 se muestra los listados obtenidos de Android.providers (Android), API nivel 14 (el nivel corresponde a la plataforma que lo implementa, en este caso Android 4.0, 4.0.1, 4.0.2 Ice_Cream_Sandwich y superiores a menos que se especifique lo contrario) (Android Developers, 2013) para diferentes aplicaciones.

Proveedor de Contenido	**Detalles**
CalendarContract.	Detalles acerca de eventos en el calendario.
CalendarContract.AttendeesColumns	Participantes del evento.
CalendarContract.CalendarAlertsColumns	Alertas programadas en la fecha del evento.
CalendarContract.CalendarCacheColumns	Zona horaria de la hora del evento.
CalendarContract.ColorsColumns	El color con el que se muestra el evento en el calendario.
CalendarContract.EventDaysColumns	Día de inicio y de finalización del evento.
CalendarContract.EventsColumns	Columnas de la tabla eventos, contiene todos los detalles enumerados.
CalendarContract.RemindersColumns	Recordatorios del evento
CalendarContract.SyncColumns	Información sobre las aplicaciones que han sincronizado los datos del calendario

Tabla 3 Proveedores de contenido en Android para la aplicación de calendario.

Proveedor de Contenido	Detalles
ContactsContract.CommonDataKinds.BaseTypes	Tipo de datos que puede contener un contacto (correo electrónico, apodo, página web, etc.)
ContactsContract.ContactNameColumns	Nombre y apellido del contacto
ContactsContract.ContactOptionsColumns	Ruta hacia un tono de llamada personalizado, fecha de la última vez que se contactó a la persona. Booleano de si el contacto debe mandarse a buzón de voz, booleano de si es un contacto favorito.
ContactsContract.ContactsColumns	El nombre que se muestra del contacto, indicador de si el contacto tiene por lo menos un número telefónico, llave de identificación de la base de datos, rutas de la foto de perfil completa y miniatura
ContactsContract.ContactStatusColumns	Capacidades de chat del contacto, estado de presencia, última actualización de estado, fecha y hora de última actualización de estado.
ContactsContract.DataColumnsWithJoins	Combinación de todas las columnas que se obtienen de consultar ContactContract.Data.
ContactsContract.DisplayNameSources	Datos utilizados para mostrar el nombre que se mostrará del contacto. En orden creciente de prioridad (email, teléfono, organización, apodo, nombre estructurado).
ContactsContract.FullNameStyle	Constantes de combinación de nombres y apellidos para generar un nombre completo. Dependiendo de la zona geográfica, los nombres y apellidos se combinan de diferentes formas para generar el nombre completo.
ContactsContract.GroupsColumns	Información de grupos de contactos y miembros de cada grupo.
ContactsContract.PhoneLookupColumns	El número de teléfono del contacto y si se encuentra marcado como personal o trabajo.
ContactsContract.PhoneticNameStyle	Constantes de estilo para la pronunciación de un nombre (japonés, coreano y chino piyin).
ContactsContract.PresenceColumns	Datos sobre la presencia del contacto en servicios de mensajería instantánea. Protocolo, cuenta, identificador.
ContactsContract.StatusColumns	Datos sobre el estado y capacidades del contacto. Estado (disponible, ausente, fuera de línea), si el contacto tiene cámara, última actualización de estado.
ContactsContract.StreamItemPhotosColumns	Datos de la foto de identificación del contacto. Ruta, identificador, nombre de archivo.

Tabla 4 Proveedores de contenido en Android para la aplicación de contactos.

Proveedor de Contenido	Detalles
MediaStore.Audio.AlbumColumns	Datos del álbum de un archivo de música. Título del álbum, identificador único, artista, año de publicación, número de canciones.
MediaStore.Audio.ArtistColumns	Detalles sobre el artista de un archivo de música. Nombre del artista, numero de álbumes y número de canciones del artista en el dispositivo.
MediaStore.Audio.AudioColumns	Detalles sobre el audio. Nombre, artista, álbum, duración, año, si está registrado como alarma o como tono de llamada.
MediaStore.Audio.GenresColumns	Genero del archivo de música.
MediaStore.Audio.PlaylistsColumns	Detalles de listas de reproducción, nombre, fecha de creación y de modificación.
MediaStore.Files.FileColumns	Campos de la tabla maestra para los archivos multimedia. Tipo de archivo (audio, video, imágenes, lista de reproducción), título, directorio.
MediaStore.Images.ImageColumns	Detalles de archivo de imagen. Latitud y longitud del lugar donde se tomó la foto, descripción de la imagen, fecha y hora de la captura de la imagen.
MediaStore.MediaColumns	Datos comunes para diferentes archivos multimedia. Fecha en la que se agregó el elemento, fecha de modificación, nombre, alto y ancho de imagen o video en pixeles, tamaño en bytes.
MediaStore.Video.VideoColumns	Datos sobre archivos de video. El álbum al que pertenece el archivo, lenguaje, duración, resolución, descripción, longitud y latitud de dónde se tomó el video.

Tabla 5 Proveedores de contenido en Android para la aplicación de administración de contenido multimedia.

Proveedor de Contenido	Detalles
AlarmClock	Permite establecer y modificar una alarma.
Browser	Clase que permite interacción con el explorador web para abrir una nueva ventana o tab.
Browser.BookmarkColumns	Clase que interactúa con los sitios marcados como favoritos en el explorador web. Da fecha de creación del favorito, dirección y título del sitio
Browser.SearchColumns	Historial de búsquedas hechas con el explorador web. Regresa los términos buscados.
CallLog	Contiene información sobre las llamadas hechas y recibidas. Fecha y hora y número telefónico
CallLog.Calls	Llamadas recientes. Número de teléfono, nombre asociado al número (en caso de existir), fecha y hora de llamada, si el usuario vio el evento, llamada recibida o hecha.
Settings	Contiene las preferencias a nivel de sistema para el dispositivo. Preferencias de fecha y hora, de almacenamiento, diccionarios internos, opciones de privacidad, operador de red, etc.
Settings.Global	Preferencias globales, se aplican del mismo modo a todos los usuarios. Las aplicaciones pueden leer el contenido pero no modificarlo. Entre otras contiene, adb habilitado o deshabilitado, modo avión, modo wifi, modo bluetooth, etc.
Settings.Secure	Preferencias dependientes por usuario, las aplicaciones solo pueden leer pero no modificar. Entre otras existen: modo accesibilidad, geo localización activada, identificador de Android, métodos de entrada, capacidad de instalar aplicaciones no oficiales, etc.
Settings.System	Otras preferencias. Entre otras incluye: estado del acelerómetro, zona horaria, formato de fecha, modo red wifi, modo de tono, http proxy, etc.
UserDictionary.Words	Contenedor de las palabras definidas por el usuario
VoicemailContract.Status	Estado del servicio de buzón de mensajes. Indica si existe un buzón de voz configurado y activo. Muestra si existen mensajes de voz esperando en el servidor.
VoicemailContract.Voicemails	Contiene datos de los mensajes de voz del usuario. Si existiera un mensaje, fecha y hora de llegada, número de teléfono de origen, duración y si ha sido o no accedido por el usuario.

Tabla 6 Proveedores de contenido en Android para las aplicaciónes de alarma, explorador, registro de llamadas, diccionario y buzón de voz.

Capítulo 3. Breve descripción del análisis digital forense.

Las ciencias forenses se refieren a la aplicación de alguna ciencia en específico con la finalidad de reconstruir hechos y obtener evidencias de escenarios donde haya ocurrido algún evento que se pueda considerar incidente. De esta forma, existen múltiples ciencias que aportan conocimientos y técnicas al campo forense, por ejemplo, medicina forense, antropología forense, odontología forense, cómputo forense, etc.

Es posible que estas ciencias puedan aplicar su conocimiento en búsqueda de evidencia gracias al cumplimiento del principio de Locard (Bay, 2011) que a grandes rasgos establece que cualquier contacto entre 2 objetos siempre deja un rastro. En el mundo digital este principio también se cumple, aunque se podría argumentar que no es necesario que una computadora tenga contacto físico con un banco para llevar a cabo un fraude bancario. Aun así, es necesario que dicha computadora intercambie datos e instrucciones en forma de bits para que un fraude se pueda llevar a cabo. Es objeto del cómputo forense digital reconstruir los eventos acontecidos que puedan explicar la forma en que sucedió un fraude o cualquier incidente relacionado con sistemas de cómputo.

Un análisis digital es una investigación que pretende responder preguntas sobre estados digitales actuales o previos, por ejemplo, depurar un programa para saber el estado de la memoria en un momento específico es una forma de análisis digital, tratar de determinar porqué una computadora se encuentra en un estado específico o cómo llegó a ese estado es una forma más general de describirlo. De hecho, los análisis digitales son muy comunes y muchos hemos llevado a cabo alguno.

Los análisis digitales forenses son análisis digitales que siguen reglas muy específicas con la finalidad de que los resultados sean aceptados en procesos legales.

El proceso de investigación en el mundo digital es muy parecido al que se lleva a cabo en el mundo físico. Los investigadores deben preservar la escena del crimen para evitar que la evidencia se contamine, analizar la evidencia en búsqueda de cosas obvias, ¿Hubo un asesinato? o ¿hubo un robo?, este proceso genera preguntas básicas a responder, por ejemplo, ¿Cómo llegó esa pistola hasta ese lugar?, ¿Desde dónde tuvo que ser disparada la bala para llegar a ese lugar? y finalmente se lleva a cabo una investigación detallada para tratar de responder esas preguntas. De la misma forma tenemos que preservar la evidencia digital haciendo copias de los bits almacenados en una computadora, intentando no modificarlos o modificándolos lo menos posible. Los pasos llevados a continuación dependerán de qué preguntas se quieran responder, se buscan palabras clave específicas, archivos y carpetas en lugares específicos y se crea una línea de tiempo con la finalidad de entender el orden en el que sucedieron los eventos. Finalmente, se trata de contestar

preguntas del estilo: ¿cómo un archivo llegó a un lugar?, ¿por qué medio se extrajo cierta información?, ¿desde dónde se conectó un usuario?, etc.

Las razones para llevar a cabo un análisis digital forense pueden ser muy variadas pero se puede notar una diferencia entre empresas privadas y públicas. Las primeras generalmente desean establecer si es que existe alguna fuga de información y detectar el uso indebido de los recursos de la empresa. En el sector de gobierno se utiliza para buscar evidencia que apoye un caso legal como secuestro o extorsión, perseguir delitos como la pedofilia o robo.

3.1 Modelo de metodología forense en dispositivos móviles.

A pesar de no existir un estándar para el análisis forense de dispositivos móviles, existen metodologías usadas comúnmente y requisitos mínimos en el Código Federal de Procedimientos Penales en México (Código Federal de Procedimientos Penales).

A continuación se muestra un modelo tomado de la presentación para el taller de computo forense impartido por el M.C. Gerardo Huerta Lozada.

Preparación. Se refiere a todo aquello que se hace antes de estar en contacto con la escena de un incidente y que permite una correcta ejecución del proceso forense en general. Incluye, adquisición de cables que pudieran ser necesarios para la extracción de la información, herramientas, información sobre el incidente, etc.

Aseguramiento de la escena. Una vez localizada la escena del incidente se debe de establecer un perímetro, controlar el acceso físico a éste y comenzar a documentar quién ha tenido acceso a qué fuentes, con qué finalidad y qué se hizo con cada cosa. Con este paso se asegura que la evidencia no ha sido modificada y se generan los primeros documentos para demostrar quién ha estado en contacto con ésta y para qué.

Entrevistas y reconocimiento. El objetivo de este paso es identificar de qué fuentes se puede obtener información relevante, se reconocen entonces tarjetas de memoria, discos duros, dispositivos de red que puedan albergar bitácoras, etc.

Documentación de la escena. Se deben tomar fotografías de la escena, diagramas, etc. Esto permitirá recrear la escena si es necesario y aportará evidencia que no puede ser levantada de otra forma.

Bloqueo de las comunicaciones. Antes de extraer información se beben bloquear las comunicaciones desde y hacia el dispositivo móvil. Esto asegura que la información contenida no vaya a ser modificada remotamente.

Recolección de datos volátiles. La información volátil es muy valiosa ya que muestra el estado en el que se encontraba el dispositivo en el momento del incidente y en caso de terminarse la batería o reiniciarse el teléfono esta información se puede perder, por lo que los primeros datos que se recolectan son los volátiles.

Recolección de datos no volátiles. A continuación se lleva la recolección de información contenida en tarjetas externas, SIM, memoria interna no volátil, respaldos, etc.

Preservación. El objetivo de este paso es empacar y transportar hacia donde se llevará acabo el análisis y almacenar de manera confiable los datos recolectados en los pasos anteriores.

Preparación para el análisis. Se realiza un respaldo de la información adquirida para asegurarse que los respaldos originales no se vayan a modificar. Se hace un análisis rápido buscando palabras clave, archivos en carpetas de sistema, etc.

Análisis. Se lleva a cabo un análisis más profundo teniendo como objetivo responder preguntas clave para el caso en cuestión. Se correlacionan eventos tomando información de diferentes fuentes, se construye una línea de tiempo y se obtienen conclusiones.

Presentación. Con las conclusiones obtenidas y los hallazgos encontrados, se elabora un reporte.

El procedimiento descrito cuenta con muchos retos a los que el investigador digital forense tiene que enfrentarse. A continuación se describen algunos:

3.1.1 Problema de adquisición de datos.

Este problema se presenta de manera más significativa en la adquisición de datos en dispositivos móviles que en computadoras tradicionales. Un dispositivo móvil puede contar con diferentes tipos de memorias externas: SD, MicroSD, Stick Pro Duo, etc. Lo que obliga a un investigador a contar con todo tipo de conectores y adaptadores que le permitan leer las diferentes tarjetas de memoria.

Para las interfaces de datos en los dispositivos móviles éste problema ha disminuido desde que se aceptó el conector “Micro USB” como interfaz estándar en los dispositivos móviles (Reardon, 2009). Aún existen dispositivos que no cuentan con esta interfaz como los de la marca Apple que cuenta con su conector propietario.

Toda esta preparación se hace con la finalidad de obtener una copia lo más fiel posible al original, por lo que es importante contar con un método de comparación que nos pueda decir si contamos con una copia exacta al original y aún más importante qué archivos fueron modificados en el proceso.

3.1.2 Problema de la cadena de custodia.

El objetivo de la cadena de custodia es poder validar que la información que se está presentando como evidencia es confiable y que no ha sido modificada durante el proceso de investigación. Por lo que cualquier persona debe poder entender cómo ésta fue identificada en la escena, qué métodos y técnicas se emplearon para hacer la adquisición, cómo se almacenó y cómo se transportó, quién ha estado en contacto con la evidencia en cada paso del proceso y con qué objetivo. Cualquier interacción con la información del incidente debe quedar registrada.

De hecho, el concepto de la cadena de custodia se comparte entre el mundo forense físico y el mundo forense digital y es algo que tiene mucho sentido cuando nos damos cuenta que en ambos casos estamos manipulando evidencia que debe llegar con la mejor calidad y fidelidad posible a un tribunal.

“La cadena de custodia iniciará donde se descubra, encuentre o levante la evidencia física y finalizará por orden de la autoridad competente.” (Código Federal de Procedimientos Penales)

3.1.3 Problema de la reproducción del análisis

Un investigador debe ser capaz de reconstruir los hechos relevantes para el caso y debe documentar todo para que alguien más sea capaz de obtener los mismos resultados al analizar la misma evidencia.

Esto también asegura que la evidencia no está siendo modificada, que el investigador está siguiendo metodologías claras y sus resultados son objetivos.

3.1.4 Problema de la presentación de datos como evidencia.

El problema de la presentación de datos como evidencia se refiere a que una prueba obtenida como resultado de un proceso forense no se tomará en cuenta como prueba plena (suficiente) para establecer la responsabilidad de alguien sobre algún hecho.

Por lo tanto, el resultado de cualquier proceso forense siempre debe ir acompañado de otras pruebas. Por sí mismo no significa prueba plena.

En el Código Federal de Procedimientos Penales (Código Federal de Procedimientos Penales) se contempla el valor de las pruebas y establece cuales son pruebas plenas:

“Artículo 280. Los documentos públicos harán prueba plena, salvo el derecho de las partes para redargüirlos de falsedad y para pedir su cotejo con los protocolos o con los originales existentes en los archivos.”

“Artículo 284. La inspección, así como el resultado de los cateos, harán prueba plena siempre que se practiquen con los requisitos legales.”

“Artículo 285. Todos los demás medios de prueba o de investigación y la confesión, salvo lo previsto en el segundo párrafo del artículo 279, constituyen meros indicios. La información, datos o pruebas obtenidas con motivo de recompensas, no podrán desestimarse por ese sólo hecho por el juzgador y deberán apreciarse y valorarse en términos del presente capítulo“.

3.2 Algunos casos apoyados por la metodología digital forense.

La metodología digital forense ha servido de apoyo para establecer responsabilidades en casos legales, por lo que es común encontrar ejemplos donde ha servido como herramienta para diferentes sistemas de justicia alrededor del mundo.

Del mismo modo, el análisis digital forense para dispositivos móviles comienza a tomar forma y aunque se pueden encontrar algunos ejemplos, aún no existen muchos casos documentados con el nivel de detalle que tiene su contraparte forense en dispositivos de cómputo tradicional.

A continuación se presentan algunos casos donde se puede observar cómo el análisis digital forense móvil ha tomado fuerza.

3.2.1 Rastreo de personas a través de dispositivos móviles.

En Septiembre del 2012, durante la conferencia de seguridad informática “44Con” llevada a cabo en Londres los investigadores Daniel Cuthbert y Glen Wilkinson de Sensepost llevaron a cabo una demostración de la información que se podía obtener de un dispositivo móvil utilizando técnicas pasivas.

“Escuchando” los paquetes que envía un dispositivo móvil cuyo radio “Wi-Fi” se encuentra encendido, los investigadores pueden reconocer el SSID y el BSSID de las redes a las que el dispositivo se ha conectado, adicionalmente se pueden emplear herramientas públicamente disponibles de geo localización para trazar una ruta de los lugares frecuentados por una persona. (Leyden, 2012)

Esto es posible porque los dispositivos móviles almacenan información de las redes inalámbricas a las que comúnmente se conectan, después, cuando no se encuentran conectados a ninguna red y su radio “Wi-Fi” se encuentra encendido, envían paquetes llamados “probe” que intentan identificar las redes a las que comúnmente se conectan. En cierta forma envían preguntas del tipo: ¿Eres tú la red llamada “FamPerez”?, un solo dispositivo hace esto para varias de las redes a las que comúnmente se conecta. Por lo que sólo es cuestión de que alguien se ponga a “escuchar” el canal para que obtenga dicha información, lleve a cabo una correlación de información para obtener nombres de establecimientos y ubicación geográfica.

Más aún, alguien puede establecer un radio Wi-Fi en modo Access Point, establecer el SSID y el BSSID a un valor específico y hacerse pasar por el Access Point de una red conocida por el dispositivo móvil, el móvil se asociará con el Access Point y si está configurado para hacerlo, empezará a sincronizar información cómo correos, notas, tareas, pendientes. Todo esto sin la interacción del dueño del móvil. En este escenario, el atacante queda en medio de la comunicación pues el controla el Access Point y puede analizar el tráfico en tránsito. Podría engañar al usuario para que acepte certificados digitales que no corresponden a los sitios que visita, modificar el código de las páginas visitadas y de esta forma obtener credenciales de usuario y obtener control sobre el móvil. (Suggy, 2012)

Si bien es cierto que mucho de esto se debe al comportamiento propio de la suite de protocolos 802.11, es a través del análisis forense que los investigadores se han dado cuenta de este tipo de comportamientos donde lo que se diseña con la finalidad de que sea conveniente y fácil de usar resulta poco seguro en la práctica.

3.2.2 Malware en el mercado de aplicaciones móviles de Google.

Una de las diferencias entre los principales desarrolladores de sistemas operativos para dispositivos móviles es el control que imponen sobre las aplicaciones que serán instaladas en dichos sistemas.

Por un lado se encuentra Apple que restringe las aplicaciones que cualquier persona puede instalar en un iPhone a lo que se encuentra únicamente en su mercado, si no hay una aplicación aprobada por la compañía entonces no podrá ser instalada en el sistema, al menos no de forma normal ya que existen diferentes métodos para hacer un “Jailbreak” a un dispositivo iOS y de esa forma acceder a un mercado de aplicaciones aún mayor. (Sell My Application)

Google en cambio, no establece tantos controles para agregar aplicaciones a su mercado, por lo menos, no hay un proceso de aprobación como el de Apple por lo que prácticamente se puede poner a disposición cualquier aplicación, sumado a esto, un usuario pueden instalar aplicaciones que no están en el mercado. Descargar una aplicación de una página web e instalándola en su dispositivo es relativamente sencillo. Esto da muchas facilidades para que alguien pueda engañar a los usuarios proveyendo de aplicaciones falsas ya sea a través del mercado oficial o fuera de éste. (Android)

Y aunque lo anterior no es nada nuevo sigue siendo motivo de análisis por muchos especialistas y firmas de seguridad como la firma McAfee que en Octubre del 2012 realizó el análisis de un virus específico para dispositivos Android, éste virus se hace pasar por aplicaciones de gran demanda como Skype, Flash Player, Opera, etc. Una vez instalado envía mensajes de texto a servicios Premium de cobro sin la autorización del usuario. El 60% del malware móvil que analiza McAfee pertenece a esta familia de virus. (Ruiz, 2012)

Una característica adicional es que esta familia de virus emplea técnicas para evitar ser detectado y analizado, modifica su código con cada descarga, emplea ofuscación de código y técnicas para evitar ingeniería inversa.

En conclusión, el análisis forense tiene muchas utilidades, desde detectar aplicaciones maliciosas antes de que un usuario las instale y contamine su dispositivo, detectar fugas de información en alguna organización o proveer evidencia en un caso legal. Los usos son muy diversos, sin embargo, para asegurar los mejores resultados debemos asegurarnos que el proceso empleado, las herramientas y los expertos son los adecuados.

3.2.3 Robos bancarios relacionados con malware en Android.

En noviembre del 2012 se reportó en Berlín que las cuentas de múltiples clientes bancarios fueron vaciadas. En todos los casos los clientes contaban con teléfonos Android y habían activado un servicio de envío de Tokens por SMS directamente con el banco. (The Local - Germany's news in English, 2012)

Estos tokens funcionan como códigos de verificación, de tal forma que cuando un usuario está a punto de realizar una transacción bancaria debe primero confirmarla con el código recibido por SMS que supuestamente sólo está en poder del cliente bancario.

Se cree que el robo sucedió de la siguiente manera: Los clientes bancarios fueron inicialmente infectados a través del equipo por el que ingresan al sitio web de su banco, generalmente sus computadoras de escritorio o sus equipos portátiles. El virus obtiene la información de la cuenta: Banco, número de cuenta, usuario, contraseña. Al detectar que el usuario está visitando la página web del banco genera una alerta que intenta imitar las alertas emitidas por el banco, avisando al cliente que debe proteger su dispositivo móvil a través de una actualización de seguridad. Solicita el número y el modelo del teléfono, a través de SMS el cliente recibe un enlace para descargar la supuesta actualización. La actualización es un programa que identifica los Tokens recibidos por el teléfono móvil y los reenvía a un tercero. Los tokens en conjunto con el malware instalado en la computadora personal del cliente bancario permiten a un externo llevar acabo transacciones sin el conocimiento de éste.

Éste tipo de ataques se basan en la “ingeniería social” que se refiere a intentar engañar a un sujeto para que éste haga algo que en una situación normal no haría. En éste caso un atacante intenta engañar a un tarjetahabiente con la finalidad de que éste instale programas maliciosos en su dispositivo móvil.

Es evidente la relación que existe entre este caso y el anterior de “Malware en el mercado de aplicaciones móviles de Google”. Sin embargo para llegar a conocer las funciones del malware o incluso obtener una muestra de éste, es necesario seguir procedimientos de análisis digital forense.

Capítulo 4. Estado del arte del análisis digital forense para dispositivos móviles con S.O. de código abierto.

El análisis digital forense en dispositivos móviles es relativamente nuevo en comparación con el análisis digital en sistemas de cómputo tradicionales. A pesar de que el número de dispositivos móviles y la información que éstos almacenan aumentan constantemente, el análisis en dispositivos móviles no cuenta con tantos procedimientos, guías y casos documentados.

Existen organizaciones gubernamentales, en diferentes países, que se encargan de desarrollar y publicar guías que sirvan de referencia para los investigadores forenses. En el campo que nos ocupa existen las guías: "Guía de buenas prácticas para evidencia electrónica basada en computadoras" (en inglés "Good practice guide for computer based electronic evidence") de la ACPO (Association of Chief Police Officers. England Wales & Nireland, 2007) en el Reino Unido. Esta guía cuenta con una diversidad de temas de aplicación como redes caseras, tecnologías inalámbricas, video cámaras de circuito cerrado, investigación de escenas de crimen y de personal y específicamente una guía de incautación y estudio de dispositivos móviles.

Existe también una guía más especializada en teléfonos celulares llamada Guía Forense para teléfonos celulares (en inglés "Guidelines on CellPhone Forensics") del NIST (National Institute of Standards and Technology. Technology Administration U.S. Department of Commerce, 2007) de Estados Unidos, ésta a pesar de ser una guía extensa y detallada, claramente establece que no debe tomarse de manera literal por los investigadores forenses ni por las unidades legales de alguna organización, más bien, debe servir como punto de referencia para que en conjunto con consejeros legales, oficiales y administradores se establezcan políticas, procedimientos y controles adecuados.

El análisis forense en sistemas de cómputo tradicional que cuentan con un disco duro emplea técnicas que se encargan de copiar la información total de dicho disco duro y posteriormente hacer un análisis sobre la copia, en el análisis forense de dispositivos móviles uno de los principales problemas es obtener dicha copia de la información. Es un problema complejo principalmente por los controles de acceso que emplean los sistemas operativos para dispositivos móviles y por la dificultad que representa extraer la unidad de almacenamiento para copiarla y analizarla.

Por lo anterior, este capítulo pretende mostrar las características de las principales guías de análisis forense para dispositivos móviles así como los alcances y las limitaciones de los métodos y herramientas con las que se cuenta actualmente para llevar a cabo dicho análisis.

4.1 Características de la Guía de buenas prácticas para evidencia electrónica basada en computadoras. ACPO

Esta guía pretende orientar a los investigadores forenses en caso de encontrarse con algún delito que contenga un elemento tecnológico, de esta forma, abarca una diversidad de elementos tecnológicos pero se centra en proporcionar recomendaciones para que la recolección de datos se lleve a cabo de manera correcta y en el momento adecuado.

Desde el inicio establece 4 principios básicos (ACPO, 2007) que deben regir el análisis y la recolección de evidencia digital sin importar la tecnología con la que se esté tratando. Estos principios son:

Principio 1. Ninguna agencia de procuración de justicia ni alguno de sus agentes puede modificar ningún dato almacenado en un sistema de cómputo o en algún medio de almacenamiento que posteriormente vaya a ser presentado frente a alguna corte.

Este principio intenta hacer cumplir que lo que se presenta ante una corte es lo que se encontró en la escena. Para cumplir con esto el documento propone aislar el dispositivo de su red. Se pretende que el dispositivo no pueda ser alterado de manera remota a través de alguna aplicación específica para ello, existen una diversidad de formas para conseguir el aislamiento, sin embargo, cada una tiene consecuencias específicas que un investigador forense debe conocer.

Se recomienda utilizar una jaula de Faraday para resguardar el dispositivo, cuidando que la batería no se agote ya que es seguro que la información volátil se perderá. Por otra parte, si existe alguna tarea programada en el dispositivo ésta se llevará a cabo afectando la información contenida. Para establecer conexiones confiables para extraer información, el uso de interfaces Bluetooth e Infra-roja no se recomiendan ya que requieren un modo de activación y autenticación que modifica la configuración en el dispositivo, la primera recomendación es una interfaz a través del cable de datos que provee el fabricante, sin embargo, por lo menos en Android es necesario activar el modo “Debug” para establecer comunicación, más adelante se explicará dicho modo.

El investigador forense debe tener presente en todo momento que el proceso muy probablemente modificará la información contenida y ésta es una característica propia del análisis en dispositivos móviles. Incluso la inacción provoca cambios en los datos almacenados en el teléfono, al cambiar el reloj o la fecha, el estado de la batería o el estado en el que se encuentra el dispositivo. Sin embargo, siempre es importante conocer cuáles son esos cambios y documentarlos apropiadamente.

Entender las herramientas con las que se cuenta y tomar decisiones en cuanto a los pasos a seguir en el análisis es muy importante. En algunas situaciones una herramienta puede requerir el reinicio del dispositivo o incluso instalar una aplicación; estas decisiones implican perdida o modificación de datos. Entender qué datos se modificarán o se perderán y cuáles de éstos son importantes para el caso legal es importante para una correcta toma de decisiones.

Por lo tanto, un investigador forense no debe sólo conocer cómo utilizar sus herramientas sino que implicaciones tiene cada una y entender el contexto del caso.

Principio 2. En caso de que sea necesario para alguien acceder a los datos originales almacenados en un sistema de cómputo o en un medio de almacenamiento, dicha persona debe ser competente para hacerlo y debe poder proporcionar evidencia de la relevancia y las implicaciones de sus actos.

La importancia de este principio reside en la necesidad de no modificar la evidencia durante el proceso forense digital. Sin embargo, en el análisis digital de dispositivos móviles existen situaciones en las que modificar la evidencia es inevitable, por ejemplo, puede ser necesario remover la batería para obtener la tarjeta SIM o la tarjeta de memoria externa y esto provoca perdida o modificación de datos.

Como ya se explicó en el principio anterior, es importante que los investigadores conozcan los dispositivos, sus principios de operación, las herramientas y el contexto del caso para que puedan tomar decisiones en cuanto al procedimiento que deben seguir durante el análisis digital de dispositivos móviles.

Principio 3. Debe crearse un registro de auditoria u otro registro de todos los procedimientos aplicados a la evidencia electrónica, mantenerse actualizado y almacenarse. Un tercero debe ser capaz de examinar los procedimientos utilizados y obtener los mismos resultados.

Este punto centra su importancia en la documentación de la evidencia desde el momento en el que se encuentra hasta las interacciones que se tienen con ella. Como medios de registro generalmente se utilizan fotografías, video y registros escritos. De esta forma se crea y se mantiene la cadena de custodia. La cadena de custodia es el documento que contiene la historia de las interacciones que se han tenido con la evidencia electrónica y permite a un tercero entender y reproducir los resultados obtenidos.

Principio 4. La persona a cargo de la investigación, llamado oficial del caso, tiene la responsabilidad de asegurarse que la ley y estos principios se cumplan.

Este principio habla de una persona que debe estar a cargo de la investigación, que pueda verificar que personal propiamente entrenado esté en contacto con la evidencia, que tenga una visión más general del caso y su contexto y por lo tanto que tenga la capacidad de decidir cuándo es preferible utilizar un método de adquisición o análisis sobre otro y de esta forma apoye la toma de decisiones.

La guía de la ACPO también presenta consideraciones finales. Habla sobre otro tipo de evidencia que se puede encontrar en estos dispositivos como las huellas dactilares y la importancia de conocer el proceso de adquisición de huellas dactilares ya que podrían dejar el dispositivo móvil inservible. Menciona la importancia de contar con cables, conectores y adaptadores para diferentes tarjetas de almacenamiento y tener una referencia de los detalles de los diferentes modelos de dispositivos móviles; saber que existen procedimientos automatizados que pueden

eliminar información de dichos dispositivos. Todo esto ayuda a llevar a cabo un mejor análisis forense y prevenir perdida de información.

Finalmente, se presenta un modelo en forma de diagrama de flujo para la incautación de dispositivos tipo asistente digital. El modelo presenta los pasos a seguir desde que se encuentra el dispositivo, cómo documentar la evidencia y su estado; almacenamiento del dispositivo y finaliza con la entrega del dispositivo a un especialista en análisis digital forense; por lo tanto, en el modelo no se muestran los pasos y procedimientos para una adquisición ni análisis de datos. El modelo es aplicable también a teléfonos inteligentes ya que se menciona una jaula de Faraday para dicho dispositivo.

El modelo presentado en la guía de la ACPO se puede consultar en el Anexo B de este trabajo.

4.2 Características de la Guía Forense para teléfonos celulares. NIST

Esta guía se centra en teléfonos celulares incluyendo teléfonos inteligentes. No se centra en algún sistema operativo en particular, sin embargo, presenta procedimientos para la preservación, adquisición, estudio, análisis y reporte de la información contenida en los teléfonos celulares. También presenta herramientas que ayudan a llevar a cabo dichas actividades.

Esta guía no pretende servir como un manual para ser implementada tal como está, pues requiere que las organizaciones en conjunto con consejeros legales y sus niveles directivos sean quienes entiendan su problemática y como consecuencia generen políticas y procedimientos.

Como antecedente la guía explica de manera breve acerca de las tecnologías celulares GSM y CDMA. A grandes rasgos cómo trabajan los operadores de redes celulares y cómo está conformada una célula de telefonía. Explica también las diferencias entre las capacidades y prestaciones de los teléfonos celulares tradicionales y los teléfonos inteligentes, qué los diferencia en cuanto a componentes y capacidades. Termina la sección de antecedentes con una explicación del Módulo de Identificación del Suscriptor (SIM por sus siglas en inglés), explica la función de identificación del usuario con la red celular incluyendo las funciones de cifrado y de almacenamiento seguro de las llaves criptográficas, los datos que un investigador forense puede encontrar como registro de contactos, de llamadas y de mensajes; explica sobre el número de identificación personal (PIN por sus siglas en inglés) y cómo sirve para permitir o negar acceso a los recursos del teléfono a través del SIM.

Los detalles del módulo SIM son únicamente aplicables para teléfonos GSM ya que los teléfonos CDMA tienen las funcionalidades de autenticación con la red incorporadas directamente en el teléfono.

El capítulo de herramientas forenses explica que el análisis digital forense es un campo relativamente nuevo y por lo tanto no existen tantas herramientas como es el caso de los sistemas de cómputo tradicional.

La guía menciona las 2 principales técnicas de adquisición de datos para dispositivos móviles:

- Adquisición física: Que consiste en hacer una copia bit a bit de algún dispositivo de almacenamiento, para el caso de los teléfonos móviles sería un chip de memoria Flash. En el caso de los sistemas de cómputo tradicionales sería una copia bit a bit de un disco duro.
- Adquisición lógica: Que consiste en hacer una copia bit a bit de una unidad lógica, por ejemplo, un sistema de archivos, un conjunto de carpetas, etc.

En el caso que se decida llevar a cabo una adquisición física, es posible recuperar archivos que han sido eliminados por el usuario (ver sección 2.2.2).

En cuanto a las herramientas para adquisición de datos, el documento lista una variedad, haciendo notar las diferencias entre herramientas comerciales, herramientas creadas por los fabricantes para la administración de un dispositivo, herramientas de código libre, herramientas independientes, herramientas de diagnóstico y herramientas hechas por terceros. De nuevo, se muestra la importancia de conocer cada herramienta que se va a utilizar, sus implicaciones en cuanto a la modificación del contenido y las capacidades de adquisición de dichos datos.

También se mencionan las diferentes interfaces que se utilizan para la comunicación con el dispositivo, siendo la interfaz de cable de datos la más confiable, sin embargo, cuando no se tiene disponible el cable apropiado o el uso de una herramienta requiere utilizar una interfaz diferente se debe tener en cuenta las consideraciones apropiadas, por ejemplo, una interfaz “bluetooth” implica que el teléfono móvil tiene que dar de alta el dispositivo con el que se va a comunicar, lo cual modifica la información de dispositivos con los que se ha comunicado.

En el listado de herramientas se presentan organizadas por función de la siguiente manera:

Herramientas de análisis del Módulo de Identidad del Suscriptor (SIM), las herramientas listadas llevan a cabo una adquisición a través del análisis directo de un lector de SIM en lugar de hacer la lectura a través del dispositivo móvil. La mayoría de estas herramientas pueden obtener los siguientes datos: Identidad Internacional del Suscriptor Móvil (IMSI), últimos números marcados (LDN), mensajes cortos (SMS) y la información de ubicación con respecto a la red celular (LOCI). Sólo algunas herramientas pueden obtener también además de la información ya listada: Mensajes cortos eliminados del teléfono, mensajes multimedia con gráficos y sonidos simples e intentan traducir ciertos datos como el código de operador y de país. Las herramientas mencionadas son: “Forensic Card Reader”, “The Forensic SIM Toolkit”, “SIMCon”, “SIMIS” y “USIM Detective”

Herramientas de análisis de dispositivos móviles: Estas herramientas están diseñadas para la adquisición de la memoria interna de los dispositivos. Generalmente surgen a partir de herramientas creadas por los fabricantes con la finalidad de administrar dispositivos móviles pero se modifican para deshabilitar la capacidad de escritura hacia el dispositivo y no cuentan con la capacidad de obtener información del módulo SIM. Nacen de las primeras herramientas de administración de Asistentes Digitales Personales (PDAs) por lo que naturalmente funcionan con sistemas operativos como Palm OS o Windows Mobile. Las herramientas mencionadas son: "PDA Seizure", "Pilot-link", "Oxygen Phone Manager" y "BitPim".

Herramientas Integradas: Son herramientas con capacidades de extracción y análisis del módulo SIM y del dispositivo móvil. Ofrecen la ventaja de integrar los resultados en un solo reporte. Las herramientas mencionadas son: "Call Seizure", "CallDek", "GSM .XRY", "MOBILedit! Forensic", "PhoneBase2", "SecureView", "TULP2G".

El estándar cuenta con un capítulo específico para procedimientos y principios. Pretende ser una serie de recomendaciones a seguir y principios a tener en cuenta para manejar investigaciones e incidentes.

El primer punto a tomar en cuenta es la preparación previa. La organización debe establecer roles específicos a cumplir durante un incidente, en la práctica quizá una sola persona tenga que cumplir con más de un rol, lo importante es que cada persona esté preparada y sepa cuál es su papel durante un incidente. Los roles que se debe cubrir son:

- Equipo de primera respuesta: Ellos son los primeros en llegar a la escena de un incidente, por lo tanto, deben tener la capacidad de evaluar el tipo y la magnitud del incidente. Son los responsables de llamar a los equipos específicos de soporte que sean necesarios y de asistir con las tareas de recolección de evidencia.
- Investigadores: El equipo de investigadores se encarga de planear y llevar a cabo la preservación, adquisición, estudio, análisis y elaboración del reporte de la evidencia digital. Debe existir un líder del equipo de investigadores quien sea el encargado de asegurarse que las tareas sean llevadas a cabo en tiempo y los resultados entregados a las personas correspondientes.
- Técnicos: Son personal altamente entrenado en la operación de herramientas de adquisición de copias de datos. Sus responsabilidades en la escena del incidente son: identificar, colectar y documentar la evidencia. Se encargan de obtener copias de los datos residentes en medios electrónicos. Por lo general se necesita más de un técnico en la escena de un incidente ya que se requiere interactuar con una diversidad de dispositivos así como de medios de almacenamiento.
- Custodios de la evidencia: Son responsables de proteger la evidencia obtenida en un incidente. Reciben la evidencia adquirida por los técnicos, verifican que se encuentre apropiadamente etiquetada y crean y actualizan registros de entrada y salida de evidencia.

- Examinadores forenses: A partir de las copias obtenidas por los técnicos se encargan de analizar la información almacenada y hacer evidentes datos específicos.
- Analistas forenses: Se encargan de analizar el resultado del proceso llevado a cabo por los examinadores forenses. Evalúan y reconocen la relevancia de los datos obtenidos para la el caso por lo que deben entender los detalles específicos del caso y tener la habilidad de relacionar eventos y sucesos.

Después de presentar los principales roles que se deben tomar en cuenta durante la etapa de planeación, la guía aborda el tema de los principios básicos a tomar en cuenta para el manejo de la evidencia digital. Menciona que toda evidencia digital tiene 2 aspectos principales: los componentes físicos, periféricos y los medios que pueden contener información y la información ya extraída de dichos medios. Cada uno de estos componentes debe contar con su propia cadena de custodia.

La guía del NIST hace referencia a otras guías y estándares para dar ejemplos de principios básicos de manejo de evidencia. Hace referencia a los 4 principios básicos de la Guía de buenas prácticas para evidencia electrónica basada en computadoras de la ACPO, los estándares propuestos para el intercambio de evidencia digital (International Organization on Digital Evidence IOCE, 2000). Los estándares de la IOCE proponen principios similares a los de la ACPO:

- Al incautar evidencia digital, ninguna acción debe modificar dicha evidencia.
- Cuando una persona tiene la obligación de acceder a la evidencia digital original, dicha persona debe ser competente.
- Toda actividad relacionada con la incautación, acceso, almacenamiento o transferencia de evidencia digital debe ser completamente documentado, preservado y estar disponible para revisión.
- Una persona es responsable por cualquier acción tomada con respecto a alguna evidencia digital cuando dicha evidencia se encuentra en su posesión.
- Cualquier agencia responsable por la incautación, acceso, almacenamiento o transferencia de evidencia digital es también responsable de cumplir con estos principios.

Se presentan también los principios de confiabilidad del método Daubert (Henry F. Fradella, 2004) que se aplican a cualquier tipo de evidencia, no sólo evidencia digital. Estos principios deben tenerse en cuenta cuando se aplica y se reporta sobre un principio científico, los principios presentados son:

- Comprobación: La teoría científica o la técnica empleada deben haber sido comprobadas.
- Aceptación: La teoría científica o la técnica debe haber sido revisada por miembros de la comunidad, esto asegura que posibles fallas puedan ser encontradas y solucionadas.

- Tasa de error: La tasa y el tipo de errores obtenidos a partir de la aplicación de una teoría o una técnica específica se conocen y se pueden predecir, esto contribuye a conocer la validez y confiabilidad de los resultados obtenidos.
- Credibilidad: La dependencia de la técnica empleada a un conjunto de habilidades y herramientas específicas o la capacidad de que los mismos resultados se puedan obtener empleando otras herramientas y por otras personas. Esto implica que personas independientes puedan verificar los resultados obtenidos.
- Claridad: La técnica empleada y sus resultados pueden ser explicados con suficiente claridad, de tal forma que un juez y un jurado puedan entender el funcionamiento básico y sus implicaciones.

Es muy importante siempre intentar cumplir con los principios del método Daubert sin importar el caso del que se trate pues es probable que al inicio de una investigación no se tenga muy clara la importancia ni la gravedad del asunto en curso pudiendo contaminar o ignorar evidencia que después tenga que ser presentada en un caso legal.

Finalmente, para el capítulo de procedimientos y principios se presentan 3 modelos que sirven de referencia cuando se interactúa con la escena de un incidente. A pesar de que los 3 modelos fueron diseñados para tratar con evidencia digital basada en computadoras se puede aplicar la misma lógica para evidencia basada en dispositivos móviles.

Se presentan: Investigación de escenas de crimen electrónico - La guía de primera respuesta (National Institute of Justice U.S., 2008), Respuesta a incidentes – Investigando crímenes basados en computadoras (Kevin Mandia, 2001) y Un estudio a los modelos forenses digitales (Mark Reith, 2002). Los 3 documentos proponen modelos que tienen cosas en común y cosas complementarias. Integrando las ideas de los 3 documentos, los puntos principales serían:

- Preparación previa al incidente: Contempla organización, definición de roles y entrenamiento. Con la finalidad de tener el equipo, el conocimiento y el personal listo para enfrentar un incidente.
- Detección de incidentes: Definir y poner en práctica una metodología para identificar incidentes.
- Primera respuesta: Llevar a cabo prácticas para asegurar la escena del incidente, reconocer el tipo de incidente e iniciar la documentación de la escena. Desde este momento cualquier interacción con la evidencia tendrá que registrarse con detalles en la documentación.
- Recolección de evidencia: Aislar, asegurar y preservar la evidencia digital llevando a cabo una copia de ésta, empaquetándola, etiquetándola y transportándola para su almacenamiento, duplicación y posterior análisis.
- Análisis: Analizar de manera integral una copia de la evidencia, intentando reconstruir hechos a partir de la información obtenida.
- Creación de un reporte: Crear un reporte de la metodología y técnicas empleadas en la investigación así como de los hechos reconstruidos y las conclusiones obtenidas.

- El último paso varía dependiendo del documento que se consulte. El documento "Respuesta a incidentes – Investigando crímenes basados en computadoras" propone aprender del incidente ocurrido y llevar a cabo medidas que puedan prevenir otro incidente similar en el futuro. Mientras que el documento "Un estudio a los modelos forenses digitales" propone asegurarse que los bienes físicos y digitales sean regresados a su dueño legítimo.

El resto de la guía Forense para teléfonos celulares del NIST se enfoca en 4 pasos muy importantes para las ciencias forenses:

- Preservación: Cuyo objetivo es obtener una muestra o un objeto de aquello que contiene la información a analizar sin modificar o alterar los datos contenidos en el dispositivo.
- Adquisición: Cuyo objetivo es obtener una copia forense de la información contenida en el objeto obtenido. Para nuestro caso será una "imagen" del contenido de un dispositivo de almacenamiento. Las características de dicha "imagen" se mencionarán en esta sección y las distintas formas de obtenerla se mencionarán posteriormente.
- Análisis: Cuyo objetivo es obtener información a partir del análisis de la copia forense.
- Reporte: Cuyo objetivo es presentar de manera clara los métodos y las técnicas empleadas durante el proceso y las conclusiones y resultados obtenidos.

El resto de esta sección se dedicará a dar una perspectiva de lo que ofrece la guía del NIST acerca de cada uno de los cuatro pasos listados y su implicación en una investigación digital forense.

4.2.1 Preservación.

La preservación es el primer paso a llevar a cabo cuando se llega a la escena de un incidente. Su objetivo principal es proteger la escena, y cualquier elemento en ella de algún agente externo que pueda modificarla. No se aprecia como una tarea fácil pues la simple interacción con la escena implica su modificación, sin embargo, se hace uso de métodos y técnicas para minimizar dichas modificaciones y registrarlas, de tal forma que estos cambios se tengan presentes cuando se presenta la evidencia y se obtengan las conclusiones.

Los puntos mencionados como importantes en la guía del NIST son:

- **Asegurar y evaluar la escena:** Su principal objetivo con respecto a los dispositivos móviles es prepararlos para una correcta recolección, de tal forma que al ser incautados aún contengan información útil para el caso. Por lo que primero se debe reconocer el estado en el que se encuentran dichos dispositivos y tomar las acciones necesarias para mantenerlos en dicho estado, poniendo especial atención en: cuidar que la batería de los dispositivos no se termine (para evitar perder información volátil), prevenir la modificación remota o automatizada del contenido (aislando el dispositivo de las redes de comunicación y evitando la ejecución de programas

automatizados que puedan eliminar o modificar su contenido), identificar cualquier medio de almacenamiento que pueda contener información del dispositivo (tarjetas de memoria externa, módulos SIM, incluso computadoras donde se pudo haber sincronizado la información).

- **Documentación de la escena:** La documentación es un proceso que debe iniciar con el reconocimiento de la escena y termina cuando concluye el caso. Su objetivo es proveer la habilidad de rastrear lo que ha sucedido con la evidencia desde su descubrimiento. Su finalidad de proteger la integridad de la información contenida en dicha evidencia. Se vale de métodos como las fotografías, sobre todo de la escena del incidente (para posteriormente reconocer dónde se encontraba físicamente la evidencia) y la escritura en bitácoras que deberá ser constantemente actualizada y responder las siguientes preguntas: ¿En qué medio se encontraba almacenada cierta información?, ¿Cómo y dónde se recolectó la información?, ¿Quién tomó posesión de la evidencia?, ¿Dónde se ha almacenado y bajo qué circunstancias?, ¿Qué medidas se han implementado para su protección?, ¿Quién la ha sacado de su almacenamiento y con qué finalidad, qué procedimientos se le han llevado a cabo y que cambios implican?
- **Recolección de la evidencia:** Una vez reconocido el estado en el que se encuentran los dispositivos y documentada la escena. Este paso se encarga de aplicar las técnicas necesarias para su correcta incautación. Su objetivo es facilitar la tarea posterior de adquisición con la finalidad de maximizar la calidad de la información que se vaya a extraer posteriormente. Se vale de técnicas como: aislar el dispositivo, recolectar medios externos de almacenamiento y cables de comunicación, realizar entrevistas a los dueños de los dispositivos (siempre que sea posible) para solicitar contraseñas y claves de acceso.
- **Empaquetado, transportación y almacenamiento:** El objetivo de este paso es hacer llegar a los analistas la evidencia obtenida de la escena del incidente en las mejores condiciones posibles, con la finalidad de que obtengan la mejor y mayor cantidad posible de información útil para el caso legal y se proteja y almacene en un lugar adecuado, manteniendo la cadena de custodia. Se vale de: empaquetar la evidencia en bolsas aislantes de radiofrecuencia, etiquetar de manera adecuada para su reconocimiento, evitar fuentes de energía magnética en su transportación, cuidar de condiciones extremas de temperatura durante el transporte y el almacenaje, cuidar de golpes y movimientos bruscos, controlar el acceso a la evidencia y actualizar constantemente la documentación con las ultimas situaciones que se han presentado en el manejo de la ésta.

4.2.2 Adquisición.

El proceso de adquisición se encarga de extraer la información contenida en los dispositivos incautados (sin preocuparse de su análisis). A esta copia de la información se le llama "imagen" y se puede obtener en la misma escena del incidente o en un laboratorio especializado. Si la "imagen" se obtiene en el lugar del incidente se podría evitar la pérdida de datos volátiles debido a la falta de baterías pero es necesario contar con las herramientas y las personas necesarias en un sitio externo (lo cual se puede complicar si existe una diversidad de dispositivos a copiar). En cambio, sí la "imagen" se obtiene en un laboratorio especializado es más probable que se cuenten con las herramientas y los expertos necesarios, sin embargo, habrá que proteger el o los dispositivos en tránsito y tener en cuenta el hecho de que las baterías del dispositivo pueden agotarse.

La guía de NIST 800-101 toma en cuenta que la obtención de la "imagen" se llevará a cabo en un laboratorio especializado y que dicho laboratorio se encuentra apropiadamente aislado electromagnéticamente o por lo menos se han empleado técnicas con la finalidad de aislar los dispositivos incautados de las distintas redes de comunicación. (Utilizar contenedores aislantes de radiofrecuencia, establecer el dispositivo en modo avión, remover el módulo SIM, etc.).

La guía menciona que no existe una gran variedad de herramientas que puedan extraer datos de un dispositivo móvil inteligente y que ninguna herramienta ha logrado extraer datos para los diferentes sistemas operativos existentes. Por lo que antes de realizar una adquisición es necesario llevar a cabo una identificación del dispositivo con el que se está trabajando, pues de esto dependerá la herramienta y los procedimientos que se puedan aplicar.

4.2.2.1 Identificación.

El proceso de identificación debe arrojar como resultados: El fabricante y el modelo del dispositivo móvil así como el proveedor que presta servicios a dicho dispositivo. Es importante que un analista forense no siempre se deje llevar por la marca ni el modelo que se encuentran marcados en la carcasa del dispositivo pues estos pueden ser fácilmente alterados, también es posible alterar la pantalla que aparece cuando se enciende el dispositivo. Por lo que siempre es más recomendable observar las características físicas del dispositivo y realizar una búsqueda utilizando las medidas de largo, ancho y peso así como tamaño de pantalla, existen algunas bases de datos públicas que ofrecen este servicio, por ejemplo: *(http://www.phonescoop.com/phones/finder.php , http://www.gsmarena.com/search.php3 , http://mobile.softpedia.com/phoneFinder).*

Otro medio de identificación puede ser el IMEI que es un número de identificación de 15 dígitos que indica el fabricante, el modelo y el país en el que se aprobó el uso del dispositivo. Los primeros 8 dígitos del IMEI especifican el modelo y el país y los últimos 7 especifican el fabricante. El IMEI se puede obtener, de un dispositivo encendido y desbloqueado, marcando la combinación **#06#*

Cuando se ha decidido apagar el teléfono antes de llevar a cabo la adquisición es importante revisar las etiquetas de identificación que se encuentran en la cavidad de la batería, en la mayoría de los dispositivos se puede encontrar el fabricante, modelo e IMEI en dichas etiquetas. Finalmente, para encontrar el proveedor de servicio se puede utilizar como indicio la pantalla que aparece al encender el dispositivo, el contenido guardado en el teléfono (tonos, imágenes, aplicaciones, etc.) y si se cuenta con el número telefónico de dicho dispositivo se puede intentar hacer llamadas al dispositivo cuando éste se encuentra fuera de la red, de tal forma que se escuche un mensaje pre-grabado que permita identificar al proveedor de servicios.

4.2.2.2. Selección de herramientas.

Una vez identificado el modelo y el fabricante del dispositivo se pueden llevar a cabo búsquedas de los manuales de uso del dispositivo, revisar cuáles son las herramientas que se encuentran disponibles para el análisis de dicho dispositivo así como las interfaces y cables necesarios para la extracción de datos. Al seleccionar una herramienta específica es importante tener en cuenta:

- Que los resultados de la herramienta se encuentren en un formato que el investigador forense pueda interpretar y entender.
- Que la calidad de los datos extraídos sea considerablemente buena.
- Que la herramienta haya sido analizada por terceros y su margen de error sea conocido y aceptable.
- Que los resultados obtenidos por la herramienta se puedan reproducir y verificar por medio de otras herramientas similares.

El estándar también recomienda experimentar previamente con la herramienta y dispositivos similares, de tal forma que el investigador se familiarice con su uso, sus características y sus limitaciones.

4.2.2.3 Adquisición.

La adquisición se encarga de hacer una copia de los datos almacenados en un dispositivo. Dicha copia puede ser por adquisición lógica o por adquisición física (ambas mencionadas en la sección 4.2).

Una vez terminada la adquisición, el investigador debe verificar que el proceso se llevó a cabo correctamente, es normal que algunos datos como mensajes almacenados como borrador y mensajes archivados no puedan ser recuperados por una herramienta que emplea adquisición lógica.

También se menciona que se pueden encontrar dispositivos bloqueados con algún método de control de acceso lógico como contraseña, patrón de rayas, PIN o reconocimiento facial. Es importante que un investigador sepa a qué tipo de control de acceso se está enfrentando y que empleé un método comprobado para obtener acceso a los datos almacenados. En el peor de los casos, cuando un usuario intenta adivinar una contraseña, un PIN o un patrón demasiadas veces sin éxito, algunos dispositivos móviles podrían borrar todos los datos almacenados y regresar a un estado preestablecido de fábrica provocando así la perdida de los datos (quedaría aún la posibilidad de realizar una adquisición física y verificar si es posible recuperar dichos datos empleando otros métodos).

Se pueden emplear diversas técnicas para acceder a los datos de un dispositivo protegido por un método de control de acceso, estas técnicas se pueden dividir en 3 categorías: investigativo, basado en software y basado de hardware.

Las técnicas basadas en software y en hardware son altamente dependientes del fabricante y del modelo del dispositivo a analizar, básicamente, explotan vulnerabilidades en los componentes del dispositivo o hacen uso de cuentas pre-establecidas por los fabricantes y desarrolladores; es relativamente fácil enterarse de dichas vulnerabilidades si uno revisa sitios especializados, sin embargo, cuando están basadas en software es posible por medio de una actualización del sistema operativo, corregir las vulnerabilidades explotadas. Métodos basados en hardware implican retirar el chip de memoria del dispositivo para analizarlo por separado.

Las técnicas investigativas no requieren herramientas forenses y están más enfocadas a obtener las contraseñas a través de otros métodos: preguntar a los propietarios del dispositivo durante las entrevistas que se llevan a cabo en la etapa de recolección de evidencia, revisar la evidencia incautada ya que existe la posibilidad de que las contraseñas o los códigos de desbloqueo se encuentren escritos entre éstos, intentar valores por default o posibles valores conocidos para los códigos de desbloqueo cuidando de no exceder número de intentos permitidos por él dispositivo.

Es importante también, como ya se mencionó antes, no olvidar hacer adquisiciones de dispositivos externos de almacenamiento como memorias extraíbles o dispositivos de cómputo hacia los cuales se pudo haber sincronizado el dispositivo móvil.

4.2.3 Estudio y análisis.

Después de haber obtenido copias de los datos almacenados en un dispositivo, se inicia el proceso de estudio. La persona encargada del estudio debe extraer datos específicos como fotografías, videos, mensajes, registros, etc. Mientras que la persona encargada del análisis debe ser capaz de analizar los datos obtenidos por el examinador y seleccionar los más relevantes para el caso legal. Por lo que el analista debe tener una mayor comprensión de los detalles del caso; es también posible que la persona encargada del estudio trabaje en conjunto con la persona encargada del análisis o que incluso sea la misma persona quien ejecute ambas tareas.

El resultado del proceso de estudio debe ser entonces un conjunto de archivos decodificados, en formatos comunes que cualquier sistema de cómputo "común" pueda interpretar, con fechas de creación y modificación, procesos y/o usuarios que los crearon.

La guía del NIST muestra un listado de los tipos de datos que se pueden extraer de un dispositivo móvil, en la sección 2.3 de este trabajo ya se mostró un listado similar específico para dispositivos Android que será en los que nos enfocaremos. En el Anexo C de este trabajo se muestra una traducción de la tabla que aparece en la guía con el tipo de información que pueden aportar dichos elementos.

El analista toma entonces los archivos que resultaron del proceso de estudio, y apoyado por los conocimientos que tiene del caso reconoce datos relevantes. De tal forma que el resultado del proceso de análisis es un subconjunto del conjunto inicial de datos y archivos, organizado por tipo, usuario y proceso que los creó y fecha de creación y modificación.

Para llegar a este resultado un analista debe, a partir de los archivos obtenidos por el examinador, generar una lista de palabras clave, fechas, lugares y personas relevantes, etc. Y buscar estos elementos en los archivos. De manera ordenada se van ligando los resultados obtenidos a personas específicas, lugares y fechas; se emplean elementos como gráficos o líneas de tiempo que permiten reconstruir y visualizar hechos.

En este proceso las herramientas con las que cuenta un analista son de gran importancia, pues realizará búsquedas de elementos clave en una gran cantidad de archivos de texto pero también en fotografías e imágenes, video y audio de tal forma que le sea considerablemente fácil reconocer elementos clave en estos medios. El proceso de organizar y posteriormente encontrar datos que puedan servir como evidencia suele ser laborioso por lo que es importante que el personal encargado de esta tarea se encuentre bien entrenado en uso de sus herramientas y en casos prácticos. En este respecto existen muchas asociaciones que periódicamente emiten "retos forenses" para analistas digitales, proveen una "imagen" producto de una adquisición y piden que sean contestadas una serie de preguntas relacionadas a un caso. Uno de los eventos más conocidos es el "Reto forense digital" organizado por el Departamento de Defensa de los E.U.A, se puede encontrar en: *http://www.dc3.mil/challenge/*

4.2.4 Reporte.

Este proceso consiste en la elaboración y presentación de un resumen incluyendo detalle de los pasos y decisiones tomadas durante la investigación y una presentación de los resultados, hechos y conclusiones obtenidas de la misma. Su objetivo es que una persona pueda entender el proceso seguido y los resultados obtenidos.

Un buen reporte debe estar fundamentado en una buena documentación, notas y fotografías obtenidas del proceso forense completo.

Existen herramientas forenses que permiten generar reportes después de haber hecho un estudio o un análisis, algunas de éstas incluso permiten personalizar dichos reportes con logos, nombres o números de caso dando una presentación más profesional a dichos reportes.

La guía muestra un listado de los datos que debe contener un reporte forense digital:

- Identidad de la agencia encargada del reporte.
- Número o identificador de caso.
- Identidad del analista a cargo del caso.
- Identidad de la persona u organización que presentó el caso.
- Fecha de recepción de solicitud de caso.
- Fecha de entrega de reporte.
- Lista descriptiva de elementos presentados para investigación, incluyendo número de serie, fabricante y modelo.
- Identidad del examinador a cargo del caso.
- Listado del equipo y herramientas utilizadas en el caso.
- Breve listado y descripción de los elementos utilizados en las búsquedas, como cadenas de texto, elementos en gráficos y recuperación de archivos eliminados.
- Documentación de la cadena de custodia y copias impresas de evidencia de apoyo.
- Detalles de los elementos encontrados:
    - Archivos específicos relacionados al caso.
    - Archivos eliminados que fueron recuperados.
    - Búsqueda de cadenas de texto, palabras clave.
    - Evidencia relacionada a comunicaciones cómo análisis de tráfico web, registros de mensajes, correos electrónicos, registro de llamadas, etc.
    - Análisis de archivos gráficos.
    - Identificadores de propiedad como datos personales encontrados y de identidades registradas en las aplicaciones.
    - Descripción de aplicaciones encontradas en el dispositivo.
    - Técnicas empleadas para ocultar datos como estenografía, criptografía, particiones escondidas y anomalías en los nombres o extensiones de los archivos encontrados.
- Conclusiones del reporte.

Es importante recordar que el proceso seguido, los resultados y las conclusiones deben ser reproducibles por un tercero e incluso por otra herramienta o conjunto de herramientas con las mismas funcionalidades; es por eso que los detalles presentados en el reporte son muy importantes; permiten evaluar el proceso en general con la finalidad de validar las conclusiones obtenidas y reproducirlas por alguien más.

Por las funciones propias del análisis forense digital es posible que existan situaciones en las que sea necesario obtener información específica en un tiempo muy corto, por ejemplo, en casos de secuestro, donde encontrar indicios del lugar en el que se localiza una víctima sea la principal importancia. Para dichos casos convendría diseñar y evaluar procedimientos específicos y de ser posible llevar a cabo un análisis forense "completo" de forma paralela que pudiera servir para apoyar un caso legal.

4.3 Herramientas y métodos actuales para la adquisición de datos en dispositivos móviles con sistema operativo Android.

Una vez descrito el estado del proceso forense digital en general y ya que este trabajo pretende contribuir al proceso de adquisición es importante entender cuáles son las herramientas disponibles al momento en que se escribe este trabajo, cuáles son sus formas de operar y los medios de los que se valen para extraer información de un dispositivo móvil; así como sus limitaciones y sus ventajas.

Esta sección toma gran parte de su material del libro "Android Forensics" de Andrew Hoog (Hoog, 2011). Que contiene un listado de las herramientas comerciales y públicas y los métodos que emplean.

Como ya se mencionó al inicio del capítulo 4, en la sección 4.1, existen principalmente 2 técnicas para ejecutar una adquisición de datos, la lógica y la física. Cada una tiene sus ventajas y desventajas propias.

Una adquisición lógica hace uso del sistema de archivos empleado en el medio de almacenamiento para obtener los datos que se encuentren almacenados en éste.

Una adquisición física realiza una lectura del medio de almacenamiento físico sin tener en cuenta el tipo de sistema de archivo empleado, por lo que es normal que una adquisición física pueda recuperar un número mayor de información que posiblemente un sistema de archivos haya marcado como eliminada.

En Android podríamos enumerar una tercera forma de adquisición que hace uso de una capa de abstracción superior al sistema de archivos; llamada SDK (Kit de desarrollo de software), es un conjunto de herramientas que provee Google para que cualquier programador pueda interactuar con un dispositivo Android y hacer uso de sus recursos. A través de peticiones a proveedores de contenido (como se explica en la sección 2.2.1.2) existe la posibilidad de acceder a la información que almacenan las aplicaciones, sin embargo, la cantidad y el tipo de información está limitada en primer lugar por lo que reconoce el sistema de archivos como datos y en segundo lugar por lo que comparte la aplicación en cuestión. Es importante tener este principio en consideración cuando evaluemos los diferentes métodos y herramientas existentes para la adquisición de datos.

Otro punto que debemos recordar es el expresado, en varias ocasiones en las guías, acerca de evitar modificar los datos y el estado del dispositivo que estamos analizando y debemos recordar que no siempre es posible cumplir con este propósito.

Imaginemos una situación en la que ha ocurrido un incidente. Un equipo de especialistas en adquisición de "imágenes forenses" llega al lugar y se encuentran con un servidor que se encuentra en operación y que no puede apagarse ya que hacerlo interrumpiría las operaciones de una empresa. El especialista debe entonces interactuar con el dispositivo encendido, establecer conexiones e iniciar procesos de lectura, acciones que por sí mismas significan modificaciones al estado del dispositivo. En los dispositivos móviles sucede algo muy similar, aun cuando apagar un dispositivo móvil no significa detener las operaciones de una empresa, esta simple acción cierra conexiones establecidas y elimina datos de la memoria volátil. De igual forma, encender un dispositivo que se encontraba apagado puede significar ejecutar tareas programadas de inicio, cambio en la última fecha y hora registradas en el dispositivo, activar controles de acceso, etc. Por lo que al trabajar con un dispositivo móvil la modificación al estado de éste es casi inevitable y la forma en que podemos compensarlo es a través del conocimiento de los procesos y las metodologías que empleamos, así como la adecuada documentación de los mismos.

4.3.1 Herramientas y técnicas de adquisición lógicas.

Las técnicas de adquisición lógicas por lo general son la primera opción a considerar durante un incidente, debido a que son relativamente fáciles de ejecutar y en la mayoría de los casos pueden proveer con suficiente información útil para un caso.

El único requisito para que funcionen estos métodos es que el dispositivo Android del que se va a extraer información tenga habilitada la función "USB Debugging". Esta función sirve para comunicar el dispositivo móvil con una computadora a través del "ADB" (Enlace Depurador de Android). ADB permite llevar a cabo varias tareas en el dispositivo:

- intercambiar datos entre el dispositivo móvil y un sistema de cómputo.
- Instalar aplicaciones sin la necesidad de contar con la autorización expresa del usuario.
- Leer datos de los registros del sistema.

Se presentan primero las herramientas disponibles gratuitamente y después las comerciales. (En el caso de la herramienta AFLogical, se encuentra disponible gratuitamente sólo para agencias gubernamentales y de procuración de justicia).

4.3.1.1 Herramientas gratuitas de adquisición lógica.

4.3.1.1.1 ADB

ADB es una herramienta disponible con el SDK de Android, básicamente es un aplicación que utiliza el modelo cliente/servidor para establecer una conexión entre un dispositivo Android y un sistema de cómputo. Requiere un cable de conexión de datos entre ambos dispositivos se utiliza, entre otras cosas, para obtener un intérprete de comandos en el dispositivo, extraer archivos almacenados en el dispositivo y reconocidos por el sistema de archivos en uso e instalar aplicaciones de forma remota en el dispositivo.

La cantidad y el tipo de datos que es capaz de extraer dependerán del usuario que ejecute la herramienta. Por configuración de fábrica, ADB se ejecuta con un usuario común por lo que no tendrá acceso a carpetas protegidas, sin embargo, se puede acceder a carpetas compartidas del sistema como "/proc" "/sys". La información que se puede obtener es historial de sitios web visitados así como información de procesos ejecutados y del sistema en general.

El siguiente es un extracto de lo que se muestra cuando se intenta ejecutar el comando *"ls"* después de obtener un intérprete de comandos con "*adb shell*" entre un dispositivo Android y un sistema de cómputo utilizando un usuario normal.

Primero para la carpeta /data donde las aplicaciones almacenan datos privados y donde se requieren permisos de administrador, la salida se muestra en la figura 5.

```
$ adb shell
shell@android:/ $ cd /data/data
shell@android:/data/data $ ls
opendir failed, Permission denied
255|shell@android:/data/data $
```

Fig. 5 Uso de ADB en carpeta */data/data* sin permisos de administrador.

Ahora, para la carpeta /proc que se encuentra disponible para cualquier usuario (Para reducir el espacio ocupado, la salida en la figura 6 se muestra truncada):

```
$ adb shell
shell@android:/ $ cd /proc
shell@android:/proc $ ls
1
1004
1007
1020
1023
1033
3478
3495
.
.
.
3498
3499
3540
3541
3542
375
tty
uid_stat
uptime
user_fault
wakelocks
zoneinfo
shell@android:/proc $
```

Fig. 6 Uso de ADB en carpeta */proc.* Sin permisos de administrador.

ADB también se puede ejecutar con permisos de administración. Ejecutar el comando *"adb pull"* con permisos de administración permite acceder a las carpetas protegidas por el sistema y obtener cualquier información que pueda reconocer el sistema de archivos, sin embargo, por defecto los dispositivos Android no permiten a los usuarios elegir libremente ejecutar sus procesos con permisos de administración.

Es a través de un proceso de obtención de permisos de administración que un usuario puede adquirir acceso privilegiado a su dispositivo, el proceso para adquirir dicho acceso se encuentra ampliamente documentado en foros de internet y a pesar de que no es muy complicado llevar a cabo dicho procedimiento la mayoría de dispositivos que se pueden encontrar en la escena de un incidente no cuentan con dichos permisos, más aun, un especialista forense no puede depender de encontrar un dispositivo con ciertas características para llevar a cabo una adquisición, sino que debe estar preparado para los diferentes escenarios que se le puedan presentar.

En la figura 7, se muestra el resultado truncado de ejecutar el comando *"ls"* con permisos de administración para la carpeta protegida "/data/data".

```
$ adb shell
shell@android:/ $ su
root@android:/ # cd /data/data
root@android:/data/data # ls
PAQUETE.culturaUNAM
android.androidVNC
cat.aat.fraseswidget
com.adobe.flashplayer
com.adobe.reader
com.android.MtpApplication
com.android.Preconfig
com.android.backupconfirm
com.android.bluetooth
com.android.browser
com.android.calendar
com.android.certinstaller
com.android.clipboardsaveservice
com.android.contacts
com.android.defcontainer
com.android.email
com.android.exchange
com.android.facelock
com.sec.android.app.shutdown
com.seven.Z7
.
.
.
com.tenable
com.twitter.android
com.uemedia.calvinandhobbes
com.ulmon.android.citymaps2go
com.unit
com.vlingo.client.samsung
com.whatsapp
com.whatsapp.wallpaper
com.wsomacp
com.wssnps
com.wssyncmldm
com.zinio.samsung.android
edu.luc.etl.android.ieee
eu.chainfire.supersu
jackpal.androidterm
lysesoft.andsmb
mx.capitandurango.metro
org.connectbot
org.leetzone.android.yatsewidgetfree
org.mozilla.firefox
org.pocketworkstation.pckeyboard
org.vudroidplus
se.maginteractive.rumble.free
st.conaculta.android.activity
stericson.busybox
websec.routerpwn
root@android:/data/data #
```

Fig. 7 Uso de ADB en carpeta */data/data.* Con permisos de Administrador

Como se puede notar, se muestran las carpetas de sistema para cada aplicación instalada en el dispositivo, se puede incluso ingresar a cada carpeta y extraer los datos almacenados por cada aplicación utilizando el comando *"adb pull"*.

A pesar de las limitantes que significan encontrar un dispositivo al que se le haya aplicado un procedimiento para obtener permisos de administración y que cuente con la opción "ADB debugging" habilitada, ADB sigue siendo una herramienta muy poderosa, gratuita, relativamente fácil de operar y bajo las circunstancias adecuadas puede ser la única necesaria para llevar a cabo una adquisición que contenga todos los datos necesarios para un caso.

4.3.1.1.2 Análisis de respaldos.

Es común que los propietarios de los dispositivos móviles quieran hacer respaldos de sus datos y de sus aplicaciones y los examinadores y analistas forenses pueden beneficiarse de ésto obteniendo información relevante al analizar respaldos almacenados dentro o fuera del dispositivo.

Existen aplicaciones en el mercado de Android que permiten generar y almacenar respaldos de las aplicaciones y sus datos en la memoria extraíble o en la memoria interna del dispositivo. En el libro "Android Forensics" se menciona la herramienta "My Backup Pro" de "RerWare". Aplicaciones de este tipo solicitan acceso a los datos almacenados en el dispositivo a través del archivo "AndroidManifest.xml", por lo que en el momento de instalación el propietario del dispositivo debe aceptar dichos permisos. Un examinador forense debe buscar en la memoria interna y en la extraíble por archivos de respaldo generados para su posterior estudio y análisis.

Existen también las aplicaciones de administración de dispositivos móviles para computadoras, en las que se puede respaldar la información de un dispositivo móvil y guardar dicho respaldo en un equipo de cómputo.

Las aplicaciones de los 3 tipos mencionados (las que hacen uso del API de Android para respaldar, las que almacenan su respaldo en el dispositivo móvil o en una memoria extraíble y las que se sincronizan con un equipo de cómputo externo) pueden obtener el siguiente tipo de información:

- Archivos de instalación de aplicaciones. (Sólo cuando se cuenta con permisos de administración en el dispositivo)
- Contactos.
- Registro de llamadas.
- Historial de navegación web y enlaces guardados.
- Mensajes cortos de texto y multimedia.
- Configuración del sistema.
- Alarmas, diccionarios, calendario.
- Listas de reproducción de audio y de video.

Las limitaciones a las que se enfrenta un examinador cuando quiere recuperar la información creada por un respaldo son: en primer lugar, que el dueño del dispositivo haya instalado dicha aplicación y haya creado por lo menos un respaldo y en segundo lugar poder localizar el o los archivos de respaldo generados.

En última instancia, un análisis digital forense puede también contemplar la instalación de una aplicación que genere respaldos para posteriormente extraerlos, examinarlos y analizarlos. Siempre teniendo en cuenta y registrando los cambios que dicha instalación implican con la finalidad de mantener la cadena de custodia.

4.3.1.1.3 AFLogical

La aplicación “AFLogical” desarrollada por “viaForensics” solicita permisos de lectura a través del archivo “AndroidManifest.xml” para hacer uso de los proveedores de contenido con la finalidad de leer los datos compartidos por el sistema (mencionados en la sección 2.3) y por otras aplicaciones sí las hubiese.

“AFLogical” sólo se encuentra disponible para agencias de gobierno y de procuración de justicia por lo que se debe descargar del sitio de “viaForensics” (con previa autorización) y se instala haciendo uso de la herramienta “ADB” (por lo que también requiere que la opción “USB debugger” se encuentre habilitada), almacena los datos a la tarjeta de memoria extraíble del dispositivo móvil en forma de archivo con formato XML y CSV.

4.3.1.2 Herramientas comerciales de adquisición lógica.

Las herramientas comerciales presentadas funcionan de una forma similar a la herramienta “AFLogical” presentada en el punto 4.3.1.1.3.

Son aplicaciones especialmente hechas para dispositivos Android, a través del archivo “AndroidManifest.xml” solicitan permisos para leer de los proveedores de contenido de las aplicaciones del sistema y de otras aplicaciones si existieran. Posteriormente, copian los datos encontrados y los almacenan en la memoria extraíble o en un agente que se ejecuta en un sistema de cómputo externo.

En comparación a su contraparte gratuita, estas aplicaciones agregan las funciones de estudio y creación de reportes, por lo que ayudan durante un proceso digital forense a reconocer los archivos encontrados, ordenarlos y generar un reporte de dichos procedimientos.

Las herramientas presentadas en el libro “Android Forensics” de Andrew Hoog son:

- Cellebrite UFED.
- Compelson MOBILedit!
- EnCase Neutrino.
- Micro Systemation XRY.
- Paraben Device Seizure.
- ViaForensics viaExtract.

En general todas las herramientas mostradas pueden extraer la siguiente información:

- Búsquedas realizadas y páginas almacenadas con el explorador web.
- Registro de llamadas.
- Números de contacto almacenados.
- Imágenes y video almacenados.
- Cuentas de mensajería instantánea
- Contactos y mensajes de mensajería instantánea.
- Mensajes multimedia.

Aunque en el libro se detalla el proceso de instalación, adquisición y reporte para cada herramienta, no es propósito de este trabajo ahondar en dichos detalles sino en los métodos de los que se valen dichas herramientas para llevar a cabo la adquisición de los datos.

4.3.2 Herramientas y técnicas de adquisición física.

Es importante recordar que las técnicas de adquisición física pretenden hacer una copia bit a bit del contenido de un medio de almacenamiento, sin importar el sistema de archivos que éste utilice, por lo que la cantidad de datos que pueden extraer es mayor que los obtenidos utilizando una técnica de adquisición lógica, recuperando datos previamente eliminados así como datos almacenados en el medio pero no mostrados por el sistema operativo o por las aplicaciones.

Para dispositivos móviles las técnicas de adquisición lógica se pueden dividir en 2 grupos:

- Por hardware, requieren de equipo y herramientas físicas especializadas pero pueden ser de mucha utilidad para evitar enfrentarse a sistemas de autenticación (dispositivos bloqueados con contraseña, patrones o código PIN) y no requieren que el dispositivo haya sido previamente modificado para obtener permisos de administración
- Por software, son herramientas lógicas que requieren permisos de administración en el dispositivo en el que se pretende extraer información. En comparación con las técnicas basadas en hardware son más fáciles de ejecutar y suponen menor riesgo de inhabilitar o descomponer un dispositivo.

4.3.2.1 Herramientas y técnicas de adquisición física por Hardware.

4.3.2.1.1 JTAG

JTAG (por sus siglas en inglés: Joint Test-Action Group), es un grupo de trabajo creado en 1980 por representantes, fabricantes y usuarios de componentes electrónicos preocupados por los crecientes retos para ejecutar pruebas que ayuden a confirmar la calidad de los circuitos impresos y sus interconexiones.

El grupo JTAG propuso entonces estructuras y procesos de diseño que permitían agregar capacidades de prueba en cada fase del desarrollo incluyendo el diseño, implementación de hardware y de software y manufactura.

En 1990 la IEEE adoptó la propuesta como el estándar 1149.1, su propósito es: Evaluar interconexiones entre circuitos integrados instalados en módulos, circuitos y otros. Evaluar los circuitos integrados y observar o modificar la actividad de los circuitos integrados durante la operación de los mismos (IEEE, 2012).

Actualmente los dispositivos que cumplen con este estándar cuentan con puertos llamados TAP (en inglés, Test Access Port) que permiten enviar y recibir señales de prueba, entre estas señales se encuentran:

- Pruebas de entrada y salida de datos
- Señal de entrada y salida de reloj
- Señal de entrada de reinicio

Estas señales sirven para realizar pruebas en el circuito impreso completo y en los dispositivos soldados en éste, por lo que se pueden realizar pruebas en el procesador o en el chip de memoria realizando un volcado de datos.

Para llevar a cabo una prueba en un circuito impreso o en un chip es necesario reconocer las terminales que el fabricante ha establecido como TAP (lo que implica desensamblar el dispositivo para examinar el circuito impreso), soldar las terminales a un dispositivo externo en el que se almacenarán los datos y conocer y aplicar el voltaje necesario para iniciar el procedimiento de prueba.

En conclusión, no es un procedimiento sencillo de realizar, implica el conocimiento de electrónica básica y de los detalles de los diseños de los circuitos para cada fabricante, su mala ejecución puede resultar en un dispositivo inservible.

4.3.2.1.2 Análisis del chip de memoria.

Esta técnica se basa en la capacidad de extraer el chip de memoria del dispositivo móvil y posteriormente analizarlo. Requiere equipo especializado y personal altamente calificado.

El procedimiento consiste en:

- Desensamblar el dispositivo móvil, de tal forma que el circuito impreso en el que se encuentran soldados los chips quede al descubierto.
- Reconocer el chip de memoria.
- Remover el chip de memoria de la placa de circuito impreso, se puede hacer desoldando el chip, aplicando aire caliente a las terminales o calentando el chip para que la soldadura se derrita. Este paso, mal ejecutado, puede hacer que el chip quede inutilizado.
- Emplear un dispositivo que permita "leer" los datos de un chip de memoria, extraerlos y almacenarlos en un medio externo.
- Incluso se puede intentar volver a soldar el chip de memoria a la placa de circuito impreso del dispositivo.

En conclusión, es un procedimiento con alto de riesgo de inutilizar el chip de memoria, requiere equipo y conocimientos muy especializados para llevarlo a cabo.

4.3.2.2 Herramientas y técnicas de adquisición física por Software.

Las técnicas de adquisición física por software tienen múltiples ventajas sobre las técnicas de adquisición por hardware, la más evidente es que son mucho más fáciles de llevar a cabo y no suponen el mismo riesgo de deshabilitar un dispositivo y dejarlo inutilizable. Sin embargo, su principal desventaja es que requieren que el dispositivo a analizar cuente con permisos de administrador, lo que, como ya hemos explicado antes, no se da en la configuración de fábrica y el usuario interesado debe llevar a cabo un procedimiento especial, no creado por el fabricante, para obtener dichos permisos.

Además de contar con permisos de administrador en el dispositivo, estas técnicas también requieren que el modo "USB Debug" se encuentre habilitado. Aunque las condiciones no son tan sencillas de cumplir, en un dispositivo que no se encuentra bloqueado por un código de seguridad (contraseña, PIN, reconocimiento de rostro) se puede fácilmente habilitar la función "USB Debug" y también se puede, aunque no tan fácilmente, obtener permisos de administrador al dispositivo, siempre conociendo y documentando los procedimientos llevados a cabo así como los cambios que éstos suponen para el dispositivo.

Cuando se cuenta con un dispositivo bloqueado por un código de seguridad (contraseña, PIN o reconocimiento de rostro) un examinador debe pensar en técnicas distintas a éstas para la extracción de los datos.

Siempre se debe verificar y confirmar que no se cuenta con la función “ADB Debug” habilitada y que no se cuenta con permisos de administrador al dispositivo antes de intentar otras técnicas como las de adquisición física por hardware. El procedimiento para hacerlo es el siguiente:

Para verificar si se cuenta con la función “ADB Debug” habilitada en el dispositivo:

Cuando se tiene habilitada la función “ADB Debug” la respuesta al comando “*adb devices*” muestra un número de dispositivo conectado y permite conectarse a su intérprete de comandos, como se muestra en la figura 8.

```
$ adb devices
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
List of devices attached
0019a0b46dbb6e  device
```

Fig. 8 Prueba exitosa de un dispositivo con ADB habilitado.

Habiendo confirmado “ADB Debug” habilitado, para verificar que se cuenta con permisos de administrador al dispositivo se intenta lo mostrado en la figura 9.

```
$ adb shell
shell@android:/ $ su
shell@android:/ #
```

Fig. 9 Prueba exitosa de permisos de administrador empleando ADB.

En caso de que se cuente con permisos de administrador, el dispositivo muestra el símbolo “#” en lugar de “$” en el intérprete de comandos.

En la figura 10 se muestra la prueba en caso de que no se cuente con permisos de administrador.

```
$ adb shell
shell@android:/ $ su
su: permission denied
```

Fig. 10 Prueba no exitosa de permisos de administrador empleando ADB.

En la figura 11 se muestra el comportamiento cuando no se tiene habilitada la función “ADB Debug” la respuesta al comando *”adb devices”* no muestra ningún número de dispositivo conectado y por lo tanto tampoco se puede verificar que existan permisos de administrador en el dispositivo.

```
$ adb devices
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
List of devices attached
$
```

Fig. 11 Prueba no exitosa de un dispositivo con ADB deshabilitado.

4.3.2.2.1 Modo de Recuperación.

El modo de recuperación de Android es un menú especial de mantenimiento, almacenado en una partición especial llamada “partición de recuperación”, se llega a él reiniciando el dispositivo y aplicando una combinación especial de teclas (que dependen del fabricante y el modelo del dispositivo).

En la tabla 7 se muestran las particiones que por configuración de fábrica se encuentran presentes en un dispositivo Android son (Timothy Vidas):

Ruta	Nombre	Sistema de Archivos	Ruta del sistema de archivos	Descripción
/dev/mtd/mtd0	pds	yaffs2 ó EXT4	/config	Datos de configuración
/dev/mtd/mtd1	misc	-	N/A	Datos de las particiones
/dev/mtd/mtd2	boot	bootimg	N/A	Partición de inicio
/dev/mtd/mtd3	recovery	bootimg	N/A	Partición de recuperación
/dev/mtd/mtd4	system	yaffs2 ó EXT4	/system	Sistemas de archivo, aplicaciones (solo lectura)
/dev/mtd/mtd5	cache	yaffs2 ó EXT4	/cache	Datos de cache del sistema
/dev/mtd/mtd6	userdata	yaffs2 ó EXT4	/data	Datos del usuario (aplicaciones)
/dev/mtd/mtd7	kpanic	-	N/A	Bitácoras de errores y fallas.

Tabla 7 Particiones predeterminadas en los dispositivos con sistema Android.

El modo de recuperación permite llevar a cabo tareas como aplicar actualizaciones al sistema, eliminar los datos almacenados por el dispositivo y regresarlo a su configuración de fábrica. Por defecto, las acciones que se pueden llevar a cabo en este menú de recuperación son muy limitadas, no permite, entre otras cosas, tener acceso con permisos de administrador.

Para acceder al modo de recuperación es necesario reiniciar el dispositivo (lo que provoca que se pierdan datos volátiles) y presionar una combinación de teclas mientras se está encendiendo el dispositivo. Esta combinación cambia para cada modelo y fabricante del dispositivo, sin embargo, la combinación es un dato público generalmente dado a conocer por los fabricantes y fácilmente localizable en foros especializados. En el anexo D se presenta una tabla con combinaciones para acceder al modo de recuperación de diferentes dispositivos.

Además de las combinaciones de teclas durante el encendido del dispositivo también se puede utilizar la herramienta ADB para iniciar un dispositivo en el modo de recuperación, como se muestra en la figura 12, primero se establece una conexión por medio de “ADB” y se envía el comando “reboot recovery”.

```
$ adb shell
shell@android:/ $ reboot recovery
```

Fig. 12 Reinicio de un dispositivo a modo de recuperación empleando ADB.

El dispositivo se reiniciara en modo de recuperación.

Para extraer información desde el modo de recuperación se verifica acceso a la herramienta "ADB" y si éste existe se verifica los permisos de administrador. Por configuración de fábrica, "ADB" y el acceso como administrador se encuentran deshabilitados, es a través de un proceso de adquisición de permisos de administrador que se puede modificar esto.

Este trabajo presenta en los capítulos siguientes una propuesta práctica de como personalizar la partición de recuperación en un sistema Android, de tal forma que se pueda contar con herramientas como "ADB" y permisos de administrador desde el modo de recuperación.

4.3.2.2.2 Técnica "AFPhysical" de viaForensics.

La técnica "AFPhysical" es un conjunto de procedimientos dados a conocer por la compañía viaForensics, hace uso de varias utilerías de extracción y copiado de datos adaptadas para ejecutarse en un dispositivo móvil.

Permite realizar adquisiciones físicas y lógicas, enviando las copias directamente a una memoria extraíble o a un dispositivo de cómputo externo.

Requiere que la utilería "ADB" se encuentre habilitada y que el examinador pueda tener acceso al dispositivo con permisos de administrador.

Una vez que se cuenta con permisos de administrador el procedimiento se puede resumir como:

1. Reconocer las particiones presentes en el dispositivo y cuales son de interés para copiar.
2. Enviar los archivos binarios ejecutables de las herramientas de copiado al dispositivo.
3. Realizar la copia de las particiones de interés utilizando los binarios previamente guardados en el dispositivo.
4. Extraer la copia del dispositivo y eliminar los binarios almacenados en éste.

Iniciando el procedimiento tenemos que reconocer las particiones presentes en el dispositivo (el nombre de la partición existente y de la ruta de la misma). Estos datos varían entre fabricantes y modelos, por lo que los datos deben ser obtenidos en cada dispositivo.

Android, como ya hemos mencionado está basado en Linux, por lo que podemos aprovechar algunos de los comandos existentes en Linux para obtener información del dispositivo. En la figura 13 se muestran algunos comandos propios de Linux ejecutados en un Samsung GT-I9100 usando permisos de administrador.

```
$ adb devices
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
List of devices attached
0019a0b46dbb6e  device

$ adb shell
shell@android:/ $ su

1|shell@android:/ # mount
rootfs / rootfs ro,relatime 0 0
tmpfs /dev tmpfs rw,nosuid,relatime,mode=755 0 0
devpts /dev/pts devpts rw,relatime,mode=600 0 0
proc /proc proc rw,relatime 0 0
sysfs /sys sysfs rw,relatime 0 0
none /acct cgroup rw,relatime,cpuacct 0 0
tmpfs /mnt/asec tmpfs rw,relatime,mode=755,gid=1000 0 0
tmpfs /mnt/obb tmpfs rw,relatime,mode=755,gid=1000 0 0
none /dev/cpuctl cgroup rw,relatime,cpu 0 0
/dev/block/mmcblk0p9 /system ext4 ro,noatime,barrier=1,data=ordered 0 0
/dev/block/mmcblk0p7 /cache ext4 rw,nosuid,nodev,noatime,barrier=1,data=ordered 0
0
/dev/block/mmcblk0p1 /efs ext4 rw,nosuid,nodev,noatime,barrier=1,data=ordered 0 0
/dev/block/mmcblk0p10 /data ext4
rw,nosuid,nodev,noatime,barrier=1,data=ordered,noauto_da_alloc 0 0
/dev/block/mmcblk0p4 /mnt/.lfs j4fs rw,relatime 0 0
/sys/kernel/debug /sys/kernel/debug debugfs rw,relatime 0 0
/dev/block/vold/259:3 /mnt/sdcard vfat
/dev/block/dm-0           /mnt/asec/com.ulmon.android.citymaps2go-1           vfat
ro,dirsync,nosuid,nodev,noatime,nodiratime,uid=1000,fmask=0222,dmask=0222,codepag
e=cp437,iocharset=iso8859-1,shortname=mixed,utf8,errors=remount-ro 0 0
```

Fig. 13 Comandos *su, mount*. En un Samsung GT-I9100 con permisos de administrador.

En el bloque de comandos de la figura 13 se crea una sesión hacia el intérprete de comandos del dispositivo Android utilizando el comando "*adb shell",* se emplean permisos de administrador con el comando *"su"* y utilizando el comando *"mount"* se muestra la lista de particiones activas en el sistema, de la salida del comando *"mount"* reconocemos algunas particiones que podrían ser de interés para un examinador (aunque quizá un procedimiento más completo implica copiar todas las particiones existentes). Las particiones reconocidas para este ejercicio se muestran en la figura 14.

```
/dev/block/mmcblk0p9 /system ext4
/dev/block/mmcblk0p7 /cache ext4
/dev/block/mmcblk0p10 /data ext4
```

Fig. 14 Algunas particiones reconocidas de la figura 13.

Donde se almacenan los datos del sistema, el cache y los datos del usuario respectivamente. Para este ejercicio realizaremos una copia de los datos almacenados por el usuario.

Generalmente, más que reconocer la partición más interesante se hace una copia de todas las particiones presentes en el dispositivo.

Recordemos también que las particiones pueden contar con diferentes sistemas de archivos que posteriormente deben ser interpretados (si se llevara a cabo una adquisición física), el comando *"mount"* también muestra el sistema de archivos utilizado por la partición, en este caso ext4.

El siguiente paso es almacenar los binarios que utilizaremos para realizar la copia de la partición seleccionada. Las herramientas a utilizar dependerán de si se desea obtener una adquisición física o lógica.

Para obtener una adquisición física existen por lo menos 2 herramientas muy útiles:

- nanddump, una herramienta especialmente diseñada para llevar a cabo un volcado de los datos almacenados en un dispositivo tipo NAND (comúnmente encontrado en los dispositivos móviles). Puede realizar copias bit a bit de un dispositivo de almacenamiento completo o sólo de particiones.
- dd, realiza copias por bloques de datos de archivos, dispositivos o particiones. Por lo que sirve también para adquisiciones físicas y no está limitado a dispositivos tipo NAND.

Para obtener una adquisición lógica existen también por lo menos 2 herramientas:

- ADB, mencionada en la sección 4.3.1.1.1. Cuenta con la ventaja de que ya ésta presente en el dispositivo Android, por lo que no hace falta almacenar una copia.
- tar, una herramienta de empaquetado y compresión de archivos. Puede empaquetar varios archivos del sistema en uno solo y usada en conjunto con otras herramientas puede comprimir el archivo final.

Para poder almacenar las utilerías en el dispositivo móvil utilizamos "ADB" por medio de su comando *"adb push"* que nos sirve para transferir desde un sistema de cómputo hacia un dispositivo móvil. Su uso se muestra en la figura 15.

```
$ adb shell
shell@android:/ # cd /dev/
255|shell@android:/dev $ su
shell@android:/dev # mkdir MIO
shell@android:/dev # exit
shell@android:/dev $ exit

[angelus@RTF5261 Android-Tools]$ adb push dc3dd /dev/MIO/.
2531 KB/s (1021168 bytes in 0.393s)
```

Fig. 15 Uso de ADB Push para almacenar utilerías en un dispositivo Android.

En el conjunto de comandos anteriores creamos primero una carpeta llamada "MIO" en la ruta "/dev" donde almacenaremos el archivo ejecutable *"dc3dd"* que nos servirá para llevar a cabo una adquisición física. Desde nuestra computadora, entonces enviamos el archivo ejecutable *"dc3dd"* hacia el dispositivo Android utilizando el comando *"adb push dc3dd /dev/MIO"*

En la figura 16, se muestra una forma para aseguramos de contar con permisos de ejecución en el archivo enviado.

```
shell@android:/dev/MIO # ll
-rwxrwxrwx shell    shell      695473 2013-03-31 13:56 dc3dd
```

Fig. 16 Revisión de permisos sobre el archivo *dc3dd* en Android.

El procedimiento para realizar la copia dependerá entonces de la herramienta que se elija.

Ya hemos mencionado la operación de "ADB", por lo que ahora lo haremos utilizando "dd". Particularmente "dc3dd", una versión modificada por el Departamento de Defensa de los E.U.A. con funciones agregadas. (Justice, 2011)

En la figura 17 se muestra cómo se lleva a cabo la copia de la partición */dev/block/mmcblk0p10* ejecutando "dc3dd" y especificando la salida a un archivo en la memoria extraíble. La sintaxis del comando es: *#dc3dd if=Dispositivo_a_Copiar of=Archivo_de_Salida*

```
shell@android:/dev/MIO # ./dc3dd if=/dev/block/mmcblk0p10
of=/sdcard/mmcblk0p10_bkp.dd

dc3dd if=/dev/block/mmcblk0p1                                          <
mmcblk0p1  mmcblk0p10 mmcblk0p11 mmcblk0p12
k0p10 of=/sdcard/mmcblk0p10_bkp.dd                                     <

dc3dd 7.1.614 started at 2013-04-01 00:59:45 +0000
compiled options:
command line: ./dc3dd if=/dev/block/mmcblk0p10 of=/sdcard/mmcblk0p10_bkp.dd
device size: 4194304 sectors (probed)
sector size: 512 bytes (probed)
2147483648 bytes (2 G) copied (100%), 310.247 s, 6.6 M/s

input results for device `/dev/block/mmcblk0p10':
   4194304 sectors in
   0 bad sectors replaced by zeros

output results for file `/sdcard/mmcblk0p10_bkp.dd':
   4194304 sectors out

dc3dd completed at 2013-04-01 01:04:56 +0000

shell@android:/dev/MIO #
```

Fig. 17 Copia de una partición de datos en Android empleando la utilería dc3dd.

El comando que se muestra en la tabla 17 crea una copia de la partición "*/dev/block/mmcblk0p10"* que contiene los datos creados por el usuario y los almacena en el archivo **"***/sdcard/mmcblk0p10_bkp.dd"* de la tarjeta de memoria extraíble. Se muestra también que la copia tarda 310.247 segundos, alrededor de 5 minutos en completarse.

Finalmente podemos extraer la memoria externa y con ella la copia o utilizar "ADB" a través de su comando "*adb pull"* para transferir la copia.

Este método es bastante ingenioso y hace uso de varias utilerías ya existentes, algunas diseñadas específicamente para Android como “ADB” y otras no como “nanddump” o “dc3dd”.

Sin embargo, para poder aplicar estas técnicas se requiere que el dispositivo a copiar tenga “USB Debugging” habilitado y permisos de administrador, lo cual no siempre sucede.

A pesar de sus limitaciones, es un procedimiento muy importante ya que puede servir como complemento a otro que logre obtener permisos de administración y habilitar “USB debugging”.

Capítulo 5. Problema actual de la adquisición de datos no volátiles en dispositivos con S.O. de código abierto.

Este capítulo pretende llevar a cabo una comparación de las técnicas y herramientas presentadas en el capítulo 4 con la finalidad de mostrar de forma clara y concisa sus ventajas, desventajas y requerimientos.

El objetivo será mostrar el área de oportunidad que existe en la investigación y desarrollo de herramientas y técnicas de adquisición de datos no volátiles en dispositivos móviles con S.O. de código abierto.

Este capítulo también pretende mostrar la importancia de continuar con dicha investigación y desarrollo, las áreas en las que éstas técnicas han ayudado y otras áreas donde también pueden ser de utilidad.

5.1 Descripción del problema actual de adquisición de datos no volátiles en dispositivos móviles de S.O. de código abierto.

Comenzamos mostrando una comparación de las ventajas y desventajas de los procedimientos actuales de adquisición y a partir de ésta derivaremos en una descripción precisa del problema que más adelante pretendemos resolver.

En la tabla 8 se muestra una comparación de propiedades de los métodos de adquisición lógica.

	Es necesario desarmar el dispositivo para ejecutar el procedimiento	Una incorrecta ejecución podría inutilizar el dispositivo	Requiere interacción del usuario para instalarse o ejecutarse	Requiere permisos de administrador para extraer la mayor parte de los datos	Requiere que el dispositivo no se encuentre bloqueado por un código o contraseña
ADB	NO	NO	SÍ	SI	NO
Análisis de respaldos	NO	NO	SÍ	NO	SÍ
AFLogical (viaForensics)	NO	NO	SÍ	NO	SÍ

Tabla 8 Comparación de propiedades de los métodos de adquisición lógica.

En la tabla 9 se muestra una comparación de propiedades de los métodos de adquisición física.

	Es necesario desarmar el dispositivo para ejecutar el procedimiento	Una incorrecta ejecución podría inutilizar el dispositivo	Requiere interacción del usuario para instalarse o ejecutarse	Requiere permisos de administrador para extraer la mayor parte de los datos	Requiere que el dispositivo no se encuentre bloqueado por un código o contraseña
JTAG	SÍ	SÍ	NO	NO	NO
Análisis del chip de memoria	SÍ	SÍ	NO	NO	NO
Modo de recuperación	NO	NO	NO	SI	NO
Técnica AFPhysical	NO	NO	NO	SI	NO

Tabla 9 Comparación de propiedades de los métodos de adquisición física.

Las últimas 3 columnas de la tabla se refieren a requerimientos necesarios para llevar a cabo un procedimiento y extraer la mayor cantidad de datos posible. Lo ideal sería contar con un procedimiento que requiera la menor cantidad de condiciones (esto es "NO" en las últimas 3 columnas).

Se observa entonces que los procedimientos JTAG y Análisis del chip de memoria son los procedimientos que menos requerimientos necesitan para extraer la mayor cantidad de información.

Las primeras 2 columnas se refieren a la necesidad de desensamblar el dispositivo para llevar a cabo la adquisición y si un procedimiento mal ejecutado podría inutilizar el dispositivo. Lo ideal sería poder contar con un procedimiento que no requiera desensamblar ni que pudiera inutilizar un dispositivo si se ejecuta de forma incorrecta. Observamos entonces que nuestros primeros procedimientos candidatos "JTAG" y "Análisis de chip de memoria" tienen 2 grandes desventajas.

Lo deseable sería entonces contar con un procedimiento que presente "NO" en las 5 columnas de la tabla. Los procedimientos que más se acercan a eso son "Modo de recuperación" y la "Técnica AFPhysical", desafortunadamente, ambos procedimientos requieren permisos de administrador en el dispositivo para extraer la mayor cantidad de datos posible.

El problema que se nos plantea es diseñar un procedimiento de adquisición en el que aún en el caso de no contar con permisos de administrador ni la interacción del propietario del dispositivo y aunque el dispositivo se encuentre bloqueado por un código o contraseña seamos capaces de obtener una copia completa de los datos no volátiles almacenados y que además no requiera desensamblar ni signifique un riesgo alto a la funcionalidad de dicho dispositivo.

En la tabla 10 se muestran las propiedades que debe cumplir el método que propone este trabajo.

	Es necesario desarmar el dispositivo para ejecutar el procedimiento	Una incorrecta ejecución podría inutilizar el dispositivo	Requiere aprobación del usuario para instalarse o ejecutarse	Requiere permisos de administrador para extraer la mayor parte de los datos	Requiere que el dispositivo no se encuentre bloqueado por un código o contraseña
Propuesta	NO	NO	NO	NO (el procedimiento obtiene permisos de administrador)	NO

Tabla 10 **Características del método de adquisición propuesto en este trabajo.**

5.2 Justificación del estudio del problema de adquisición de datos no volátiles en dispositivos móviles en México.

En esta sección nos enfocaremos a justificar la importancia de estudiar la resolución del problema enunciado en la sección 5.1.

Los temas que se plantearan en ésta sección son:

1. Agencias gubernamentales y privadas en México que se pueden beneficiar de esta investigación.
2. Aplicaciones actuales y posibles aplicaciones futuras para procedimientos de adquisición, extracción, análisis y presentación de resultados digitales forenses.

5.2.1 Organizaciones públicas y privadas en México que pueden beneficiar de este trabajo.

En México se ha reconocido la importancia que tiene el análisis forense digital en la prevención del delito y en la procuración de justicia, es por eso que existen agencias que dedican personal exclusivamente a esta tarea, algunas de estas agencias gubernamentales son:

- Procuraduría General de la Republica (PGR), a través de la Coordinación de servicios periciales cuenta con la especialidad de Informática y Telecomunicaciones. Asigna 34 peritos en el interior del país y 26 en el Distrito Federal. Cuentan con laboratorios especializados y hacen uso de herramientas como las descritas en la sección 4.3. (Procuraduria General de la República)

- La Secretaría de Seguridad Pública del Distrito Federal, a través de la Policía de Ciber-delincuencia preventiva. Integrada por 30 elementos enfocados a prevenir delitos como pornografía infantil, trata de personas, fraude, extorsión y "cyber-bulling". (Policia Ciberdelincuencia)
- El CERT de la UNAM, del departamento de seguridad en cómputo de la UNAM se dedica a llevar a cabo investigaciones forenses dentro y fuera de la UNAM para entidades de gobierno, privadas y financieras. (UNAM CERT, 2005)

Las empresas privadas de consultoría también se han dado cuenta de las áreas de oportunidad que presenta el análisis digital forense y ofrecen servicios relacionados con éste. Es importante notar que de las empresas listadas a continuación sólo una de ellas es mexicana y el resto son consultoras trasnacionales:

- Mattica, empresa mexicana especializada en investigaciones digitales. Fundadores del primer laboratorio de investigaciones digitales en américa latina. Han aumentado gradualmente su presencia en Latinoamérica, especialmente Colombia. (Mattica)
- PricewaterhouseCoopers, una de las 4 consultoras financieras más importantes, ofrece servicios de investigación de fraudes y disputas comerciales. (PwC)
- Ernst & Young, otra de las grandes 4 consultoras, ofrece servicio de investigación de fraudes y asistencia en litigios, específicamente servicios de tecnología y descubrimiento forense. (Ernst & Young)
- KPMG, otra de las 4 grandes consultoras, ofrece servicios de prevención, investigación y remediación de fraudes a través de cómputo forense y de análisis de datos. (KPMG)
- Deloitte, otra de las 4 grandes consultoras, ofrece diferentes servicios relacionados al cómputo forense. (Deloitte)

Las empresas públicas y privadas listadas basan muchos de sus procesos en herramientas generadas por terceros, listadas en la sección 4.3.

5.2.2 Aplicaciones actuales y posibles aplicaciones futuras.

A través de éste trabajo hemos presentado algunas aplicaciones que se benefician del proceso forense digital. En ésta sección listaremos dichas aplicaciones y propondremos otras que también podrían beneficiarse de éste.

Las aplicaciones que actualmente se benefician del proceso digital forense son:

- Apoyo a procesos legales. Recolectando, reconociendo y presentando evidencia presente en sistemas y medios digitales de almacenamiento.
- Apoyo en casos de fraude. Identificando indicios que dentro de un contexto especifico pudieran significar uso indebido de los bienes de una empresa.
- Apoyo en búsqueda de malware. Identificando, asilando y extrayendo archivos maliciosos para su posterior análisis.
- Apoyo en identificación de intrusos. Reconstruyendo eventos suscitados en sistemas de cómputo que pudieran significar accesos no permitidos o violaciones a las políticas de uso de dichos sistemas.

Propuestas de aplicaciones para el proceso digital forense:

- Obtención de información en caso de siniestros automovilísticos. Las aseguradoras y las agencias de transito podrían obtener información acerca del uso de dispositivos móviles durante los siniestros. Se estima que en el D.F. 11% de los accidentes automovilísticos son producidos por utilizar algún dispositivo móvil (Animal Politico, 2012). Aplicaciones como las presentadas en este trabajo podrían ayudar a identificar y entender de mejor manera la problemática a la que nos enfrentamos.
- Evaluaciones de uso y aprovechamiento de recursos. En las organizaciones privadas y públicas se asignan recursos de cómputo móvil y fijo con la finalidad de aumentar la productividad. Una herramienta como las presentadas en este trabajo que pudiera extraer el tipo y la cantidad de datos almacenados en un dispositivo para posteriormente evaluar su uso y aprovechamiento serviría para definir mejores políticas y aumentar la productividad dentro de las organizaciones.
- Detección de fugas de información y espionaje corporativo. Políticas de revisión continua de la información almacenada en dispositivos móviles complementadas con controles de acceso podrían ayudar a prevenir, identificar y disminuir fugas de información en organizaciones.

Finalmente somos los usuarios finales quienes siempre debemos estar conscientes de la importancia que tienen nuestros datos personales y ser selectivos con la información que compartimos y los medios que utilizamos para hacerlo. Es posible que aplicaciones como la propuesta en este trabajo puedan ser utilizadas de forma directa o indirecta para llevar a cabo suplantación de identidad, espionaje, extorsiones, chantajes, secuestros, etc.

Capítulo 6. Resolución del problema presentado.

La solución al problema planteado tendrá como resultado un procedimiento que permita extraer la mayor cantidad de datos no volátiles de un dispositivo móvil sin la necesidad de que el dispositivo al que se le aplique dicho procedimiento cumpla con las siguientes condiciones:

- Contar con permisos de administrador.
- Que el dispositivo no se encuentre bloqueado con una contraseña, pin o patrón de acceso.
- Que se pueda emplear aún sin la interacción del dueño de dicho dispositivo.
- Que pueda aplicarse a una gran variedad de dispositivos móviles con S.O. de código abierto, independientemente del fabricante y de la versión del S.O.
- Que su aplicación sea relativamente sencilla y una vez que se haya validado el procedimiento éste no implique un riesgo mayor de descomponer o inutilizar el dispositivo.

Es de notar que en el Capítulo 4 se describen procedimientos que cumplen con algunas de estas características pero no con todas, por ejemplo:

El modo de recuperación y la técnica “AFPhysical”,

- Permiten extraer la mayor cantidad de datos no volátiles.
- No están limitados por una contraseña, pin o patrón presente en el dispositivo móvil.
- No implican un alto riesgo de inutilizar el dispositivo
- Pero requieren acceso con permisos de administrador para extraer los datos almacenados por él usuario. Lo cual no se encuentra presente por configuración de fábrica en los dispositivos móviles.

Si pudiéramos generar un procedimiento para obtener permisos de administrador en cualquier dispositivo con S.O. de código abierto podríamos emplear el modo de recuperación o la técnica AFPhysical o una combinación de ambas para extraer los datos no volátiles.

Una analogía para entender esta propuesta serían nuestras computadoras de escritorio con sistema operativo Windows. Cuando olvidamos nuestra contraseña o nuestro sistema operativo se daña de tal forma que no se puede recuperar, una práctica muy común para acceder a los datos almacenados en la computadora es emplear un Disco Compacto que se conoce como “Live CD” (Martinez, 1998) generalmente con un sistema operativo Linux que se almacena en la memoria RAM de la computadora. Estas distribuciones nos permiten tener un sistema operativo con condiciones controladas y conocidas ejecutándose en nuestra computadora y haciendo uso de sus recursos, de tal forma que si contamos con las utilerías necesarias podemos acceder a la información almacenada en el disco duro.

De igual forma, la propuesta que se presenta en este trabajo pretende aprovechar el hecho de que Android es un S.O. de código abierto y se encuentra disponible públicamente para copiar y modificar, de tal forma que podemos generar un S.O. Android con condiciones controladas por nosotros y aplicarlo a un dispositivo móvil.

Para aplicar la nueva distribución de Android creada por nosotros es necesario sobre-escribir los datos almacenados en la partición de sistema (/system) asegurándonos de no modificar la partición de datos del usuario. En la partición del sistema se almacenan los archivos necesarios para que Android funcione por lo que al modificar sus archivos modificamos los parámetros de funcionamiento. No estamos modificando los datos del usuario.

En los dispositivos móviles los tamaños y características de sus pantallas son diferentes, al igual que la distribución de sus teclados, la cantidad y el tipo de sensores, el formato de compresión y empaquetamiento de los archivos del sistema, el sistema de archivos, etc. Por lo que el análogo del "Live CD" para un dispositivo Android que en éste trabajo llamaremos "imagen de recuperación" tiene que ser creada específicamente para un fabricante y un modelo de dispositivo móvil. Es probable que bajo ciertas circunstancias un misma "imagen de recuperación" funcione para dos modelos de dispositivos Android diferentes, pero esto más que una regla es una excepción.

6.1 Modelo de adquisición propuesto.

El modelo propuesto que detalla los pasos desde la identificación del dispositivo hasta la extracción de los datos se muestra en la figura 18.

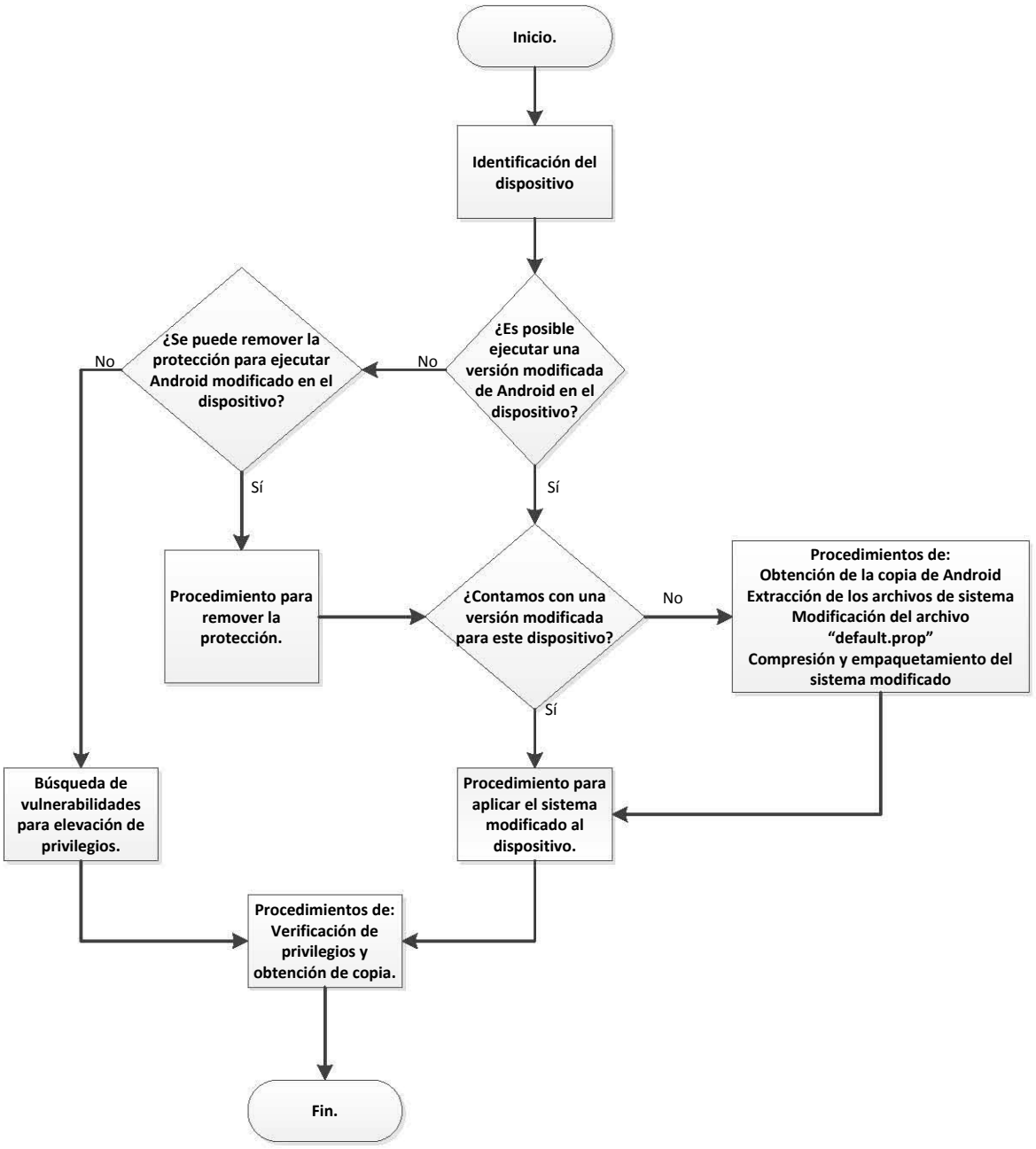


Fig. 18 Propuesta de modelo para adquisición digital forense en dispositivos móviles.

A continuación se detallan los procedimientos listados en el modelo.

6.1.1 Identificación del dispositivo y del S.O. que está ejecutando.

Este procedimiento inicia con la identificación del dispositivo del que se pretenden extraer los datos de usuario y la versión del sistema operativo con la que cuenta.

Generalmente, en el compartimiento de la batería se encuentra el nombre del fabricante y el modelo del dispositivo, otra forma de identificar el dispositivo puede ser por la pantalla de inicio (la que aparece cuando se enciende el dispositivo móvil), dicha pantalla también podría ayudar a identificar el nombre del prestador de servicios. Finalmente, por medio del modo de recuperación se puede obtener información acerca del modelo y de la versión de Android que se está ejecutando. El procedimiento para la identificación a través del modo de recuperación es:

1. Establecer el dispositivo móvil en modo de recuperación (ver sección 4.3.2.2.1 y anexo D).
2. Establecer una conexión por medio de ADB entre el dispositivo móvil (en modo de recuperación) y una computadora (ver sección 4.3.1.1.1).
3. A través de un intérprete de comandos en ADB ejecutar los comandos de la figura 19.

```
>$adb shell
$getprop ro.build.display.id
$getprop ro.build.fingerprint
```

Fig. 19 Comandos para verificación de propiedades en Android.

De tal forma que se debe obtener una salida como la mostrada en la figura 20.

```
$ getprop ro.build.display.id
GINGERBREAD.XXKPH
$ getprop ro.build.fingerprint
samsung/GT-S5830/GT-S5830:2.3.4/GINGERBREAD/XXKPH:user/test-
keys
```

Fig. 20 Propiedades de dispositivo Samsung GT-S5830 con S.O. Android.

De la figura 20 podemos identificar que el dispositivo móvil es un Samsung GT-S5830 con un sistema operativo Android 2.3.4 GingerBread XXKPH.

La información obtenida nos servirá para obtener la versión específica del S.O. que está ejecutando el dispositivo, modificarla y aplicarla de nuevo al dispositivo.

6.1.2 Obtención de la copia del S.O.

Copias del S.O Android se pueden obtener en internet. Principalmente compartidas por comunidades de personas interesadas en entender cómo funciona y en realizar modificaciones a sus dispositivos móviles. Algunos fabricantes también ponen a disposición pública el software compilado y el código fuente del que se creó. Finalmente Google, empresa propietaria de Android,

distribuye el código fuente de Android (sin modificaciones de ningún fabricante) con la finalidad de que otras personas lo estudien y lo adapten a sus necesidades.

Google distribuye Android haciendo uso de la licencia "Creative Commons v2.5" (Commons) que entre otras cosas permite copiar, distribuir y transmitir el trabajo de manera libre, adaptar el trabajo a tus necesidades y hacer uso comercial del mismo.

Los principales sitios para obtener copias compiladas y del código fuente de Android son:

- http://source.android.com/ Sitio oficial donde se puede descargar el código fuente de Android (sin compilar), contiene tutoriales y guías para personalizar Android y aplicarlo a diferentes dispositivos.
- http://www.xda-developers.com/ Comunidad de desarrolladores en dispositivos móviles que principalmente se dedican a personalizar la interfaz y la funcionalidad de sus dispositivos, de cualquier fabricante. Muy buena fuente para buscar distribuciones de Android específicas de los fabricantes y operadores así como el código fuente.
- http://www.sammobile.com/ Comunidad de desarrolladores enfocada a dispositivos móviles Samsung con S.O. Android. Se pueden encontrar noticias, herramientas y software compilado de Android, con distribuciones específicas por región geográfica y operadores móviles.
- http://opensource.samsung.com/ Sitio oficial de Samsung donde se distribuye el código fuente de diferentes S.O. para dispositivos Samsung incluyendo televisiones, teléfonos inteligentes, reproductores de música, cámaras fotográficas y de video, impresoras y multifuncionales y equipos de red.
- http://www.techknow.me/forum/ Comunidad especializada en tabletas, se pueden encontrar herramientas, tutoriales y software necesario para personalizar una gran gama de tabletas con S.O. Android.

Evidentemente, la oferta de sitios que contienen código, información y software necesario para personalizar dispositivos Android es muy grande. Es importante que al descargar la distribución de Android específica para nuestro dispositivo nos aseguremos de ubicar el modelo, la versión y la distribución adecuada.

El formato en el que se encuentran los archivos es diferente para cada fabricante y en algunos casos puede variar dependiendo del modelo y de la versión de Android en uso. Samsung utiliza archivos .tar, Sony archivos .ftf, algunas tabletas usan archivos sin extensión.

Es claro que no existe un estándar y que cada fabricante emplea el que más le convenga.

Para el ejemplo que se muestra en la sección 6.1.1, utilizando el sitio http://www.sammobile.com/ se descargó un archivo llamado "S5830LUMKP3_S5830LUMKP3_S5830LTCEKP3_HOME.tar" correspondiente a la versión 2.3.4 de Android para el dispositivo S5830.

6.1.3 Extracción de los archivos de sistema Android.

Una vez que contamos con los archivos del sistema Android nuestro objetivo es localizar el archivo que especifica el contenido de la partición de recuperación. A través de la partición de recuperación intentaremos llevar a cabo un procedimiento similar al método AFPhysical.

Por lo que el objetivo de este paso es reconocer, extraer y modificar el archivo necesario para obtener permisos de administrador a través del modo de recuperación o en su defecto del modo de operación normal.

El archivo del sistema Android que descargamos se encuentra generalmente comprimido en formato ZIP o empaquetado en formato TAR. Haciendo uso de utilerías como WinZip (WinZip International) o Winrar (RarLab Winrar) para Windows se puede descomprimir o desempaquetar en los archivos que forman el sistema. En sistemas Linux se pueden utilizar los comandos *unzip* y *tar*

Para el ejemplo del archivo descargado en la sección 6.1.2 llamado "S5830LUMKP3_S5830LUMKP3_S5830LTCEKP3_HOME.tar" descomprimiendo los archivos mostrados en la figura 21.

```
Directory of S5830LUMKP3_S5830LUMKP3_S5830LTCEKP3_HOME

14/05/2013  09:21 a.m.    <DIR>          .
14/05/2013  09:21 a.m.    <DIR>          ..
16/07/2011  05:47 a.m.        18,804,736 amss
16/07/2011  05:15 a.m.           464,928 arm11boot
16/07/2011  05:15 a.m.         6,361,088 boot.img
16/07/2011  05:47 a.m.        10,362,880 csc.rfs
16/07/2011  05:47 a.m.           786,432 mibib
16/07/2011  05:47 a.m.           566,408 oemsbl
16/07/2011  05:47 a.m.           368,640 qcsbl
16/07/2011  05:15 a.m.         6,774,784 recovery.img
16/07/2011  05:15 a.m.       166,809,600 system.rfs
               9 File(s)    211,299,496 bytes
               2 Dir(s)  186,956,189,696 bytes free
```

Fig. 21 Archivos obtenidos de descomprimir el paquete de actualización S5830LUMKP3_S5830LTCEKP3_HOME.tar

El contenido de los archivos de sistema varían por fabricante y por modelo de dispositivo pero en general, debemos ser capaces de reconocer por lo menos uno de los siguientes archivos, el archivo "boot.img" y el archivo "recovery.img".

El archivo "boot.img" define los parámetros y la estructura de archivos que tendrá el sistema al ingresar al modo normal (el modo común de operación de todos los dispositivos).

El archivo “recovery.img” define los parámetros y la estructura de archivos que tendrá el sistema al ingresar al modo de recuperación (modo especial explicado en la sección 4.3.2.2.1 de este trabajo).

Preferentemente modificaremos el archivo que define los parámetros del sistema en modo de recuperación. Sin embargo, hay que tener presente que en algunos modelos de dispositivos móviles el modo de recuperación no se encuentra presente por lo que es posible que no exista un archivo “recovery.img”. En ese caso tendríamos que modificar los parámetros del archivo “boot.img”.

Es importante notar que los archivos que contienen los parámetros de Android son necesarios para iniciar un dispositivo móvil, sin embargo, los nombres de estos archivos, el formato de compresión y de empaquetamiento pueden cambiar (y suelen ser diferentes) entre diferentes fabricantes e incluso entre diferentes modelos del mismo fabricante.

Cada uno de los archivos mencionados (*boot.img, recovery.img*) contiene por lo menos una cabecera, copia del kernel y el “ramdisk”. Nuestro objetivo es modificar el “ramdisk” pues contiene los parámetros y la estructura de archivos del sistema Android.

El “ramdisk” consiste en un espacio de memoria RAM en la que se almacenan archivos necesarios para iniciar el S.O. En el “ramdisk” de Android existen archivos que establecen las propiedades del sistema que se ejecutará en el dispositivo.

Para desempaquetar las imágenes se pueden emplear utilerías estándar como “abootimg” (Grandou, 2010).

abootimg es una herramienta que funciona para imágenes creadas siguiendo los lineamientos definidos en el archivo bootimg.h del código de Android. Al no ser un estándar y dado que Google permite que los fabricantes tomen el código fuente y lo modifiquen a su gusto, es posible que existan imágenes que se encuentren empaquetadas de otra manera y por lo tanto no puedan desempaquetarse empleando esta utilería.

abootimg permite: obtener información de una imagen, descomprimir una imagen existente en los archivos que la componen, actualizar una imagen (modificando solo uno de los archivos que la componen) y crear una nueva imagen a partir de sus archivos.

En la figura 22 se muestra el método para obtener información del archivo “recovery.img”

```
$ ./abootimg -i recovery.img
Android Boot Image Info:
* file name = recovery.img
* image size = 8388608 bytes (8.00 MB)
page size  = 2048 bytes
* Boot Name = ""
* kernel size       = 3002744 bytes (2.86 MB)
ramdisk size      = 1639626 bytes (1.56 MB)
```

Fig. 22 Información del archivo *recovery.img* para el dispositivo Samsung S5830 con Android.

En la figura 23 se muestra el método para desempaquetar una imagen “recovery.img” en sus componentes (kernel y ramdisk).

```
$ abootimg -x recovery.img
writing boot image config in bootimg.cfg
extracting kernel in zImage
extracting ramdisk in initrd.img
```

Fig. 23 Desempaquetado de archivo *recovery.img* para dispositivo Samsung S5830.

La herramienta extrae el kernel en forma comprimida en el archivo zImage y el “ramdisk” en el archivo “initrd.img”.

El “ramdisk” contiene archivos y parámetros necesarios para el inicio del sistema como el proceso init, el archivo “init.rc” y el archivo “default.prop” (DLS, 2013).

La idea es extraer esos archivos del “ramdisk” modificar los parámetros de inicio de Android, actualizar la imagen y finalmente aplicarla al teléfono.

En la figura 24 se puede observar que Linux reconoce el archivo “initrd.img” como un archivo comprimido utilizando el formato gzip (Loup, 2003):

```
$ file initrd.img
initrd.img: gzip compressed data, from Unix, last modified: Mon May 13
09:59:17 2013
```

Fig. 24 Detalles del archivo *initrd.img*

En la figura 25 se muestra el método para descomprimir y obtener un archivo empaquetado usando el formato cpio (Free Software Foundation, 2010)

```
$ gunzip initrd.img
$ ls
initrd
$ file initrd
initrd: ASCII cpio archive (SVR4 with no CRC)
```

Fig. 25 Desempaquetado del archivo *initrd.img* y detalles del archivo *initrd*

Una vez descomprimido lo desempaquetamos con la utilería cpio, como se muestra en la figura 26.

```
$ cpio -i -F initrd
15408 blocks
```

Fig. 26 Desempaquetado del archivo *initrd.img* y detalles del archivo *initrd*

En la figura 27 se muestran los archivos obtenidos de desempaquetar *initrd.*

```
$ ls -l
total 8372
-rwxrwxrwx. 1 root root       0 May 31 17:52 COOPER.rle
drwxrwxrwx. 1 root root       0 May 31 17:52 data
-rwxrwxrwx. 1 root root    3440 May 31 17:52 default.prop
drwxrwxrwx. 1 root root       0 May 31 17:52 dev
drwxrwxrwx. 1 root root     160 May 31 17:52 etc
-rwxrwxrwx. 1 root root  222472 May 31 17:52 fsua
-rwxrwxrwx. 1 root root   94380 May 31 17:52 init
-rwxrwxrwx. 1 root root    3048 May 31 17:52 init.qcom.post_boot.sh
-rwxrwxrwx. 1 root root    6563 May 31 17:52 init.qcom.sh
-rwxrwxrwx. 1 root root    1485 May 31 17:52 init.rc
drwxrwxrwx. 1 root root     144 May 31 17:52 lib
drwxrwxrwx. 1 root root       0 May 31 17:52 proc
-rwxrwxrwx. 1 root root  331588 May 31 17:52 recovery
drwxrwxrwx. 1 root root     240 May 31 17:52 res
drwxrwxrwx. 1 root root     448 May 31 17:52 sbin
drwxrwxrwx. 1 root root       0 May 31 17:52 sys
drwxrwxrwx. 1 root root       0 May 31 17:52 system
drwxrwxrwx. 1 root root       0 May 31 17:52 tmp
-rwxrwxrwx. 1 root root       0 May 31 17:52 ueventd.goldfish.rc
-rwxrwxrwx. 1 root root    5759 May 31 17:52 ueventd.rc
```

Fig. 27 Contenido del archivo *initrd*

El archivo “default.prop” es 1 de los 4 archivos de los cuales Android lee las propiedades con las que iniciará el sistema. El proceso de inicio de Android y su interacción con el archivo “default.prop” es el siguiente:

Cuando se enciende el dispositivo móvil se ejecuta el “Boot ROM” que se encuentra almacenado en el ASIC del CPU. Las funciones del “Boot ROM” son:

- Detectar el medio que se emplea de almacenamiento y dentro de éste localizar el “boot loader”
- Almacena en memora RAM el “boot loader” y comienza la ejecución de éste.

El “bootloader” es un programa especial que no forma parte de Android, sino que se ejecuta antes de éste. Tiene la función de preparar la memora RAM y cargar el Kernel que el sistema ejecutará. La ejecución del “bootloader” se divide en 2 fases:

- 1ª fase del “bootloader”: Detecta la memoria RAM presente en el dispositivo y almacena una copia de su código en memoria. Inicia ejecución de la 2ª fase del “bootloader”.
- 2ª fase del “bootloader”: Esta fase establece los espacios en memoria RAM que se necesitaran para los sistemas de archivos y reconocerá dispositivos de memoria adicionales.
- La 2ª fase del “bootloader” reconoce los botones que se presionan mientras se inicia el dispositivo. De ésta forma selecciona el kernel apropiado del medio de almacenamiento, establece sus parámetros y lo almacena en memoria. Ésta función permite ingresar a modos especiales de operación como el “modo de recuperación” o el modo “fastboot”.
- Finalmente la 2ª fase del “bootloader” descomprime el kernel almacenado en memoria, aplica los parámetros correspondientes y el kernel comienza a ejecutarse en el dispositivo.

El kernel se encarga de cargar servicios de administración de memoria, preparar espacios protegidos de memoria (que se usarán más adelante), cargar controladores, buscar y ejecutar el proceso “init” localizado en /system/core/init en Android.

El proceso “init” se considera el proceso “padre” del resto de los procesos en Android ya que éstos serán ejecutados por “init” o por un proceso que previamente fue lanzado por él.

Dentro de los servicios que ejecuta el proceso “init” se encuentra el administrador de propiedades de Android.

Las propiedades en Android son muy importantes ya que establecen los parámetros con los que se ejecutarán las aplicaciones y los servicios. A través del administrador de propiedades los procesos deben ser capaces de obtener el valor de éstas y en algunos casos modificarlas.

El administrador de propiedades reserva un espacio de 32,620 bytes en memoria RAM y lee el contenido de 4 archivos de los que extrae y copia las propiedades que servirán para la ejecución del sistema. Los 4 archivos en orden de lectura del sistema son:

- /default.prop
- /system/build.prop
- /system/default.prop
- /data/local.prop

El tamaño del espacio de memoria que se utiliza para almacenar las propiedades del sistema y los archivos desde los que se leerán éstas se encuentran definidos en el archivo "_system_properties.h" localizado en "/libc/include/sys/".

El archivo "system_properties.h" define las funciones que sirven al sistema para leer y establecer el valor de las propiedades.

Una copia de los archivos "_system_properties.h" y "system_properties.h" se encuentran en el Anexo G.

El Administrador de propiedades de Android lee los valores de los 4 archivos listados y guarda en memoria las propiedades y los valores definidos en éstos. Una vez que se establece una propiedad cuyo nombre comienza con "ro" su valor no puede ser modificado.

Después de establecer los valores de las propiedades del sistema, "init" debe inicializar servicios y programas para Android. Con éste fin se sirve de los archivos:

- "init.rc" – Provee instrucciones genéricas para inicializar un dispositivo Android.
- "init.<nombre_código_del_dispositivo>.rc" – Provee instrucciones específicas para el modelo Android.

En la figura 28 se muestra un extracto del archivo init.rc de Android (Dummann, 2010):

```
## Daemon processes to be run by init.
##
service ueventd /sbin/ueventd
    critical

service console /sbin/sh
    console
    disabled
    user shell
    group log

on property:ro.secure=0
    start console

service fbsetup /sbin/fucktheframebuffer.sh
    oneshot


service recovery /sbin/recovery
    oneshot

# adbd is controlled by the persist.service.adb.enable system property
service adbd /sbin/adbd
    disabled

# adbd on at boot in emulator
on property:ro.kernel.qemu=1
    start adbd

on property:persist.service.adb.enable=1
    start adbd

on property:persist.service.adb.enable=0
    stop adbd
```

Fig. 28 Extracto del archivo *init.rc* para sistemas Android.

En este archivo se leen propiedades del sistema y se inician servicios con parámetros que dependen del valor leído en las propiedades.

Al establecer valores de propiedades dentro del archivo "/default.prop" obligamos a Android a iniciar el servicio de "consola" localizado en "/sbin/sh" que nos dará acceso al sistema con privilegios de administración.

6.1.4 Modificación del archivo "default.prop"

El archivo "default.prop" se puede leer y modificar con un editor de textos. Algunos fabricantes pueden agregar parámetros a este archivo, sin embargo el contenido básico se muestra en la figura 29.

```
#
# ADDITIONAL_DEFAULT_PROPERTIES
#
ro.secure=1
ro.allow.mock.location=0
ro.debuggable=0
persist.service.adb.enable=1
```

Fig. 29 Contenido del archivo *default.prop* para sistemas Android.

Los parámetros que contiene son:

- La opción "ro.allow.mock.location" con la finalidad de realizar pruebas en aplicaciones que lo requieran, permite establecer al dispositivo móvil en un lugar diferente al que se encuentra.
- La opción "ro.debuggable" es la que habilita la función ADB para comunicarse con una computadora.
- La opción "ro.secure" permite ejecutar aplicaciones con permisos de administrador.
- La opción "persist.service.adb.enable" habilita la función "ADB" desde el inicio del sistema. Con este valor habilitado aunque se reinicie el dispositivo, el servicio ADB siempre estará disponible.

Nos interesa entonces modificar el archivo "default.prop" de tal forma que su contenido sea el mostrado en la figura 30.

```
#
# ADDITIONAL DEFAULT PROPERTIES
#
ro.secure=0
ro.allow.mock.location=0
ro.debuggable=1
persist.service.adb.enable=1
```

Fig. 30 Contenido propuesto para el archivo *default.prop*

Con este contenido en el archivo "default.prop" nos aseguramos de contar con el servicio ADB habilitado y permisos de administrador para acceder a los recursos protegidos por el sistema.

6.1.5 Compresión y empaquetamiento de la imagen del sistema.

El objetivo de éste paso es reconstruir el archivo boot.img o recovery.img con nuestra versión del archivo default.prop presente.

Con el archivo "default.prop" modificado debemos crear primero el archivo "initrd.img" de donde lo extrajimos y después empaquetar éste último en el archivo "recovery.img" o el archivo "boot.img" dependiendo de cuál empleamos en la sección 6.1.3. Para nuestro ejemplo hemos empleado el archivo "recovery.img".

Para crear el archivo "initrd.img" necesitamos llevar a cabo el proceso inverso que aplicamos durante el desempaquetado.

1. En la figura 31 se muestra el método para empaquetar todos los archivos del "ramdisk" utilizando la utilería cpio

```
$ find . | cpio -o -H newc > initrd_tmp
cpio: File ./initrd_tmp grew, 333312 new bytes not copied
16059 blocks
```

Fig. 31 Empaquetado de los archivos que forman el *ramdisk*

La salida del comando "*find .* " es un listado de todos los archivos en la carpeta actual y sus subcarpetas. Reenviamos la salida a la utilería "*cpio*", cuyas opciones (*-o –H newc*) especifican que cree un nuevo archivo empaquetando todo lo que obtiene en la entrada con el formato 'newc' (cpio(1) - Linux man page). Y escribimos la salida al archivo llamado *initrd_tmp*

Finalmente, en la figura 32 se muestra el método para comprimir la salida empaquetada utilizando la herramienta "gzip" y nombrar "initrd.img" al archivo resultante.

```
$ gzip -c initrd_tmp > ../initrd.img
```

Fig. 32 Compresión de los archivos empaquetados en la figura 31.

2. En la figura 33 se muestra la actualización de la imagen "recovery.img" con el archivo "initrd.img" que acabamos de crear.

```
$ abootimg -u recovery.img -r initrd.img
reading ramdisk from initrd.img
Writing Boot Image recovery.img
```

Fig. 33 Actualización del archivo original *recovery.img* agregando la versión modificada del archivo *initrd.img*

Y obtenemos el archivo "recovery.img" actualizado con el "initrd.img" modificado por nosotros.

6.1.6 Aplicar el sistema modificado al dispositivo móvil.

Este paso es específico por fabricante.

Cada fabricante genera sus propias herramientas y procedimientos para aplicar actualizaciones a sus dispositivos por lo que es muy importante para un laboratorio forense conocer y familiarizarse con las distintas herramientas, dispositivos y procedimientos.

En este caso estamos empleando un teléfono Samsung S5830. Samsung creó la herramienta llamada "Odin" que se sirve para aplicar actualizaciones a sus dispositivos. Este trabajo no pretende ahondar en las particularidades de la herramienta "Odin", sin embargo, existen muchos recursos públicamente disponibles en Internet que explican a detalle la herramienta. (Android Zone, 2013)

"Odin" permite aplicar actualizaciones de archivos específicos a sus dispositivos, es decir, se puede actualizar un dispositivo utilizando solo el archivo que contiene dicho cambio. Esto permite reducir el tamaño del archivo final a aplicar y permite a un laboratorio mantener una gran cantidad de archivos para diferentes dispositivos.

Un archivo que contiene solamente una imagen "recovery.img" actualizada utiliza aproximadamente 6 MB, en cambio, un archivo de actualización que contiene todos los archivos del sistema Android utiliza aproximadamente 200 MB.

Aplicar una actualización a través de "Odin" toma aproximadamente 5 min, se lleva a cabo un reinicio del dispositivo e inmediatamente después se puede verificar que se cuenten con los permisos de administrador necesarios para obtener una copia de las diferentes particiones del sistema.

6.1.7 Verificación de privilegios y obtención de copias.

Una vez que se ha completado el procedimiento para obtener permisos de administrador es necesario verificar dichos privilegios y posteriormente realizar una copia de las particiones que contienen los datos que nos interesan.

1. En la figura 34 realizamos una conexión a través de adb y verificamos los detalles de usuario.
   Si contamos con permisos de administrador debemos obtener el siguiente prompt: "*#*" y al ejecutar el comando "*id*" debemos obtener la siguiente salida:

```
# id
id
uid=0(root) gid=0(root)
```

Fig. 34 Comando *id* para verificar detalles de usuario y grupo.

2. En la figura 35 realizamos una copia de la partición */dev/block/stl12* empleamos la utilería "*dd*" para realizar una copia de las diferentes particiones presentes en el dispositivo.

```
$adb shell
#
# cd /dev/block
cd /dev/block
# dd if=stl12 of=/sdcard/stl12.dd
dd if=stl12 of=/sdcard/stl12.dd
429568+0 records in
429568+0 records out
219938816 bytes transferred in 66.512 secs (3306753 bytes/sec)
#
```

Fig. 35 Prueba exitosa de adquisición física empleando *dd. Con permisos de administrador*

En la tabla 11 se muestran algunos de los datos depurados que se recuperaron al emplear la metodología propuesta.

Tipo de información	Valores depurados
Historial del explorador WEB	-http://www.adnkronos.com/IGN/News/Spettacolo/Cinema-morta-Sara-Montiel-la-star-spagnola-che-conquisto-Hollywood_3268577790.html -http://www.adnkronos.com/IGN/News/Spettacolo/Cinema-morta-Sara-Montiel-la-star-spagnola-che-conquisto Hollywood_3268577790.html
Búsquedas en explorador WEB	"microsoft translator", "google"
Datos del calendario	-Tramitar en San Angel la certificación del numero d seguridad social', 'Plaza San Jacinto','Llevar acta d nacimiento e IFE, Mexico_City -Enviar al prof. Nuestro proyecto d investigación. tapiaarguello@derecho.unam.mx, Mexico_City
Datos de contactos	-N:Mamá TEL;TYPE=CELL,VOICE,PREF:005215534721000 -N:Ernestito TEL;TYPE=CELL,VOICE,PREF:005215539967082 -N:Dose;Alberto TEL;TYPE=CELL,VOICE,PREF:5530355087
Listado de archivos multimedia	-Archivo "Mission started.mp3" almacenado en "/system/media/audio/ringtones" -Archivo "Modern catwalk.mp3" almacenado en "/system/media/audio/ringtones" -Archivo: "video-2013-04-01-20-42-42.mp4" almacenado en "/mnt/sdcard/DCIM/Camera
Correos electrónicos	'Hola Adriana, Recibiste un mensaje de Laura '
Mensajes cortos	"Ya estoy pasando cu. No esta tan pesado insurgentes, solo la parte de bodega. Nos vemos al rato ma. Te quiero mucho :)"
Historial de Google Maps	-"Librería el Sotano cid" -Longitud: 19.4035, latitud: -99.17029, Marysol casa. -"Museo jose luis cuevas mexico df"

Tabla 11 Datos depurados, recuperados de extracción en Samsung S5830 con Android.

En el anexo E se muestran algunos de los datos no depurados que se pueden recuperar a partir de una extracción usando el procedimiento propuesto en este trabajo. Un listado completo de la información almacenada por un dispositivo Android se puede consultar en la sección 2.3

6.1.8 Procedimiento para remover la protección del dispositivo móvil con la finalidad de ejecutar un sistema modificado.

En algunos dispositivos, el sistema que permite guardar y ejecutar imágenes personalizadas de Android se encuentra bloqueado, sin embargo, en algunos casos existe la posibilidad de desbloquearlo.

Tal es el caso de los dispositivos Sony de la línea Xperia (Sony Mobile Communications AB, 2013). Sony provee un procedimiento que permite remover dicha restricción pero advierte que seguir el procedimiento de desbloqueo significará perder la garantía y funcionalidades propias del dispositivo como la posibilidad de aceptar actualizaciones de software a través de la red celular.

Para realizar el procedimiento un usuario debe contar con el IMEI de su dispositivo (el cual se puede encontrar en la etiqueta del compartimiento de la batería), enviar un correo electrónico a Sony y esperar la confirmación con un código de desbloqueo. Finalmente, es necesario establecer una conexión entre el dispositivo y un sistema de cómputo para ejecutar una herramienta provista por Sony agregando el código de confirmación obtenido por correo electrónico. Este último paso invalida la garantía del dispositivo y elimina los bloqueos para poder ejecutar software Android personalizado en él dispositivo.

6.2 Particularidades al procedimiento general.

Durante todo el procedimiento que se ha presentado es posible que existan modificaciones propias de cada fabricante.

En algunos casos algunos fabricantes prefieren emplear la herramienta “TAR” en lugar de “cpio” para empaquetar, otros cambian los nombres de los archivos “initrd.img” por “ramdisk”, algunos agregan parámetros al archivo “default.prop”. Y como hemos visto, cada uno emplea sus propias utilerías para aplicar actualizaciones a sus dispositivos.

Por estas razones y para asegurarse de siempre estar al tanto de las particularidades que cada fabricante agrega a sus dispositivos es importante que un laboratorio forense se encuentre siempre trabajando con los dispositivos disponibles presentes en el mercado y que constantemente realicen extracciones forenses en equipos de prueba.

Las particularidades más importantes al procedimiento presentado se encuentran en la forma de reconocer los archivos de interés, desempaquetarlos y re-empaquetarlos en un archivo actualizable para el dispositivo.

Para los dispositivos Samsung (el mismo fabricante del ejemplo en la sección 6.1) de la serie "Galaxy S" los archivos del sistema (obtenidos en la sección 6.1.2) no contienen los archivos "recovery.img" ni "boot.img". Un listado de los archivos contenidos en los paquetes Android es:

```
E:\I9100UHMS8_I9100IUSMS4_I9100UHMS1_HOME>dir
 Volume in drive E is Documents
 Volume Serial Number is EC6D-65C7

 Directory of E:\I9100UHMS8_I9100IUSMS4_I9100UHMS1_HOME

18/05/2013  07:41 a.m.    <DIR>          .
18/05/2013  07:41 a.m.    <DIR>          ..
31/01/2013  01:03 a.m.           131,072 boot.bin
07/02/2013  08:27 a.m.        27,029,676 cache.img
31/01/2013  01:03 a.m.       480,158,944 factoryfs.img
07/02/2013  08:27 a.m.       391,629,240 hidden.img
15/01/2013  03:28 a.m.        12,583,168 modem.bin
31/01/2013  01:03 a.m.         1,277,952 param.lfs
31/01/2013  01:03 a.m.         1,310,720 Sbl.bin
31/01/2013  01:03 a.m.         8,387,840 zImage
               9 File(s)    932,629,954 bytes
               3 Dir(s)  179,244,081,152 bytes free
```

Fig. 36 Listado de archivos extraídos para el paquete de actualización *I9100IUSMS4_I9100UHMS1_HOME* para el Samsung GT-I9100

En este caso, a partir de desempaquetar (XDA Developers, 2011) los diferentes archivos y verificar su contenido se puede ver que es el archivo zImage el que contiene el archivo de sistema que nos interesa modificar.

Un listado del archivo zImage desempaquetado:

```
E:\INITRAMFS_zImage_EXTRACTED>dir

 Directory of E:\INITRAMFS_zImage_EXTRACTED

18/05/2013  07:34 a.m.    <DIR>          .
18/05/2013  07:34 a.m.    <DIR>          ..
17/05/2013  11:02 p.m.    <DIR>          data
18/05/2013  07:34 a.m.               129 default.prop
17/05/2013  11:02 p.m.    <DIR>          dev
17/05/2013  11:02 p.m.             1,010 fstab.smdk4210
17/05/2013  11:02 p.m.           113,512 init
17/05/2013  11:02 p.m.               581 init.bt.rc
17/05/2013  11:02 p.m.             2,344 init.goldfish.rc
17/05/2013  11:02 p.m.            24,231 init.rc
17/05/2013  11:02 p.m.            12,768 init.smdk4210.rc
17/05/2013  11:02 p.m.             6,209 init.smdk4210.usb.rc
17/05/2013  11:02 p.m.             1,637 init.trace.rc
17/05/2013  11:02 p.m.             3,915 init.usb.rc
17/05/2013  11:02 p.m.    <DIR>          lib
17/05/2013  11:02 p.m.             1,524 lpm.rc
17/05/2013  11:02 p.m.    <DIR>          proc
17/05/2013  11:02 p.m.             1,532 recovery.rc
17/05/2013  11:02 p.m.    <DIR>          res
17/05/2013  11:02 p.m.    <DIR>          sbin
17/05/2013  11:02 p.m.    <DIR>          sys
17/05/2013  11:02 p.m.    <DIR>          system
17/05/2013  11:02 p.m.    <DIR>          tmp
17/05/2013  11:02 p.m.               272 ueventd.goldfish.rc
17/05/2013  11:02 p.m.             3,879 ueventd.rc
17/05/2013  11:02 p.m.             4,149 ueventd.smdk4210.rc
17/05/2013  11:02 p.m.    <DIR>          vendor
              15 File(s)        177,692 bytes
              12 Dir(s)  179,244,077,056 bytes free
```

Fig. 37 Contenido del archivo *zImage desempaquetado.*

El archivo “default.prop” se encuentra presente, sin embargo, podemos empezar a observar diferencias:

- No existe una separación entre los archivos del modo de recuperación y los archivos del modo normal de operación.
- El procedimiento de desempaquetado de los archivos es diferente y por lo tanto.
- El procedimiento para empaquetar los archivos también cambia.

El archivo “zImage” es el “kernel” de Android compilado específicamente para un dispositivo. Por lo que al hacer alguna modificación a dicho archivo es necesario re-compilar el código fuente del kernel para obtener un archivo “zImage” compilado y modificado.

El proceso de re-compilar un kernel es mucho más elaborado que simplemente empaquetar y comprimir utilizando “*cpio*” y “*gzip*”. Sin embargo, Google y los diferentes fabricantes ofrecen públicamente en internet el código fuente necesario para llevar a cabo dicha compilación (véase la sección 6.1.2). De igual forma existen guías que permiten familiarizarse con el procedimiento (XDA Developers, 2011), (XDA Developers, 2012).

Sony es otro ejemplo de cómo cada fabricante emplea los formatos y las utilerías que considera apropiadas. Los archivos de actualización de Android para los dispositivos Sony tienen la extensión “.ftf”, se pueden extraer con utilerías como “7-zip”, modificarse con editores de texto estándares y aplicarse al dispositivo haciendo uso de herramientas especiales para Sony como “Flashtool” (Xperia Gamer, 2012).

De la misma forma, Motorola cuenta con el formato “SBF” para sus archivos de sistema, sus propias utilerías para desempaquetar y para aplicar a sus dispositivos (AND Developers, 2011).

A pesar de las particularidades en los procedimientos de desempaquetado y empaquetado de los archivos de sistema, el procedimiento general sigue siendo el mismo. Desempaquetar los archivos de sistema de Android, modificar los valores contenidos en el archivo “default.prop”, empaquetar y aplicar al dispositivo.

Capítulo 7. Conclusiones y trabajo futuro.

Los teléfonos inteligentes han tenido una gran penetración en el mundo y se espera que estas tendencias aumenten. De igual forma han aumentado los dispositivos que cuentan con un sistema operativo Android, desde teléfonos inteligentes, tabletas, televisiones, consolas de videojuegos y próximamente lentes de realidad aumentada (Google).

Como frecuentemente sucede, un aumento en la facilidad de uso generalmente se acompaña de una disminución en una o más de las 3 propiedades intrínsecas de la información, vista desde el punto de vista de la seguridad: confidencialidad, integridad y autenticidad.

En este caso somos los mismos dueños de los dispositivos móviles quienes por conveniencia almacenamos datos como contactos, mensajes, correos electrónicos, fotografías, etc. en nuestros dispositivos móviles.

Este trabajo ha mostrado que es posible, con la configuración de fábrica de un dispositivo móvil, extraer los datos almacenados en dicho dispositivo. Y que los datos obtenidos así como el procedimiento utilizado son confiables, reproducibles y verificables.

La intención de este procedimiento es apoyar a las agencias de procuración de justicia para la extracción de información en dispositivos móviles (sin importar si algún sistema de control de acceso se encuentra habilitado en el dispositivo). Sin embargo, es claro que este procedimiento también puede tener otros usos que atentan contra la confidencialidad de la información de cualquier persona que hace uso de un dispositivo móvil con S.O. basado en Linux.

Podemos también concluir que un laboratorio de análisis forense digital debe aplicar este procedimiento a una gran gama de dispositivos con el fin de familiarizarse con su operación y para crear una base de “imágenes de recuperación” a aplicar.

7.1 Resumen de contribuciones.

Este trabajo contribuye con los métodos de adquisición de datos no volátiles en dispositivos móviles con S.O. de código abierto permitiendo obtener acceso con permisos de administración al dispositivo.

El acceso con permisos de administración al dispositivo permite aplicar métodos de extracción de datos a dispositivos con las siguientes características.

- Dispositivos bloqueados con contraseñas, PIN, patrones o algún otro control de acceso impuesto por el usuario del mismo.

- Dispositivos cuyas interfaces de entrada o salida se encuentran dañadas como pantallas rotas o interfaces de entrada defectuosas.
- Dispositivos que no cuentan con permisos de administración ya que uno de los objetivos de este procedimiento es obtener dichos permisos.

De esta forma, el modelo propuesto en este trabajo permite ampliar el número de dispositivos a los que se les puede aplicar alguna técnica de extracción de datos y aumentar la cantidad de datos que se pueden extraer.

7.2 Líneas de investigación y trabajo futuro.

Algunas de las líneas de investigación que se pueden desarrollar a partir de este trabajo son:

- Complemento del proceso forense: Este trabajo se enfoca a la adquisición de la copia de los datos de usuario, sin embargo, hace falta interpretar las bases de datos específicas para las diferentes aplicaciones, generar herramientas que puedan analizar grandes cantidades de texto y archivos multimedia para extraer términos relevantes a una situación, fechas o contexto.
- Para datos críticos, el diseño de un dispositivo móvil que pueda operar con almacenamiento en unidades de red. De tal forma que comprometer la integridad del dispositivo no signifique comprometer los datos del usuario.
- Estudio de aplicaciones maliciosas existentes en el mercado de Android. A través del procedimiento aquí descrito es posible obtener una copia de las aplicaciones instaladas en el dispositivo para posteriormente analizarlas y poder identificar aplicaciones maliciosas y estudiar su comportamiento.

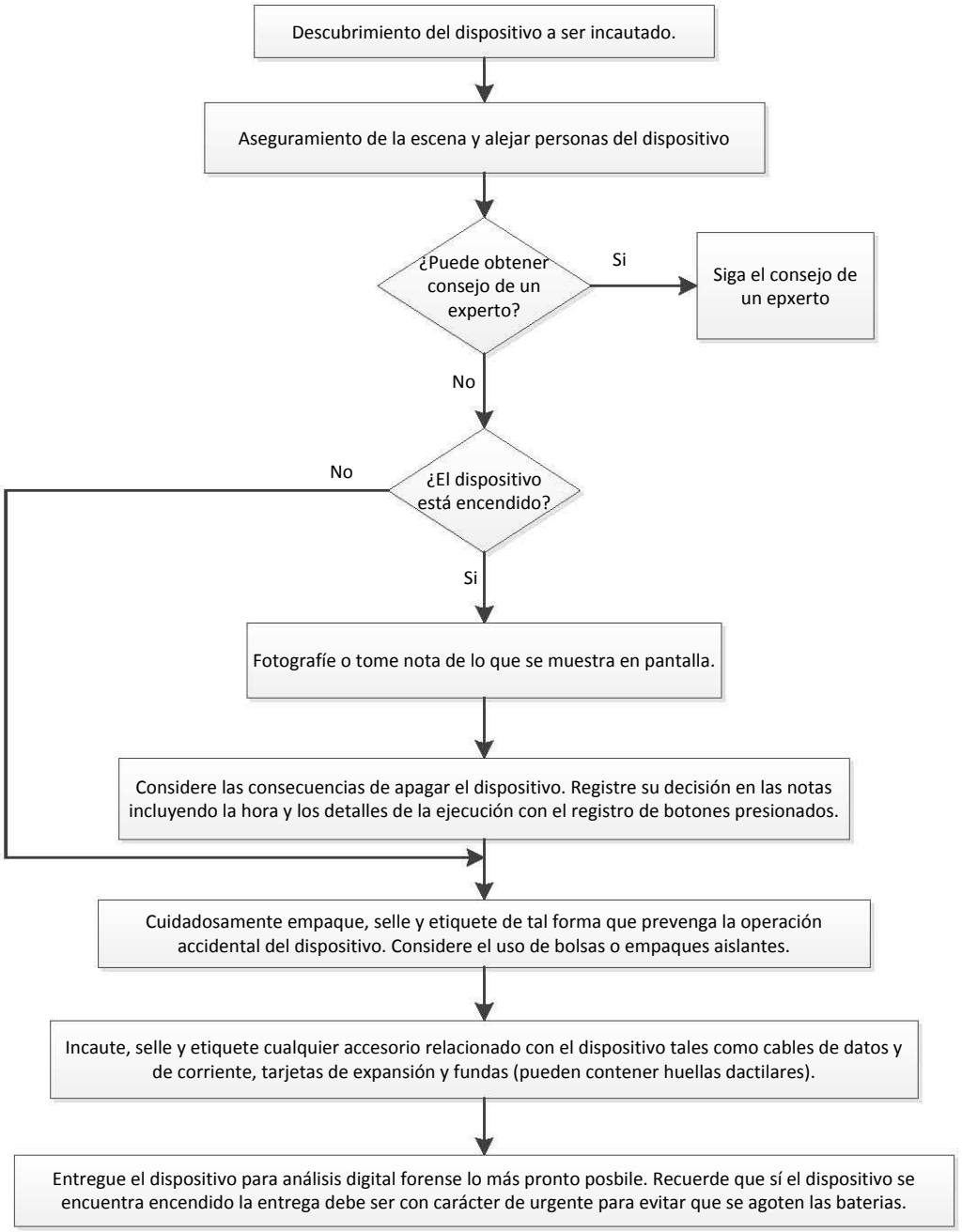
Anexo A.

Se muestra la estructura general del archivo AndroidManifest.xml con cada elemento que puede contener. (Android)

```
<?xml version="1.0" encoding="utf-8"?>
 <manifest>

    <uses-permission />
    <permission />
    <permission-tree />
    <permission-group />
    <instrumentation />
    <uses-sdk />
    <uses-configuration />
    <uses-feature />
    <supports-screens />
    <compatible-screens />
    <supports-gl-texture />

    <application>

        <activity>
            <intent-filter>
                <action />
                <category />
                <data />
            </intent-filter>
            <meta-data />
        </activity>

        <activity-alias>
            <intent-filter> . . . </intent-filter>
            <meta-data />
        </activity-alias>
        <service>
            <intent-filter> . . . </intent-filter>
            <meta-data/>
        </service>

        <receiver>
            <intent-filter> . . . </intent-filter>
            <meta-data />
        </receiver>

        <provider>
            <grant-uri-permission />
            <meta-data />
            <path-permission />
        </provider>

        <uses-library />

    </application>

</manifest>
```

Anexo B.

Modelo traducido de la Guía de buenas prácticas para evidencia electrónica basada en computadoras de la ACPO. Diseñado para la incautación de Asistentes Personales Digitales.



Anexo C.

Listado de recursos que se pueden obtener del proceso de estudio de un dispositivo móvil y el tipo de preguntas que pueden ayudar a responder. Traducido de “Guidelines on Cell Phone Forensics” Special Publication 800-101 del NIST. (National Institute of Standards and Technology. Technology Administration U.S. Department of Commerce, 2007).

	¿Quién?	¿Qué?	¿Dónde?	¿Por qué?	¿Cómo?
Identificadores de suscriptor	X				
Registros de llamadas	X			X	
Lista de contactos	X				
Calendario	X	X	X	X	X
Mensajes	X	X	X	X	X
Ubicación geográfica			X	X	
Historial de navegación WEB	X	X	X	X	X
Imágenes y video	X	X	X	X	X
Otro contenido	X	X	X	X	X

Anexo D.

Tabla de combinación de teclas a presionar durante el encendido de un dispositivo para acceder al modo de recuperación.

Información recopilada de:

http://forum.xda-developers.com/
http://forums.androidcentral.com/
http://htcmania.com/

SAMSUNG	**Combinación de teclas**
Galaxy S3	Vol. subir + Vol. bajar + Inicio + Encendido
Galaxy S3 mini	Vol. subir + Inicio + Encendido
Galaxy S2	Vol. subir + Inicio + Encendido
Galaxy Note 2	Vol. subir + Inicio + Encendido
Galaxy Note	Vol. subir + Inicio + Encendido
Galaxy Ace	Inicio + Encendido
Galaxy Pro	Vol. subir + Inicio + Encendido
SONY	**Combinación de teclas**
Xperia X10	Botón Regresar
Xperia X10 mini	Botón Regresar de manera intermitente
Xperia X10 mini pro	Botón Regresar de manera intermitente
Xperia X8	Mantener presionado encendido + regresar intermitente.
Xperia ARC	Vol. Bajar
Xperia ARC S	Vol. Bajar + Encendido
Xperia S	Con el LED azul presiona Vol. Subir
Xperia P	Con el LED azul presiona Vol. Subir
MOTOROLA	**Combinación de teclas**
Droid	Botón X + Encendido
Droid 2	Botón X + Encendido
Defy	Encendido + Vol. Bajar
Xoom	Vol. Bajar

Anexo E.

Listado de algunos de los datos encontrados en la extracción mostrada en el capítulo 6.

Dispositivo Samsung Galaxy Ace S5830.

Todos los datos se encontraron almacenados en archivos con extensión “db”. Android hace uso de SQLite (ver sección 2.2.1.3 Base de datos SQLite) por lo que se utilizó SQLite para Linux con la finalidad de interpretar los datos extraídos.

Ruta de almacenamiento y Extracto de datos encontrados	Datos interpretados
/data/data/com.android.browser/databases/browser.db INSERT INTO "bookmarks" VALUES(9689,'http://www.adnkronos.com/IGN/News/Spettacolo/Cinema-morta-Sara-Montiel-la-star-spagnola-che-conquisto Hollywood_3268577790.html','http://www.adnkronos.com/IGN/News/Spettacolo/Cinema-morta-Sara-Montiel-la-star-spagnola-che-conquisto-Hollywood_3268577790.html',1,1365476739220,0,NULL,0,NULL,NULL,NULL,0,99); INSERT INTO "searches" VALUES(2,'microsoft translator',1337963446319); INSERT INTO "searches" VALUES(3,'google',1338019668219);	**Para el explorador web:** **Se agrega a los favoritos el sitio:** http://www.adnkronos.com/IGN/News/Spettacolo/Cinema-morta-Sara-Montiel-la-star-spagnola-che-conquisto Hollywood_3268577790.html **Se agrega al historial de busquedas:** microsoft translator google Fechas codificadas
/data/data/com.android.providers.calendar/databases/calendar.db INSERT INTO "Events" VALUES(5,'local','local','MyCalendar2',NULL,NULL,NULL,1,NULL,1,NULL,'Tramitar en San Angel la certificación del numero d seguridad social','Plaza San Jacinto','Llevar acta d nacimiento e IFE',NULL,1,NULL,1346677200000,1346680800000,'America/Mexico_City',NULL,0,0,0,1,1,NULL,NULL,NULL,NULL,NULL,NULL,NULL,1346680800000,1,0,0,1,1,'',0,NULL,NULL,NULL,NULL); INSERT INTO "Events" VALUES(15,'local','local','MyCalendar12',NULL,NULL,NULL,1,NULL,1,NULL,'Enviar al prof. Nuestro proyecto d investigaciòn','','tapiaarguello@derecho.unam.mx',NULL,1,NULL,1347026400000,1347030000000,'America/Mexico_City',NULL,0,0,0,1,1,NULL,NULL,NULL,NULL,NULL,NULL,NULL,1347030000000,1,0,0,1,1,'',0,NULL,NULL,NULL,NULL);	**Para el calendario:** **Se agregan eventos:** Tramitar en San Angel la certificación del numero d seguridad social','Plaza San Jacinto','Llevar acta d nacimiento e IFE, Mexico_City Enviar al prof. Nuestro proyecto d investigaciòn','','tapiaarguello@derecho.unam.mx, Mexico_City Fechas codificadas

/data/data/com.android.providers.contacts	**Para los contactos:** **Se agregan:** N:Mamá FN:Mamá TEL;TYPE=CELL,VOICE,PREF:005215534721000 N:Ernestito FN:Ernestito TEL;TYPE=CELL,VOICE,PREF:005215539967082 N:Dose;Alberto FN:Alberto Dose TEL;TYPE=CELL,VOICE,PREF:5530355087
/data/data/com.android.providers.media/databases/internal.db INSERT INTO "audio_meta" VALUES(64,'/system/media/audio/ringtones/Mission started.mp3','Mission started.mp3',1336423,'audio/mpeg',1167610325,1217592000,'Mission started',QQQQQQQQQQQQQQQ',33384,1,NULL,4,0,NULL,1,0,0,0,'<unknown>',0,NULL); INSERT INTO "audio_meta" VALUES(65,'/system/media/audio/ringtones/Modern catwalk.mp3','Modern catwalk.mp3',984293,'audio/mpeg',1167610326,1217592000,'Modern catwalk',QQQQQQQQQQQQQQ',24581,1,NULL,4,0,NULL,1,0,0,0,'<unknown>',0,NULL);	**Para el listado de archivos multimedia almacenados en la memoria interna:** El archivo "Mission started.mp3" almacenado en le ruta "/system/media/audio/ringtones" El archivo "Modern catwalk.mp3" almacenado en le ruta "/system/media/audio/ringtones"
/data/data/com.android.providers.media/databases/external-33386164.dbINSERT INTO "video" VALUES(97,'/mnt/sdcard/DCIM/Camera/video-2013-04-01-20-42-42.mp4','video-2013-04-01-20-42-42.mp4',6163680,'video/mp4',1364870593,1364870592,'video-2013-04-01-20-42-42',30119,'<unknown>','Camera','640x480',NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,1364870592000,8776904487311107402,'1506676782','Camera',30070);INSERT INTO "audio_meta" VALUES(10,'/mnt/sdcard/WhatsApp/Media/WhatsApp Audio/AUD-20120520-WA0010.amr','AUD-20120520-WA0010.amr',24390,'audio/amr',1338184937,1337563716,'AUD-20120520-WA0010'Q',15240,1,NULL,2,0,NULL,0,1,0,0,'<unknown>',0,NULL);	**Para el listado de archivos multimedia almacenados en la memoria externa:**El archivo de video: "video-2013-04-01-20-42-42.mp4" almacenado en la ruta: "/mnt/sdcard/DCIM/Camera"El archivo de audio: "AUD-20120520-WA0010.amr" almacenado en la ruta: "/mnt/sdcard/WhatsApp/Media/WhatsApp Audio"

/data/data/com.android.email/databases/EmailProviderBody.db INSERT INTO "Body" VALUES(963,1674,'Hola Adriana, Recibiste un mensaje de Laura Haz click aquí para ver el mensaje y el anuncio. Para poder leer el contenido de los mensajes, necesitas ser usuario superior...La mayoría de nuestros usuarios superiores han encontrado un cuarto o compañero de cuarto por medio de nuestro sitio. ¡Suerte en tu búsqueda! El Equipo de CompartoDepa ',NULL,NULL,NULL,'0',NULL);	**Para el listado de Correos electrónicos (en formato html):** 'Hola Adriana, Recibiste un mensaje de Laura Haz click aquí para ver el mensaje y el anuncio. Para poder leer el contenido de los mensajes, necesitas ser usuario superior...La mayoría de nuestros usuarios superiores han encontrado un cuarto o compañero de cuarto por medio de nuestro sitio. ¡Suerte en tu búsqueda! El Equipo de CompartoDepa
/data/data/com.android.providers.telephony/databases/mmssms.db INSERT INTO "canonical_addresses" VALUES(48,'Banamex'); INSERT INTO "canonical_addresses" VALUES(49,'5543881873'); INSERT INTO "canonical_addresses" VALUES(50,'TAE Amigo');INSERT INTO "words_content" VALUES(3,831,'Ya estoy pasando cu. No esta tan pesado insurgentes, solo la parte de bodega. Nos vemos alrato ma. Te quiero mucho :)',831,1); INSERT INTO "words_content" VALUES(4,832,'VEN A LA VENTA DE GALA A TU TELCEL MAS CERCANO! DEL 26 AL 29 DE ABRIL APROVECHA HASTA 70% DE DESCUENTO EN AMIGO KIT,CONSULTA MODELOS PARTICIPANTES',832,1);	**Para el listado de mensajes: Mensajes en el historial:** "Ya estoy pasando cu. No esta tan pesado insurgentes, solo la parte de bodega. Nos vemos alrato ma. Te quiero mucho :)" "VEN A LA VENTA DE GALA A TU TELCEL MAS CERCANO! DEL 26 AL 29 DE ABRIL APROVECHA HASTA 70% DE DESCUENTO EN AMIGO KIT,CONSULTA MODELOS PARTICIPANTES" Fechas y números de teléfono codificados
/data/data/com.android.providers.userdictionary/databases/user_dict.db CREATE TABLE words (_id INTEGER PRIMARY KEY,word TEXT,frequency INTEGER,locale TEXT,appid INTEGER); COMMIT;	**Para el listado de palabras agregadas al diccionario del usuario:** No existe ninguna palabra agregada por el usuario

/data/data/com.google.android.apps.maps/databases/search_history.db INSERT INTO "suggestions" VALUES(19,'libreria el sotano cid:0,0,9432477775444710623',NULL,'libreria el sotano'); INSERT INTO "suggestions" VALUES(20,'loc:19.4035,-99.17029 (andrea)',NULL,'loc:19.4035,-99.17029 (Andrea)'); INSERT INTO "suggestions" VALUES(21,'museo jose luis cuevas mexico df cid:0,0,15857043359476663628',NULL,'museo jose luis cuevas mexico df');	**Para el listado de historial de busquedas en Google Maps:** En sugerencias de búsqueda: "Librería el Sotano cid" En sugerencias de búsqueda: Longitud: 19.4035, latitud: -99.17029 En sugerencias de busqueda: "Museo jose luis cuevas mexico df"

Anexo F.

Listado de los dispositivos móviles a los que se les ha aplicado el procedimiento propuesto en éste trabajo y sus particularidades.

Marca y modelo	Samsung I9100
Versión o versiones de Android	4.0.3, 4.1.2

Los dispositivos Samsung permiten, por configuración de fábrica, la carga y ejecución de un S.O. personalizado.

En este caso se siguió el procedimiento descrito en la sección 6.2 que implica obtener el código fuente del sitio oficial de Android y compilar una versión específica para este modelo.

Para la versión 4.0.3 se compiló un sistema a partir de la distribución oficial de Telcel: "I9100UMLP6_I9100TCELP6" y para aplicar el sistema modificado al dispositivo se utilizó la herramienta oficial de Samsung Odin v1.85

Para la versión 4.1.2 se compiló un sistema a partir de la distribución oficial de Samsung: "I9100XWLS8_I9100XXLS8" y para aplicar el sistema modificado al dispositivo se utilizó la herramienta oficial de Samsung Odin v3.04

Marca y modelo	Samsung I9070
Versión o versiones de Android	2.3.6

En este caso se siguió el procedimiento descrito en la sección 6.2 que implica obtener el código fuente del sitio oficial de Android y compilar una versión específica para este modelo.

Se compiló un sistema a partir de la distribución oficial de Telcel: "I9070UBLK1_I9070-TCELI1" y para aplicar el sistema modificado al dispositivo se empleó la herramienta oficial de Samsung: Odin v3.04

Marca y modelo	Samsung S5830
Versión o versiones de Android	2.3.6

En este caso se siguió el procedimiento descrito en las subsecciones 6.1.x

No fue necesario compilar el código fuente de Android para generar una imagen, se emplearon utilerías provistas por Android como "abootimg" con la finalidad de descomprimir y comprimir los archivos de sistema.

Para aplicar la imagen modificada se empleó la herramienta oficial de Samsung: Odin v4.42

Marca y modelo	Tablet WonderMedia 8850
Versión o versiones de Android	4.1.1

Esta es una Tableta fabricada por la empresa WonderMedia. El software lo crea MaPan quien provee distintas distribuciones en su sitio: http://www.maixin-china.com/software-download.asp

Una de ellas es una distribución que ya cuenta con permisos de administración. Dicha distribución fue modificada con la finalidad de que mantener los permisos de administración durante su aplicación y además evitar que se modificaran los archivos del sistema. El procedimiento de modificación es el mismo que el listado en las subsecciones 6.1.x

Para la actualización de estas tabletas se tienen que colocar todos los archivos de sistema en una memoria extraíble “Micro SD”, apagar la tableta, colocar la memoria en la ranura correspondiente y encender la tableta. No es necesario presionar alguna combinación de botones para ingresar a un modo especial de carga de software.

Anexo G.

Archivo /libc/include/sys/_system_properties.h (Google - Android)

```
/*
 * Copyright (C) 2008 The Android Open Source Project
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *  * Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *  * Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
 * OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
 * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */
#ifndef _INCLUDE_SYS__SYSTEM_PROPERTIES_H
#define _INCLUDE_SYS__SYSTEM_PROPERTIES_H
#ifndef _REALLY_INCLUDE_SYS__SYSTEM_PROPERTIES_H_
#error you should #include <sys/system_properties.h> instead
#else
#include <sys/system_properties.h>
typedef struct prop_area prop_area;
typedef struct prop_msg prop_msg;
#define PROP_AREA_MAGIC   0x504f5250
#define PROP_AREA_VERSION 0x45434f76
#define PROP_SERVICE_NAME "property_service"
/* #define PROP_MAX_ENTRIES 247 */
/* 247 -> 32620 bytes (<32768) */
#define TOC_NAME_LEN(toc)       ((toc) >> 24)
#define TOC_TO_INFO(area, toc)  ((prop_info*) (((char*) area) + ((toc) &
0xFFFFFF)))
struct prop_area {
    unsigned volatile count;
    unsigned volatile serial;
    unsigned magic;
    unsigned version;
    unsigned reserved[4];
    unsigned toc[1];
};
#define SERIAL_VALUE_LEN(serial) ((serial) >> 24)
#define SERIAL_DIRTY(serial) ((serial) & 1)
struct prop_info {
    char name[PROP_NAME_MAX];
    unsigned volatile serial;
    char value[PROP_VALUE_MAX];
};
```

```
struct prop_msg
{
    unsigned cmd;
    char name[PROP_NAME_MAX];
    char value[PROP_VALUE_MAX];
};
#define PROP_MSG_SETPROP 1

/*
** Rules:
**
** - there is only one writer, but many readers
** - prop_area.count will never decrease in value
** - once allocated, a prop_info's name will not change
** - once allocated, a prop_info's offset will not change
** - reading a value requires the following steps
**   1. serial = pi->serial
**   2. if SERIAL_DIRTY(serial), wait*, then goto 1
**   3. memcpy(local, pi->value, SERIAL_VALUE_LEN(serial) + 1)
**   4. if pi->serial != serial, goto 2
**
** - writing a value requires the following steps
**   1. pi->serial = pi->serial | 1
**   2. memcpy(pi->value, local_value, value_len)
**   3. pi->serial = (value_len << 24) | ((pi->serial + 1) & 0xffffff)
**
** Improvements:
** - maintain the toc sorted by pi->name to allow lookup
**   by binary search
**
*/
#define PROP_PATH_RAMDISK_DEFAULT  "/default.prop"
#define PROP_PATH_SYSTEM_BUILD     "/system/build.prop"
#define PROP_PATH_SYSTEM_DEFAULT   "/system/default.prop"
#define PROP_PATH_LOCAL_OVERRIDE   "/data/local.prop"
#endif
#endif
```

Archivo /libc/include/sys/system_properties.h (Google - Android)

```
/*
 * Copyright (C) 2008 The Android Open Source Project
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *  * Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *  * Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
```

```
 * OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
 * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */
#ifndef _INCLUDE_SYS_SYSTEM_PROPERTIES_H
#define _INCLUDE_SYS_SYSTEM_PROPERTIES_H
#include <sys/cdefs.h>
__BEGIN_DECLS
typedef struct prop_info prop_info;
#define PROP_NAME_MAX   32
#define PROP_VALUE_MAX  92
/* Look up a system property by name, copying its value and a
** \0 terminator to the provided pointer.  The total bytes
** copied will be no greater than PROP_VALUE_MAX.  Returns
** the string length of the value.  A property that is not
** defined is identical to a property with a length 0 value.
*/
int __system_property_get(const char *name, char *value);
/* Set a system property by name.
**/
int __system_property_set(const char *key, const char *value);
/* Return a pointer to the system property named name, if it
** exists, or NULL if there is no such property.  Use
** __system_property_read() to obtain the string value from
** the returned prop_info pointer.
**
** It is safe to cache the prop_info pointer to avoid future
** lookups.  These returned pointers will remain valid for
** the lifetime of the system.
*/
const prop_info *__system_property_find(const char *name);
/* Read the value of a system property.  Returns the length
** of the value.  Copies the value and \0 terminator into
** the provided value pointer.  Total length (including
** terminator) will be no greater that PROP_VALUE_MAX.
**
** If name is nonzero, up to PROP_NAME_MAX bytes will be
** copied into the provided name pointer.  The name will
** be \0 terminated.
*/
int __system_property_read(const prop_info *pi, char *name, char *value);
/* Return a prop_info for the nth system property, or NULL if
** there is no nth property.  Use __system_property_read() to
** read the value of this property.
**
** This method is for inspecting and debugging the property
** system.  Please use __system_property_find() instead.
**
** Order of results may change from call to call.  This is
** not a bug.
*/
const prop_info *__system_property_find_nth(unsigned n);
__END_DECLS
#endif
```

Glosario.

ARM - Se refiere a la arquitectura de procesadores diseñados por la compañía británica "ARM holdings".

ASIC - Circuito integrado específico para una aplicación. Por sus siglas en inglés: "Application Specific Integrated Circuit".

Bluetooth - Estándar para comunicación inalámbrica de área personal. Su número de estándar es IEEE 802.15

Bootloader - Código que se ejecuta al encender un dispositivo de cómputo (antes de la ejecución del sistema operativo). Su función es iniciar el sistema operativo.

BSSID - Identificador único de un punto de acceso inalámbrico, generalmente la dirección física de la tarjeta inalámbrica del punto de acceso. Por sus siglas en inglés "Basic Service Set Identifier".

CPU - Unidad central de procesamiento de un sistema de cómputo. Por sus siglas en inglés: “Central Processing Unit”.

CSV - Formato de archivo en el que los datos se almacenan separados por comas. Por sus siglas en inglés: "Comma separated values"

EEPROM - Chip de memoria que puede ser borrado mediante la aplicación de una corriente eléctrica pero mantiene su información en ausencia de ésta. Por sus siglas en inglés: "Electrically Erasable Programmable Read Only Memory".

EXT3 - Tercera versión del sistema de archivos extendido.

EXT4 - Cuarta versión del sistema de archivos extendido

Fastboot - Es un modo de operación de Android en el que se puede modificar el sistema operativo que ejecuta un dispositivo sin necesidad de usar una herramienta adicional.

FAT - Sistema de archivos desarrollado para "MS-DOS" y "Windows" contiene una tabla que sirve como índice para los archivos almacenados en el dispositivo. Por sus siglas en inglés "File Allocation Table".

FAT32 - Evolución del sistema de archivos FAT permite el uso de discos de hasta 2 terabytes.

Flash - Memoria flash se refiere a un medio de almacenamiento que puede ser borrado y reprogramado en bloques en lugar de bytes.

HFS+ - Sistema de archivos jerárquico desarrollado por Apple Inc. Por sus siglas en inglés: "Hierarquical File System"

HTML5 - La quinta revisión del lenguaje de marcado para el desarrollo de páginas WEB.

IMEI - Es el número de identificación único de los dispositivos móviles GSM. Por sus siglas en inglés: "International Mobile Equipment Identity".

init - Es el primer proceso que se ejecuta en los sistemas operativos basados en Linux. Este proceso ejecutará otros que iniciaran servicios en el sistema.

Jailbreak - Es un procedimiento que permite a los usuarios de “iPhone”, “iPod touch” e “iPad” ejecutar aplicaciones distintas a las alojadas en la tienda de aplicaciones oficial de Apple.

Kernel - Parte del código del sistema operativo que se encarga de gestionar los recursos del sistema como administración de memoria, comunicación con dispositivos de entrada y salida, etc.

Malware - Software malicioso. Pretende llevar a cabo actividades en un equipo de cómputo sin el conocimiento o aprobación de su propietario.

Micro SD - Formato específico para tarjetas de memoria tipo flash. Miden 15mm x 11 mm x 0.7 mm. Generalmente se utiliza en teléfonos inteligentes, reproductores de música mp3 y sistemas de posicionamiento global.

MP3 - Formato de compresión de archivos de audio. Aprovecha redundancias estadísticas y perceptuales del oído humano para reducir el tamaño de un archivo de audio.

NTFS - Sistema de archivos diseñado para el sistema operativo Windows NT. Agregó mejoras al sistema FAT32 utilizado anteriormente como la posibilidad de especificar permisos de lectura y escritura en archivos y carpetas individuales.

OTP, memoria - Se refiere a un tipo de memoria que se puede grabar sólo una vez y que no puede ser borrada . Por sus siglas en inglés: "One Time Programable".

RAM - Memoria volátil de rápido acceso. Sirve al procesador para mantener datos y programas durante su ejecución.

Ramdisk - Es una zona de memoria RAM que se usa como almacenamiento secundario para un sistema. En Android se descomprime un sistema de archivos mínimo para que funcione el sistema operativo, se almacena en un espacio de la memoria RAM y desde ahí se ejecuta.

ROM - Tipo de memoria no volátil de sólo lectura. Permite almacenar datos que deben mantenerse en el sistema a pesar de que éste pierda su fuente de alimentación. En los sistemas de cómputo el BIOS se almacena en una memoria tipo ROM. Por sus siglas en inglés: "Read Only Memory".

SD - Es un formato de memoria flash generalmente utilizado en cámaras fotográficas digitales, asistentes personales digitales, computadoras portátiles, etc. Sus dimensiones son 32 mm x 24 mm x 2.1 mm

SDK - Conjunto de herramientas que permiten a un desarrollador crear aplicaciones para una plataforma especifica. Por sus siglas en inglés: "Software Development Kit".

SIM - Tarjeta de memoria que de forma segura almacena el IMSI, por sus siglas en inglés: “International Mobile Subscriber Identity” y las llaves necesarias para identificar a un usuario en la red celular. Por sus siglas en inglés: "Subscriber Identity Module".

SMS - Servicio de mensajes cortos para teléfonos celulares. Permite el envío y recepción de mensajes de hasta 160 caracteres. Por sus siglas en inglés: "Short message system".

SQL - Lenguaje de consulta estructurado. Principalmente se utiliza para la gestión y la comunicación con bases de datos. Por sus siglas en inglés: "Structured query language"

SSID - Es el nombre asociado a una red inalámbrica 802.11. No es necesariamente un identificador único. Por sus siglas en inglés: "Service Set Identifier".

Stick Pro Duo - Formato de memoria flash diseñada por la compañía Sony. Tiene un tamaño de 31 mm x 20 mm x 1.6 mm.

Token - Dispositivo físico o programa que ayuda a probar la identidad de un sujeto en un esquema de control de acceso.

USB - Estándar que define las propiedades de los cables, conectores y protocolos de comunicación que se emplean en un bus de comunicación entre computadoras y dispositivos electrónicos. Por sus siglas en inglés: "Universal Serial Bus".

Wifi - Tipo de red inalámbrica basada en el estándar IEEE 802.11 Permite el intercambio de datos entre dispositivos de computo. Por sus siglas en inglés: "Wireless Fidelity"

XML - Lenguaje extensible de marcado. Desarrollado por el "W3 Consortium" para permitir la descripción de información contenida en internet a través de formatos y estándares comunes. Por sus siglas en inglés: "Extensible Markup Language".

Referencias

(s.f.). Recuperado el Abril de 2013, de Mattica: http://www.mattica.com/

(Google), D. B. (2008). *Google IO.* Recuperado el 18 de Febrero de 2013, de Dlavik VM Internals Video/Slides: https://sites.google.com/site/io/dalvik-vm-internals

ACPO. (2007). Good Practice Guide for Computer-Based Electronic Evidence. England, Wales & Nireland.

AND Developers. (Octubre de 2011). *sbf*. Recuperado el Mayo de 2013, de And developers: http://and-developers.com/sbf

Android . (s.f.). *Content Provider Basics*. Recuperado el 18 de Febrero de 2013, de Android Developer: http://developer.android.com/guide/topics/providers/content-provider-basics.html

Android. (s.f.). *android.database.sqlite*. Recuperado el 17 de Febrero de 2013, de Android Developer: http://developer.android.com/reference/android/database/sqlite/package-summary.html

Android. (s.f.). *android.provider*. Recuperado el 18 de Febrero de 2013, de Adroid Developer: http://developer.android.com/reference/android/provider/package-summary.html

Android Developers . (s.f.). *Governance Philosophy*. Recuperado el Junio de 2013, de The Android Source Code: https://source.android.com/source/#governance-philosophy

Android Developers. (26 de Febrero de 2013). *<uses-sdk>.* Recuperado el 1 de Marzo de 2013, de Android Developers: http://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels

Android Developers. (s.f.). *Android Compatibility*. Recuperado el 2013, de Android Developers: https://source.android.com/compatibility/index.html

Android Developers. (s.f.). *Android Developers.* Recuperado el 5 de Febrero de 2012, de Android 2.2 Platform Highlights: http://developer.android.com/about/versions/android-2.2-highlights.html

Android Project. (s.f.). *Android Open Source Project*. Recuperado el 8 de Febrero de 2013, de Source Android: http://source.android.com/

Android. (s.f.). *Publishing Checklist for Google Play.* Recuperado el 15 de Enero de 2013, de Android Developers: http://developer.android.com/distribute/googleplay/publish/preparing.html

Android. (s.f.). *The AndroidManifest.xml File*. Recuperado el 17 de Febrero de 2012, de Android Developers: http://developer.android.com/guide/topics/manifest/manifest-intro.html

Android Zone. (Febrero de 2013). *¿Qué es? y ¿Cómo usar Odin?* Recuperado el Mayo de 2013, de Android Zone: http://androidzone.org/2013/02/tutorial-que-es-y-como-usar-odin3/

Animal Politico. (Enero de 2012). *Por uso de celular, 11% de accidentes automovilísticos en el DF*. Recuperado el Abril de 2013, de Animal Politico: http://www.animalpolitico.com/2012/01/por-uso-de-celular-11-de-accidentes-automovilisticos-en-el-df/#axzz2QA3oytrj

Ashe, D. (23 de Enero de 2013). *Mozilla Revelals first Firefox OS mobile devices.* Recuperado el 7 de Febrero de 2013, de Complex Tech: http://www.complex.com/tech/2013/01/mozilla-reveals-first-developer-firefox-os-devices

Association of Chief Police Officers. England Wales & Nireland. (5 de Julio de 2007). *Good Practice Guide for Computer-Based Electronic Evidence.* Recuperado el 21 de Febrero de 2013, de 7Safe: http://www.7safe.com/electronic_evidence/ACPO_guidelines_computer_evidence_v4_web.pdf

Auckland, R. K. (s.f.). *Trusted Platform Module and Privacy: Promises and Limitations.* Recuperado el Junio de 2013, de Department of Computer Science: http://www.cs.auckland.ac.nz/compsci725s2c/archive/termpapers/skim.pdf

Bay, D. J. (Noviembre de 2011). *Forensic Magazine*. Recuperado el Abril de 2013, de The Digital Forensics Cyber Exchange Principle: http://www.forensicmag.com/articles/2011/12/digital-forensics-cyber-exchange-principle#.UavK_9hMeQ4

Bornstein, D. (s.f.). *Technical University of Lodz.* Recuperado el 18 de Febrero de 2013, de Dalvik VM Internals: http://fiona.dmcs.pl/podyplomowe_smtm/smob3/Presentation-Of-Dalvik-VM-Internals.pdf

Boxall, A. (16 de Enero de 2012). *Samsung's Bada goes boom, gets integrated into Tizen.* Recuperado el 6 de Enero de 2013, de Digital Trends: http://www.digitaltrends.com/mobile/samsungs-bada-goes-boom-gets-integrated-into-tizen/

Carrier, B. D. (2009). Digital Forensics Works. *IEEE Security & Privacy*, 26-29.

Código Federal de Procedimientos Penales. (s.f.).

Commons, C. (s.f.). *Creative Commons v2.5*. Recuperado el Mayo de 2013, de http://creativecommons.org/licenses/by/2.5/

cpio(1) - Linux man page. (s.f.). Recuperado el Mayo de 2013, de Android Documentation: http://linux.die.net/man/1/cpio

Darren Quick, Mohammed Alzaabi. (2011). Forensic Analysis of the Android File system YAFFS2. *Australian Digital Forensics Conference* (pág. 11). University of South Australia.

Defreez, D. (2012). *Android Privacy Through Encryption.* Ashland, Oregon: Department of Computer Science.

Deloitte. (s.f.). *Análisis y Computación Forense*. Recuperado el Abril de 2013, de Deloitte: http://www.deloitte.com/view/es_MX/mx/servicios/asesoria-financiera/61a220adf64db210VgnVCM3000001c56f00aRCRD.htm

Developers, A. (s.f.). *Android Developers*. Recuperado el Marzo de 2013, de Using the Backup API: http://developer.android.com/training/cloudsync/backupapi.html

DLS, A. (7 de Mayo de 2013). *Android Wiki*. Recuperado el 30 de Mayo de 2013, de HOWTO: Unpack, Edit, and Re-Pack Boot Images: http://android-dls.com/wiki/index.php?title=HOWTO:_Unpack%2C_Edit%2C_and_Re-Pack_Boot_Images#Structure_of_boot_and_recovery_images

Dummann, K. (2010). *android_device_samsung_galaxys / recovery.rc*. Recuperado el Junio de 2013, de GitHub: https://github.com/coolya/android_device_samsung_galaxys/blob/master/recovery.rc

Ehringer, D. (Marzo de 2010). The Dalvik virtual machine architecture.

Elenkov, N. (Febrero de 2013). *Secure USB debugging in Android 4.2.2*. Recuperado el Junio de 2013, de Android Explorations: http://nelenkov.blogspot.mx/2013/02/secure-usb-debugging-in-android-422.html

Enck, W. (Jan/Feb 2009). Understanding Android Security. *IEEE Security & Privacy*, 50-57.

Ernst & Young. (s.f.). *Servicios de Aseguramiento*. Recuperado el Abril de 2013, de Ernst & Young: http://www.ey.com/MX/es/Services/Assurance/Fraud-Investigation---Dispute-Services

Flurry. (s.f.). Obtenido de Flurry: http://www.flurry.com/

Free Software Foundation. (2010). *GNU's home page*. Recuperado el 2013, de cpio: http://www.gnu.org/software/cpio/

Google - Android. (s.f.). *android-source-browsing*. Recuperado el Junio de 2013, de Google Project Hosting: https://code.google.com/p/android-source-browsing/source/browse/?repo=platform--bionic&r=9ec0f03a0d0b17bbb94ac0b9fef6add28a133c3a

Google. (s.f.). *Google Glass Project*. Recuperado el Junio de 2013, de Google Glass Start: http://www.google.com/glass/start/

Grandou, G. (2010). *Gitorious*. Recuperado el Mayo de 2013, de Readme abootimg in AC100: http://gitorious.org/ac100/abootimg/blobs/master/README

Henry F. Fradella, L. O. (2004). The Impact of Daubert on Forensic Science. *Pepperdine Law Review, 31*(2).

Hoog, A. (2011). *Android Forensics. Investigation, analysis and mobile security for Google Android.* Massachusetss: Syngress.

IDC. (1 de Noviembre de 2012). *IDC Press Release*. Obtenido de IDC WorldWide Mobile Phone Tracker: http://www.idc.com/getdoc.jsp?containerId=prUS23771812#.UPBIZayYIfc

IEEE. (Diciembre de 2012). *Official IEEE Std. 1149.1 Standard Working Group*. Recuperado el Marzo de 2013, de IEEE Grouper: http://grouper.ieee.org/groups/1149/1/

International Organization on Digital Evidence IOCE. (Abril de 2000). *FBI - Digital Evidence: Standards and Principles by SWDGE and IOCE Forensic Science Communications.* Recuperado el Marzo de 2013, de Federal Bureau of Investigation: http://www.fbi.gov/about-us/lab/forensic-science-communications/fsc/april2000/swgde.htm/

J, P. (Mayo de 2009). *What Percentage of iPhones Are Jailbroken? Less Than 10 Percent?* Recuperado el Abril de 2013, de iSource.com: http://isource.com/2009/05/27/what-percentage-of-iphones-are-jailbroken-less-than-10-percent/

Justice, U. D. (2011). *Test Results for Forensic Media Preparation Tool: dc3dd: Version 7.0.0.* National Institute of Justice.

Kevin Mandia, C. P. (2001). *Incident Response: Investigating Computer Crime.* McGraw Hill Osborne Media.

Kingsley-Hughes, A. (Febrero de 2013). *Why do iPhone and iPad users jailbreak? Freedom* . Recuperado el Abril de 2013, de ZDNet: http://www.zdnet.com/why-do-iphone-and-ipad-users-jailbreak-freedom-7000011114/

KPMG. (s.f.). *Forensic*. Recuperado el Abril de 2013, de KPMG: http://www.kpmg.com/mx/es/servicios/advisory/rc/forensic/paginas/default.aspx

Leyden, J. (14 de Septiembre de 2012). *Smartmobe Wi-Fi blabs FAR TOO MUCH about us, warn experts.* Recuperado el 15 de Enero de 2013, de The Register: http://www.theregister.co.uk/2012/09/14/smartphone_tracking_research/

Lin, F. (2009). Operating System Battle in the Ecosystem of Smartphone Industry. *International Symposium on Information Engineering and Electronic Commerce* (pág. 5). IEEE Computer Society.

Loup, J. (2003). *GZIP*. Recuperado el Mayo de 2013, de Gzip Home Page: http://www.gzip.org/#intro

Mark Reith, C. C. (2002). An Examination of Digital Forensic Models. *International Journal of Digital Evidence, 1*(3).

Martinez, R. (1998). *Distribuciones 'Live CD'*. Recuperado el 2013, de www.linux-es.org/livecd

McHugh, M. (1 de Noviembre de 2011). *Ubuntu's going mobile: Will it survive?* Recuperado el 6 de Enero de 2013, de Digital Trends: http://www.digitaltrends.com/mobile/ubuntus-going-mobile-will-it-survive/

Méndez, M. A. (3 de Enero de 2013). *Ubuntu mobile OS estrena el año de los aspirantes al smartphone.* Recuperado el 6 de Febrero de 2013, de Gizmodo: http://es.gizmodo.com/ubuntu-mobile-os-estrena-el-ano-de-los-aspirantes-al-smartphone-1236310

Micron. (4 de Octubre de 2006). *NAND Flash 101: An Introduction to NAND Flash and How to Design it in to your next product.* Recuperado el 5 de Febrero de 2013, de http://download.micron.com/pdf/technotes/nand/tn2919.pdf

Microsoft Research. (1993). *Simon*. Obtenido de Buxton Collection: http://research.microsoft.com/en-us/um/people/bibuxton/buxtoncollection/detail.aspx?id=40

Milward, S. (Marzo de 2013). *Report: Jailbreaking Declining in China, Now Down to 32.3% of iOS Devices*. Recuperado el Abril de 2013, de TechInAsia: http://www.techinasia.com/jailbreaking-declining-china-32-percent-in-february-2013/

National Institute of Justice U.S. (Abril de 2008). *Electronic Crime Scene Investigation: A Guide for First Responders, Second Edition.* Recuperado el Marzo de 2013, de National Institute of Justice: www.ojp.usdoj.gov/nij

National Institute of Standards and Technology. Technology Administration U.S. Department of Commerce. (Mayo de 2007). *Guidelines on Cell Phone Forensics. Special Publication 800-101.* Recuperado el 22 de Febrero de 2013, de Recommendations of the National Institute of Standards and Technology: http://csrc.nist.gov/publications/nistpubs/800-101/SP800-101.pdf

Noyes, K. (3 de Enero de 2013). *Samsung: Tizen Linux phones will arrive in 2013.* Recuperado el 6 de Febrero de 2013, de PC World: http://www.pcworld.com/article/2023649/samsung-tizen-linux-phones-will-arrive-in-2013.html

Policia Ciberdelincuencia. (s.f.). Recuperado el Abril de 2013, de Policia DF. Ciudad de México: http://www.ssp.df.gob.mx/notasnewsslider/Pages/Polic%C3%ADa-.aspx

Procuraduria General de la República. (s.f.). *Informática y Telecomunicaciones* . Recuperado el Abril de 2013, de PGR: http://www.pgr.gob.mx/periciales/PlantillaHTML/especialidades.asp?id=5

PwC. (s.f.). *Fraude y disputas comerciales*. Recuperado el Abril de 2013, de PwC: http://www.pwc.com/mx/es/servicios-asesoria-financiera/fraude-disputas-comerciales.jhtml

RarLab Winrar. (s.f.). *Winrar*. Recuperado el Mayo de 2013

Reardon, M. (17 de Febrero de 2009). *Universal Cell phone chargers coming soon.* Obtenido de CNet : http://reviews.cnet.com/8301-13970_7-10165603-78.html

Ruiz, F. (4 de Octubre de 2012). *'FakeInstaller' Leads the Attack on Android Phones.* Recuperado el 15 de Enero de 2013, de McAfee Blog Central: http://blogs.mcafee.com/mcafee-labs/fakeinstaller-leads-the-attack-on-android-phones

Scott, D. (2 de Julio de 2012). *Mozilla Announces Carrier and Hardware Partners for Firefox OS.* Recuperado el 6 de Enero de 2013, de Complex Tech: http://www.complex.com/tech/2012/07/mozilla-announces-carrier-and-hardware-partners-for-firefox-os

Sell My Application. (s.f.). *How To Submit a iPhone App to the Appstore.* Recuperado el 15 de Enero de 2013, de Sel lMy Application - The Official App Code Marketplace: http://www.sellmyapplication.com/how-to-submit-a-iphone-app-to-the-appstore/

Sony Mobile Communications AB. (2013). *Unlocking the boot loader*. Recuperado el Mayo de 2013, de Sony. Developer World: http://unlockbootloader.sonymobile.com/

SQLite. (s.f.). Recuperado el 17 de Febrero de 2013, de SQLite: http://www.sqlite.org/

Suggy, C. A. (11 de Septiembre de 2012). *44Con Trip Report.* Recuperado el 15 de Enero de 2013, de Security Geek: http://www.securityg33k.com/blog/?p=629

The Local - Germany's news in English. (14 de Noviembre de 2012). *Bank accounts emptied by phone Trojan.* Recuperado el 2013 de Enero de 15, de The Local: http://www.thelocal.de/sci-tech/20121114-46169.html#.UPXH6_KYIfc

Tim Bray. (9 de Diciembre de 2010). *Saving Data Safely.* Recuperado el 4 de Marzo de 2013, de Android Developers Blog: http://android-developers.blogspot.mx/2010/12/saving-data-safely.html

Timothy Vidas, C. Z. (s.f.). *Towards a General Collection Methodology for Android Devices.* Carnegie Mellon .

Trusted Computing Group. (2013). *Trusted Platform Module*. Recuperado el Mayo de 2013, de Trusted Computing Group Developers: http://www.trustedcomputinggroup.org/developers/trusted_platform_module/

UNAM CERT. (Septiembre de 2005). *Experiencias de análisis forense en México.* Recuperado el Abril de 2013, de RedIris ES: http://www.rediris.es/cert/doc/reuniones/af05/Experiencias_analisis_forense_en_mexico.pdf

Welte, H. (8 de Agosto de 2010). Anatomy of contemporary GSM cellphone hardware.

WinZip International. (s.f.). *Descarga de WinZip*. Recuperado el Mayo de 2013, de http://www.winzip.com/es/downwz.htm

XDA Developers. (Diciembre de 2011). *[TUTORIAL] Setting up and Compiling JB and ICS from AOSP*. Recuperado el Mayo de 2013, de XDA Developers: http://forum.xda-developers.com/showthread.php?p=20194713#post20194713

XDA Developers. (Agosto de 2011). *[TWEAKS][SCRIPTS] Collection of 'em all - build.prop; init.d; etc.* Recuperado el Mayo de 2013, de XDA Forum: http://forum.xda-developers.com/showthread.php?t=1227269

XDA Developers. (Mayo de 2011). *Extract initramfs from zImage*. Recuperado el Mayo de 2013, de Forum XDA: http://forum.xda-developers.com/wiki/Extract_initramfs_from_zImage

XDA Developers. (Julio de 2012). *[Tutorial] Compile JB on Ubuntu*. Recuperado el Mayo de 2013, de XDA Developers: http://forum.xda-developers.com/showthread.php?t=1762641

Xperia Gamer. (2012). *XPERIA Gamer*. Recuperado el Mayo de 2013, de Installing and Using FlashTool: http://www.xperiagamer.com/Beginners-Guides/installing-and-using-flashtool.html