ELABORACIÓN DE UNA METODOLOGÍA PARA LA REALIZACIÓN DEL ANÁLISIS FORENSE EN DISPOSITIVOS MÓVILES BASADOS EN *ANDROID*

TRABAJO FINAL DE MÁSTER

Alumno: José Antonio Pérez Salvador

Director del proyecto: Marco Antonio Lozano Merino

Universidad/Empresa: UOC/INCIBE

Programa: MISTIC

Fecha de entrega: 09/01/2017

Enero de 2017

A mis padres
que modelaron los
primeros veinte años
de mi vida.

Resumen

En esta memoria se analizan los métodos y técnicas existentes para el análisis de dispositivos móviles en el SO *Android*, con el objetivo de crear una metodología para el análisis forense de dichos terminales.

Durante este trabajo también veremos cómo afectan las medidas de seguridad destinadas a proteger los datos de los usuarios al análisis forense (cifrado de disco, borrado remoto y bloqueo por código de un terminal, principalmente) y cómo podemos maximizar los resultados del mismo.

Uno de los requisitos para la validez de los resultados de un análisis forense es el seguimiento de una metodología validada y aceptada por la comunidad.

Por tanto, en este trabajo se definen y analizan las principales etapas que se llevan a cabo durante un proceso de análisis forense.

Además, se describen los contenidos y estructura del informe forense. También se hace un repaso de los métodos más utilizados actualmente para la adquisición de datos en la plataforma *Android*.

En relación a la tarea de análisis, el análisis de datos adquiridos, veremos cómo y dónde se almacenan datos como llamadas, contactos y mensajes en la plataforma *Android*.

Finalmente, veremos cómo se puede recuperar y analizar el espacio borrado del almacenamiento del dispositivo.

Por tanto, con el desarrollo de esta metodología se pretende lo siguiente:

- Realizar adquisiciones forenses de terminales móviles *Android*.
- Localizar e interpretar la información relevante contenida en un dispositivo móvil.
- Identificar las etapas con las que cuenta un análisis forense y las tareas de las que se compone cada una de ellas.
- Conocer el estado del arte para la adquisición e interpretación de evidencias existentes para la plataforma móvil *Android*.
- Completar procesos de análisis forense y plasmar sus resultados en un informe forense.

Índice de contenidos

Lista de figuras

Lista de tablas

1. INTRODUCCIÓN

1.1. CONTEXTO Y JUSTIFICACIÓN DEL TFM

La irrupción de los dispositivos móviles ha supuesto un cambio en la manera que teníamos de interactuar con *Internet* hasta el momento, y ha propiciado que los cibercriminales hayan dirigido sus ataques hacia esta clase de dispositivos.

El SO *Android* es el más popular del mercado, pero también es el que cuenta con un mayor número de riesgos y amenazas. Por esa razón, y porque no existe un modelo único que nos permita llevar a cabo un análisis forense en dispositivos móviles, debido a que los procedimientos y las normas para su análisis aún se encuentran en desarrollo, se ha propuesto una metodología que se adapte a las condiciones del ámbito tecnológico actual, y que permita resolver problemas que no se dan en el análisis forense tradicional de ordenadores de sobremesa y soportes de datos convencionales.

1.2. OBJETIVOS DEL TFM

El objetivo general del proyecto consiste en elaborar una metodología de análisis forense para la plataforma de dispositivos móviles con SO *Android*.

En cuanto a los objetivos específicos a alcanzar en este trabajo se incluyen:

- Estudiar la arquitectura interna del SO *Android*.
- Identificar los riesgos y amenazas más recientes y su impacto, en el SO *Android*.
- Analizar las metodologías existentes actualmente para el análisis forense digital en dispositivos móviles, y conocer las herramientas para realizar las investigaciones.
- Desarrollar una metodología orientada al análisis forense en dispositivos móviles basados en *Android*.
- La preparación de un entorno de pruebas, donde analizar las vulnerabilidades y realizar el análisis forense.
- Realizar el informe pericial de un caso de estudio ficticio en el que se aplique la metodología propuesta.

1.3. METODOLOGÍA DEL TFM

Uno de los requisitos para la validez en el resultado del análisis forense es el seguimiento de una metodología validada y aceptada por la comunidad.

Por tanto, en esta metodología se van a definir las etapas que se llevan a cabo durante un análisis en un proceso forense. Además, se describirán los contenidos y la estructura del informe forense.

También se describirán los métodos más utilizados actualmente para la adquisición de datos en el análisis forense de la plataforma móvil *Android*.

En relación al análisis de datos adquiridos, se van a identificar las localizaciones y forma en la que se almacenan los datos, como llamadas, contactos y mensajes en la plataforma

Android.

Finalmente, también veremos como se puede recuperar y analizar el espacio borrado del almacenamiento del dispositivo.

Lo que se pretende con el diseño de la metodología es, por tanto:

- Poder realizar adquisiciones forenses en terminales móviles *Android*.
- Localizar e interpretar la información relevante contenida en un dispositivo móvil.
- Identificar las etapas con las que cuenta un análisis forense y las tareas de que se compone cada una de ellas.
- Conocer el estado del arte para la adquisición e interpretación de evidencias existentes en *Android*.
- Completar procesos de análisis forense y plasmar sus resultados en un informe forense.

Por tanto, para lograr los objetivos propuestos, se ha propuesto una metodología que consta de seis fases, algunas de las cuales se pueden realizar en paralelo. Las fases planteadas se presentan a continuación:

- Fase de preparación.
- Fase de adquisición.
- Fase de gestión de evidencias.
- Fase de examen.
- Fase de análisis.
- Fase de presentación.

1.4. LISTADO DE TAREAS

Entre las tareas a realizar durante el proyecto se encuentran las siguientes:

- Comienzo.
  - Propuesta de trabajo.
  - Determinar el ámbito del proyecto.
  - Calcular los tiempos y fechas de entrega.
  - Definir recursos preliminares.
  - Obtener recursos preliminares.
  - Elaboración del plan de trabajo.
  - Entrega del plan de trabajo.
- Elaboración de la metodología.
  - Búsqueda y recopilación de información.
  - Selección y estudio de la tecnología.
  - Análisis y especificación de las fases de peritaje.

    - Borrador de las especificaciones preliminares.
    - Revisar las especificaciones.
    - Actualización parcial de la memoria de trabajo.
    - Entrega de la PEC 1.
- Preparación del laboratorio virtual.
    - Determinar la estrategia a seguir.
    - Obtener los recursos necesarios.
    - Instalar y configurar las herramientas necesarias.
    - Análisis forense de un dispositivo *Android*.
    - Actualización parcial de la memoria de trabajo.
    - Entrega de la PEC 2.
- Entrega final.
    - Documentación y revisión final de la memoria de trabajo.
    - Elaboración del video de presentación.
    - Entrega final.

1.5. RIESGOS PRELIMINARES

Todo depende de las circunstancias. Por ello, se establece una lista de riesgos que abarca la realidad del proyecto, aunque afortunadamente no todos darán lugar a riesgos durante el mismo. A continuación, se detallan algunos de los más importantes:

Id	Riesgo	Tipo de riesgo	Clase
1	Proyecto demasido complicado.	Tamaño del proyecto	Organización
2	Calendario muy ajustado. Riesgo de incumplimiento.	Calendario del proyecto	Organización
3	Proyecto sobredimensionado.	Calendario del proyecto	Organización
4	Perímetro funcional no determinado.	Cierre del perímetro funcional	Funcional
5	Perímetro demasiado amplio.	Cierre del perímetro funcional	Funcional
6	Funcionalidades ya cubiertas por estándares existentes.	Cierre del perímetro funcional	Funcional
7	Procesos aparentemente parecidos a los de una metodología existente, pero cuyas diferencias implican replantearse los fundamentos.	Entendimiento de los procesos	Funcional
8	Funcionalidades no necesarias.	Adhesión	Funcional
9	Funcionalidades ausentes.	Adhesión	Funcional
10	Disponibilidad de las herramientas.	Uso de herramientas de terceros	Técnica

Id	Riesgo	Tipo de riesgo	Clase
11	Términos de licencias de las herramientas incompatibles con las características del proyecto.	Uso de herramientas de terceros	Técnica
12	Ausencia de conocimiento dentro del proyecto.	Conocimientos de la tecnología	Técnica
13	Herramientas defectuosas o no adaptadas	Conocimientos de las tecnologías	Técnica
14	Ausencia de documentación o de soporte.	Plataforma de desarrollo	Técnica

Cuadro 1: Riesgos preliminares.

Sobre los riesgos detectados, se ha realizado un plan de contingencia. Es decir, se ha confeccionado una tabla que resuma los riesgos identificados y su peso (valores), a partir de los cuales se ha podido evaluar de manera razonable la probabilidad de bloqueo o fracaso del proyecto.

Id	Acción de mitigación	Probabilidad (1-5)	Impacto	Total
1, 3, 5	No complicarse en el proyecto. No opacar el trabajo real que se debe hacer. No perseguir los riesgos irrelevantes y evitar sobrecargar el plan con información innecesaria.	Medio-alta (4)	Medio-alto (5)	20
2	Trabajo rápido. Ser concisos. Saltarse los pasos que tienen poco impacto en el proyecto.	Medio-alta (4)	Medio-alto (5)	20
4, 8	Evaluar el trabajo real que se debe hacer. Eliminar la funcionalidad innecesaria.	Media (4)	Medio-bajo (4)	16
6, 7	Identificar qué actividad ya está cubierta. Innovar. Ser crea-Investigar por *Internet*. Ser proactivo.	Media (4)	Medio-bajo (4)	16
9	Evaluar el trabajo real que se deba hacer. Añadir la funcionalidad necesaria.	Media (4)	Medio-bajo (4)	16

Id	Acción de mitigación	Probabilidad (1-5)	Impacto	Total
10, 11, 13	Descargar las herramientas necesarias del sitio oficial. Búsqueda de información en foros especializados. Buscar herramientas alternativas. Investigar.	Alta (5)	Alto (5)	25
12, 14	Documentarse. Buscar información en *Internet*. Visualizar videotutoriales.	Media (4)	Medio-bajo (4)	16

Cuadro 2: Resumen de riesgos identificados con su peso.

1.6. PLANIFICACIÓN TEMPORAL DETALLADA

Debido a la naturaleza del trabajo a realizar, la planificación viene determinada por la fecha de inicio, que comienza el 21 de septiembre de 2016, la entrega inicial del plan de trabajo, la entrega parcial de dos PEC (Pruebas de Evaluación Continua) de que consta la asignatura, y de una entrega final de carácter esencial e improrrogable, que tiene como fecha de cierre el 9 de enero de 2017.

Por tanto, la estimación se impone, dado que las fechas de entrega se imponen también. En este sentido, se ha diseñado la solución buscando adecuar el tiempo de realización de cada una de las entregas, para alcanzar los objetivos fijados y ajustar el perímetro del proyecto evitando el riesgo.

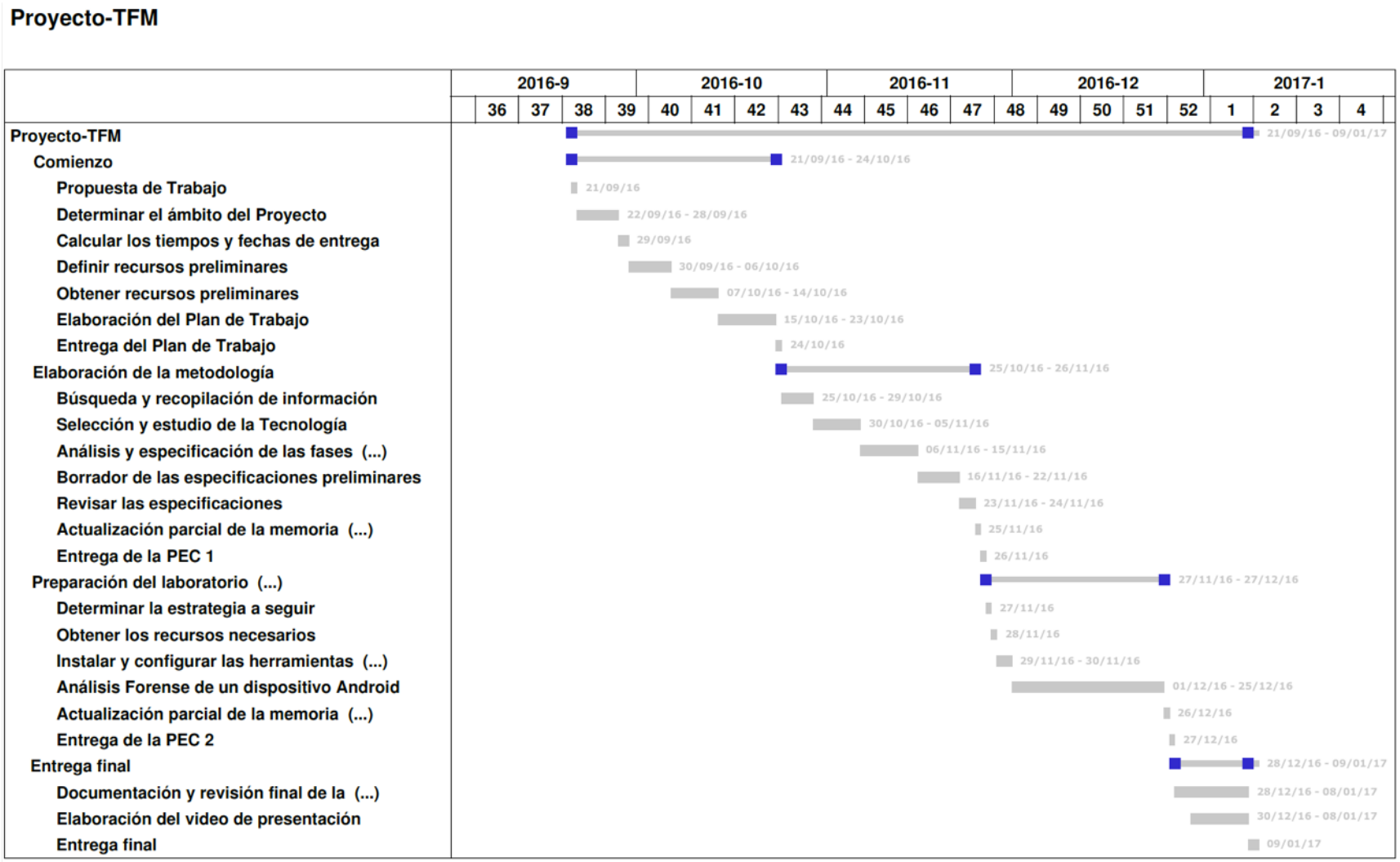


Figura 1: Planificación temporal detallada.

1.7. ORGANIZACIÓN DE LA MEMORIA DEL PROYECTO

El TFM divide la memoria en nueve capítulos ordenados cronológicamente, la cual presenta la siguiente estructura:

Capítulo 1 - Introducción. En este primer capítulo se hace una breve introducción al TFM, haciendo una descripción del contexto y justificación del trabajo, los objetivos, el enfoque y método elegido, los productos obtenidos, la planificación de las tareas a realizar, así como de una lista amplia, pero no exhaustiva, de posibles riesgos potenciales que pueden afectar al proyecto.

Capítulo 2 - Pasos previos e introducción a *Android*. En este segundo capítulo, se muestra la situación actual de la informática forense y se realiza un estudio en profundidad del SO *Android*: el problema de la fragmentación en *Android*, su arquitectura, el sistema de archivos, el proceso de arranque y la gestión de la seguridad. Finalmente, se analizan las diferencias entre el análisis forense tradicional y el análisis forense en dispositivos móviles.

Capítulo 3 - Metodología propuesta para el análisis de dispositivos móviles basados en *Android*. El tercer capítulo se centra principalmente en el desarrollo de cada una de las seis fases de que consta la metodología propuesta.

Capítulo 4 - Laboratorio. En general, este capítulo se basa en: la preparación de un entorno virtual donde realizar el proceso forense; la obtención de las herramientas necesarias; la captura de imágenes de dispositivos móviles; el proceso de adquisición, extracción y análisis de la información del dispositivo móvil y recomendaciones sobre cómo realizar un informe forense.

Además de lo anterior, se realiza un caso de estudio ficticio: análisis forense de un dispositivo basado en *Android*. En este caso, se plantea una situación no real, en la que se analiza un dispositivo móvil incautado siguiendo los pasos de la metodología propuesta.

Capítulo 5 - Conclusiones y trabajo futuro. En este capítulo se analizará el trabajo realizado, las dificultades que plantea el análisis de este tipo de dispositivos, las posibles ampliaciones del trabajo, etc.

Capítulo 6 - Recomendaciones. El séptimo capítulo se centra esencialmente en recomendaciones de relacionadas con el proceso de análisis forense en dispositivos *Android*.

Capítulo 7 - Referencias. Bibliografía, *webgrafía* y otras fuentes de información.

Capítulo 8 - Glosario. Este penúltimo capítulo se centra en el glosario, el cual está concebido a modo de complemento para ayudar a comprender los tecnicismos utilizados en este trabajo.

Capítulo 9 - Anexos. Este capítulo podría incluir toda la documentación adicional de la memoria si fuese necesario.

1.8. BREVE SUMARIO DE LOS PRODUCTOS OBTENIDOS

Para terminar el Trabajo Final de Máster, hay que concretar la generación de un conjunto de piezas intermedias. Estas piezas son los entregables parciales, que son las piezas que forman el "resultado" del proyecto. Entre los productos a entregar se encuentran:

- La planificación del proyecto.

- Un entregable correspondiente a la primera prueba de evaluación continua (PEC 1).
- El entregable de la PEC 2.

Finalmente, entre los productos incluidos en la entrega final figuran los siguientes:

- La memoria del trabajo, la cual se va a realizar en LaTeX.
- Un video demostrativo del trabajo realizado.

2. PASOS PREVIOS E INTRODUCCIÓN

Según el principio de intercambio de *Locard*, cualquier interacción física entre dos objetos implica la transferencia de material de uno a otro.

Este principio fue desarrollado por un criminalista francés llamado *Edmond Locard* en 1934.

Este principio es el precursor del análisis forense como campo científico, donde:

- El criminal deja pruebas en el escenario durante la consecución del delito.
- El perito puede modificar las pruebas durante el proceso de adquisición o análisis.

Figura 2: Principio de *Locard*.

Como referencia indicar que el término analista forense no existe como tal en la legislación española. Por tanto, es preferible utilizar en su lugar la denominación de perito forense.

2.1. INTRODUCCIÓN AL ANÁLISIS FORENSE

En entornos informáticos, el análisis forense es el conjunto de procedimientos de recopilación y análisis de evidencias que se realizan con el fin de conocer las causas de un incidente en el que se ve envuelto un sistema informático.

Dependiendo del tipo de incidente, el proceso de análisis forense se realiza con diferentes objetivos:

- Si el incidente está relacionado con un hecho delictivo e intervienen las fuerzas de seguridad y cuerpos judiciales, el objetivo del análisis forense es la presentación de las pruebas en un tribunal.
- Si se trata de un incidente de seguridad informática, los diferentes procedimientos ejecutados, como el análisis, tendrán como objetivo la respuesta eficiente ante el incidente.

En este sentido, el proceso de análisis forense llevado a cabo dentro de una organización en caso de incidente informático es compatible con el proceso legal que se pueda derivar del propio incidente, y en muchos casos ayuda a esclarecer los hechos y atribución del mismo.

2.2. OBJETIVOS DEL ANÁLISIS FORENSE

De forma general, los objetivos del análisis forense se dividen en dos:

- Conocer lo que ha sucedido realmente en un sistema informático. Según el tipo de investigación, los hechos a estudiar pueden ser diferentes:
  - En el caso de una intrusión informática, conocer el procedimiento que se llevó a cabo para acceder al sistema y el alcance de los daños generados.
  - En delitos que no han sido ejecutados mediante medios informáticos, ayuda a obtener información sobre la propietaria del dispositivo (por ejemplo, para el caso de una coartada, etc.).
- Conocer al responsable de cada acción o evento descubierto durante el análisis. Por cada uno de los hechos identificados, es necesario identificar al responsable del mismo:
  - Para delitos cometidos mediante medios informáticos, esta tarea es en ocasiones muy compleja, debido a la existencia de técnicas que permiten conservar el anonimato de los atacantes (utilización de *botnets*, la red *Tor*, etc.).

2.2.1. MOTIVACION DEL ANÁLISIS FORENSE

La informática forense es una parte integral de los procedimientos de respuesta ante incidentes:

- Se aplica después de que un delito o incidente de seguridad haya sucedido.
- Permite reconstruir los sucesos o acciones que han llevado a un incidente de seguridad para mejorar los procesos de protección existentes en una organización.

La informática forense también se puede utilizar de forma activa en el contexto de una organización para:

- Auditar las propiedades de seguridad de un sistema (mantenimiento de privacidad, envío de datos sensibles, etc.).
- Revisar el cumplimiento de normativas y estándares de seguridad.
- Aseguar que se cumplen los procedimientos para la destrucción de datos sensibles en el sistema.

2.2.2. PARTICULARIDADES DEL ENTORNO MÓVIL

Uno de los principales problemas de la informática forense es que debe adaptarse a la constante aparición de nuevos dispositivos:

- Durante sus inicios, la informática forense trataba delitos que se cometían a través de medios informáticos. Por lo tanto, las investigaciones se centraban especialmente en estaciones de trabajo, servidores y redes.
- La aparición de teléfonos móviles ofreció nuevos datos (*SMS* y llamadas) y empezó a acercar la informática forense a delitos que se suceden fuera de los medios telemáticos.

- La aparición de los *smartphones* amplió el abanico de información a recolectar de un dispositivo (mensajes, correos electrónicos, localización, etc.).
- Los nuevos dispositivos conectables (*wearables*, vehículos, domótica, etc.) ofrecen aún más información que puede ser de gran importancia durante una investigación judicial.

Cada uno de estos tipos de dispositivos tiene un conjunto de particularidades que hacen del análisis forense una tarea compleja y dificultosa.

En particular, el análisis de los entornos móviles y *smartphones* supone un desafío por las siguientes razones:

- **Diferentes sistemas operativos**: pese a que *Android* es el SO móvil de uso mayoritario, existen otros que también tienen una importante cuota de mercado y que, por tanto, deben ser conocidos en profundidad, para poder llevar a cabo el proceso de toma de evidencias. Algunos de ellos son: *iOS*, *Windows Phone* y *BlackBerry OS*.
- **Consideraciones legales**: durante el proceso es fundamental cumplir en todo momento con la normativa vigente, con el fin de mantener la validez legal de las pruebas en caso de que se requiera.
- **Técnicas anti-forense**: al igual que sucede con otros dispositivos, como en el caso de los ordenadores, es posible realizar diferentes acciones para dificultar la identificación de pruebas en un proceso forense, como por ejemplo: la destrucción, ocultación o falsificación de las evidencias.

Además, los sistemas operativos ofrecen por defecto sistemas de protección y cifrado que dificultan la adquisición y análisis de datos:

- El bloqueo por código de un terminal evita el acceso al dispositivo, incluso por cable en algunos sistemas.
- El borrado remoto permite eliminar todas las pruebas de un dispositivo sin tener acceso físico al mismo.
- El cifrado de disco imposibilita la lectura de las memorias a través del acceso físico al *chip*.

Por útlimo, podemos añadir que existen millones de aplicaciones disponibles para cada dispositivo, cada una con un mecanismo de almacenamiento de información diferente.

2.2.3. EVIDENCIAS RELEVANTES EN EL ENTORNO MÓVIL

Casi cualquier elemento digital se puede considerar una evidencia. Algunos ejemplos son:

Figura 3: Evidencias relevantes en el entorno móvil.

2.2.4. ALMACENAMIENTO DE EVIDENCIAS

Es imprescindible definir los métodos adecuados para el almacenamiento y etiquetado de las evidencias. Una vez que se cuenta con todas las evidencias del incidente, es necesario conservarlas intactas. Para ello, se debe seguir el siguiente proceso:

- Deben realizarse varias copias de la evidencia. Como primer paso, se debe realizar dos copias de las evidencias obtenidas, generar también una suma de comprobación de la integridad de cada copia mediante el empleo de funciones *hash*. Incluir estas firmas en la etiqueta de cada copia de la evidencia, etiquetando la fecha y la hora de creación de la copia, poniendo un nombre a cada copia, por ejemplo “Copia A” y “Copia B”, para distinguirlas claramente del original.
- Deben almacenarse en un lugar seguro y a salvo de accesos no autorizados.
- Deben garantizarse los sistemas necesarios para la preservación de las mismas.
- Debe establecerse una cadena de custodia. Es en la cadena de custodia donde se establecen las responsabilidades y controles de cada una de las personas que manipulen la evidencia. Para ello, se debe preparar un documento en el que se registren los datos personales de todos los implicados en el proceso de manipulación de las copias, desde que se tomaron hasta su almacenamiento.
- El informe y la cadena de custodia debe ser completa, correcta, auténtica y convincente, para que en caso de proceso legal sea admitida en un juzgado.

Figura 4: Etiquetado de evidencias.

2.3. INTRODUCCIÓN AL SISTEMA OPERATIVO *ANDROID*

En esta introducción se ofrece una versión general del SO (sistema operativo) *Android*. En concreto:

- Arquitectura del SO.
- Librerías disponibles.
- Sistema de ficheros y particiones existentes.
- Estructura de un binario.
- Componentes de una aplicación.
- Comunicación entre aplicaciones.
- Proceso de compilación y creación de una aplicación.

2.3.1. ¿QUÉ ES *ANDROID*?

Android es un SO móvil originalmente desarrollado por *Android Inc.* (empresa fundada por *Andy Rubin* [antiguo empleado de *Apple*] y comprada por *Google* en 2005), que se basa en una versión modificada de *Linux*. Se desarrolló inicialmente por una *startup* (un negocio con amplias posibilidades de crecimiento) del mismo nombre, *Android*, *Inc.* En 2005, como parte de su estrategia para entrar en el mundo móvil, *Google* compró *Android* y se hizo cargo de su trabajo de desarrollo (así como de su equipo).

Su desarrollo actual está gestionado por el *Android Open Source Project* (*AOSP*), mantenido por *Google* y promocionado por la *Open Handset Alliance* desde el año 2007.

El proyecto *AOSP* se rige por dos condicionados maestros del código libre: la Licencia General Pública *GNU*, versión 2(*GPLv2*), y la Licencia de *Software Apache* 2.0 (*ASL* 2.0).

Google se decidió por la licencia *Apache* 2.0, que permite a los productores comerciales sacar sus productos al mercado sin revelar secretos industriales vinculados con el proceso

de elaboración de aquellos. De este modo se consiguió un equilibrio entre la libertad del usuario y los intereses legítimos de los fabricantes.

Por tanto, *Android* es un sistema de código abierto que está generalmente personalizado con *software* propietario de fabricantes y operadores.

Para obtener más información sobre el SO *Android* se puede visitar la siguiente dirección:

http://source.android.com

Android ha pasado por una serie de actualizaciones desde que se lanzó por primera vez. La imagen siguiente muestra algunas de las diferentes versiones de *Android* y sus nombres en clave.

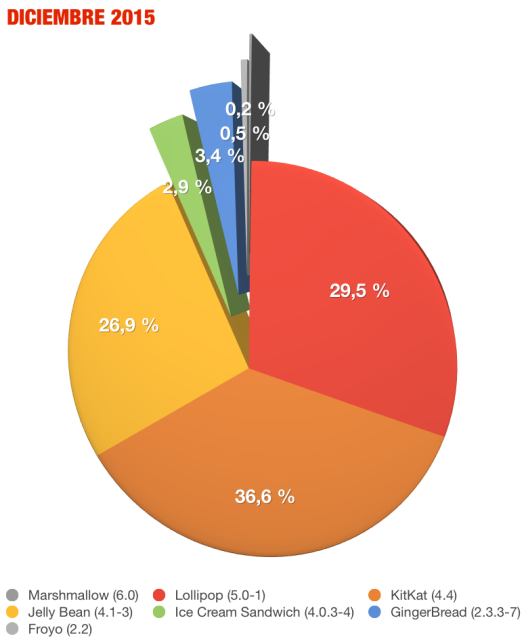


Figura 5: Distribución de versiones.

Como curiosidad sobre las primeras versiones del SO de *Android*, indicar que la primera versión de *Android* (*1.0*, sin denominación publicitaria de proyecto) fue liberada a finales de 2008 para utilizarse exclusivamente en *smartphones*. Incluía soporte para productos típicos de *Google* como *Gmail*, *Youtube* y *Google Maps*, además de telefonía y mensajes *SMS*.

Seis meses más tarde *Google* hizo pública una primera revisión, denominada *Cupcake*. La nueva versión incorporaba mejoras de diseño y prestaciones adicionales, como grabación de vídeo y aplicaciones auxiliares de pantalla (*widgets*), como relojes, indicadores meteorológicos, *tickers* de bolsa, etc.

En la versión 1.6 (*Donut*) se incluyó soporte para mayores resoluciones de pantalla y navegación gestual.

Android 2.0 (*Eclair*), tras haber corregido algunos defectos de seguridad anteriores, añadió soporte para *Microsoft Exchange* y *Bluetooth 2.1*. Al cabo de pocos meses, a mediados de

2010, era liberado *Android 2.2* (*Froyo*) con nuevos avances: *tethering* o capacidad para habilitar el terminal como punto de acceso inalámbrico portátil, soporte para memoria *RAM* por encima de *256 MB*, borrado remoto del terminal y posibilidad de instalar aplicaciones en la tarjeta *SD*.

2.3.2. ESPECIFICACIONES TÉCNICAS

Especificaciones técnicas	
Arquitecturas válidas	*ARM 32* Y *64 bits*, *x86 32* y *64 bits*, *MIPS* y *NEON*. (la mayoría de los dispositivos *Android* utilizan microprocesadores *ARM* basados en arquitectura *RISC* [*reduced instruction set computer*, ordenador con conjunto de instrucciones reducido]).
Kernel	*Kernel de Linux 3.X* dependiendo de la versión de *Android* y del dispositivo.
Sistema de ficheros	Soporta varios, pero principalmente utiliza *EXT4* o *JFFS2* para el sistema de ficheros internos. Acepta tarjetas de memoria formateadas en *FAT32*. También soporta cifrado de disco desde la versión 4.3.
Ejecutables	*Bytecode*. Ejecutable por la máquina *Dalvik* o por el más reciente *Android Runtime* (*ART*).
Plataforma del sistema	Basado en *Linux*.
Licencia	Licencia *Apache* 2.0. Los fabricantes modifican el código del sistema y lo adaptan a los dispositivos con total libertad dentro de unos parámetros mínimos comunes.

Cuadro 3: Especificaciones técnicas.

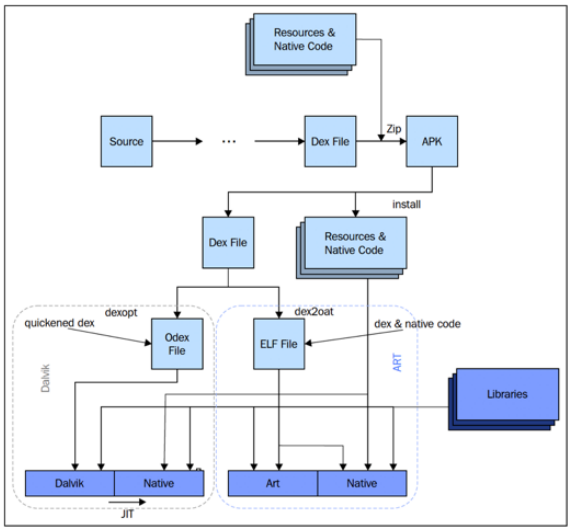


Figura 6: *Android Runtime* (*ART*).

2.3.3. ARQUITECTURA INTERNA DEL SISTEMA OPERATIVO *ANDROID*

El conocimiento de la arquitectura de lo que se está investigando es la primera base del análisis forense.

El SO de *Android* se organiza en las siguientes capas:

- **Aplicaciones**: aquellas instaladas por el usuario o pre-instaladas en el sistema.
- ***Application Framework*** (entorno de aplicaciones): ofrecen servicios a las aplicaciones (están desarrollados en *Java*).

  Gracias al *framework* las aplicaciones que funcionan en el último nivel de la pila *Android* tienen acceso a las funcionalidades básicas del sistema: administración de actividades, geolocalización, notificaciones, proveedores de contenido, etc.

- **Librerías**: módulos que ofrecen servicios al *application framework* (marco de aplicación). Están desarrollados en *C/C++* y compilados directamente en el código nativo de cada plataforma, y son utilizadas tanto por el sistema operativo como por las aplicaciones.

  Las librerías son archivos que contienen rutinas de código reutilizables para la ejecución de funciones específicas, evitando al programador el tener que incluir el mismo código cada vez que escribe un programa. De este modo se evita la redundancia de *software* a la vez que se ahorra memoria *RAM* y espacio de almacenamiento en los soportes de datos.

- ***Android Runtime***: cada aplicación ejecuta su propia instancia de una máquina virtual de *Java*, específica de la plataforma de dispositivos móviles *Android* y denominada *Dalvik VM* (ha sido diseñada por *Dan Bornstein* con la contribución de otros ingenieros de *Google*).
- ***Kernel***: ejecuta código dependiente del dispositivo (*ARM*, *x86*, *MIPS*, etc.). Ofrece servicios de seguridad a las capas superiores.

  El *kernel* es la capa más baja y está en contacto directo con el *hardware*. El *kernel* gestiona procesos, memoria y mecanismos de seguridad del sistema de archivos *Linux*. A través de diversos controladores – pantalla, teclado, cámara de vídeo, adaptador *WiFi*, memoria *flash*, audio, *Binder IPC* y administrador de energía -, el *kernel* pone todas las funcionalidades del *hardware* al servicio del sistema operativo.

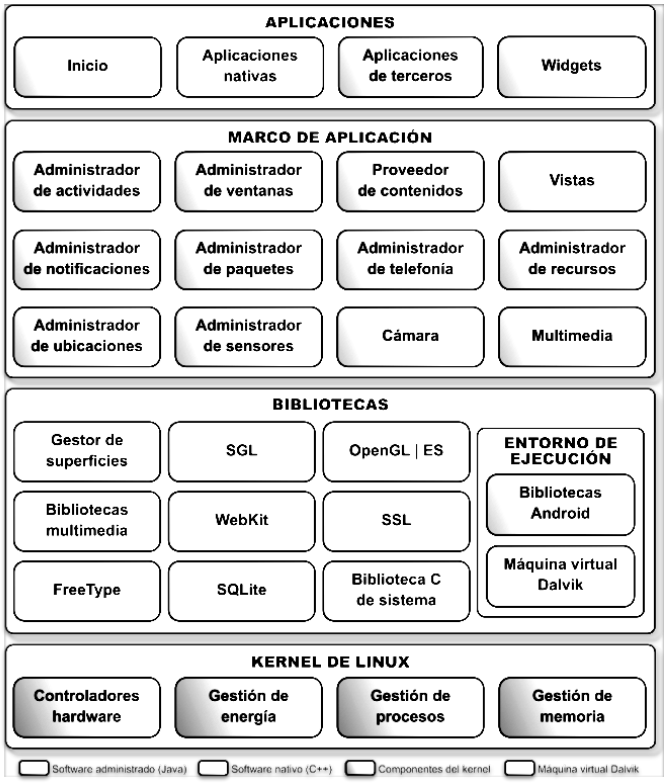


Figura 7: Arquitectura *Android*.

2.3.4. LIBRERÍAS DISPONIBLES

Las aplicaciones acceden a las librerías a través del *Application Framework*:

- ***Package Manager***: controla la instalación de paquetes.
- ***Activity Manager***: gestiona las actividades que se muestran en pantalla.
- ***Location Manager***: ofrece información sobre la localización del dispositivo.
- ***Notification Manager***: permite gestionar las notificaciones recibidas por una aplicación.
- ***Content Providers***: ofrece acceso a datos almacenados por aplicaciones en bases de datos.

  Los *content providers* (proveedores de contenido) son estructuras de datos a las que han de acceder en algún momento las aplicaciones *Android*. Ejemplos típicos son los contenedores para mensajes de correo electrónico, contactos, coordenadas del sistema de geolocalización o archivos almacenados en un soporte de datos y accesibles a través de peticiones o enlaces.

  En todos los casos, los datos son guardados por lo general en tablas *SQLite*, accesibles desde una aplicación a través de funciones exportadas por librerías.

- ***View System***: controla las diferentes vistas que se muestran al usuario.

- ***Bluetooth API***: controla las conexiones *Bluetooth* del dispositivo.

Para más información sobre las librerías se recomienda visitar la siguiente dirección en ***http://developer.android.com/guide/index.html***.

2.3.5. SISTEMA DE FICHEROS Y PARTICIONES EXISTENTES

El *kernel* de *Linux*, a través de módulos adecuados, puede manejar la mayor parte de los sitemas de archivos utilizados en la actualidad. Aunque *Android* reconoce gran cantidad de sistemas de archivos, en la práctica solo utiliza tres o cuatro.

Android, por defecto, utiliza múltiples particiones, normalmente formateadas en *Journal Flash File System 2* (*JFFS2*). Si bien es cierto que en los primeros tiempos, *Android* se servía de *YAFFS*, para aquellas particiones en las que venía instalado el sistema operativo con aplicaciones de fábrica. También guardaba los datos del usuario, - contactos, cuentas de correo, aplicaciones instaladas desde el *Android Market*, archivos temporales, etc. - . En la actualidad, *YAFFS* es un sistema de archivos en desuso, y ha sido reemplazado por *Linux EXT4*, principalmente por razones de rendimiento y compatibilidad con las nuevas *CPU multicore*.

Indicar que cualquier fabricante puede utilizar otro sistema de ficheros o modificar la estructura de ficheros o particiones.

Principales directorios (las particiones se explicarán con mayor detalle en la fase de análisis):

- ***/system***: directorio donde se guarda la *ROM* del SO (solo lectura).
- ***/proc***: información de los procesos en ejecución. El acceso a */proc* y otros sistemas de archivos virtuales de *Android* require procedimientos de *rooting*, lo cual significa modificar el dispositivo en mayor o menor medida. Esto implica un riesgo de alterar medios probatorios, a resultas de lo cual la parte contraria podría impugnar los elementos de evidencia presentados.
- ***/mnt***: punto de montaje para otros tipos de almacenamiento.
- ***/sdcard***: redirige a */mnt/sdcard*, punto de montaje de la tarjeta *SD*.
- ***/cache***: guarda la caché de datos de las aplicaciones y del sistema.
- ***/data***: directorio en el que se almacenan las aplicaciones.

Indicar también que cada aplicación se almacena en un directorio en el que solamente ella dispone de permisos de acceso. En *Android 6.0*, existe la posibilidad de revocar los permisos después de haber sido aceptados.

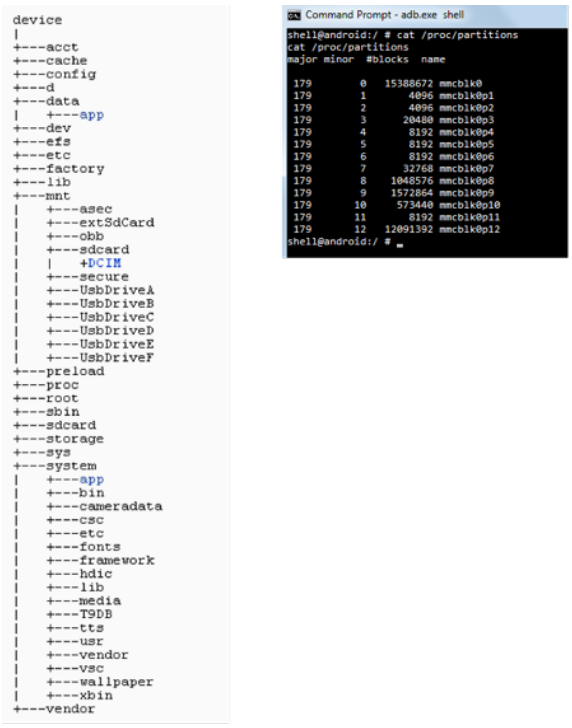


Figura 8: Sistema de ficheros y particiones.

Estos sistemas de archivos constituyen el objetivo prioritario de la investigación forense. Dentro de ellos suele haber documentos, imágenes, archivos de sonido y/o vídeo, datos de archivos borrados, pero aún accesibles a bajo nivel y otros elementos de evidencia.

2.3.6. ESTRUCTURA DE UN BINARIO

Las aplicaciones *Android* se empaquetan en ficheros *APK* (*zip*), los cuales contienen los siguientes elementos:

- ***META-INF***: directorio que contiene un listado de ficheros (*MANIFEST.MF*). El certificado de la *app* (*CERT.RSA*), una lista de recursos de la *app* y *hashes SHA-1* de los ficheros indicados en el listado.
- ***lib***: directorio con el código nativo utilizado por la *app* para plataformas específicas (*ARM*, *x86*, *MIPS*).
- ***assets***: directorio con diferentes elementos utilizados por la *app*.
- ***res***: directorio con recursos utilizados por la *app* (iconos, imágenes, constantes, etc.).
- ***resources.arsc***: ficheros *XML* precompilados con definiciones del interfaz de usuario.
- ***classes.dex***: código compilado de la aplicación para el *runtime* de *Android*.
- ***AndroidManifest.xml***: fichero de información de la aplicación.

EL FICHERO *ANDROIDMANIFEST*

Toda aplicación *Android* viene acompañada de un archivo llamado *AndroidManifest.xml*. El *manifest* de una aplicación contiene toda la información necesaria para la instalación y ejecución de la *app* por parte de *Android*:

- Versión mínima de *Android* con la que la aplicación es compatible.
- Nombre del paquete de la aplicación y versión.
- Componentes incluidos, donde se definen las actividades, servicios y *broadcast receivers* que utiliza la aplicación. Los últimos también pueden declararse dinámicamente durante la ejecución de la aplicación.
- Permisos que la aplicación solicita al usuario para ejecutarse (anterior a la versión de *Android 6.0*).
- Capacidades del dispositivo necesarias para ejecutar la aplicación. Algunos ejemplos son: cámaras, acelerómetro, etc.
- Librerías del sistema que necesita cargar la aplicación para que funcione.

Este archivo es de tipo texto y está escrito en formato *XML*. La información incluida en el manifiesto se utiliza para instalar el programa. A través de un menú se informa al usuario sobre los permisos que debe autorizar.

En función de lo que se haya decidido, el sistema determina si aquella es instalada o no. La lectura de estos avisos y la autorización de permisos es un elemento clave en el sistema de seguridad de los dispositivos *Android*.

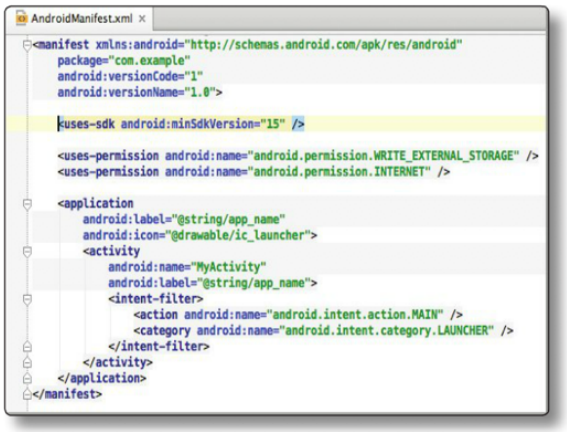


Figura 9: Aspecto del archivo *AndroidManifest.xml*.

2.3.7. COMPONENTES DE UNA APLICACIÓN

Los componentes de una aplicación pueden ser:

- Públicos: otras aplicaciones pueden interactuar con ellos.
- Privados: solo los componentes de la misma aplicación (mismo user *ID*) pueden interactuar con ellos.

Todos los componentes dentro de una aplicación se ejecutan dentro del mismo proceso, a no ser que el desarrollador especifique lo contrario.

Indicar también que el desarrollador puede definir restricciones para el paso de mensajes a través de permisos:

- Los que envíen mensajes pueden requerir a las aplicaciones que lo reciban de cierto permiso.
- Los que reciban mensajes pueden aceptar mensajes solo de aplicaciones con ciertos permisos.

A continuación, se van a describir algunos de los componentes más importantes que se encuentran presentes en aplicaciones *Android*:

ACTIVITY

Las actividades constituyen la capa de presentación de la aplicación y ofrecen el interfaz visible de la aplicación. Una aplicación puede tener cero o más actividades. Aunque, por lo general, las aplicaciones tienen una o más actividades, y el objetivo principal de una actividad es interactuar con el usuario.

Las actividades ocupan toda la pantalla. Cuando se quiere mostrar elementos de información que no ocupan toda la pantalla se utilizan los *Fragments*.

Cada actividad contiene una jerarquía de vistas (*Views*) con las que el usuario puede interactuar.

Toda aplicación puede tener una Actividad principal, que será la que se lanza cuando se toca el icono de la aplicación en la pantalla de *apps* del dispositivo móvil.

Además, cada actividad de la aplicación tiene un ciclo de vida, como se muestra a continuación:

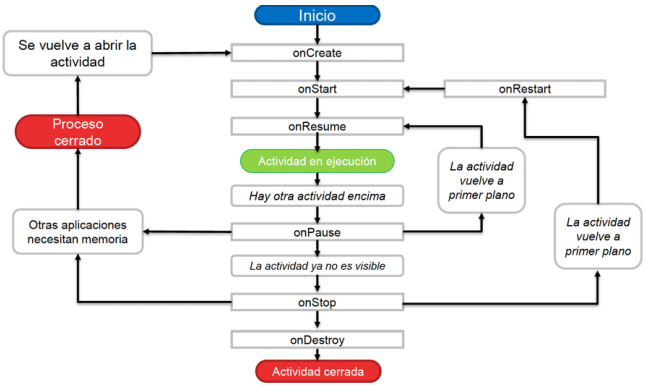


Figura 10: Ciclo de vida de una aplicación.

SERVICES

Los servicios son los encargados de realizar operaciones en el *background* (se ejecutan en segundo plano) sin una interfaz explícita de interacción con el usuario. Cuando una aplicación deja de estar en primer plano puede seguir ejecutándose a través de servicios.

En sentido estricto no son imprescindibles para el funcionamiento del terminal, pero sin ellos resulta imposible realizar acciones como descargar archivos de *Internet* en segundo plano o escuchar música mientras el usuario ejecuta una aplicación o navega por *Internet*.

Normalmente, son utilizados para realizar cómputos, guardar o recibir datos de *Internet* o disco. Es decir, pueden monitorizar procesos de comunicación, gestionar el tráfico de la red

y suministrar información – por ejemplo, coordenadas geográficas – a otras aplicaciones.

Una de las formas de ejecutar tareas de servicios es por medio de la utilización de *IntentServices*:

- Por cada *IntentService* el sistema crea un único *thread* separado de la interfaz de usuario.
- Cada tarea enviada al *IntentService* (por medio de un *Intent*) es ejecutada en el *thread* (no hay ejecución concurrente de varios *Intents*).

CONTENT PROVIDER

Los *Content Providers* están diseñados para compartir datos estructurados entre aplicaciones, mediante una interfaz, para que otras aplicaciones puedan acceder a los datos almacenados por la aplicación.

Las aplicaciones que quieren compartir datos con el resto deben contar con sus propios *providers*. Por lo que es una buena práctica de seguridad requerir permisos, para restringir las aplicaciones que pueden acceder a los permisos declarados por la aplicación.

El SO ofrece por defecto una serie de *Providers* genéricos, para el acceso a datos del sistema, como los contactos, listado de llamadas y *SMS*. Para acceder a ellos, la aplicación tiene que declarar los permisos correspondientes.

BROADCAST RECEIVERS

Algunas *activities* pueden actuar como receptoras de eventos y procesar directamente la información contenida en los *intents*. En caso de que no sea así, el elemento adecuado para cumplir dicha función es un *broadcast receiver* o receptor de multidifusión.

Los *Broadcast Receivers* son tareas que se ejecutan cuando llegan mensajes (*Intents*) generados por otros componentes de la aplicación o por otras aplicaciones.

Las aplicaciones pueden crear *Intents*, para enviar mensajes a actividades, servicios o *Broadcast Receivers*.

Cada *Broadcast Receiver* es configurado, para escuchar un tipo específico de *Intents*.

Por tanto, los *broadcast receivers* recogen el *intent*, analizan los datos transportados por el mismo y en función de aquellos lanzan la aplicación o *activity* encargada de procesarlos.

2.3.8. COMUNICACIÓN ENTRE APLICACIONES

Las aplicaciones en *Android* se pueden comunicar mediante cualquier mecanismo de comunicación entre procesos, el cual ha sido heredado de *UNIX*:

- Sistema de ficheros, *sockets*, etc.
- Los permisos de la aplicación siempre se aplican antes de la comunicación.

Además, *Android* ofrece dos mecanismos adicionales de comunicación entre aplicaciones:

- ***Binder***: sistema *RPC* (*Remote Procedure Call*) implementado sobre *Linux* con un *driver* específico. Las aplicaciones lo utilizan a través de objetos Intent que son enviados al sistema para lanzar Servicios, Actividades y *Broadcast Receivers*.
- ***Content Providers***: operaciones de lectura/escritura sobre bases de datos de otras aplicaciones.

Un *Intent* es un objeto para el envío de mensajes, el cual contiene:

- Información acerca de la operación que se quiere realizar, a través del parámetro *action* o *component*.
- Datos sobre los que se quiere realizar la operación, a través de extras o una *URI* de origen de los datos.
- Información adicional que puede ser de utilidad para el receptor.

Los *Intents* son enviados al sistema operativo que se encarga de entregarlos al receptor correspondiente, dado que no pueden ser enviados directamente.

Existen dos tipos de *Intents*:

- **Explícitos**: especifican el receptor a través del nombre del componente.
- **Implícitos**: indican una acción genérica a realizar (por ejemplo, enviar un mensaje) y es el sistema quien se encarga de encontrar un receptor. Los *intents* implícitos no tienen asignado un destino, por lo que son puestos a disposición del gestor de paquetes de *Android* para que sea este quien decida a qué aplicaciones debe hacerlos llegar.

```
Intent i1 = new Intent(this, ActivityB.class);
i.putExtra("KEY", "Value");
startActivity(i);
```

Figura 11: Ejemplo de *Intent* implícito.

Indicar también, en cuanto a la comunicación entre apliaciones en el SO *Android*, que sólo se pueden comunicar aplicaciones que compartan el permiso del recurso que comparten.

2.3.9. PROCESO DE COMPILACIÓN Y CREACIÓN DE UNA APLICACIÓN

Las aplicaciones *Android* están programadas en *Java* y, al igual que en este lenguaje, lo que se ejecuta no es código compilado para el *hardware* de la máquina, sino una especie de código intermedio en un formato especial, válido para la misma máquina virtual implementada en distintas plataformas *hardware* y, consiguientemente, portable.

Sin embargo, existen algunas diferencias importantes con respecto a *Java*. En el modelo de desarrollo *Android*, los diferentes ficheros fuente de *java* son compilados en un fichero *JAR* (*Java archive*). Este formato combina múltiples ficheros *.class* compilados en *bytecode* en un fichero, utilizando la compresión *zip*.

El *bytecode* es transformado al formato "Ejecutable Dalvik", para su ejecución en *Android*. Es decir, es ejecutado por *Dalvik* y *ART*, el runtime introducido en las últimas versiones

de *Android*.

El resultado del proceso es un archivo *.dex* que contiene todas las clases pertenecientes al programa.

El fichero *dex* resultante será comprimido en un fichero *zip* (paquete *APK*) junto con los recursos, *assets* y *manifest* de la aplicación.

Posteriormente, si se quiere publicar la aplicación en *Google Play* será necesario firmarla con un certificado. Éste puede ser auto firmado, pero es requisito imprescindible que todas las versiones de la *app* deban ser firmadas con el mismo certificado.

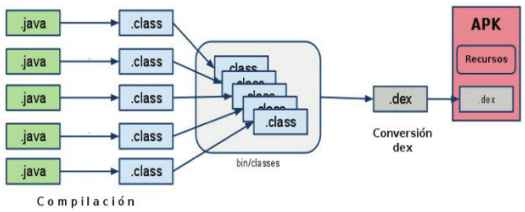


Figura 12: Proceso de compilación y creación de una aplicación.

3. METODOLOGÍA PROPUESTA

3.1. INTRODUCCIÓN

Dado que no existe una metodología estandarizada que se centre en el análisis forense de dispositivos móviles, existen diferentes guías que pueden orientar el proceso. Entre las más importantes se encuentran:

- ***Guidelines on Mobile Device Forensics*** del *NIST*.
- ***Developing Process for Mobile Device Forensics*** del *SANS*.
- ***Best Practices for Mobile Phone Forensics*** del *Scientific Working Group on Digital Evidence* (*SWGDE*).
- ***Good Practice Guide for Mobile Phone Seizure & Examination*** de la Interpol.
- ***ISO/IEC 27037:2012***, Orientaciones para la idenficación, recogida, adquisición y preservación de la evidencia digital (*Guidelines for identification, collection, acquisition and preservation of digital evidence*).
- ***RFC 3227***, no hace mención directa a los dispositivos móviles, pero es un estándar de facto en el proceso forense de ordenadores. Los artículos *RFC* o *Request for Comments* son documentos públicos sometidos al debate de la comunidad, para estandarizar procesos.
- ***UNE 197010:2015***, criterios generales para la elaboración de informes y dictámenes periciales sobre Tecnologías de la Información y las Comunicaciones (*TIC*). Esta norma tiene por objeto establecer los requisitos formales que deben tener los informes y dictámenes periciales en el ámbito de las *TIC*, sin determinar los métodos y procesos específicos para la elaboración de los mismos.

En este sentido, el proceso de análisis forense se va a basar en la creación y el seguimiento de una metodología y la utilización de unas herramientas aceptadas por la comunidad.

Hay que indicar que la metodología utilizada durante el análisis forense de sistemas informáticos ha sido heredada de los procesos forenses tradicionales.

Estas herramientas deben cumplir dos requisitos principales:

- Repetibilidad: capacidad para repetir exactamente los mismos resultados a partir de las mismas condiciones iniciales, en ejecuciones sucesivas separadas, utilizando el mismo método y herramientas. Es decir, que si un grupo de peritos tienen que repetir una prueba, lleguen a las mismas conclusiones en su informe.
- Reproducibilidad: capacidad de obtener los mismos resultados a partir de las mismas condiciones iniciales, utilizando el mismo método, pero medios diferentes (es decir, utilizando otras herramientas o creándolas desde cero).

Además de lo anterior, se dice que un proceso de análisis forense es "forensically sound", si el proceso de análisis forense asegura que las evidencias no han sido modificadas o destruidas. Es decir, si el proceso para la recogida, manejo, almacenamiento y análisis de evidencias puede asegurar que no han sido modificadas o destruidas durante el proceso de análisis.

Por tanto, el proceso de análisis forense se ha dividido en seis etapas.

3.2. ETAPAS DEL ANÁLISIS FORENSE

El proceso de análisis forense seguido en esta metodología se divide en seis etapas, las cuales se muestran a continuación:

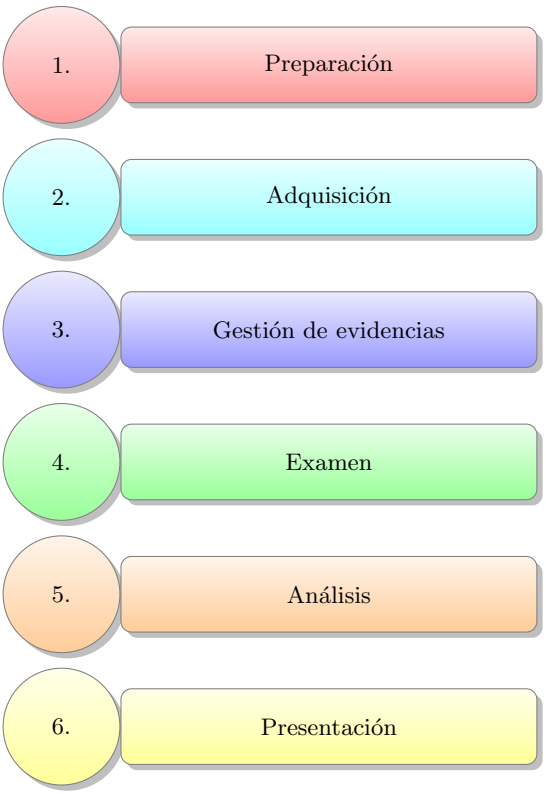


Figura 13: Etapas del análisis forense.

Como se puede observar, el proceso de análisis forense se ha dividido en seis fases. En este sentido, las fases del análisis forense de dispositivos móviles que más diferencias guardan con respecto a los procedimientos llevados a cabo en el análisis forense en equipos de escritorio son las fases de adquisición y de examen.

Para la correcta consecución del proceso de análisis se recomienda la toma de notas durante cada una de las fases del análisis.

Las notas, cuanto más detalladas mejor, pueden incluir, por ejemplo:

- Capturas de pantalla.
- Localización de evidencias encontradas.
- Notas manuscritas.
- Utilización de sistemas de anotación dentro de la propia aplicación forense.

Date	Time	Task	Description	Comments
3/3/15	900	Reception	Det. Smith requested an analysis of a mobile device	Device was obtained from the property room and transported to lab. Det. Smith completed processing request form.
3/3/15	945	Open evidence	Mobile device and SIM	Photographed, sealed, and then when the bag was opened—mobile device was off.
3/2/15	1015	Document	UICC and mobile device	Gathered mobile device information and UICC information using photographs.
3/3/15	1100	Collection	SIM	UICC was photographed and collected using USIM Detective. File system and PIN data recovered.
3/3/15	1130	Isolation	Mobile device	Mobile device was isolated from cellular by its lack of UICC. Network was isolated after pressing power button and selecting Airplane Mode while in Faraday box. Device was unlocked.
3/3/15	1300	Collection	Mobile device	Device was collected using non-invasive means physically using UFED. Binary file was produced of the userdata partition.
3/4/15	900	Collection	Mobile device	Device was collected logically using UFED. SMS, MMS, Calls, Contacts.

Figura 14: Ejemplo de sistema de anotación.

3.2.1. FASE DE PREPARACIÓN

Esta etapa se ejecuta de forma previa al proceso de análisis y consiste en identificar los elementos físicos que se van a analizar y las evidencias que se buscarán en cada uno de los elementos analizables.

Éstas dependen del objetivo del análisis forense. Por ejemplo:

- El análisis de una intrusión a través de un dispositivo móvil requerirá, por ejemplo, el análisis del almacenamiento del dispositivo en busca de pruebas de acceso ilegítimo a los sistemas.
- El análisis de un dispositivo para la comprobacion de una coartada requerirá, por ejemplo, el análisis del almacenamiento del dispositivo para comprobar las localizaciones en las que ha estado el dueño del dispositivo.

En ocasiones, esta tarea se realiza de forma conjunta con la de adquisición de las propias evidencias debido a la necesidad de una respuesta rápida ante el incidente, para evitar la eliminación de pruebas. Así, por ejemplo, durante la incautación de un dispositivo móvil, la primera tarea que se realiza es la preservación inicial de la evidencia mediante su inserción en una “Jaula de Faraday”, para aislarlo de señales externas.

Figura 15: Bolsa de *Faraday*.

Si no se dispone de una jaula de *Faraday*, el aislamiento se puede conseguir por diversos métodos, por ejemplo, introduciendo el dispositivo móvil en un recipiente métalico que impida el paso de las ondas de radio. Algunos de los accesorios que los peritos utilizan para este fin son:

- Bolsas especiales. Estas bolsas distan de ser perfectas, por lo que el perito debería tomar la precaucación de envolver el dispositivo móvil en papel de plata – tres capas como mínimo – para apantallar por completo la señal.
- Papel de alumino. Otro truco consiste en envolver el dispositivo móvil en papel de aluminio de buena calidad, cuanto más grueso sea el papel mejor. Si con una capa no basta, será necesario envolver el dispositivo en dos vueltas de papel de aluminio.

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

  En este sentido, tampoco se recomienda envolver el dispositivo en papel de alumino con el cable de alimentación puesto, dado que este actuaría como una antena externa dejando pasar la señal de radio.

- Inhibidor de frecuencias. Si se dispone de algo más de presupuesto se puede adquirir a través de *Internet* un dispositivo capaz de bloquear transmisiones de telefonía móvil y redes inalámbricas, impidiendo así que el teléfono utilice los canales de control para establecer comunicación con el exterior. Uno de estos aparatos puede generar hasta 6 vatios de potencia, haciendo imposible el funcionamiento de cualquier dispositivo móvil o cliente *WiFi* en un radio de 10 metros.

  Se recomienda antes de adquirir un inhibidor de frecuencias estar al tanto de lo que las leyes y normativas estatales establecen sobre el uso de estos dispositivos.

Figura 16: Inhibidor de frecuencias

3.2.2. FASE DE ADQUISICIÓN - CONSIDERACIONES

En un entorno ideal, la adquisición de datos del dispositivo no debería modificar el estado físico del dispositivo.

Por desgracia, esto no siempre es posible. Dependiendo de su estado, el tipo de adquisición y las herramientas utilizadas, el estado del dispositivo se verá afectado:

- Fecha y hora de acceso a ficheros.
- Borrado o creación de nuevos ficheros.
- Modificación de la memoria del dispositivo, para la carga de aplicaciones de volcado.

Para que la validez del análisis no se vea afectada, es necesario documentar todos los tipos de adquisición realizadas y sus consecuencias sobre el dispositivo analizado. Es decir:

- La adquisición manual creará ficheros de captura de pantalla.
- La adquisición lógica puede modificar la fecha de acceso a los ficheros.

También es importante indicar, que durante el proceso de adquisición de datos en un dispositivo móvil se debe proceder, de forma general, de medio más volátil a medio menos volátil.

3.2.3. FASE DE ADQUISICIÓN - MÉTODOS DE ADQUISICIÓN

Una vez enumeradas las evidencias que se van a adquirir, hay que obtener los datos de los dispositivos que van a ser objeto del informe forense.

El método utilizado para adquirir los datos del dispositivo variará dependiendo de:

- La plataforma de la que se van a adquirir los datos (en este caso, *Android*).
- La versión específica del *hardware*, *software* y configuración del dispositivo (versión del dispositivo, configuración de desbloqueo activa, etc.).
- El estado en el que se encontró el dispositivo (apagado o encendido, bloqueado o sin bloquear, etc.).
- El tipo de datos a adquirir y su volatilidad (capturar datos en almacenamiento persistente o en memoria).

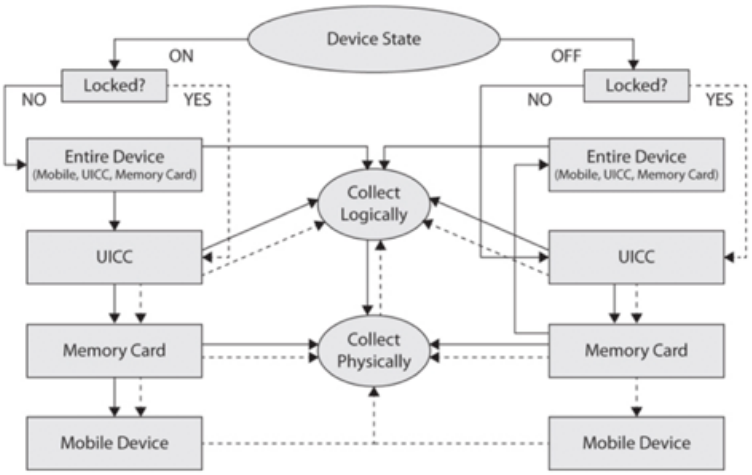


Figura 17: Métodos de adquisición 1.

3.2.4. FASE DE ADQUISICIÓN - TIPOS DE ADQUISICIÓN

Dependiendo de las variables anteriores (vistas en el apartado anterior) se pueden realizar tres tipos de adquisición: manual, lógica y física.

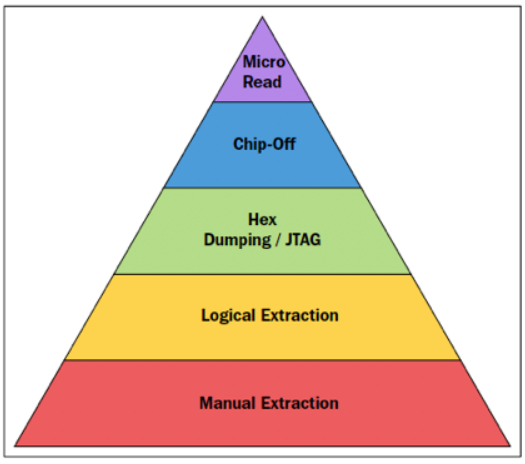


Figura 18: Tipos de adquisición.

ADQUISICIÓN MANUAL

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

Este tipo de adquisición cuenta con ciertas ventajas, pero también con desventajas.

Por una parte, entre las ventajas que puede proporcionar encontramos:

- No requiere de herramientas adicionales.

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

Por otra parte, algunas de las desventajas son:

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

ADQUISICIÓN LÓGICA

La adquisición lógica consiste en copiar los archivos y directorios del sistema de archivos del dispositivo. Este tipo de adquisición forense requiere inevitablemente la ayuda de la interfaz *USB*.

Para ello, se utilizan:

- Las propias *API* de acceso al sistema de ficheros del dispositivo objeto de análisis. El SO del dispositivo se encargará de copiar a otro dispositivo los ficheros y directorios solicitados.
- Las *API* del SO de la herramienta de adquisición. La unidad se conecta al dispositivo que hay que analizar. Los datos del sistema analizado seguirán siendo leídos por el *firmware* del dispositivo analizado.

Algunos de los puntos a favor de la adquisición lógica:

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

Y como puntos en contra, tenemos:

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

Por último, indicar que la adquisición lógica de un dispositivo *rooteado* se puede realizar mediante la copia de todos los archivos existentes en el sistema.

ADQUISICIÓN FÍSICA

Consiste en el copiado *bit* a *bit* del dispositivo físico de almacenamiento (permite obtener una copia *bit* a *bit* del contenido de los *chips* de memoria del dispositivo analizado), por lo que requiere de acceso completo al dispositivo de almacenamiento.

En el dispositivo móvil, por lo general, el sistema de almacenamiento se encuentra soldado al resto de los componentes del teléfono y no es accesible de forma física.

Además, dadas las medidas de seguridad incluidas en los actuales sistemas operativos, en muchas ocasiones, es necesario ejecutar *exploits* sobre el sistema, para realizar el copiado

a bajo nivel.

Esto presenta ciertas ventajas, como por ejemplo, que permite accdeder a todos los bloques del soporte físico copiado, incluyendo los archivos borrados y bloques que no han sido marcados como utilizados.

Entre sus inconvenientes se encuentra que, normalmente, el proceso es más complejo, por lo que no siempre es posible llevarlo a cabo.

En este sentido, también hay que matizar que la adquisición física depende de los tipos de almacenamiento con los que cuenta el dispositivo móvil:

- La memoria *NAND* es el tipo de memoria *flash* más utilizada para el almacenamiento en los dispositivos móviles (es un tipo de memoria similar a la de las antiguas tarjetas *CompactFlash* y los actuales *pendrives USB*, que una vez grabada conserva su contenido aunque no circule la corriente). Se puede leer y escribir en bloques. Normalmente, es utilizada de forma genérica para el almacenamiento del SO, la partición de datos del sistema y otras memorias extraíbles.

  Dependiendo de la marca y el modelo del dispositivo, la memoria *NAND* es ampliable mediante tarjetas *MicroSD*. Por motivos de seguridad casi siempre se encuentra formateada con el sistema de archivos *FAT32*.

- La memoria *NOR* es otro tipo de memoria *flash* optimizada, para la ejecución de código (la unidad mínima de acceso a la memoria *NOR* es el *byte*). Permite la lectura y ejecución de *bytes* de forma independiente. Sin embargo, en los últimos años, su utilización se está viendo reducida a favor de las memorias *NAND*, para usos más genéricos.

- Las tarjetas de memoria. Estas tarjetas utilizan memorias *NAND*. Normalmente, están formateadas en *FAT32*.

- Tarjetas *SD*. En *Android*, como en otros dispositivos (como por ejemplo, *Windows Phone* y *BlackBerry*), se permite la utilización de tarjetas *SD*, dependiendo del modelo (por ejemplo, los dispositivos *iOS* no permiten la utilización de tarjetas *SD*).

  La tarjeta *MicroSD* es de tipo no volátil y está fabricada también con tecnología *NAND*. Este elemento posee gran importancia para la investigación forense. El perito debe tenerlo en cuenta cuando analiza un dispositivo, tomando nota de la partición correspondiente, el sistema de archivos – generalmente *FAT32* -, el punto de montaje y otras características de configuración.

3.2.5. FASE DE ADQUISICIÓN - MAXIMIZANDO LA ADQUISICIÓN

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

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

Para maximizar la adquisición de datos a obtener en un dispositivo, es fundamental seguir un conjunto de pasos iniciales. Si bien, los pasos concretos varían de una plataforma a otra, este conjunto de procedimientos se puede realizar con cualquier terminal, independientemente del SO o fabricante.

DISPOSITIVO DESBLOQUEADO

Si el dispositivo ha sido incautado, los pasos a realizar para desbloquearlo serán los siguientes:

- Aislar el dispositivo de la red. Es decir, habilitar el modo avión y extraer la tarjeta *SIM*. Aquí, es recomendable introducirlo en un recipiente que aísle el campo electromagnético (por ejemplo, en una "Jaula de Faraday"). También hay que activar todas las opciones posibles, para permitir el acceso físico al dispositivo. Es decir:
  - Eliminar el código de bloqueo (si es posible).
  - Activar la depuración a través de *USB*.
  - Desactivar el bloqueo por inactividad (activando la opción "siempre activo").
- Obtener todos los medios extraíbles, tarjeta *SD*, *SIM* (la principal función de la tarjeta *SIM* en el proceso de cifrado y control de integridad de las comunicaciones en redes inalámbricas celulares es utilizar la clave de autenticación *Ki*, para realizar una operación criptográfica cada vez que es requerida, si la tarjeta está desbloqueada) o copias de seguridad en dispositivos asociados (ordenadores).

DISPOSITIVO BLOQUEADO

Si el dispositivo se encuentra bloqueado solo podremos:

- Aislar el dispositivo de la red, extraer la tarjeta *SIM* o introducirlo en una jaula de *Faraday*.
- Comprobar si el dispositivo tiene activada la depuración a través de *USB*. En caso de que la conexión *USB* se encuentre activa, es posible que podamos cargar *boot loaders*, para modificar el sistema de arranque del dispositivo y permitir de este modo, el acceso físico al mismo.
- Si el dispositivo no tiene activada la depuración *USB*, ejecutar un ataque para la extracción del código de bloqueo (*smudge attack* o fuerza bruta).

  El ataque de suciedad (*smudge attack*) consiste en examinar las marcas existentes sobre una pantalla táctil para descubrir trazas del patrón de bloqueo que el usuario ha utilizado para proteger su terminal.

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

  En este caso, más que técnicas *hacking*, se aplican métodos basados en el proceso digital de imágenes. El éxito del procedimiento requiere evitar todo contacto manual

con los dispositivos incautados. Por lo que es buena idea acudir al escenario de los hechos provisto de bolsas para recogida de pruebas y de un lápiz capacitivo con punta de goma, para realizar las manipulaciones más urgentes (como por ejemplo, activar el modo avión), antes de trasladar el dispositivo al laboratorio en el que se van a hacer las fotografías digitales.

- Obtener todos los medios extraíbles: tarjeta *SD*, *SIM* o copias de seguridad en dispositivos asociados (ordenadores).
- Si el dispositivo se encuentra apagado se pude proceder directamente a extraer todos los medios extraíbles y encender el teléfono.

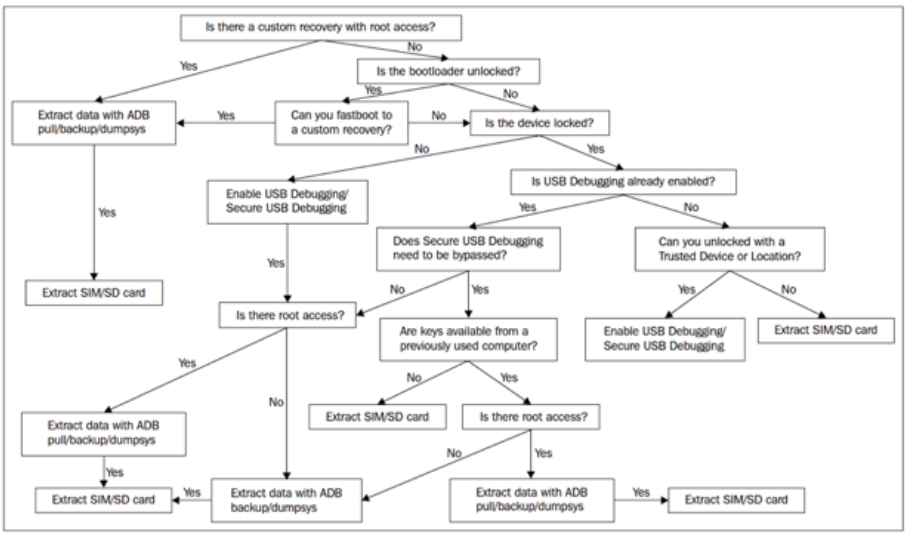


Figura 19: Métodos de adquisición 2.

3.2.6. FASE DE ADQUISICIÓN - *ANDROID*

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

A continuación, se describe un posible orden de adquisición según su volatilidad, aplicándose a dispositivos móviles los marcados en negrita (***RFC 3227*** [1]):

- Registros o cachés.
- Tablas de enrutamiento, lista de procesos y memoria.
- Sistemas de ficheros temporales.
- Disco.
- Sistemas de monitorización remota.
- Topología de red y configuración básica.
- Medios físicos externos.

[1] *RFC 3227* se encuentra disponible en ***http://www.ietf.org/rfc/rfc3227.txt***

La adquisición consiste en obtener o capturar las evidencias enumeradas en la fase de preparación.

Las evidencias se pueden agrupar en dos grupos, atendiendo a su tiempo de vida:

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

Siempre que sea posible, se debe llevar a cabo un duplicado forense:

- Consiste en realizar una copia *bit* a *bit* de la información de la fuente.
- Una vez obtenida la copia se obtiene su *hash*, para poder validar que es una copia exacta.
- Se puede comprimir, para optimizar su almacenamiento.
- Generalmente, se realiza mediante la utilización de *hardware* específico.

Dependiendo del estado del dispositivo y el tipo de evidencia, se requiere la utilización de diferentes técnicas y herramientas:

- Si el dispositivo está encendido y desbloqueado, se pueden utilizar técnicas de monitorización de red o volcado de memoria para capturar evidencias en tiempo real. Hay que indicar, que algunas de estas técnicas modifican levemente el sistema analizado, por lo que la validez de la prueba depende de la cantidad de cambios generados por la herramienta de adquisición. Así, por ejemplo, el programa ***dd***, para el volcado de memoria, se debe cargar en la memoria que se volcará para su ejecución.
- Si el dispositivo se encuentra en reposo, la adquisición de información se puede realizar *in situ* o en el laboratorio tras la incautación del dispositivo.

Por cada evidencia recogida es fundamental:

- Especificar las herramientas y procedimientos utilizados para su adquisición.
- Especificar la evidencia exacta que se ha recogido:
  - Tráfico de red: duración, hora de inicio, tipo de paquetes, datos obtenidos, etc.
  - Disco duro: porcentaje recuperado, método de recuperación, etc.

También es necesario utilizar algún mecanismo, para asegurar que los datos adquiridos no son modificados, y si lo son, que los cambios puedan ser trazados:

- Generalmente, se hace un resumen de los datos obtenidos mediante una función resumen (*SHA-256*).
- Dependiendo de la finalidad de la investigación el resumen puede ser firmado con una nueva clave privada del investigador.

ADQUISICIÓN MANUAL

Solo es realizable si el teléfono se encuentra desbloqueado.

En la adquisición manual hay que listar todas las aplicaciones existentes y realizar capturas de pantalla de los elementos que se consideren pertinentes.

Es posible que para el acceso a los datos de algunas aplicaciones sea necesario conectar el dispositivo a *Internet*. Esto se debe realizar sólo como último recurso, ya que el dispositivo puede ser bloqueado remotamente.

Además de las aplicaciones, es importante acceder a los ajustes y capturar la información de cada una de las cuentas existentes en el dispositivo.

En cuanto al método utilizado para la realización de capturas de pantalla, éste puede diferir dependiendo de la versión del sistema. Por lo general, se puede realizar manteniendo pulsado al mismo tiempo el botón de bajar el volumen y el botón de inicio, hasta que se realiza la captura de pantalla.

ADQUISICIÓN LÓGICA

Se puede realizar mediante *adb* con el siguiente comando

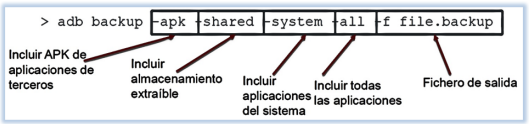


Figura 20: *Android full backup* sin *root* con *adb*.

Hay que tener en cuenta que las aplicaciones que no permitan el *backup* no se copiarán.

Una vez obtenida la copia de seguridad se puede extraer utilizando *Android Backup Extractor*, disponible en ***https://sourceforge.net/projects/adbextractor/***.

Y el comando:

```
> java -jar abe.jar unpack file.backup file.tar
```

Figura 21: Extracción de *file.backup* en *file.tar*.

También se puede realizar a través de *apps* (pero, solo se podrá acceder a información del teléfono accesible mediante permisos), mediante la herramienta *AFLogical*, disponible en *Santoku Linux*.

ADQUISICIÓN FÍSICA

Sólo se puede realizar si se tiene acceso de administrador al dispositivo o mediante acceso físico y una conexión al interfaz *JTAG*[2] del *chip* de memoria.

[2]La extracción mediante *JTAG* resulta complicada, al requerir equipos especiales y personal experimentado.

Figura 22: Interfaz *JTAG* del *chip* de memoria.

Si el dispositivo está *rooteado*, se puede realizar con ***dd***.

Para realizar el proceso, en primer lugar se busca la ruta de toda la memoria *flash*, mediante:

```
> cat /proc/partitions
```

Figura 23: Búsqueda de la ruta de toda la memoria *flash* con *cat*.

La primera entrada (***mmcblk0*** normalmente) corresponde a la totalidad de la memoria *flash*.

Para averiguar el tamaño del bloque del sistema se utiliza:

```
> df /data
```

Figura 24: Averiguar el tamaño del bloque del sistema con *df*.

Y con lo obtenido bajo la columna *Blksize* ejecutamos *dd*[3], del siguiente modo:

```
> dd if=/dev/block/mmcblk0 of=/sdcard/blk0.img bs=4096
```

conv=notrunc,noerror,sync

Figura 25: Copia del *backup* a la memoria externa con *dd*.

ADQUISICIÓN FÍSICA - Memoria

La implementación de *dd* para *Android* no está convenientemente preparada para leer la memoria *RAM* del dispositivo.

La memoria del dispositivo se puede adquirir utilizando la herramienta *Linux Memory Extractor* (*LiME*), que se encarga de realizar tres tareas fundamentales:

[3]Si la imagen del disco se guarda en la tarjeta *SD*, hay que asegurarse de que la tarjeta introducida ha sido borrada (formateada) convenientemente (todo a *0*'s).

- Averiguar las direcciones físicas de los rangos de direcciones de la *RAM* inspeccionando el *Kernel*.
- Transformar las direcciones físicas en virtuales.
- Copiar el contenido de las direcciones virtuales a un *socket* de red o a la tarjeta *SD* para su extracción.

Hay que indicar que *LiME* es una extensión del *Kernel* que se debe cargar a través del *adb*.

Matizar también que la adquisición de Memoria *RAM* de procesos separados, es posible en dispositivos marcados como producción (o el emulador), a través del *Android Device Monitor*.

3.2.7. FASE DE GESTIÓN DE EVIDENCIAS

La gestión de evidencias es un proceso fundamental para la validez de todo el proceso de análisis forense.

Una buena gestión de evidencias asegura que la cadena de custodia sea respetada y que, por lo tanto, las evidencias no han sido comprometidas.

Indicar que la cadena de custodia es el conjunto de procedimientos encaminados a la recogida, el traslado y la custodia de las evidencias relativas a una investigación.

La cadena de custodia tiene como objetivo garantizar la autenticidad, inalterabilidad e indemnidad de las evidencias.

Por tanto, la cadena de custodia permite:

- Trazar los elementos físicos correspondientes a una evidencia en particular.
- Identificar el origen del elemento físico utilizado como evidencia.
- Asegurar que el acceso a una evidencia es controlado y registrado.
- Documentar todos los procesos realizados para extraer las evidencias.
- Demostrar que los procesos anteriores son reproducibles y replicables.

3.2.8. FASE DE EXAMEN

El examen consiste en identificar las evidencias a partir de la información obtenida en la fase de adquisición.

En el análisis de un disco:

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

En el análisis de red:

- Descartar paquetes que no sean relevantes.

En el análisis de la memoria:

- Descartar procesos que no sean relevantes.
- Extraer la información relevante de los procesos.

3.2.9. FASE DE ANÁLISIS

El análisis consiste en obtener conclusiones a partir de las evidencias obtenidas.

Es la fase más completa del proceso, y la que más libertad ofrece, por lo que suele variar en función del analista.

En ocasiones, el análisis de las evidencias puede originar una nueva fase de examen y extracción para hacer visibles nuevas evidencias.

Para ello, se procede mediante un proceso iterativo:

- Construir una hipótesis en base a la información existente sobre el caso.
- Probar la hipótesis con las evidencias existentes. Hay que tener en cuenta también la posible existencia de contra-evidencias.

Un ejemplo, podría ser el siguiente:

- **Hipótesis**: el sujeto se encontraba en el lugar del crimen a una hora determinada.
- **Evidencias**: El historial de localizaciones del dispositivo indica que estaba a 15 km.
- **Técnicas antiforense**: el dispositivo ha sido manipulado y la fecha de última modificación del fichero del historial de localizaciones es inconsistente con la fecha de los eventos.

3.2.10. FASE DE ANÁLISIS - FORMATO DE DATOS

Durante la etapa de análisis se revisan y estudian las evidencias adquiridas. Dada la ingente cantidad de información que almacenan los teléfonos móviles hoy en día, no se recomienda utilizar una estrategia en la que se extraiga toda la información posible del dispositivo sin orden ni justificación.

Dependiendo del origen de las evidencias y el caso concreto, se deberán formular una serie de hipótesis, donde se describan:

- Los principales tipos de datos que se pueden encontrar en un dispositivo.
- La forma de analizarlos dependiendo del tipo de evidencia adquirida.
- Los principales tipos de datos en las plataformas predominantes.

Independientemente de la plataforma o sistema operativo, muchas aplicaciones utilizan los mismos formatos para el almacenamiento de información persistente.

El conocimiento de la estructura y componentes de estos tipos de archivo puede servir para identificar la existencia de información almacenada en un formato específico, incluso cuando ha sido borrado del sistema.

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

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

FICHEROS *XML*

Los ficheros *XML* (en inglés *eXtensible Markup Language*) son ficheros de texto que contienen información estructurada a través de lo que se denominan marcas.

Se utilizan principalmente para el almacenamiento de las preferencias.

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

- Dependiendo de la plataforma, el contenido de los ficheros será diferente.
- Normalmente, siempre empiezan con la siguiente línea:

<?xml version="1.0" encoding="UTF-8"?>

Por último, decir que los ficheros *XML*, normalmente tienen extensión *xml*, pero esto no siempre es así, dado que podemos encontrar también otras extensiones (por ejemplo, *plist* en *iOS*).

Además, los ficheros *plist* incluyen también una cabecera y tienen *tags* (etiquetas) específicos:

- ***<?plist version="1.0"?>***.
- ***<key><dict><integer>***.

FICHEROS *SQLite*

Los ficheros *SQLite* están organizados en páginas de tamaño fijo, que se rellenan desde abajo.

Al igual que en los sistemas de archivos, cuando el contenido de la página no es necesario, se marca como vacía, pero no se borra (eficiencia). Algunos editores permiten inspeccionar este contenido, como por ejemplo, *SQLite Viewer*[4].

El almacenamiento se realiza en ficheros con diferente extensión, siendo *sqlite* y *db* las más utilizadas:

[4]Disponible en ***http://www.sqliteviewer.org/***

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

Independientemente de su extensión, todos los ficheros *SQLite* empiezan con el *string SQLite format 3* (para la versión 3 del formato). Puede ser utilizado para buscar ficheros *sqlite* borrados del sistema.

FOTOGRAFÍAS - *EXIF*

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

En el caso de los dispositivos móviles, además del modelo de dispositivo y configuración de la cámara, los datos *EXIF* también pueden ofrecer información sobre la localización en la que fue tomada una imagen.

Este tipo de información puede ser muy importante a la hora de establecer líneas de tiempo y localizar el dispositivo en lugares que estén relacionados con los hechos que se están investigando.

FICHEROS DE TEXTO PLANO

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

- Texto de notas.
- Configuración de aplicaciones, etc.

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

En muchas ocasiones, las claves y la ocurrencia a buscar tendrá que ver con el caso específico que se esté investigando.

Algunas de las claves que pueden ser interesantes incluyen:

- *Password*, *pass*, *pass= password=*.
- *User*, *location*.
- Nombres de personas, lugares, etc.

3.2.11. FASE DE ANÁLISIS - TIPOS DE ANÁLISIS

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

- Análisis de archivos binarios ejecutables.
- Análisis de sistema de ficheros.

- Análisis de espacio borrado – *File carving*.
- Análisis de memoria.
- Análisis de *backup*.

ANÁLISIS DE ARCHIVOS BINARIOS EJECUTABLES

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

- Una instrucción por *malware*.
- Necesidad de extracción de datos de una aplicación específica.

Específicamente, los siguientes elementos pueden resultar de interés de cara a una investigación forense:

- Credenciales almacenadas por la aplicación.
- Datos de la aplicación, como por ejemplo, el historial de conversaciones (*WhatsApp*), historial de compras, etc.
- Interacción de la aplicación con las *API* del sistema.

Una vez identificados los elementos de interés, utilizando diferentes técnicas de análisis, como el análisis estático y dinámico de aplicaciones *Android*, se procederá al análisis de los mismos.

Por último, indicar que para dar validez al análisis forense es necesario documentar y validar el proceso de extracción de información, si no ha sido documentado previamente por otros peritos.

ANÁLISIS DE SISTEMA DE FICHEROS

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

La localización de los diferentes elementos dependerá de la plataforma, versión, dispositivo, etc. Tiene el beneficio de que, normalmente es consistente entre todos los dispositivos de la misma plataforma y versión.

Además, el análisis del sistema de ficheros se realiza por lo general, mediante el montaje de las imágenes adquiridas en modo de sólo lectura. De esta manera, se puede navegar por la estructura de ficheros del sistema en busca de datos o artefactos de interés.

Finalmente, dependiendo del sistema de adquisición de datos, una vez montado el disco, también se puede realizar un análisis del espacio no utilizado por el mismo.

ANÁLISIS DE ESPACIO BORRADO – *FILE CARVING*

Normalmente, en los sistemas de ficheros tradicionales, borrar un archivo sólo marca como disponibles los bloques del disco en los que estaba almacenado el archivo. Por tanto, el contenido de los bloques permanece intacto hasta que el sistema de ficheros los necesita.

Dependiendo del tipo de archivo, su tamaño y el estado de los bloques en los que se encontraba almacenado, se podrán recuperar aquellos datos en bloques, que no hayan sido

sobreescritos por otros archivos del sistema para el análisis.

Para el análisis de espacio borrado hay que tener en cuenta los tipos de archivos que se quieren recuperar. Por lo que, dependiendo del tipo de archivo e información a recuperar podremos proceder de un modo u otro.

La mayoría de ficheros de interés tienen un inicio de cabecera específico (*MAGIC NUMBER*):

- *SQLite Format 3* (en notación *ASCII*), para ficheros *SQLite*.
- *%PDF* (en notación *ASCII*), para ficheros *pdf*.
- *\211PNG\r \n* (en notación *ASCII*, para archivos *png*).
- *FFD8* (en hexadecimal), para archivos *jpeg*.

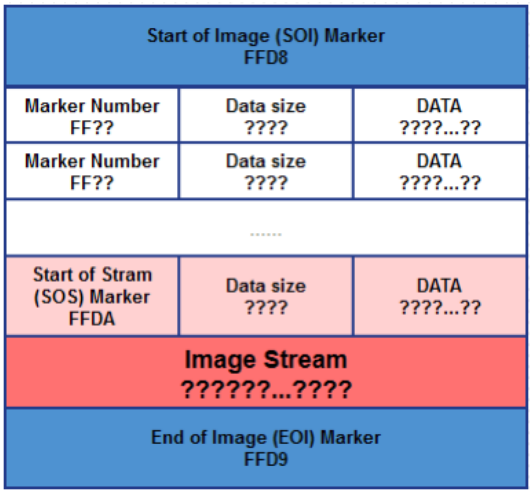


Figura 26: Estructura de una archivo *JPEG*.

A su vez, el tamaño del archivo es descrito en la cabecera del mismo:

- Si el archivo es menor que el tamaño del bloque no hará falta buscar.
- Si el archivo es mayor, primero se buscará en los bloques contiguos y después, en otros bloques del disco, si no ha tenido éxito (con diferentes heurísticas).

Afortunadamente, existen herramientas como *Scalpel* (disponible en *Santoku*), que realizan esta tarea de forma automática.

Por útlimo, indicar que en algunas ocasiones, no siempre es necesario recuperar el archivo completo. Por ejemplo, para recuperar una contraseña se pueden realizar búsquedas de cadenas como *password*, *pass*, etc. Este tipo de búsqueda se puede realizar con el programa de consola *strings* (disponible en *Santoku*) o con un editor hexadecimal.

ANÁLISIS DE MEMORIA

Consiste en analizar un volcado de la memoria:

- El volcado puede ser de la memoria completa del dispositivo.
- O de un proceso únicamente.

Se puede realizar de dos maneras:

- En bruto: analiza la memoria como un *stream* de *bytes*. Permite buscar *strings* y otros datos, pero el análisis de variables, de código, etc... Es más complejo.
- Organizado: utiliza un mapa de la memoria, para interpretar los diferentes valores y estructura del fichero de imagen capturado. Permite distinguir las partes de código y datos de la memoria. Al igual que en el sistema de ficheros, la organización de la memoria del dispositivo depende del terminal y versión del sistema operativo utilizado.

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

ANÁLISIS DE *BACKUP*

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

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

La estructura y localización de los ficheros almacenados en la copia de seguridad es diferente de la correspondiente estructura física del dispositivo.

Para comprobar la precisión de los datos almacenados en la copia de seguridad se puede utilizar un dispositivo para volcar la copia.

En el caso de que la copia de seguridad haya sido cifrada, será necesario averiguar la contraseña de la copia de seguridad:

- *Phone PasswordBreaker*, permite extraer la contraseña utilizada mediante ataques de de fuerza bruta y se encuentra disponible en *https://www.elcomsoft.com/eppb.html*.
- Indicar que aunque es posible en *Android*, este tipo de herramientas no es capaz de extraer la copia de seguridad cifrada en otros dispositivos, como por ejemplo en *BlackBerry 10*. Para ello, se necesitan las credenciales de *BlackBerry Link*.

3.2.12. FASE DE ANÁLISIS - ANÁLISIS DE DATOS

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

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

- Ficheros de texto en claro.
- Ficheros binarios: imágenes.

Los ficheros se pueden almacenar en:

- Almacenamiento interno del dispositivo (protegidos del acceso de otras aplicaciones si no está rooteado).
- Almacenamiento externo del dispositivo (accesible sin problemas por el resto de aplicaciones).

SISTEMA DE FICHEROS

El sistema de ficheros de *Android* está dividido en varias particiones:

- *boot loader*: partición de sólo lectura. El primer código que se ejecuta al encender el teléfono. Carga el *kernel* de *Android*.
- *boot*: incluye el *kernel* de *Android*.
- *splash*: guarda la imagen que se muestra al iniciar el dispositivo.
- *userdata*: almacena todos los datos del usuario (aplicaciones, fotografías, etc.).
- *System*: incluye las librerías, las aplicaciones del sistema y el *framework* de *Android*.
- *cache*: almacena los ficheros utilizados temporalemente por las diferentes aplicaciones (incluyendo la máquina virtual de *Dalvik*).

Partition Name	Description
boot	RAMDisk, kernel
cache	App and OS cache area: may include artifacts
data	User data, settings, applications, and third-party apps for some Samsung devices; can be represented as userdata as well
dbdata	User data, settings, and some stock applications and settings (Samsung)
emmc	Internal media card
misc	System feature settings; used by the device for configuration and hardware settings
modem	Firmware for modem; hardware dependent
radio	Firmware for radio, cellular, GPS, data connection, and Bluetooth; hardware dependent
recovery	Device stock recovery image often used as alternative boot partition by mobile forensic tools; no user data here unless hidden by user
sdcard	Internal or external media card; for some phones, other partitions could be sd, emmc, and so on, depending on device type and whether internal or external to the device
system	Operating system and settings, built-in application settings
userdata	User data, settings, applications and third-party apps for some Samsung devices; can also be represented as data
wimax	Firmware for WiMAX; hardware dependent

Figura 27: Particiones.

Todos los procesos tienen el acceso a ciertos directorios restringidos (incluido el proceso de *adb*).

DIRECTORIOS DE INTERÉS – APLICACIONES DEL SISTEMA

Las aplicaciones del sistema en *Android* se localizan en ***/system/app***. Hay que tener en cuenta que esta partición es de sólo lectura. Sólo almacena los ficheros *APK* y ficheros *odex* (código de la aplicación preparado para su ejecución en la máquina virtual).

```
root@condor_umts:/ # cd /system/app/
3c_main.apk
3c_main.odex
AonIntLT.apk
AonIntLT.odex
BasicDreams.apk
BasicDreams.odex
Bluetooth.apk
Bluetooth.odex
BluetoothExt.apk
BluetoothExt.odex
Books.apk
BrowserProviderProxy.apk
Bug2GoStub.apk
Calculator.apk
Calculator.odex
CellBroadcastReceiver.apk
CellBroadcastReceiver.odex
CertInstaller.apk
CertInstaller.odex
Chrome.apk
```

Figura 28: Directorios de interés 1.

Los datos generados por las aplicaciones son guardados en ***/data/data/***.

```
root@condor_umts:/data/data # ls -las
total 552
drwxr-x--x u0_a3    u0_a3            1970-02-26 01:42 com.android.backupconfirm
drwxr-x--x bluetooth bluetooth         1970-02-26 01:44 com.android.bluetooth
drwxr-x--x u0_a49   u0_a49           1970-02-26 01:43 com.android.browser.provider
drwxr-x--x u0_a50   u0_a50           1970-02-26 01:43 com.android.calculator2
drwxr-x--x u0_a6    u0_a6            1970-02-26 01:44 com.android.calendar
drwxr-x--x u0_a51   u0_a51           1970-02-26 01:44 com.android.cellbroadcastreceiver
drwxr-x--x u0_a52   u0_a52           1970-02-26 01:43 com.android.certinstaller
drwxr-x--x u0_a53   u0_a53           2016-02-13 18:02 com.android.chrome
drwxr-x--x u0_a9    u0_a9            1970-02-26 01:42 com.android.contacts
drwxr-x--x u0_a12   u0_a12           2016-02-13 15:43 com.android.defcontainer
drwxr-x--x u0_a14   u0_a14           1970-02-26 01:44 com.android.deskclock
drwxr-x--x u0_a16   u0_a16           1970-02-26 01:44 com.android.dialer
drwxr-x--x u0_a55   u0_a55           1970-02-26 01:43 com.android.documentsui
drwxr-x--x u0_a47   u0_a47           1970-02-26 01:43 com.android.dreams.basic
drwxr-x--x u0_a79   u0_a79           1970-02-26 01:43 com.android.dreams.phototable
drwxr-x--x u0_a18   u0_a18           1970-02-26 01:44 com.android.email
drwxr-x--x u0_a57   u0_a57           1970-02-26 01:44 com.android.exchange
drwxr-x--x u0_a19   u0_a19           2016-02-13 16:16 com.android.externalstorage
drwxr-x--x u0_a65   u0_a65           1970-02-26 01:43 com.android.htmlviewer
```

Figura 29: Directorios de interés 2.

DIRECTORIOS DE INTERÉS – APLICACIONES DE TERCEROS

Los archivos *APK* se encuentran en ***/data/app***.

```
root@condor_umts:/data/app # ls -las
total 105028
-rw-r--r-- system   system   32940855 2016-02-13 20:35 com.facebook.katana-1.apk
-rw-r--r-- system   system   46874450 2016-02-13 20:39 com.snapchat.android-1.apk
-rw-r--r-- system   system   27726782 2016-02-13 20:37 com.spotify.music-1.apk
```

Figura 30: Directorios de interés 3.

Las *sandbox* están localizadas en */data/data/*.

```
drwxr-x--x u0_a74   u0_a74          1970-02-26 01:43 com.motorola.motosignature.app
drwxr-x--x u0_a84   u0_a84          1970-02-26 01:43 com.motorola.pgmsystem2
drwxr-x--x radio    radio           1970-02-26 01:43 com.motorola.programmenu
drwxr-x--x u0_a38   u0_a38          1970-02-26 01:45 com.motorola.setup
drwxr-x--x u0_a41   u0_a41          2016-02-13 17:48 com.motorola.so
drwxr-x--x u0_a45   u0_a45          1970-02-26 01:44 com.motorola.wappushsi
drwxr-x--x system   system          1970-02-26 01:44 com.qualcomm.atfwd
drwxr-x--x u0_a67   u0_a67          1970-02-26 01:43 com.qualcomm.interfacepermissions
drwxr-x--x system   system          1970-02-26 01:43 com.qualcomm.location
drwxr-x--x u0_a85   u0_a85          1970-02-26 01:43 com.qualcomm.qcom_qmi
drwxr-x--x radio    radio           1970-02-26 01:44 com.qualcomm.qcrilmsgtunnel
drwxr-x--x system   system          1970-02-26 01:44 com.qualcomm.qualcommsettings
drwxr-x--x system   system          1970-02-26 01:44 com.qualcomm.services.location
drwxr-x--x u0_a89   u0_a89          2016-02-13 15:43 com.qualcomm.timeservice
drwxr-x--x u0_a86   u0_a86          1970-02-26 01:43 com.quickoffice.android
drwxr-x--x u0_a95   u0_a95          2016-02-13 20:37 com.spotify.music
drwxr-x--x u0_a93   u0_a93          2016-02-13 18:02 eu.chainfire.supersu
drwxrwx--- media    media           1970-02-26 01:42 media
drwxr-x--x bluetooth bluetooth        1970-02-26 01:43 org.codeaurora.bluetooth
```

Figura 31: Directorios de interés 4.

Se puede comprobar como cada aplicación se encuentra asociada a un usuario específico.

ESTRUCTURA DE LA *SANDBOX* DE UNA APLICACIÓN

El directorio de cada aplicación de *Android* está estructurado en las siguientes carpetas (no todas están presentes en todas las aplicaciones):

- ***files***: almacena los ficheros que la aplicación pueda generar y necesitar durante su ejecución. Puede servir de almacenamiento para fotos u otros elementos utilizados por la aplicación.
- ***lib***: almacena enlaces a los directorios en los que se encuentran las librerías compiladas específicamente para la plataforma que necesita la aplicación.
- ***shared_prefs***: almacena los ficheros de *SharedPreferences* creados por la aplicación.
- ***databases***: almacena los ficheros *sqlite* (*providers*) que utiliza la aplicación.
- ***cache***: ficheros temporales utilizados por la aplicación.

```
root@condor_umts:/data/data/com.spotify.music # ls
app_MixpanelAPI.Images.DecideChecker
app_MixpanelAPI.Images.ViewCrawler
cacert.pem
cache
code_cache
databases
files
lib
shared_prefs
```

Figura 32: Estructura de la *Sandbox* de una aplicación.

DATOS DE INTERÉS - FOTOGRAFÍAS

El directorio de almacenamiento de las fotografías en *Android* depende en gran medida del fabricante y del tipo de *ROM* exacta:

- */storage/emulated/DCIM*, si el dispositivo no tiene una tarjeta *SD*.
- */storage/sdcardX/DCIM*, si el dispositivo tiene una tarjeta *SD*.

Dependiendo del uso del teléfono, es posible que haya fotografías en ambas localizaciones.

En el interior de la carpeta *DCIM*, las fotos generadas con la cámara del dispositivo se guardan en la carpeta *Camera*.

```
root@condor_umts:/storage/emulated/0/DCIM # ls
Camera
root@condor_umts:/storage/emulated/0/DCIM # cd Camera
root@condor_umts:/storage/emulated/0/DCIM/Camera # ls
IMG_20160213_201017879.jpg
IMG_20160213_201022555.jpg
root@condor_umts:/storage/emulated/0/DCIM/Camera #
```

Figura 33: Contenido de la carpeta *DCIM*.

Independientemente del dispositivo de almacenamiento, las fotografías pueden ser accedidas desde el propio equipo del analista a través de la conexión *USB* (o la imagen de la tarjeta *SD*).

DATOS DE INTERÉS – CACHÉ DEL TECLADO

Android guarda en un *Content Provider* las palabras seleccionadas por el usuario como parte del sistema predictivo del teclado:

/data/data/com.android.providers.userdictionary/database/user_dict.db

Las palabras introducidas en campos de contraseñas no se almacenan en este diccionario.

Además, la caché del teclado tampoco contiene marcas de tiempo.

DATOS DE INTERÉS – CONTRASEÑAS Y CONFIGURACIONES

En el caso de que se haya configurado el código de bloqueo:

- Se encuentra en ***/data/system/gesture.key***, si se trata de un patrón de bloqueo:
  - A cada punto del patrón se le asigna un número (empezando por el cero desde la esquina superior izquierda).
  - Se hace un resumen de la concatenación del valor en *byte* del patrón. Por ejemplo, *01254*.
  - Dado el reducido número de combinaciones posibles, se puede obtener fácilmente a través de la generación de todas las combinaciones en *SHA1*[5].

[5] *http://forensics.spreitzenbarth.de/2012/02/28/cracking-the-pattern-lock-on-android/*

- Se encuentra en ***/data/system/password.key***, si se trata de un código numérico:
  - Almacena el resumen del código numérico resumido junto con una semilla en *SHA* y *MD5* (ambos concatenados).
  - El lugar de almacenamiento de la semilla depende de la versión de *Android*.
    - ***/data/data/com.android.providers.settings/databases/settings.db***
    - ***/data/system/locksettings.db***
  - En ambos ficheros se guarda en *lockscreen.password_salt*.
  - Una vez obtenida la semilla y el *hash* final, se puede realizar un ataque de diccionario para obtener el *PIN* o *password* original[6].

REDES *WIFI* CONECTADAS

Los datos de las redes *WiFi* a las que ha tenido acceso el dispositivo se encuentran en */data/misc/wifi*.

El fichero *wpa_supplicant.conf* almacena la información relativa a la configuración de los puntos de acceso *WiFi*.

Además de la lista de últimas redes *WiFi* a las que se ha conectado el dispositivo, también es posible encontrar las contraseñas de acceso a las mismas. Entre estos datos, se pueden encontrar claves de acceso a entornos corporativos.

```
root@condor_umts:/data/misc/wifi # cat wpa
wpa_supplicant.conf wpa_supplicant/
t wpa_supplicant.conf
mot_wpa_conf_version=3
ctrl_interface=/data/misc/wifi/sockets
disable_scan_offload=1
driver_param=use_p2p_group_interface=1
update_config=1
device_name=condor_retgb
manufacturer=motorola
model_name=XT1021
model_number=XT1021
serial_number=ZX1B23QDHQ
device_type=10-0050F204-5
config_methods=physical_display virtual_push_button
p2p_disabled=1
p2p_no_group_iface=1
country=US

network={
        ssid="SK    5BA"
        psk="          "
        key_mgmt=WPA-PSK
        priority=1
}
```

Figura 34: Información de las redes *WiFi* conectadas.

DATOS DE INTERÉS - CALENDARIO

La agenda de eventos del sistema se almacena en un *Content Provider* localizado en ***/data/data/com.android.providers.calendar/databases/calendar.db***.

Esta base de datos incluye información sobre eventos, sus asistentes, recordatorios y alertas.

[6] *http://forensics.spreitzenbarth.de/2015/08/12/breaking-the-screenlock-a-short-update/*

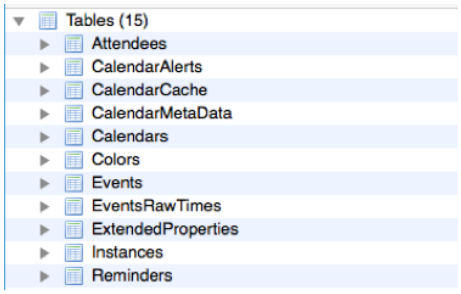


Figura 35: Datos de interés 1 (calendario).

La información de esta tabla puede ser de especial interés, dado que contiene el calendario que el usuario ha sincronizado con su cuenta de *Google*.

DATOS DE INTERÉS – MENSAJES DE TEXTO

Los mensajes de texto se almacenan en un *Content Provider*[7].

Los mensajes de texto se almacenan dentro de la tabla *SMS*.

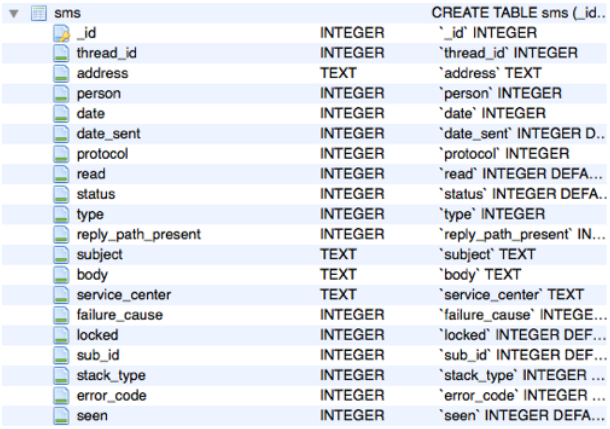


Figura 36: Datos de interés 2 (mensajes de texto).

El *provider* también ofrece otras tablas, para acceder a los mensajes (a través de *threads*).

En cuanto a la *URI* de los *MMS*, ésta se encuentra en la tabla *attachments*.

DATOS DE INTERÉS – NAVEGADOR

En *Android*, el navegador por defecto depende del fabricante y de la versión del SO:

- ***/data/data/com.android.chrome*** para *Chrome*:
- Los datos de interés se almacenan en la carpeta ***app_chrome/Default/***.
- Ficheros de interés en la misma carpeta:
  - *Login Data*: Fichero *SQLite* con credenciales de acceso a páginas *web*.

[7] Localizado en ***/data/data/com.android.providers.telephony/databases/mmssms.db***.

- *Cookies*: Fichero *SQLite* con las *cookies* de los sitios visitados.
- *Bookmarks*: Fichero *JSON* con los favoritos guardados en el navegador.
- *History*: Fichero *SQLite* con el historial de páginas visitadas.
- *Web Data*: Fichero *SQLite* con información de auto-relleno de formularios.

- ***/data/data/com.android.browser*** para navegadores por defecto que no sean *Chrome*:
  - La base de datos que almacena las contraseñas se guarda en:

/data/data/com.android.browser/databases/webview.db

DATOS DE INTERÉS – CONTACTOS Y LLAMADAS

Los contactos y llamadas del dispositivo se almacenan en el *ContentProvider*, el cual se encuentra en ***/data/data/com.android.providers.contacts/***.

Dependiendo del fabricante y de la operadora, la localización puede variar.

La base de datos que almacena esta información se encuentra en ***databases/contacts2.db***:

- Los contactos se almacenan en las tablas *contacts*, *raw_contacts* y *deleted_contacts*.
- Las llamadas recientes se almacenan en la tabla *call*.

Además, la carpeta *photos* almacena todas las fotos asociadas a contactos en el dispositivo.

La lista de llamadas recientes del dispositivo se encuentra en el mismo archivo *databases/contacts2.db*, en la tabla *call*.

DATOS DE INTERÉS – CORREO ELECTRÓNICO

Al igual que en el caso del navegador, depende del tipo de cuenta que tenga configurada el usuario.

Para las cuentas de *Gmail*, una base de datos por cada cuenta:

- ***/data/data/com.google.android.gm/databases/mailstore***.[CUENTA].***db***.
- Incluye tablas como:
  - *Converstations*.
  - *Attachtments*.
  - *Messages*.

Las demás cuentas de correo, se pueden localizar en el *Content Provider*:

/data/data/com.android.email/databases/EmailProvider.db

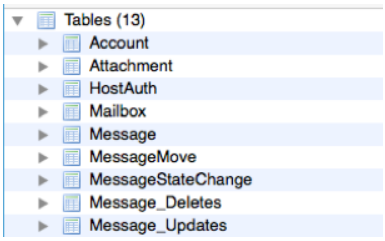


Figura 37: Datos de interés 3 (correo electrónico).

DATOS DE INTERÉS – DATOS GEOGRÁFICOS

Los datos geográficos en *Android* se almacenan principalmente en la aplicación de *Google Maps*.

Su carpeta se encuentra en ***/data/data/com.google.android.apps.maps***.

Dentro de esta carpeta existen varios datos de interés:

- La carpeta *cache* almacena imágenes de partes de los mapas y *Street View*, vistos anteriormente.
- En el fichero ***databases/gmm_myplaces.db*** se almacenan los lugares que el usuario ha añadido a favoritos en la aplicación.

3.2.13. FASE DE PRESENTACIÓN

La presentación consiste en describir los diferentes sucesos probados y las evidencias que los corroboran.

Normalmente, esta fase consiste en la elaboración de un informe forense, el cual va a ser leído por personal que no es técnico (jueces, ejecutivos, etc.), por lo que debe ser claro y contener un lenguaje que se adapte al perfil adecuado.

En caso de que sea escrito para un proceso judicial, es posible que sea necesaria su defensa ante el juez.

3.3. HERRAMIENTAS BÁSICAS

Si bien el análisis forense depende en gran medida de la plataforma para la que se está realizando, existen un conjunto de herramientas básicas que pueden ayudar durante todo proceso de análisis forense.

En este sentido, se van a presentar las principales herramientas y programas de utilidad que se pueden necesitar durante las diferentes fases del análisis forense.

SUITES FORENSES

El *software* de desarrollo de *Android* – gratuito y disponible en *Internet* – y las herramientas de código libre son útiles para resolver la mayor parte de las situaciones. Su dominio

es condición imprescindible para dedicarse profesionalmente a la investigación de dispositivos móviles *Android*. No obstante, es posible que un perito necesite adquirir uno o varios productos comerciales.

Las *suites* forenses de nivel comercial, incluyen de forma general, varias de estas herramientas integradas en un único producto, facilitando así las tareas de análisis:

- *EnCase Forensic* (*Guidance Software*) – *https://www.guidancesoftware.com*.
- *Oxygen Forensics* (*Oxygen Forensics*) – *http://www.oxygen-forensic.com/*.
- *Forensic ToolKit* (*Access Data*)[8].
- *UFED* (*Cellebrite*) – *http://www.cellebrite.com/Mobile-Forensics/Applications*.

Tool	Logical	File System	Physical (NI)*	Physical (I)*	Limited Support
BlackLight	X	X			X
UFED 4PC	X	X	X	X	
Device Seizure	X	X	X	X	
EnCase	X	X	X		
Lantern	X	X	X		X
MOBILedit Forensic	X	X			
MPE+	X	X	X	X	
Oxygen Forensic Analyst	X	X	X	X	
Secure View	X				
XRY	X	X	X	X	

*(NI = non-invasive; I = invasive)

Figura 38: *Suites* forenses de nivel comercial.

HERRAMIENTA DE ADQUISICIÓN - *dd*

La herramienta *dd* es una herramienta de consola disponible en la mayoría de sistemas *UNIX*. De hecho, está incluida en *Santoku Linux*, *Android* y dispositivos *iOS* con *jailbreak*. Esta herramienta puede escribir y leer de dispositivos directamente a través del *driver* de bajo nivel sin pasar por el SO.

Esta característica hace que sea una herramienta de especial interés para el copiado en bruto de discos duros, memorias *flash* y *RAM*, pues copia *bit* a *bit* los datos ofrecidos por el driver de bajo nivel del dispositivo copiado.

En el caso de la memoria *RAM* debe cargarse en la misma para su ejecución, por lo que es modificada (la huella de la utilidad en memoria es mínima).

Para su ejecución basta con:

[8]Disponible en *http://accessdata.com/solutions/digital-forensics/forensic-toolkit-ftk*

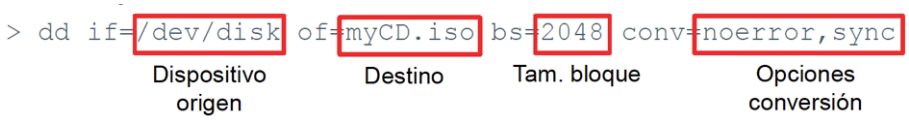


Figura 39: Herramienta *dd*.

HERRAMIENTA DE ANÁLISIS – Visor hexadecimal

Durante el análisis forense, en más de una ocasión será necesario analizar ficheros de datos en formato *raw*. La inspección de estos ficheros con editores de texto no es posible, dado que muchos de los caracteres mostrados no son imprimibles.

Un editor hexadecimal permite mostrar el contenido de un fichero con dos vistas:

- Una muestra la conversión de los datos a hexadecimal.
- Otra muestra los caracteres imprimibles si los tiene.

De esta manera, se pueden realizar búsquedas e incluso reemplazar el contenido de un fichero binario editando directamente sus caracteres imprimibles o valores en hexadecimal (según convenga).

Algunos de los editores hexadecimales existentes son:

- *iHex* (*Mac OS X*) – Disponible en la *App Store*.
- *Bless* (*Linux*) – ***http://home.gna.org/bless/donwloads.html***
- *HxD* (*Windows*) – ***https://mh-nexus.de/en/hxd/***

HERRAMIENTA DE ANÁLISIS – Editor *SQLite*

SQLite es un motor muy popular de base de datos que se utiliza en la mayoría de aplicaciones móviles, para la persistencia de datos.

Las bases de datos *SQLite* se almacenan en ficheros con extensión *sqlite* (aunque también utilizan otras extensiones como *db*, *sqlitedb*, *sqlite3*, etc.).

Para inspeccionar el contenido de este tipo de ficheros existe un visor *SQLite*. Además, existen multitud de editores *SQLite* para todas las plataformas:

- *DB Browser*, es un proyecto de *software* libre disponible en todas las plataformas – ***http://sqlitebrowser.org***.
- *Sqliteman* – Disponible en *Santoku Linux*.

HERRAMIENTA DE ANÁLISIS – Editor de textos

El editor de textos sirve para acceder a la información que se encuentra almacenada en ficheros de texto durante el transcurso del análisis.

Esta información puede incluir, entre otras cosas:

- Ficheros *XML*.
- Ficheros de configuración.
- Ficheros con texto utilizados por aplicaciones.

Actualmente, existen multitud de editores disponibles en cualquier plataforma:

- *Atom* (multiplataforma) – ***https://atom.io***.
- *Sublime Text* (multiplataforma) – ***https://www.sublimetext.com/***
- *Leafpad* – Disponible en *Santoku Linux*.

HERRAMIENTA DE ANÁLISIS – Herramientas de consola

Existen multitud de herramientas de consola que pueden ser útiles para el perito forense:

- ***Grep***: Herramienta para la búsqueda de expresiones regulares. El comando *grep* permite realizar búsquedas de caracteres en el interior de un archivo, en un directorio determinado, un árbol de directorios o incluso en todo el sistema de archivos. Una particularidad de *grep* es que puede distinguir entre mayúsculas y minúsculas.
- ***Strings***: Identifica las cadenas de texto imprimible en un archivo binario.
- ***Exiftool***: Extrae los metadatos de una fotografía.

Estas herramientas se encuentran instaladas en cualquier distribución de *Linux* (*Santoku* incluido).

SLEUTH KIT Y AUTOPSY

The Sleuth Kit (*TSK*) es una colección de herramientas de consola y una librería que permiten el análisis de imágenes de disco y la recuperación de archivos de las mismas.

Autopsy es un interfaz que utiliza *Sleuth Kit* para la gestión de casos mediante *Autopsy*. El perito forense puede crear un nuevo caso forense, cargar imágenes de adquisiciones, generar hashes *MD5* de diferentes elementos de la imagen, navegar por la estructura de ficheros o por los bloques de la imagen, añadir notas sobre el análisis forense que está realizando, etc.

Sleuth Kit y *Autopsy* están disponibles en *Santoku Linux*. Para abrirlo basta con ejecutar en consola "autopsy" (debido a la instalación, es necesario ejecutarlo en modo *root*).

La versión más reciente de *Autopsy* sólo se encuentra disponible en entornos *Windows* – *http://www.sleuthkit.org/*.

3.4. EL INFORME FORENSE

El producto final de la actividad del perito es un informe con los resultados de su trabajo, que posiblemente tenga que ser expuesto y defendido ante un tribunal.

En general, el informe forense debe incluir las siguientes secciones:

- Sumario o resumen del caso (descripción detallada y exacta de los hechos).

- Herramientas utilizadas.
- Adquisición de evidencias (los elementos de evidencia hallados y cualquier otro extremo que pudiera resultar de interés).
- Procesado de evidencias.
- Análisis de evidencias.
- Conclusiones.

Además de lo anterior, el informe forense debe incluir los resúmenes criptográficos de todas las evidencias y ficheros recogidos que se han mencionado en el análisis.

Indicar también, que el perito no debe hacer conjeturas ni aventurar conclusiones sobre las cuales corresponda decidir a jueces y fiscales. Su labor consiste en hallar elementos de evidencia que habrán de ser interpretados de modo profesional en el contexto técnico dentro del cual desempeña su labor.

INFORME EJECUTIVO E INFORME TÉCNICO

El informe ejecutivo debe ser claro, conciso y no debe contener lenguaje técnico, dado que por lo general va dirigido a gerentes, fiscales y/o jueces, que tienen poca relación con la informática.

El informe técnico debe detallar todos los procedimientos realizados, utilizar información técnica que permita a cualquier persona que siga esos pasos conseguir los mismos resultados que hemos conseguido.

SUMARIO O RESUMEN DEL CASO

Cada actuación debe ajustarse a derecho y a catálogos de buenas prácticas, para que el informe no pueda ser impugnado por omisiones o defectos de forma. Aunque dependiendo del objetivo y ámbito del mismo, puede ser necesario ajustar la estructura del informe forense.

Esta sección debe mencionar:

- Las razones por las que se está llevando a cabo el análisis forense.
- Como han llegado las pruebas al analista (cadena de custodia). Debe existir una trazabilidad adecuada de todo aquello que haya servido como material para elaborarlo. Especial cuidado ha de ponerse en el elemento fundamental de toda la ciencia forense: el mantenimiento de la cadena de custodia.
- Quién ha solicitado el informe forense.
- Las fechas más importantes en relación al informe: fecha de solicitud, recepción de evidencias y tiempo utilizado para la elaboración del informe.

En algunos casos, el sumario incluye también un resumen de los principales resultados del análisis, por lo que hay que tener cuidado con la introducción de esta información para no predisponer al lector.

Indicar también que el informe no solo es un resumen del proceso de investigación; forma parte de él, y como tal ha de cumplir unos determinados requisitos de solvencia: aceptabilidad, integridad, credibilidad, existencia de una relación causa-efecto, carácter repetible y

una documentación coherente y completa. Por ese motivo, numerosos investigadores elaboran sus informes de acuerdo con estándares reconocidos, como por ejemplo la Directiva *RFC 3227* o la Norma *ISO/IEC 27037*.

HERRAMIENTAS UTILIZADAS

El resumen del caso también debe describir todas las herramientas de terceros utilizadas para el análisis.

Para cada una de las herramientas será necesario especificar:

- La versión de la herramienta utilizada (incluyendo la plataforma).
- El fabricante.
- La tarea para la que se ha utilizado.

Si se ha desarrollado alguna herramienta específica para el análisis, se deberá mencionar en esta sección, pero también se deberá añadir un anexo en el que se demuestre la necesidad y validez de la herramienta.

En algunos casos, esta sección puede ser dividida a su vez en subsecciones en el informe.

ADQUISICIÓN DE EVIDENCIAS

Se debe detallar el proceso de interacción con las evidencias. Para ello, se deben documentar los siguientes pasos de la forma más detallada posible:

- **Momento** en el que el analista entra en contacto con las evidencias.
- **Estado** en el que se reciben las evidencias (con fotos identificativas y describiendo los números de serie de los dispositivos si los tienen).
- **Procesos ejecutados** para preservar cada una de las evidencias recibidas. Estos procesos deben incluir la configuración de los dispositivos o entornos en los que se preservarán las evidencias.
- **Marcadores de integridad** de todas las copias y evidencias recolectadas. Aunque algunas herramientas utilizan *MD5*, es recomendable utilizar estándares superiores como *SHA-256*, dado que *MD5* presenta colisiones o combinaciones de estándares.

  SHA-256 se recomienda por su robustez criptográfica y su mayor rendimiento. Y también porque los criptógrafos opinan que *MD5* es vulnerable a determinados ataques que en teoría permitirían obtener dos *hashes* idénticos a partir de archivos diferentes. Esta vulnerabilidad no tiene apenas consecuencias en la práctica, pero el hecho de que exista, y de que *SHA-256* se vea de manera comprobada totalmente libre de tales ataques, resulta suficiente para dar prioridad a este último método.

El contenido de esta sección debe probar que la integridad de las evidencias no se ha visto comprometida y que se ha respetado la cadena de custodia.

PROCESADO DE EVIDENCIAS

Se describen los pasos ejecutados para la extracción de información que no se encuentra de forma explícita en la evidencia:

- Bloques del sistema de ficheros eliminados.
- Ficheros o datos después de las marcas de fin de fichero.

Se deben documentar los siguientes pasos:

- Proceso realizado para pasar de una imagen forense a una copia de trabajo. Es necesario asegurar que no se modifica la copia original y que la copia de trabajo es idéntica *bit* a *bit* al original.
- Procesos ejecutados por cada elemento de evidencia extraído de la copia de trabajo.
- Cada evidencia extraída debe poder trazarse de forma unívoca de los datos originales.

ANÁLISIS DE EVIDENCIAS

Esta sección del análisis forense se construye con las evidencias analizadas que son relevantes para el caso en concreto.

Además, se presentan y razonan las evidencias que confirman o desmienten las diferentes hipótesis que se han analizado durante el proceso de análisis.

Para cada una de las hipótesis que se han analizado:

- Se declara la hipótesis inicial y la información previa que llevó a plantearla.
- Se enumeran los artefactos y evidencias que durante el análisis se han utilizado para verificar o desmentir la hipótesis.
- Se ofrece una conclusión sobre si se ha verificado la hipótesis definida.

Las hipótesis que no son corroboradas durante el análisis también deben incluirse en el informe, si son relevantes de cara al caso.

También, es recomendable incluir todos aquellos elementos (capturas de pantalla, *logs*, etc.) que nos ayuden a entender y que sean necesarios durante el proceso de verificación de la hipótesis.

CONCLUSIONES

Esta sección incluye las conclusiones a las que ha llegado el perito tras la realización de todas las tareas del análisis forense.

El objetivo final de un análisis forense es describir los hechos de forma objetiva. De hecho, todas las conclusiones que se enumeren en esta sección deben estar soportadas por evidencias obtenidas y mostradas durante el informe.

También es recomendable recordar al lector las razones por las que se ha realizado el informe forense, dado que posiblemente (si este fuera el caso) serán los tribunales posteriormente los que comprueben minuciosamente los elementos de evidencia presentados por los peritos forenses, y también sus métodos de trabajo.

Por tanto, el profesional debe contribuir al sostenimiento de su credibilidad haciendo afirmaciones solventes, cuidándose de que la presentación de las pruebas sea impecable y no haciendo nada que implique una vulneración de derechos fundamentales ni un ataque contra la privacidad ni las leyes de protección de datos.

4. LABORATORIO

4.1. LABORATORIO DE ADQUISICIÓN DE DATOS

4.1.1. INTRODUCCIÓN

El objetivo del laboratorio es mostrar las diferentes técnicas de extracción y análisis de información en dispositivos móviles *Android*.

En concreto, se van a realizar las siguientes actividades:

- Adquisición de una imagen forense de un dispositivo Android.
- Adquisición de una imagen de una tarjeta *SD*.
- Adquisición lógica de un dispositivo *Android*.
- Adquisición de la memoria de un dispositivo *Android*.

En esta ocasión, los laboratorios son descritos como un conjunto de pasos que se pueden ir ejecutando a la vez que se van mostrando por pantalla.

4.1.2. IMAGEN FORENSE DE UN DISPOSITIVO *ANDROID*

CONEXIÓN DEL DISPOSITIVO

Para la realización de este laboratorio es necesario disponer de un dispositivo *rooteado*.

La adquisición de la imagen del dispositivo se realizará mediante la conexión *USB*, por lo que procedemos a conectarlo al equipo de análisis.

Vamos a proceder a realizar la imagen desde desde *Santoku*, por lo que en *VirtualBox*, conectamos el dispositivo a la máquina virtual.

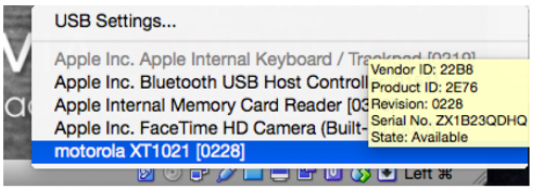


Figura 40: Conexión del dispositivo.

En la opción de *Settings*, activamos la opción de *USB 2.0* ó *3.0*, dependiendo de nuestro dispositivo. Para esto, debemos descargar una extensión de *VirtualBox*:

http://www.virtualbox.org/wiki/Downloads

PREPARACIÓN DE LA ADQUISICIÓN

Una vez conectado en *Santoku*, abrimos una *shell* en el dispositivo y cambiamos al usuario *root*:

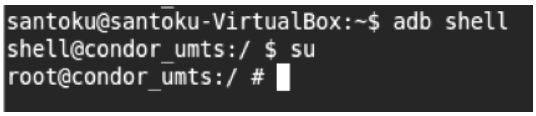


Figura 41: Preparación de la adquisición 1.

Dado que la única partición que puede modificarse por el usuario es aquella que está montada en data, utilizamos *mount*, para averiguar el dispositivo que le corresponde:

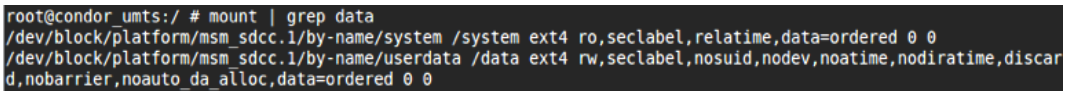


Figura 42: Preparación de la adquisición 2.

Ahora, averiguamos el tamaño del bloque:

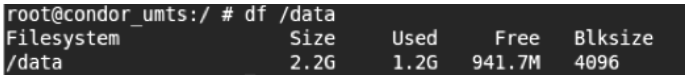


Figura 43: Preparación de la adquisición 3.

ADQUISICIÓN DE LA IMAGEN

La adquisición necesita de una tarjeta *SD* vacía en el dispositivo.

Dependiendo de la versión de *Android* y del dispositivo, la tarjeta se montará en un directorio.

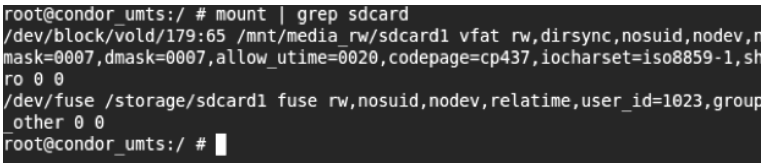


Figura 44: Adquisición de la imagen 1.

Para realizar la adquisición, ejecutamos *dd* con los parámetros correspondientes.

```
> dd if=/dev/block/platform/msm_sdcc.1/by-name/userdata
of=/storage/sdcard1/user.img bs=4096
```

Figura 45: Adquisición de la imagen 2.

Una vez se haya realizado la imagen desde la consola de la máquina de análisis escribimos:

```
> adb pull /storage/sdcard1/user.img
```

Figura 46: Adquisición de la imagen 3.

Obteniendo la imagen de la partición del usuario.

4.1.3. IMAGEN DE UNA TARJETA *SD*

INSTALACIÓN DE *BUSYBOX*

Para la adquisición de imágenes de tarjetas *SD* necesitamos otro dispositivo en el que almacenar la imagen capturada.

Mediante la utilidad *BusyBox* podemos redirigir los datos generados por *dd* directamente a un puerto donde los reciba la máquina de análisis.

Se puede instalar *BusyBox* a través de *Google Play*:

https://play.google.com/store/apps/details?id=stericson.busybox

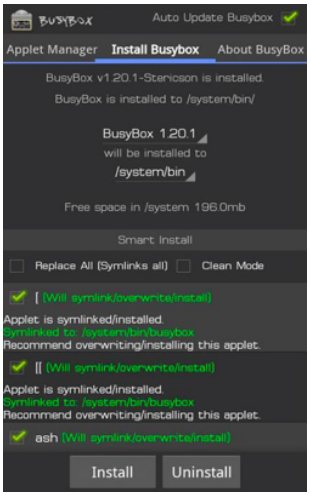


Figura 47: Instalación de *BusyBox*.

Una vez instalado, lo ejecutamos y procedemos a la instalación sin aceptar los mensajes publicitarios.

OBTENCIÓN DE LA IMAGEN

Averiguamos el dispositivo correspondiente a la tarjeta de memoria:

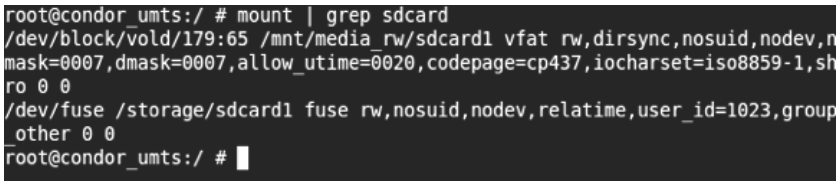


Figura 48: Obtención de la imagen 1.

Dado que no podemos guardar la imagen en la tarjeta *SD*, la vamos a transmitir a la máquina de análisis a través de un *socket*. Para ello, abrimos una nueva ventana del terminal en la máquina de análisis y escribimos:

Figura 49: Obtención de la imagen 2.

De esta manera, nos aseguramos de que todo lo que le llega a *adb* por el puerto *8888*, es transmitido a la máquina de análisis por el mismo puerto.

Ejecutamos:

Figura 50: Obtención de la imagen 4.

Y en la máquina de análisis empezamos a recibir la información mediante el siguiente comando:

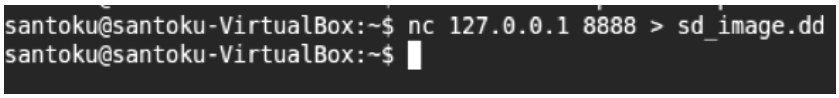


Figura 51: Obtención de la imagen 4.

4.1.4. ADQUISICIÓN LÓGICA DE UN DISPOSITIVO *ANDROID*

INSTALACIÓN DE *AFLOGICAL*

En este laboratorio vamos a utilizar la herramienta *AFLogical*, para la adquisición lógica de evidencias.

AFLogical es una aplicación de *Android* que contiene los permisos necesarios para extraer toda la información accesible mediante permisos en un sistema *Android*:

- Historial de llamadas.
- Contactos.
- Mensajes *SMS*, *MMS* y sus adjuntos.

Dado que se ejecuta a través de una aplicación normal, no requiere disponer de un teléfono *rooteado*.

Para su instalación, es necesario escribir en la línea de comandos de *Santoku*:

```
> aflogical-ose
```

Figura 52: Instalación de *AFLogical*.

De este modo, se instalará y ejecutará la aplicación.

EJECUCIÓN DE *AFLOGICAL*

Una vez ejecutada la aplicación, en el dispositivo deberemos marcar la información que queremos extraer en el dispositivo.

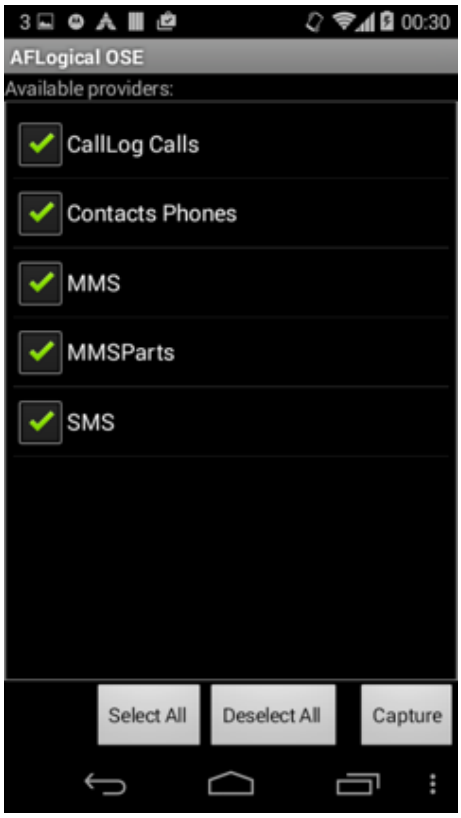


Figura 53: Ejecución de *AFLogical* 1.

Una vez obtenida la información en el dispositivo, continuamos en la consola para transferir los datos a nuestro dispositivo.

```
santoku@santoku-VirtualBox:~$ aflogical-ose
Make sure android device is connected to USB
[sudo] password for santoku:

697 KB/s (28794 bytes in 0.040s)
        pkg: /data/local/tmp/AFLogical-OSE_1.5.2.apk
Success

Starting: Intent { cmp=com.viaforensics.android.aflogical_ose/com.viaforensics.a
ndroid.ForensicsActivity }

Press enter to pull /sdcard/forensics into ~/aflogical-data/

pull: building file list...
pull: /sdcard/forensics/20160215.2430/SMS.csv -> /home/santoku/aflogical-data/20
160215.2430/SMS.csv
pull: /sdcard/forensics/20160215.2430/MMS.csv -> /home/santoku/aflogical-data/20
160215.2430/MMS.csv
```

Figura 54: Ejecución de *AFLogical* 2.

RESULTADOS

Los resultados pueden ser inspeccionados utilizando el navegador de archivos de *Santoku Linux*.

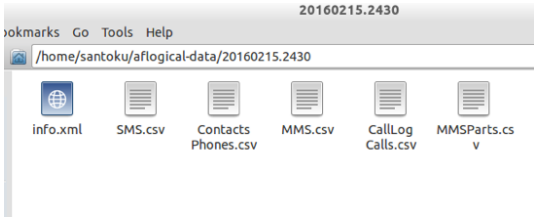


Figura 55: Resultado obtenido.

4.1.5. ADQUISICIÓN DE MEMORIA EN *ANDROID*

INSTALACIÓN Y EJECUCIÓN DE *LIME*

Al igual que en el laboratorio anterior, en esta ocasión será necesario contar con un dispositivo *rooteado*.

En primer lugar, debemos descargar y compilar *LiME*. La guía de instalación se encuentra en:

https://github.com/504ensicsLabs/LiME/tree/master/doc

Una vez compilado, es necesario copiar el módulo al dispositivo del que se va a realizar el proceso de adquisición de la memoria:

```
> adb push lime.ko /storage/sdcard1/lime.ko
```

Figura 56: Instalación y ejecución de *LiME* 1.

Modificamos los puertos para redirigir la salida de *LiME*:

```
> adb forward tcp:4444 tcp:4444
```

Figura 57: Instalación y ejecución de *LiME* 2.

Abrimos una *shell*:

```
> adb shell
```

Figura 58: Instalación y ejecución de *LiME* 3.

Accedemos a *root* y ejecutamos el módulo de *kernel*:

```
> su
> insmod /sdcard/lime.ko "path=/storage/sdcard1/ram.lime
format=lime"
```

Figura 59: Instalación y ejecución de *LiME* 4.

TRANSMISIÓN DE LA IMAGEN

La imagen se crea en formato *lime* en la *sdcard* del sistema.

Para obtenerla en la máquina de análisis utilizamos *adb*:

```
> adb pull /storage/sdcard1/ram.lime
```

Figura 60: Transmisión de la imagen 1.

La imagen obtenida puede ser analizada con *volatility*. Requiere de su instalación en *Santoku*:

```
> sudo apt-get install volatility
```

Figura 61: Transmisión de la imagen 2.

4.2. LABORATORIO DE ANÁLISIS DE DATOS

4.2.1. INTRODUCCIÓN AL LABORATORIO

ANÁLISIS DE UNA IMAGEN DE UN DISPOSITIVO *ANDROID*

Durante este laboratorio se va a completar el análisis forense de una imagen *Android*.

El análisis se basa en una simulación en la que se debe asumir el papel de perito forense en un caso ficticio.

El objetivo final del análisis es redactar un informe forense.

Por tanto, la realización del laboratorio está dividida en tareas. Además, es posible que durante el análisis se encuentren más evidencias de las que se indican, dado que dada la cantidad de evidencias que generan los dispositivos móviles esto es completamente normal (las restricciones de tiempo y espacio obligan a mostrar sólo algunas de ellas).

PREPARACIÓN DE *AUTOPSY*

Durante la realización del laboratorio, utilizaremos *Santoku Linux*. Por lo tanto, es recomendable descargar todo el material necesario para la realización del mismo desde la propia máquina virtual de *Santoku*.

Entre las herramientas que se van a utilizar se encuentran:

- *Autopsy*.
- *Sqliteman*.
- *Exiftool*

En el caso de *Autopsy*, existe un pequeño *bug* en la versión de *Santoku* que debemos subsanar antes de comenzar. Para ello, sólo hay que escribir en la consola lo siguiente:

```
> sudo ln -s /usr/bin/icat /usr/bin/icat-sleuthkit
> sudo ln -s /usr/bin/ils /usr/bin/ils-sleuthkit
```

Figura 62: Corrección de un pequeño *bug* en la versión de *Santoku*.

Esto permite a *sleuthkit* utilizar *icat* e *ils*, para mostrar información de los ficheros.

4.2.2. PRESENTACIÓN DEL CASO

ESCENARIO

Debido a un soplo de un soplo de un confidente a la policía, se conoce que unos delincuentes de habla hispana pueden estar preparando una serie de robos en la ciudad de Londres.

Tras un primer robo en una de las joyerías más famosas de la ciudad, uno de los delincuentes de forma descuidada deja su cara descubierta y es captado por las videocámaras existentes

en la ciudad. En el robo participan cuatro delincuentes.

El seguimiento del sospechoso a través de las cámaras existentes por la ciudad permite a la policía identificarle en una casa, aparentando ser su residencia habitual.

Tras una redada en el domicilio del presunto delincuente, se recupera un dispositivo *Android* que el sospechoso estaba utilizando en el momento de la redada.

ESTADO DEL DISPOSITIVO

El dispositivo incautado no tenía configurado ningún sistema de bloqueo y además estaba *rooteado*.

El dispositivo no tenía insertada ninguna tarjeta *SD*.

Aprovechando el estado actual del dispositivo, justo después de la incautación, un experto de la policía llevó a cabo las siguientes tareas:

- Formatear una tarjeta SD de *8GB*, para la adquisición de datos forenses.
- Conectar desde un equipo de análisis forense, a través de un cable *USB* al dispositivo incautado.
- Utilizar *adb* y el comando *dd*, para volcar el contenido físico de la partición de disco que ha sido montada en */data* (partición de datos de usuario) a la tarjeta *SD*.
- Extraer la imagen capturada desde la tarjeta *SD* al equipo del analista mediante el comando *adb pull*.
- El *hash MD5* de la imagen obtenida es ***235fdf8cdba7584ac5c8a10fc9e11c56***.

ENUNCIADO

La imagen obtenida ha sido enviada a nuestro laboratorio de análisis forense para el análisis.

Se nos pide un informe forense de la imagen encontrada, que incluya:

- Localizaciones de posibles futuros objetivos del grupo criminal.
- Nombres o cualquier otro elemento identificativo de posibles cómplices del sospechoso.
- Información adicional (nombres de cuentas, etc.) que, tras la obtención de la correspondiente orden judicial, pueda ofrecer más información sobre el sospechoso y sus actividades.

4.2.3. CREACIÓN DEL CASO

CARGA DE *AUTOPSY*

Durante la resolución del caso vamos a utilizar como herramienta principal *Autopsy*.

Para lanzar *Autopsy*, en una consola de *Santoku Linux* escribimos (necesita ejecutarse como administrador en la instalación).

```
> sudo autopsy
```

Figura 63: Carga de *Autopsy* 1.

```
santoku@santoku-VirtualBox:~$ sudo autopsy
================================================================================
                    Autopsy Forensic Browser
                http://www.sleuthkit.org/autopsy/
                          ver 2.24
================================================================================
Evidence Locker: /var/lib/autopsy
Start Time: Mon Feb 15 13:23:09 2016
Remote Host: localhost
Local Port: 9999

Open an HTML browser on the remote host and paste this URL in it:

    http://localhost:9999/autopsy

Keep this process running and use <ctrl-c> to exit
```

Figura 64: Carga de *Autopsy* 2.

Y abrimos el navegador y escribimos la dirección indicada.

ABRIENDO UN NUEVO CASO

En la ventana del navegador presionamos en "New Case".

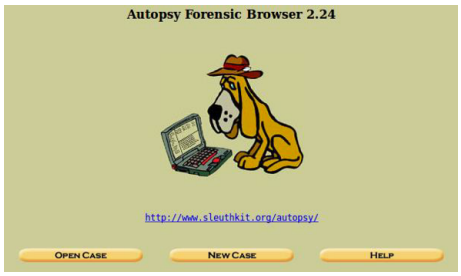


Figura 65: Abriendo un nuevo caso en *Autopsy* 1.

Y rellenamos los datos del caso.

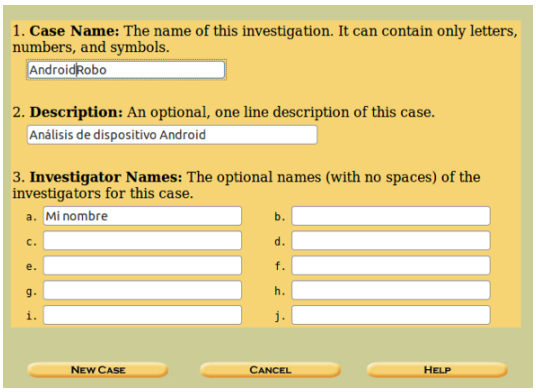


Figura 66: Abriendo un nuevo caso en *Autopsy* 2.

AÑADIENDO DISPOSITIVOS

Llegaremos a una ventana en la que se nos informa de los directorios de configuración y del caso.

En este caso, solo vamos a analizar un dispositivo así que presionamos en "Add Host".

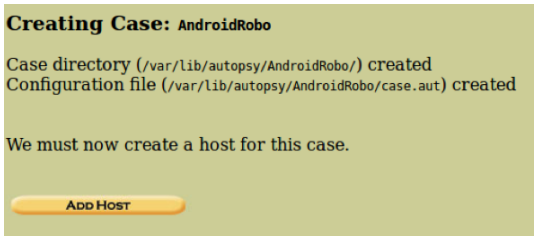


Figura 67: Añadiendo dispositivos en *Autopsy*.

DETALLES DEL DISPOSITIVO

Añadimos los detalles del dispositivo:

Figura 68: Detalles del dispositivo en *Autopsy*.

AÑADIENDO LA IMAGEN

Seleccionamos "Add Image":

Figura 69: Añadiendo la imagen en *Autopsy* 1.

Y a continuación, “Add Image File”:

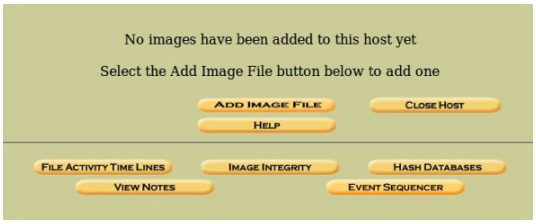


Figura 70: Añadiendo la imagen en *Autopsy* 2.

Escribimos la localización del fichero en nuestro sistema de archivos.

Como se especificó durante la descripción del caso, la imagen se corresponde con una partición del disco, por lo que seleccionamos la opción correspondiente.

Para no copiar o mover todo el fichero, seleccionamos *Symlink*.

Figura 71: Añadiendo la imagen en *Autopsy* 3.

INTEGRIDAD DE LA IMAGEN

Como analistas forenses, debemos asegurarnos de que la imagen se correponde exactamente con la que se obtuvo del volcado del teléfono.

Para ello, disponemos del *hash MD5*, por lo que procedemos a verificar su integridad antes de incorporarla al análisis:

235fdf8cdba7584ac5c8a10fc9e11c56

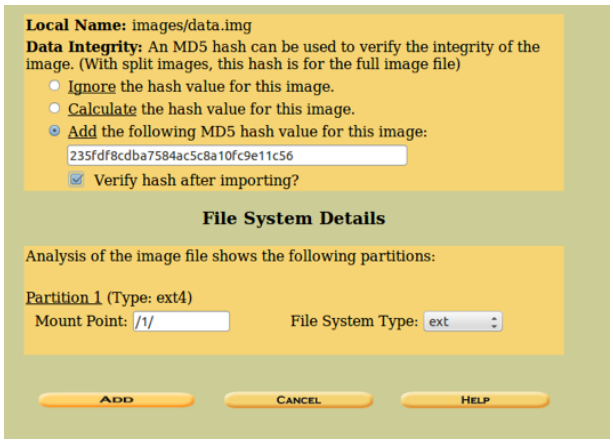


Figura 72: Integridad de la imagen en *Autopsy* 1.

Una vez comprobada la integridad de la imagen deberíamos obtener un resultado como el mostrado a continuación:

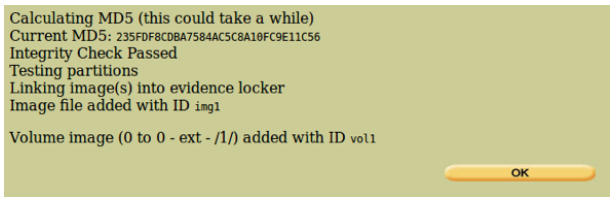


Figura 73: Integridad de la imagen en *Autopsy* 2.

Teniendo como resultado la imagen cargada dentro del caso.

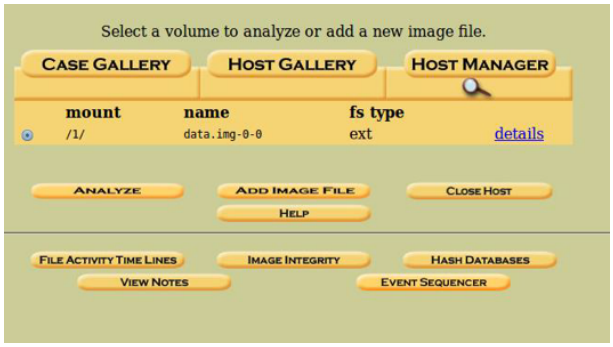


Figura 74: Integridad de la imagen en *Autopsy* 3.

ANÁLISIS INICIAL

Para facilitar la extracción de evidencias futuras, procedemos a:

- Analizar el espacio del dispositivo.
- Generar un índice de *strings*, para la realización de búsquedas de forma eficiente.

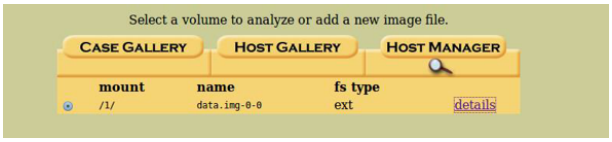


Figura 75: Análisis inicial en *Autopsy* 1.

Para ello, en la ventana del caso abrimos los detalles de la imagen:

Y seleccionamos "Extract Strings" y "Extract Unallocated". Una vez extraído el espacio borrado, se pueden extraer también los *strings* del mismo.

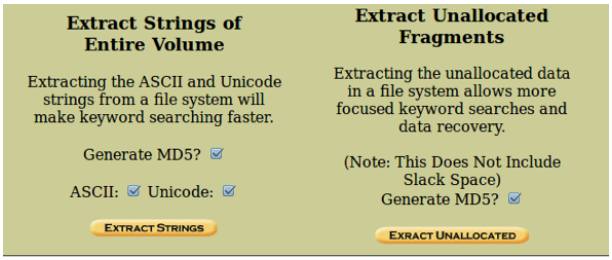


Figura 76: Análisis inicial en *Autopsy* 2.

4.2.4. EXTRACCIÓN DE INFORMACIÓN

DETERMINANDO LOS ELEMENTOS INICIALES A EXAMINAR

Una vez cargada la imagen en *Autopsy*, el primer paso de la investigación será decidir que información contenida en la imagen será de interés para el caso en cuestión. Para ello, hay que tener en cuenta el objetivo con el que se ha encargado el informe forense en el enunciado.

Por tanto, es necesario elaborar una lista con los elementos de información de interés para el caso en cuestión. Es decir, se deberá obtener una lista con la información que pudiese contener el dispositivo que permita razonar sobre los posibles futuros objetivos, los cómplices del sospechoso o información adicional, para continuar las investigaciones.

Volviendo al caso, el informe que se nos ha solicitado tiene como objetivo recabar información sobre tres elementos en particular:

- Posible localización de nuevos objetivos.
- Las conexiones personales del sospechoso con otros cómplices.
- La información adicional sobre su presencia *online*.

En lo relativo a la localización de nuevos objetivos, serán de interés:

- Información de aplicaciones de localización. Los pares de coordenadas almacenados en las diferentes aplicaciones nos pueden ofrecer datos de interés.

- Puntos *WiFi* a los que se ha conectado el dispositivo. Si las conexiones *WiFi* han sido a redes públicas, podremos ubicar el dispositivo en un entorno determinado.
- Mensajes que incluyan información sobre localizaciones. La información que haya podido intercambiar con otras personas sobre las localizaciones puede ser de interés también para el análisis.

En lo relativo a la presencia *online* del sospechoso:

- Nombres de usuario (y credenciales si es posible) del sospechoso en otros servicios *online*. Nos permitirá acceder a información adicional, para determinar el grado de participación del sospechoso en los hechos.

Dado que ciertos elementos de información pueden ser generados por diferentes aplicaciones, deberemos también elaborar un inventario de las aplicaciones instaladas en el dispositivo.

LISTA DE ELEMENTOS A EXAMINAR

- Lista de aplicaciones instalada con especial detalle en aquellas aplicaciones que tengan relación con:
  - Localización.
  - Mensajería.
  - Servicios en la red (redes sociales, etc.).
- Localizaciones incluidas en aplicaciones de localización.
- Redes *WiFi* a las que se ha conectado el dispositivo.
- Agenda del teléfono.
- Historial de llamadas.
- Contenido y destinatarios de los mensajes de las diferentes aplicaciones de mensajería existentes en el dispositivo.
- Listado de credenciales de acceso a redes sociales de aplicaciones instaladas en el dispositivo a tal efecto.
- Fotografías que incluyan información relevante sobre alguno de los elementos anteriores.

EXTRACCIÓN DE ELEMENTOS DE INFORMACIÓN

Una vez listados todos los elementos a extraer durante el análisis, procedemos a su extracción.

Cada una de las siguientes tareas está enfocada a la obtención de uno de los elementos mencionados en la tabla anterior.

Para proceder a la fase de análisis se deberá extraer la información relativa a cada uno de ellos.

Es posible que durante la fase de análisis sea necesario realizar la extracción de nuevos

elementos de información. Esta tarea es normal dentro de los procesos de análisis forense.

De cara a la elaboración del informe forense, se deberán tomar notas sobre todos los pasos y procesos que se realicen durante estas tareas.

EXTRACCIÓN DEL LISTADO DE APLICACIONES

El primer paso a realizar para acotar el resto de tareas es identificar las aplicaciones que se encuentran instaladas en el dispositivo.

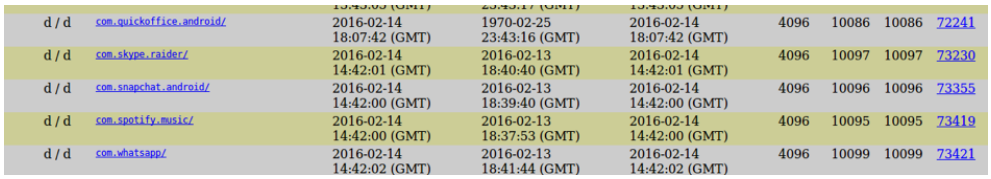
Por tanto, hay que elaborar una lista de las aplicaciones instaladas en el dispositivo, haciendo énfasis en las aplicaciones que puedan incluir información de localización, mensajes o acceso a redes sociales. Es decir, un listado de aplicaciones especificando el tipo de información que se puede extraer de cada una de ellas.

Las aplicaciones instaladas en un dispositivo *Android* se encuentran en: */data/data*.

La imagen obtenida en el caso, es la partición *data*, por lo que sólo tendremos que inspeccionar el contenido de la carpeta *data*, para averiguar las aplicaciones instaladas.

En *Autopsy*, navegamos a *data* y comprobamos la lista de aplicaciones.

De entre la multitud de aplicaciones existentes, se observan algunas interesantes, como las que se muestran a continuación:



d / d	com.quickoffice.android/	2016-02-14 18:07:42 (GMT)	1970-02-25 23:43:16 (GMT)	2016-02-14 18:07:42 (GMT)	4096	10086	10086	72241
d / d	com.skype.raider/	2016-02-14 14:42:01 (GMT)	2016-02-13 18:40:40 (GMT)	2016-02-14 14:42:01 (GMT)	4096	10097	10097	73230
d / d	com.snapchat.android/	2016-02-14 14:42:00 (GMT)	2016-02-13 18:39:40 (GMT)	2016-02-14 14:42:00 (GMT)	4096	10096	10096	73355
d / d	com.spotify.music/	2016-02-14 14:42:00 (GMT)	2016-02-13 18:37:53 (GMT)	2016-02-14 14:42:00 (GMT)	4096	10095	10095	73419
d / d	com.whatsapp/	2016-02-14 14:42:02 (GMT)	2016-02-13 18:41:44 (GMT)	2016-02-14 14:42:02 (GMT)	4096	10099	10099	73421

Figura 77: Extracción del listado de aplicaciones en *Autopsy* 1.

En el listado de aplicaciones, comprobamos que hay una que ha sido borrada del dispositivo. En este sentido, podría ser interesante investigar la utilidad de esa aplicación de cara al informe.

		15:48:30 (GMT)	23:43:01 (GMT)	15:48:30 (GMT)				
d / d	com.motorola.wappushsi/	1970-02-25 23:44:42 (GMT)	1970-02-25 23:43:04 (GMT)	1970-02-25 23:44:42 (GMT)	4096	10045	10045	72133
d / r	com.pinellascodeworks.securewipe	2016-02-14 14:50:04 (GMT)	2016-02-14 14:50:04 (GMT)	2016-02-14 14:50:04 (GMT)	20480	10021	10021	73691 (realloc)
d / d	com.qualcomm.atfwd/	1970-02-25 23:44:46 (GMT)	1970-02-25 23:43:18 (GMT)	1970-02-25 23:44:46 (GMT)	4096	1000	1000	72261
d / d	com.qualcomm.interfacepermissions/	1970-02-25	1970-02-25	1970-02-25	4096	10067	10067	72195

Figura 78: Extracción del listado de aplicaciones en *Autopsy* 2.

Se puede comprobar que el listado de aplicaciones es muy grande:

- Hay que tener en cuenta que en esta carpeta se almacenan todas las aplicaciones, incluidas las que vienen por defecto en el sistema.
- En el informe se deberían listar todas, pero especificar las que están relacionadas directamente con el caso.

Lista de aplicaciones de interés:

- *com.quickoffice.android – QuickOffice.*
  - Puede almacenar documentos de interés.
- *com.skype.raider – Skype.*
  - Puede almacenar contactors, fotografías, mensajes y localizaciones.
- *com.snapchat.android – Snapchat.*
  - Puede almacenar contactos, fotografías, mensajes y localizaciones.
- *com.whatsapp.*
  - Puede almacenar contactos, fotografías, mensajes y localizaciones.
- *com.instagram.android.*
  - Puede almacenar contactos, fotografías, mensajes y localizaciones.
- *com.google.android.gm.*
  - Puede almacenar correos electrónicos de interés.
- *com.android.browser.*
  - Almacena los datos de navegación.
- *com.google.android.apps.maps.*
  - Puede almacenar localizaciones.
- *com.facebook.katana.*
  - Puede almacenar contactos, fotografías, mensajes y localizaciones.
- *com.android.email.*
  - Puede almacenar correos electrónicos de interés.
- *com.android.chrome.*
  - Almacena los datos de navegación.

Además, dentro del dispositivo se pueden encontrar los siguientes providers de interés forense (mencionados anteriormente):

- *com.android.providers.calendar.*
- *com.android.providers.telephony.*
- *com.android.providers.contacts.*

EXTRACCIÓN DE DATOS DE INTERÉS

Una vez recuperada la lista de aplicaciones de interés del dispositivo, pasamos a analizarlas en busca de datos que puedan ser relevantes para la investigación.

Una vez extraídos los datos relevantes de cada aplicación, pasamos a realizar las tareas de análisis.

Es conveniente documentar y etiquetas correctamente la información extraída durante este proceso, para facilitar las posteriores tareas de análisis.

En la mayoría de los casos, la validez de la información extraída durante este proceso ha sido corroborada por la comunidad. Como en los datos que se han observado durante el estudio de los diferentes elementos analizables.

En el caso de que hubiese que extraer información de aplicaciones o servios no documentados, sería necesario validar que la información extraída se corresponde con la existente en el o los dispositivos (en este caso, este tipo de validación ha quedado fuera del alcance de este laboratorio).

HISTORIAL DE LLAMADAS Y CONTACTOS

Hay que localizar y extraer la información relativa al historial de llamadas y contactos del teléfono. Es decir, hay que obtener el fichero que identifica las últimas llamadas realizadas desde el dispositivo.

Tanto el historial de llamadas, como los contactos del dispositivo se encuentran en:

/data/com.android.providers.contacts/databases/contacts2.db

Procedemos a extraer el fichero (su correspondiente journal está vacío). Por tanto, tomamos notas de todo el proceso para su documentación de cara al informe.

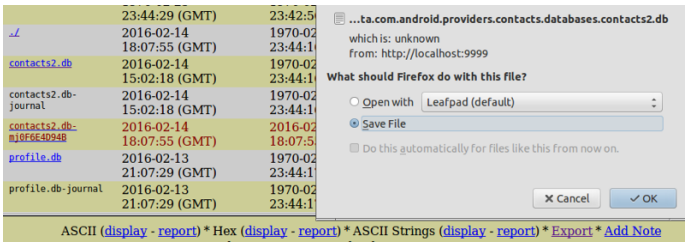


Figura 79: Historial de llamadas y contactos en *Autopsy*.

MENSAJES DE TEXTO

Se debe localizar y extraer la información relativa a los mensajes de texto existentes en el dispositivo. Es decir, el fichero que identifica los mensajes de texto existentes en el

dispositivo.

Los mensajes de texto del dispositivo se encuentran en:

/data/com.android.providers.telephony/databases/mmssms.db

Procedemos a extraer el fichero (y su correpondiente *journal*) y tomamos notas de todo el proceso para su documentación de cara al informe.

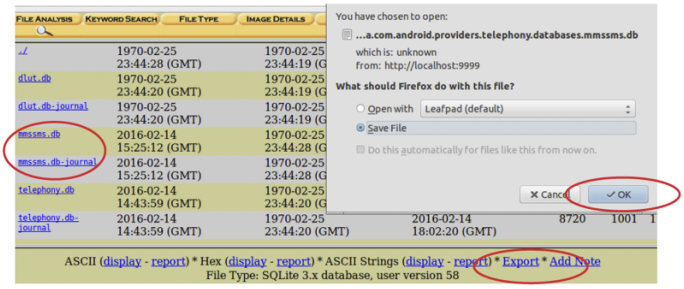


Figura 80: Mensajes de texto en *Autopsy*.

REDES *WIFI*

Es necesario localizar y extraer la información de aquellas redes *WiFi* a las que se ha conectado el dispositivo. Es decir, el fichero que identifica las redes *WiFi* a las que se ha conectado el dispositivo.

La información de las redes *WiFi* a las que se ha conectado el dispositivo se puede encontrar en:

/misc/wifi/wpa_supplicant.conf

En este caso, podemos ver la información contenida en el mismo antes de la extracción.

Extraemos el fichero y tomamos notas de todo el proceso, para su documentación de cara al informe.

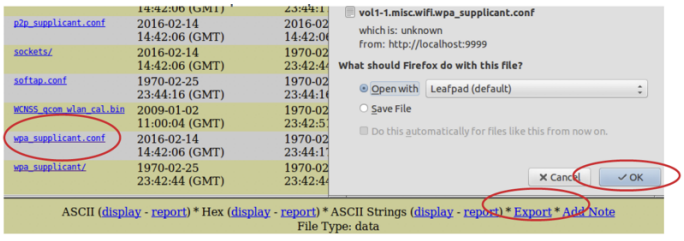


Figura 81: Redes *WiFi* en *Autopsy*.

LOCALIZACIONES EN LA APLICACIÓN DE MAPAS

En este apartado, hay que localizar y extraer la información relativa a las localizaciones existentes en la aplicación de mapas. Es decir, el fichero que identifica las localizaciones

existentes en la aplicación de mapas.

Accedemos a la carpeta de la aplicación de mapas, para buscar información relacionada con las localizaciones:

/data/com.google.android.apps.maps/

En la carpeta *cache/http* localizamos dos imágenes:

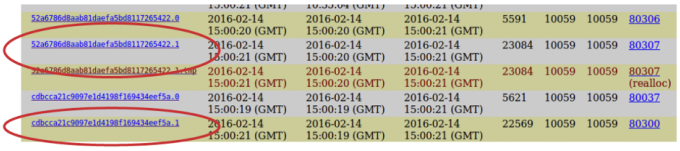


Figura 82: Localizaciones en la aplicación de mapas en *Autopsy* 1.

Además de extraerlas, podemos añadir notas en las mismas, para facilitar la creación del informe. Es decir, además de un texto de información, podemos añadir las fechas que queremos añadir a la nota, para la creación de una línea de tiempo de eventos.

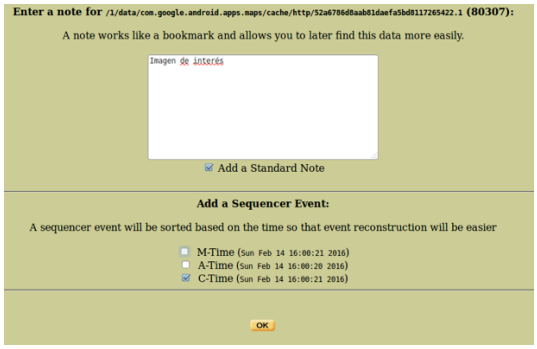


Figura 83: Localizaciones en la aplicación de mapas en *Autopsy* 2.

En *files/share_history.xml* podemos encontrar un listado con las aplicaciones con las que se han compartido ubicaciones:

- *Gmail*.
- *Snapchat*.

Más adelante, comprobaremos las ubicaciones compartidas.

En la carpeta databases encontraremos las bases de datos que pueden resultar de nuestro interés. Por tanto, las guardamos y anotamos.

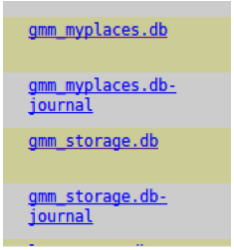


Figura 84: Localizaciones en la aplicación de mapas en *Autopsy* 3.

FOTOGRAFÍAS

Se localizan y extraen las fotografías existentes en la imagen del dispositivo. Es decir, los ficheros de fotografías existentes en la imagen del dispositivo.

En primer lugar, accedemos al directorio donde se guardarán por defecto las fotografías, para inspeccionar su contenido: /media.

Comprobamos que hay imágenes de interés en:

- */media/0/WhatsApp/Media/WhatsApp Images/*
- */media/0/DCIM/Camera/*
- */media/0/Pictures/Screenshots/*

Autopsy nos permite también buscar todas las imágenes existentes en la imagen de forma automática.

En *File Type*:

Figura 85: Fotografías en *Autopsy* 1.

Seleccionamos la opción "Sort Files by Type" y marcamos las opciones mostradas en la captura de *Autopsy*, para extraer todos los tipos de archivos relevantes:

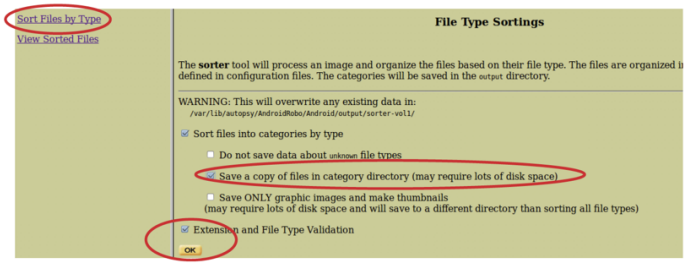


Figura 86: Fotografías en *Autopsy* 2.

Una vez generada la lista de ficheros, se puede navegar a ella accediendo a la *URL* mostrada en la opción "View Sorted Files".

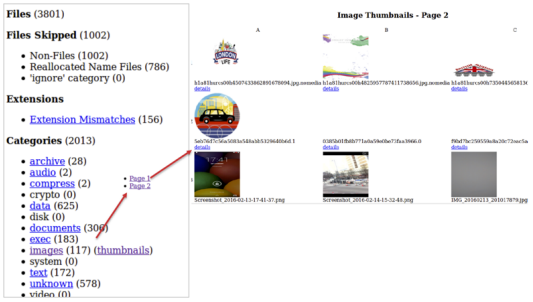


Figura 87: Fotografías en *Autopsy* 3.

CORREOS ELECTRÓNICOS

Hay que localizar y extraer los correos electrónicos existentes en la imagen del dispositivo. Es decir, la base de datos de correos electrónicos en la imagen del dispositivo.

Existen dos aplicaciones de correo en el dispositivo:

- */data/com.android.email/*
- */data7com.google.android.gm/*

Comprobamos el contenido de las bases de datos y también lo referente a la primera aplicación. En este caso, no hay ningún contenido adicional de interés, salvo el código de creación de las tablas, por lo que añadimos las notas correspondientes, para la posterior elaboración del informe.

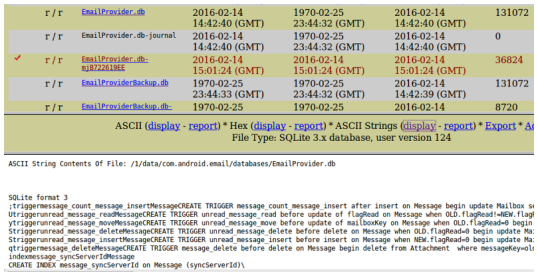


Figura 88: Correos electrónicos en *Autopsy* 1.

En el caso de la aplicación de *Gmail* existen ficheros de bases de datos que hacen referencia a una cuenta de *Gmail*. Por tanto, los anotamos como importantes y procedemos a su descarga:

Figura 89: Correos electrónicos en *Autopsy* 2.

DATOS DE NAVEGACIÓN *WEB*

Se tiene que localizar y extraer los datos de navegación *web* existentes en la imagen del dispositivo. Es decir, la base de datos de correos electrónicos existentes en la imagen del dispositivo.

La imagen analizada solo tiene instalado un navegador en: ***/data/com.google.chrome/***.

Inspeccionamos el contenido y lo anotamos, para el análisis posterior de todos los ficheros existentes en la carpeta interna: ***app_chrome/Default***.

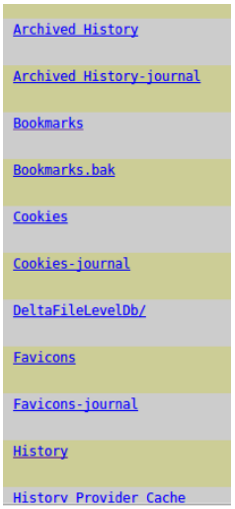


Figura 90: Datos de navegación *web* en *Autopsy*.

INFORMACIÓN RELATIVA A LAS REDES SOCIALES

Hay que localizar y extraer del dispositivo la información que pueda ser de interés y esté relacionada con las aplicaciones de redes sociales. Es decir, los ficheros de bases de datos, fotografías y aquellos otros ficheros existentes en las cachés de las aplicaciones relacionadas con redes sociales.

Para realizar esta tarea debemos inspeccionar los contenidos de todas las aplicaciones que están relacionadas con redes sociales.

Procedemos a entrar en cada una de las aplicaciones y anotamos los ficheros relevantes de cada una de las mismas.

Las aplicaciones que se van a analizar son:

- *com.quickoffice.android – QuickOffice.*
- *com.skype.raider – Skype.*
- *com.snapchat.android – Snapchat.*
- *com.whatsapp.*
- *com.instagram.android.*
- *com.facebook.katana.*

Durante la etapa de análisis se analizan los contenidos de las mismas en busca de pruebas.

4.2.5. ANÁLISIS

TAREAS DE ANÁLISIS

Durante esta parte del laboratorio se van a inspeccionar los contenidos marcados en las tareas anteriores y añadir anotacions con respecto a la información contenida en los mismos.

El objetivo de esta tarea es la obtención de la información que necesitamos, para la correcta elaboración del informe.

Para ello, partimos de la hipótesis de que el dispositivo ha sido utilizado para que los miembros de una banda criminal intercambien algún tipo de información sobre sus objetivos.

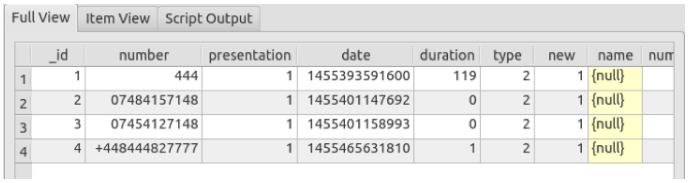
Y los datos que necesitamos para que las investigaciones puedan continuar son:

- Información sobre los posible cómplices en la ejecución de los hechos.
- Localizaciones de los posibles futuros objetivos.
- Información sobre las cuentas existentes en el propio dispositivo, para la posterior adquisición de las mismas a través de la correspondiente orden judicial.

INFORMACIÓN SOBRE LOS CÓMPLICES

Se tienen que analizar los datos obtenidos, para establecer una lista de contactos frecuentes en el dispositivo y el tipo de información que han intercambiado. Es decir, una parte del informe forense que especifique los contactos de interés encontrados en el dispositivo, así como los mensajes que han sido intercambiados.

Analizando la tabla *call* de la base de datos de llamadas *contacts2.db*:


Full View | Item View | Script Output

	_id	number	presentation	date	duration	type	new	name	num
1	1	444	1	1455393591600	119	2	1	{null}	
2	2	07484157148	1	1455401147692	0	2	1	{null}	
3	3	07454127148	1	1455401158993	0	2	1	{null}	
4	4	+448444827777	1	1455465631810	1	2	1	{null}	

Figura 91: Información sobre los cómplices en *Autopsy* 1.

Además, navegando a la tabla de contactos observamos lo siguiente:

- Se ha ejecutado la sentencia ***Select display_name, sync1 from raw_contacts***, para reducir el número de columnas.

display_name	
J	https://www.google.com/m8/feeds/contacts/joh
J	447401089370@s.whatsapp.net
j.thebest@gmail.com	https://www.google.com/m8/feeds/contacts/joh

Figura 92: Información sobre los cómplices en *Autopsy* 2.

La inspección de la base de datos de mensajes (*mmssms.db*), nos ofrece la siguiente información:

- Se ha ejecutado la sentencia ***Select address,person,body from sms***, para simplificar la salida obtenida y facilitar su lectura.

	address	person	
			You can also tap on this link to verify your phone: v.whatsapp.com/492
8	Snapchat	{null}	Snapchat Code: 727829. Happy Snapping!
9	7401 089370	{null}	Movil nuevo. Aun no me hago con el. La app de Snapchat esta muy bien
10	+447401089370	1	Hola John! Móvil nuevo :) aunque aún no me hago con el. Snapchat esta
11	7401 089370	{null}	Siii
12	+447401089370	1	También estoy llegando. No tengo tarifa de datos ya

Figura 93: Información sobre los cómplices en *Autopsy* 3.

INFORMACIÓN SOBRE LOS CÓMPLICES Y OTROS TELÉFONOS

Tras la información analizada de las bases de datos de *SMS*, contactos y teléfonos, podemos concluir que se han realizado llamadas a 4 números diferentes, información que completaremos a partir de las redes sociales:

- El 444 parece ser un número de información de la operadora.
- Se han realizado dos llamadas telefónicas a teléfonos móviles muy similares, pero ninguna de ellas ha sido respondida.
  - La búsqueda en *Internet* no proporciona información de utilidad.
- Se han enviado varios mensajes sospechosos a un contacto con el número de teléfono *7401089370*.
- Se ha realizado una llamada al teléfono *+448444827777*.
  - Una búsqueda en *Internet* muestra que el teléfono pertenece al servicio de Palacios Históricos de Reino Unido.
- Existe un contacto de correo con la dirección *j.thebest@gmail.com*.
- Se han recibido mensajes de texto, para el alta en servicios como *WhatsApp* y *Snapchat*.

INFORMACIÓN SOBRE LAS LOCALIZACIONES

Hay que analizar la información sobre las localizaciones en las que haya podido estar el dispositivo. Es decir, una parte del informe que especifique las localizaciones encontradas

y su interés en relación con el caso de estudio.

Analizando la base de datos *ggm_myplaces.db* observamos que existen dos entradas en la tabla *sync_item* con *URL* pertenecientes a mapas de *Google*.

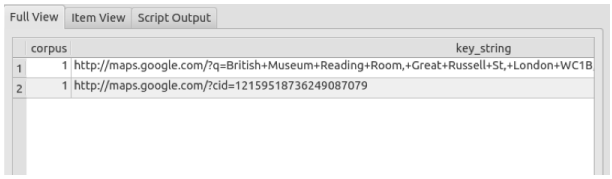


Figura 94: Información sobre las localizaciones en *Autopsy* 1.

Comprobamos el resultado que devuelven las *URL* y analizamos aquellas que apuntan a dos localizaciones específicas de la ciudad de Londres.

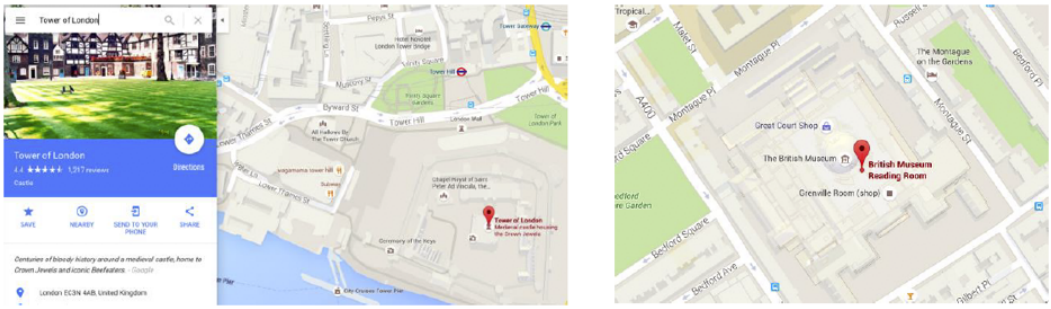


Figura 95: Información sobre las localizaciones en *Autopsy* 2.

El teléfono de contacto de "La Torre de Londres" coincide con el encontrado en el historial de llamadas.

Además, si analizamos uno de los ficheros encontrados en la caché de la aplicación vemos que la imagen coincide con la mostrada en la figura:

cache/http/52a6786d8aab81daefa5bd8117265422

Figura 96: Información sobre las localizaciones en *Autopsy* 3.

Analizamos ahora el contenido del fichero de redes *WiFi* y comprobamos que tiene datos de una conexión (con contraseña incluida), pero el nombre del punto de acceso no ofrece ninguna información sobre su localización más allá del país de origen (Gran Bretaña).

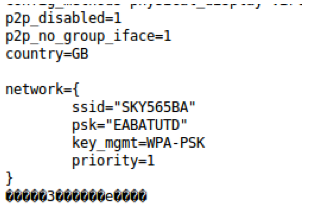


Figura 97: Información sobre las localizaciones en *Autopsy* 4.

Anotamos todos los descubrimientos realizados en esta tarea para la posterior realización del informe.

Las fotografías tomadas por la cámara no muestran las coordenadas *GPS* entre su información *EXIF*.

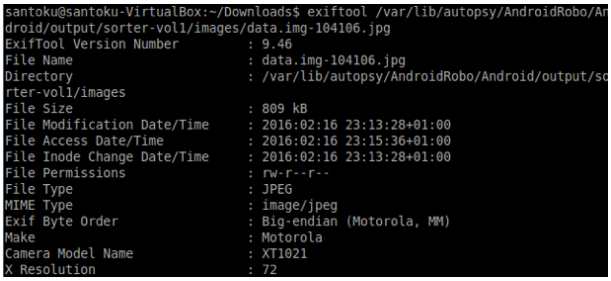


Figura 98: Información sobre las localizaciones en *Autopsy* 5.

Pero, su contenido muestra que han sido tomadas en un museo con importantes piezas de arte. Por tanto, dada la información de localización obtenida anteriormente, se trata con toda seguridad del Museo Británico.

Figura 99: Información sobre las localizaciones en *Autopsy* 6.

Anotamos todos los descubrimientos realizados en esta tarea, para la posterior realización del informe.

INFORME SOBRE LAS CUENTAS EXISTENTES

Se analizan los datos en cada una de las aplicaciones de interés encontradas en el dispositivo, para establecer una lista de contactos frecuentes y el tipo de información que han enviado mediante la utilización de las mismas. El resultado es una parte del informe que especifique la información de interés encontrada en cada una de las aplicaciones analizadas y si tiene alguna relación con los datos analizados anteriormente.

Durante esta tarea es necesario analizar la información existente en el resto de aplicaciones instaladas en el dispositivo. El análisis de estas aplicaciones debe ser validado con dispositivos de control:

- Por cada una de las aplicaciones, se deberán generar evidencias en el dispositivo de control y se deberá verificar que las evidencias generadas corresponden de verdad con los hechos que se puedan observar de forma directa en el teléfono.
- En este caso, se asume que la validación del significado de los datos ya ha sido realizada con anterioridad.

En esta sección del análisis mostramos sólo una parte de los posibles resultados a obtener. Por tanto, solamente quedaría la revisión del posible conjunto de evidencias adicionales.

INFORME SOBRE LAS CUENTAS EXISTENTES - *SPOTIFY*

Tras un primer análisis, se comprueba que las siguientes aplicaciones no han sido utilizadas y por lo tanto, no incluyen información de interés:

- *Facebook*.
- *Instagram*.
- *Skype*.

Tras el análisis de la aplicación de *Spotify* se localiza un fichero que contiene el usuario y el *token* de autenticación de la cuenta.

/data/com.spotify.music/files/settings/prefs

```
autologin.canonical_username="johntoppysmith"
autologin.username="johntoppysmith"
autologin.saved_credentials="{\"johntoppysmith\":[\"johntoppysmith\",\"ajpHPT8ErZCWmrR3x0NUPmbfsSKbL+krHEWhyqKJqrjlqbDUobsfuO65eWVKXEIn\"]}"
language="en_GB"
autologin.blob="ajpHPT8ErZCWmrR3x0NUPmbfsSKbL+krHEWhyqKJqrjlqbDUobsfuO65eWVKXEIn"
core.clock_delta=-1
```

Figura 100: Información sobre las cuentas existentes en *Autopsy* 1.

INFORME SOBRE LAS CUENTAS EXISTENTES - *GMAIL*

El análisis de los correos de *Gmail* se puede comprobar en el fichero:

databases/mailstore.johntoppysmith@gmail.com.db

Su contenido muestra una serie de mensajes del contacto *j.thebest@gmail.com* compartiendo la ubicación del museo y después mencionando unas joyas. Se ha realizado la siguiente consulta para facilitar la lectura del resultado:

Select to Addresses, snippet from messages

5	"johntoppysmith" <johntoppysmith@gmail.com>	Hey johntoppysmith! Before you start Snapping, it's a
6	"" <j.thebest@gmail.com>	British Museum Reading Room Great Russell St Lond
7	"" <j.thebest@gmail.com>	Ya he llegado
8	"" <j.thebest@gmail.com>	Despues iremos a ver las joyas

Figura 101: Información sobre las cuentas existentes en *Autopsy* 2.

Dadas las evidencias encontradas hasta la fecha es posible que tengan que ver con "La Torre de Londres".

INFORME SOBRE LAS CUENTAS EXISTENTES - *WHATSAPP*

Se analizan las bases de datos de la aplicación *WhatsApp*. Los chats de la aplicación se encuentran en el fichero: */databases/msgstore.db*.

Analizando el contenido se pueden observar las siguientes entradas (se ha realizado la siguiente consulta para facilitar su lectura):

Select key_remote_jid, data, latitude, longitude from messages

	key_remote_jid	data	latitude	longitude
1	-1	{null}	0	0
2	447401089370@s.whatsapp.net	Hola!	0	0
3	447401089370@s.whatsapp.net	{null}	51.52393464	-0.07597850985
4	447401089370@s.whatsapp.net	Voy de camino	0	0
5	447401089370@s.whatsapp.net	{null}	0	0
6	447401089370@s.whatsapp.net	{null}	51.5211384	-0.1149357
7	447401089370@s.whatsapp.net	Genial!	0	0

Figura 102: Información sobre las cuentas existentes en *Autopsy* 3.

Se comprueba que hay una comunicación mediante mensajes de *chat* con el mismo teléfono con el que se intercambiaron *SMS*.

Se han compartido dos localizaciones. Tras la búsqueda de las coordenadas se comprueba:

- La primera corresponde a la dirección *10 Redchurch St, Londres E2 7DD*.
- La segunda corresponde a la dirección *A401, Londres WC1X 8NX*.

INFORME SOBRE LAS CUENTAS EXISTENTES - *SNAPCHAT*

De la aplicación de *Snapchat* analizamos el fichero de base de datos principal:

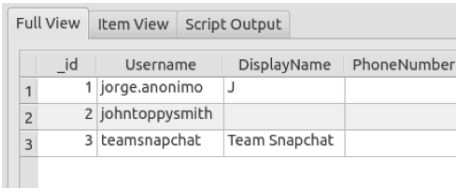


Figura 103: Información sobre las cuentas existentes en *Autopsy* 4.

databases/tcspahn.db

Y comprobamos que también ha tenido contacto con ***J*** mediante la aplicación de *Snapchat*.

No se ha podido acceder a los ficheros borrados por *Snapchat*. Esto puede que se deba a la utilización de la aplicación *com.pinellascodeworks.secureswipe*, que permite el borrado seguro de los bloques no utilizados en la memoria interna.

4.3. EL INFORME FORENSE

4.3.1. INTRODUCCIÓN

Como último paso para la realización del laboratorio es necesario realizar un informe con las evidencias encontradas y las conclusiones a las que se ha llegado tras la realización del análisis.

Para la realización del informe nos podemos ayudar de todas las notas e información que se haya recopilado durante el análisis, incluyendo la que se haya obtenido durante la ejecución de cada uno de los pasos.

4.3.2. RESUMEN DEL CASO

En esta sección se debe incluir:

- Una portada que incluya el nombre y apellidos del perito.
- Los antecedentes concretos que se conozcan del caso. Enunciado del mismo descrito con tus propias palabras.
- Estado de las evidencias que te fueron entregadas. El tamaño y descripción del archivo de evidencias recibido con los datos que puedan permitir la comprobación de su integridad por terceros.
- Las limitaciones del análisis que se están realizando dado el estado y conjunto de evidencias recibidas. Hay que tener en cuenta que en este caso, no se tiene acceso al dispositivo real y que solo se está analizando una de las particiones del dispositivo.
- Lo que se pide es corroborar o comprobar, como perito forense.

Dado el caso simulado del informe, no ha sido necesario incluir:

- Quién ha solicitado el informe forense.
- Las fechas más importantes en relación al informe.
- Datos personales del perito.

4.3.3. HERRAMIENTAS UTILIZADAS

Se deberán describir todas las herramientas utilizadas, además de las utilizadas durante la resolución del caso:

- *Autopsy*.
- *Sqliteman*.
- *Firefox*.
- *Exiftool*.

Se deberán añadir todas las herramientas adicionales que se hayan utilizado. Por cada una de ellas hay que especificar:

- La versión de la herramienta utilizada (incluyendo la plataforma).
- El fabricante.
- La tarea para la que ha sido utilizada.

Si se ha utilizado alguna herramienta reciente o poco conocida, hay que incluir un test de validación de la misma.

4.3.4. ADQUISICIÓN DE EVIDENCIAS

Se deberá incluir el proceso que se ha llevado para añadir la imagen obtenida a *Autopsy*.

4.3.5. PROCESADO DE EVIDENCIAS

En esta sección se debe describir como se han extraído los diferentes ficheros de evidencias que se utilizarán posteriormente, para demostrar los razonamientos durante la fase de análisis.

Si se ha realizado alguna operación para la recuperación de archivos en el espacio borrado, se deberá describir el proceso llevado a cabo.

Además, por cada elemento de información extraído de la imagen se deberá obtener su resumen *MD5* y añadirlo al informe forense:

- Cada evidencia extraída debe poder trazarse de forma unívoca a los datos originales.
- Además, de esta manera se podrá demostrar que las pruebas no se han modificado.

4.3.6. ANÁLISIS Y CONCLUSIONES

En esta sección se deben razonar, basándose en la información extraída en la sección anterior ("Procesado de evidencias"), los diferentes hechos que se demuestran con la información existente en la imagen analizada.

En este caso, el conjunto de razonamientos e hipótesis vienen dirigidos por el personal que nos ha encargado el caso, por lo que nuestro trabajo en ese sentido está limitado.

Cada una de las conclusiones que se extraiga del análisis deberá estar justificada por alguna

evidencia que se haya identificado durante el proceso de extracción y análisis.

Es posible que dadas las evidencias existentes, no se puedan realizar ciertas afirmaciones con rotundidad, por lo tanto se deben limitar nuestras afirmaciones a lo que indican las evidencias.

5. CONCLUSIONES Y TRABAJO FUTURO

CONCLUSIONES

El análisis forense de dispositivos móviles es importante por varias razones.

En primer lugar, por su protagonismo en la escena delictiva y en todo tipo de conflictos entre particulares y empresas.

Los dispositivos móviles, son auténticos ordenadores equipados con diversas interfaces de comunicación inalámbrica a redes públicas y privadas, memoria de almacenamiento interna y externa, giroscopio y sensor de inercia.

Tienen más potencia que los *PC* de sobremesa de hace más de cinco años y se ven involucrados en la escena del crimen de formas diversas: como objetivo de ataque para la instalación de *malware*, el robo de datos y las intrusiones en redes. También, como objeto de valor susceptible de ser sustraído y, sobre todo, como herramienta de piratería informática.

Con un dispositivo móvil es posible enviar anónimos, transportar información confidencial y llevar a cabo otras acciones ilícitas.

Por tanto, el estudio forense de dispositivos móviles constituye una necesidad perentoria para cualquier perito forense, dado que un dispositivo móvil plantea dificultades que afectan a principios de la informática forense tradicional, como por ejemplo el requisito de no alterar las pruebas.

Además, en la mayor parte de los casos resulta imposible realizar adquisiciones forenses con un terminal que no esté en funcionamiento, aparte de otras circunstancias que se han visto en los distintos capítulos de esta memoria, correspondientes a la tecnología del *smartphone* y los procedimientos de investigación.

Finalmente, indicar que los terminales móviles, por su proximidad a la persona que los utiliza, contienen información que afecta a su esfera de privacidad y a derechos reconocidos por las leyes.

TRABAJO FUTURO

En muchas ocasiones, en el entorno móvil, el fichero binario no puede ser analizado directamente. Este hecho puede deberse, por ejemplo a la forma en la que está empaquetado o al cifrado que protege el código ejecutable del binario.

Por esa razón, el siguiente trabajo o línea de investigación podría consistir en realizar el análisis estático de aplicaciones empaquetadas (tanto las que se encuentran en un proyecto compilable con el código fuente disponible, como las que están en un fichero binario que no ha sido instalado).

Por otra parte, dado que el mundo de los dispositivos móviles cuenta con distintas plataformas, cada una basada en un sistema operativo, quizá podría resultar interesante trabajar con otro tipo de dispositivos distintos de *Android* y analizar las diferencias.

6. RECOMENDACIONES

BORRADO REMOTO

La mayor parte de los fabricantes disponen de medios de borrado remoto para evitar fugas de información ante el robo o perdida de los dispositivos.

Por ejemplo, *Android* dispone de forma nativa de *Android Device Manager* (*ADM*) desde la version *2.3* (*Gingerbread*) que permite opciones de bloqueo, localización o borrado remoto, emparejando el dispositivo a la cuenta de *Gmail*.

También existen soluciones *MDM* (*Mobile Device Management*) para entornos corporativos como *Microsoft Intune* o *Kaspersky Mobile Security* que pueden permitir el borrado remoto de los datos desde una consola centralizada.

Lógicamente, si un dispositivo es borrado de forma remota, puede complicarnos mucho un posible análisis; en algunos casos, es posible que necesitemos realizar un análisis a bajo nivel del *hardware* (dependiendo del tipo de borrado).

Otra posibilidad que debemos de conocer es el alcance del borrado remoto; dependiendo del dispositivo y su versión, es posible que el borrado remoto solo sea de los datos, de las aplicaciones o que no sea posible borrar la tarjeta *SD*, por lo que todavía podríamos acceder a información almacenada en ella durante nuestro análisis.

Con el uso actual de los dispositivos móviles, dado que la mayor parte del tiempo se encuentran conectados a *Internet*, se deben tomar precauciones a la hora de incautar o analizar un dispositivo para evitar que sea borrado de forma remota.

Dado que las herramientas de borrado remoto suelen funcionar a través de la conexión de datos, lo primero es quitarle la conexión a la red de datos. No obstante, podría existir una aplicación que iniciara el proceso ante la recepción de una llamada o *SMS* concreto, por lo que la opción más rápida es poner el terminal en modo avión.

El poner el terminal en modo avión implica que su conectividad al exterior queda casi eliminada, no obstante, el terminal puede conectarse mediante *WiFi* o *Bluetooth* de forma manual tras la activación del modo avión, por lo que, para evitar cualquier tipo de conexión, lo mejor es disponer de una jaula de *Faraday* donde ubicar el terminal requisado de forma segura. Lógicamente, esto puede hacernos replantear nuestro proceso de análisis, al menos durante el proceso de adquisición y transporte de las evidencias.

Una vez disponemos del terminal aislado, o al menos, lo más aislado posible, debemos hacer una imagen del sistema, de la tarjeta *SD* (en caso de disponer de ella), de la memoria, y de cualquier otro medio que pueda contener información de interés para el análisis. Para ello, podemos usar herramientas libres como las que se han visto, u otras comerciales como *Paraben* u *Oxigen Forensic*, que nos permiten la adquisición de datos y su posterior análisis.

CIFRADO DE DISCO

En *Android*, el cifrado completo de disco se implementó con la versión *3.0* (*Honeycomb*), una versión para *tablets* muy poco utilizada.

A partir de la versión *4.4* (*KitKat*) se empezó a generalizar (para el cifrado de datos

personales). Y desde la versión *5.0* ya se podía cifrar casi todo el dispositivo[9].

Si obtenemos un terminal con el disco completamente cifrado, esto tiene un impacto alto en el resultado del análisis forense, dado que limita todos los análisis que intentemos realizar en el dispositivo. En este sentido, habría que realizar un trabajo extra para intentar obtener el código de desbloqueo, trabajo que puede resultar arduo.

En la mayoría de casos se utilizan ataques de ingeniería social, para obtener el código. Otras veces, se realizan ataques por fuerza bruta.

De todos modos, una de las recomendaciones para reducir su impacto durante el proceso de análisis es, si encontramos el dispositivo desbloqueado, intentar mantenerlo así hasta la extracción de datos. Otra recomendación es utilizar una jaula de *Faraday* para evitar que se pueda acceder al dispositivo de forma remota y de este modo se produzca un borrado de los datos.

Por otra parte, existen algunas alternativas, como por ejemplo *Passware Kit Forensic* y *Cellebrite*, que permiten mitigar el cifrado de disco.

BLOQUEO POR CÓDIGO

Otro problema se presenta cuando el usuario del terminal ha configurado un bloqueo de seguridad, para evitar el acceso por parte de terceros.

Entonces, no solo será imposible visualizar datos, aplicaciones y otros elementos de evidencia, sino que tampoco podremos conectar el dispositivo a un ordenador para acceder a dispositivos móviles protegidos por bloqueo. Cada uno de ellos supone riesgos técnicos y jurídicos.

En este sentido. La posibilidad de sortear con éxito un código de bloqueo depende de las circunstancias, la marca y modelo del terminal, y también de si este ha sido *rooteado* o sometido a manipulaciones que puedan llegar a alterar los datos o poner en peligro la cadena de custodia.

Por tanto, en un dispositivo protegido mediante un sistema de bloqueo, nuestra investigación habrá llegado a un punto muerto si no somos capaces de hallar una manera que nos permita acceder al terminal sin conocer el código. De este modo, lo único que se puede hacer es poner el terminal en modo avión, impidiendo así el acceso desde el exterior.

Pero, si tenemos suerte y la depuración *USB* está activada, podemos acceder a través de *Android SDK*, aunque esté protegido por bloqueo (*PIN*, patrón o contraseña). En este caso, existe la posibilidad de eliminar el bloqueo borrando los archivos que contienen los *hashes* del patrón o la contraseña/*PIN*[10].

En tales circunstancias, al menos una parte de los contenidos del terminal estará disponible para ser adquirida por medios forenses. Si además, el dispositivo está *rooteado*, entonces habremos tenido suerte. No hace falta decir, sin embargo, que la depuración *USB* activa en un teléfono - y no digamos el *rooting* - no es un caso que se presente con frecuencia.

[9] Los datos en la tarjeta de memoria no son cifrados.

[10] los códigos de bloqueo se guardan en dos archivos cuyas rutas en el árbol de directorios *Android* son mbox*/data/system/gesture.key* (para el patrón) y */data/system/password.key* (para el *PIN* o la contraseña)

7. REFERENCIAS

Srinivasa Rao Kotipalli; Mohammed A. Imran (julio de 2016). *Hacking Android. Packt Publishing Ltd.*

Rohit Tamma; Donnie Tindall (abril de 2015). *Learning Android Forensics. Packt Publishing Ltd.*

Soufiane Tahiri (mayo de 2016). *Mastering Mobile Forensics. Packt Publishing Ltd.*

Vijay Kumar Velu (marzo de 2016). *Mobile Application Penetration Testing. Packt Publishing Ltd.*

Prashant Verma; Akshay Dixit (junio de 2016). *Mobile Device Exploitation Cookbook. Packt Publishing Ltd.*

Oleg Afonin; Vladimir Katalov (septiembre de 2016). *Mobile Forensics - Advanced Investigate Strategies. Packt Publishing Ltd.*

Richard Boddington (mayo de 2016). *Practical Digital Forensics. Packt Publishing Ltd.*

Heather Mahalik; Rohit Tamma; Satish Bommisetty (mayo de 2016). *Practical Mobile Forensics. Packt Publishing Ltd.*

Lee Reiber (noviembre de 2015). *Mobile Forensic Investigations: A Guide to Evidence Collection, Analysis, and Presentation. McGraw Hill Education.*

Francisco Lázaro Domínguez (octubre de 2015). Investigación forense de dispositivos móviles *Android. Ra-Ma.*

Páginas *web*:

Estas son solo algunas de las páginas *web* de referencia más importantes citadas en el texto. Para más información sobre forénsica de *Android* existe en *Internet* gran número de enlaces disponibles directamente a través de *Google* y otros buscadores:

- *http://www.ietf.org/rfc/rfc3227.txt*
- *http://developer.android.com/guide/index.html*
- *https://sourceforge.net/projects/adbextractor/*
- *http://www.sqliteviewer.org/*
- *https://www.elcomsoft.com/eppb.html*
- *http://forensics.spreitzenbarth.de/2012/02/28/cracking-the-pattern-lock-on-android/*
- *http://forensics.spreitzenbarth.de/2015/08/12/breaking-the-screenlock-a-short-update/*
- *http://www.cellebrite.com/Mobile-Forensics/Applications*
- *http://www.oxygen-forensic.com/*
- *http://home.gna.org/bless/donwloads.html*
- *http://sqlitebrowser.org*
- *https://play.google.com/store/apps/details?id=stericson.busybox*

8. GLOSARIO

Binder IPC – Es un *driver* cuyo cometido consiste en administrar el intercambio de información entre procesos. En *Android* las aplicaciones no se ejecutan dentro del mismo espacio de memoria y recursos, sino que cada una de ellas tiene asignada su propia zona (*sandboxing*), para no interferir con otros programas y también por razones de seguridad. La comunicación se establece a través de mensajes *IPC* (*inter process communication*). *Binder* es el controlador que gestiona este intercambio de señales (está basado en una arquitectura servidor-cliente).

EnCase Forensic – Es una herramienta de pago referente en el mundo del análisis forense. Entre su amplio abanico de funcionalidades incluye la de identificar ficheros cifrados y la de intentar descifrarlos mediante *Passware Kit Forensic*, una utilidad que incorpora *backups* o imágenes.

EXT4 – Es un sistema de archivos estándar con soporte para permisos, comandos y procedimientos característicos de *Linux*.

LiME – *Linux Memory Extractor* es un *software* que permite la obtención de un volcado de memoria volátil de un dispositivo basado en *Linux*, como es el caso de los dispositivos móviles *Android*. Asimismo, presenta la ventaja de que puede ser ejecutado remotamente a través de la red.

Oxygen Forensics – Es una herramienta de pago capaz de obtener información de más de 10.000 modelos diferentes de dispositivos móviles e incluso obtener información de servicios en la nube e importar *backups* o imágenes.

Oxygen extrae información sobre los contactos, *SMS*, llamadas, calendarios, tareas, notas, todo el *filesystem*, diccionarios de usuario, *apps* y *passwords*, historial de conexiones *wireless*, geo-coordenadas, etc.

Perito - Persona que, poseyendo determinados conocimientos científicos, artísticos, técnicos o prácticos, informa, bajo juramento, al juzgador sobre puntos litigiosos en cuanto se relacionan con su especial saber o experiencia.

El perito debe ser aséptico a todo entorno exterior de la pericia. Debe reflejar en sus informes de forma exacta y precisa, los datos obrantes en su poder sin extrapolar ni interpolar. Fiel reflejo de la realidad debe ser su informe.

La ley de Enjuiciamiento Civil se convierte en Ley marco como norma procesal y proclama en su artículo 3, más allá de los procesos civiles, su extensión a todos los ámbitos y materias, deviniendo supletoria únicamente frente a disposiciones específicas que regulen los procesos penales, contencioso-administrativos, laborales y militares. Es por ello que la relación del perito con la Administración de Justicia, salvo las especificidades aludidas y demás leyes interrelacionadas, se encuentra fundamentalmente regulada en ella, con independencia de la jurisdicción en la que actúe.

Sandboxing – Consiste en el aislamiento de aplicaciones por parte de los sistemas operativos en entornos móviles, dado que por lo general, las aplicaciones son consideradas como no confiables. Por tanto, es un mecanismo de seguridad que se utiliza en aplicaciones tan conocidas como *Google Chrome* y que limita el acceso al sistema operativo y a otras aplicaciones. En este sentido, la interacción entre aplicaciones se lleva a cabo siempre con la mediación del SO y el acceso a recursos importantes de éste es controlado por un sistema

de permisos.

El *sandboxing* implica que una aplicación puede escribir sólo en una carpeta especialmente asignada dentro del almacenamiento compartido.

Santoku Linux – Distribución de *Linux* especializada en el análisis de seguridad en dispositivos móviles. Incluye multitud de herramientas que son de utilidad para la realización del análisis dinámico y estático. Se encuentra disponible en ***http://santoku-linux.com/download/***.

YAFFS (*yet another flash file system*)– Es un sistema de archivos nativo de *Android* que fue desarrollado para adaptarse a las características específicas de las memorias *NAND*, dispositivo de almacenamiento de datos utilizado por la mayor parte de los terminales *Android* y otros aparatos móviles con sistemas operativos empotrados. En la actualidad, *YAFFS* es un sistema de archivos en desuso, dado que carece de soporte multihilo, lo cual genera cuellos de botella en el rendimiento de los nuevos dispositivos provistos de procesadores multinúcleo.

9. ANEXOS