Escuela Politécnica Superior de Jaén

Ingeniería en Informática

UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Proyecto Fin de Carrera

HERRAMIENTA DE APOYO PARA EL ANÁLISIS FORENSE DE COMPUTADORAS

Alumno: José Arquillo Cruz

Tutor: Prof. D. Manuel José Lucena López

Dpto: Informática

Septiembre, 2007

D. Manuel José Lucena López, del Departamento de Informática de la Universidad de Jaén,

INFORMA

Que la memoria titulada "*Herramienta de apoyo para el análisis forense de computadoras*" ha sido realizada por **Dª. José Arquillo Cruz** bajo mi dirección y se presenta como memoria del Proyecto Fin de Carrera relizado para optar al grado de Ingeniera en Informática.

Jaén, 14 de septiembre de 2006

Vº Bº

Fdo: Manuel José Lucena López

INDICE DE CONTENIDOS

Agradecimientos

A mis padres por aguantarme y apoyarme en todo momento. A mi tutor del proyecto, Manuel José Lucena, por orientarme y animarme en los momentos decisivos, así como por aportarme sus enormes conocimientos sobre sistemas operativos y seguridad informática. A todos los profesores a los que he sacado de quicio, en especial a Luís Martínez, por ayudarme con el apartado del análisis y diseño de la aplicación y a Arturo Montejo por mostrarme el proceso de creación de un CD-Live. Y por último, a mi compañera de clase y amiga Eli, sin cuya ayuda nada de esto hubiera sido posible.

PARTEI: INVESTIGACIÓN

1 INTRODUCCIÓN

1.1 Contexto

El **Análisis Forense** es la aplicación de métodos científicos en investigaciones criminales. Es un campo único de estudio que deriva de todas las áreas de la ciencia, desde la entomología a la genética, desde la geología a las matemáticas, con el único objetivo de resolver un misterio. Esto levanta una gran expectación para el público en general. Gracias a las series de TV, millones de personas están familiarizadas con cómo una marca de rifle en una bala puede identificar el arma de un asesinato y como el "luminol" se usa para revelar manchas de sangre en el baño.

El **Análisis Forense de Ordenadores** estudia como éstos están involucrados en la realización de un crimen. Entre estos podemos citar el fraude de contabilidad, chantaje, robos de identidad, pornografía infantil o las intrusiones de un hacker black-hat en un sistema. En todos estos casos, los contenidos de un disco duro pueden contener evidencias críticas de un crimen. El análisis de discos y el rastreo de e-mails entre varias personas se han convertido en herramientas comunes para las fuerzas de la ley en todo el mundo. Una definición de "Computer Forensics" tambien llamado Análisis forense de computadores, Informática forense, Análisis de computadores, Recuperación de datos, etc., es la que encontramos en [1]:

"El análisis forense de computadoras es el proceso de examinar dispositivos metódicamente (discos duros, disquetes, etc.) en busca de evidencias.

Después de que ha ocurrido un crimen o incidente que implique una computadora, un especialista adiestrado en informática forense puede examinar la misma para encontrar pistas de lo que ha pasado. Este es el papel del examinador forense de computadoras. Este especialista podría trabajar para el estado como agente de la ley, o para una empresa privada en algunos casos, como los incidentes de seguridad en un sistema. Aunque en cada uno de los dos casos la ley es diferente, la estrategia de investigación para el especialista es más o menos la misma.

En los últimos años ha habido una explosión del interés sobre el estudio de evidencias digitales. Este crecimiento ha provocado acalorados debates sobre herramientas, terminología, definiciones, estándares, ética, y otros muchos aspectos de este campo en desarrollo. Según la bibliografía que se consulte se obtendrá la opinión subjetiva del autor en su obra, ya que no existe un método exacto que nos permita obtener la "verdad científica" para todos los casos. Existen en todo caso recomendaciones y "buenas prácticas" que generalmente son aceptadas y que todo examinador debería conocer.

1.2 Un poco de historia

Las pruebas extraídas de las computadoras se admiten como prueba en un juicio desde los años 70, pero en su fase más temprana las computadoras no se consideraban más que un dispositivo para almacenar y reproducir registros de papel, que constituían la evidencia real. Las versiones impresas de registros de contabilidad eran aceptadas como el equivalente de expedientes de negocio conservados en mano o escritos a máquina, pero no se contaba con los datos almacenados en la computadora.

El análisis forense de computadoras (Computer Forensics) es una ciencia relativamente nueva, por lo que aún no hay estándares aceptados. Sus orígenes se remontan a los Estados unidos a mediados de los años 80. Respondiendo al crecimiento de crímenes relacionados con las computadoras, los Estados Unidos comenzaron a desarrollar programas de adiestramiento y a construir su propia infraestructura para ocuparse del problema. Estas iniciativas derivaron en centros como SEARCH, Federal Law Enforcement Center (FLETC), y el National White Collar Crime Center (NW3C).

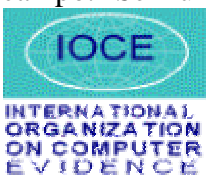
En **1985** se crea el FBI Magnetic Media Program, que más tarde pasará a ser el Computer Analysis and Response Team (CART)

En **1990**, el Laboratorio de Inspección Postal de los Estados Unidos se traslada a una nueva instalación en Dulles, Virginia, y entre 1996 y 1997 establece una unidad de Informática Forense. Trabaja junto con el FBI durante muchos años en el desarrollo de sus habilidades en informática forense.

En **1993** se celebra la primera conferencia anual sobre evidencias de computadoras (First International Conference on Computer Evidence).

En **1994**, el juicio de O.J. Simpson expuso muchas de las debilidades de la investigación criminal y la ciencia forense. La investigación fue entorpecida desde el inicio con colecciones de evidencias, documentación y preservación de la escena del crimen incompletas. Como resultado de estos errores iniciales, científicos forenses especializados estaban confundidos y sus interpretaciones solo incrementaron la duda de los miembros del jurado. La controversia que rondaba este caso puso de manifiesto que investigadores y científicos forenses no eran fiables como previamente se creía, socavando no solo su credibilidad sino también su profesión. Esta crisis motivó a muchos laboratorios y agencias de investigación a revisar sus procedimientos, mejorar su entrenamiento y hacer otros cambios para evitar situaciones similares en el futuro.

Por esa época hubo muchos desarrollos notables hacia la estandarización en este campo. Se fundó la Organización Internacional de Evidencias de Computadoras



a mediados de los 90 que anunció “asegurar la armonización de métodos y prácticas entre naciones y garantizar el uso de evidencias digitales de un estado en las cortes de otro estado”.

En España se crea en 1995 la Brigada de Investigación Tecnológica, perteneciente al Cuerpo Nacional de Policía. Comenzaron con 3 agentes de policía.

En **1997**, los países del G8 declararon que “la policía debe estar adiestrada para hacer frente a delitos de alta tecnología” en el Comunicado de Moscú de diciembre. En Marzo del año siguiente, el G8 designa al IOCE para crear principios internacionales para los procedimientos relacionados con la evidencia digital.

Ese mismo año se crea el Grupo de Delincuencia Informática de la Guardia Civil, que pasó a llamarse Grupo de Investigación de Delitos de Alta Tecnología antes de tomar su nombre actual de Grupo de Delitos Telemáticos.

Los directores del Laboratorio Federal de Crímen en Washington, DC, se reunieron dos veces en **1998** para discutir asuntos de interés mutuo. Se formó lo que es ahora conocido como el Scientific Working Group Digital Evidence (SWGDE). El concepto de encontrar “evidencias latentes en una computadora” se pasó a llamar infomática forense. El concepto de evidencia digital, que incluye audio y video digital se llevó ante los directores del laboratorio federal el 2 de Marzo de 1998, en un encuentro albergado por Servicio de Inspección Postal de los Estados Unidos y la División de Servicios Técnicos. La primera discusión se centraba principalmente en la fotografía digital. El resultado de esa reunión fue que se necesitaba personal experto para abordar el tema, por lo que e12 de Mayo de ese año se reunieron de nuevo con expertos del FBI y de otros grupos especializados en el tema. De ese encuentro surgió la formación de otro Grupo de trabajo técnico para tratar los asuntos relacionados con la evidencia digital.

El 17 de Junio de 1998, el SWGDE celebra su primer encuentro, dirigido por Mark Pollitt, agente especial del FBI y Carrie Morgan Whitcomb, del departamento forense del Servicio de Inspección Postal de los Estados Unidos. Como laboratorios forenses invitados estuvieron los del Departamento de Alcohol, Tabaco y Armas de Fuego(ATF), el Departamento de Control de Drogas(DEA), Inmigración(INS), Hacienda(IRS), la NASA, los Servicios Secretos(USSS) y el servicio de Inspección Postal. Decidieron algunos procedimientos administrativos y desarrollaron documentos relevantes. Se establece que “La evidencia digital es cualquier información de valor probatorio que es almacenada o transmitida en formato binario”. Más tarde “binario” cambió a “digital”. La evidencia digital incluye hardware, audio, video, teléfonos móviles, impresoras, etc.

Ese mismo año se celebra el primer Simpósium de ciencia forense de la INTERPOL.

En **1999**, la carga de casos del FBI CART excede los 2000 casos, habiendo examinado 17 terabytes de datos. El IOCE presenta un borrador con estándares sobre informática forense al G8.

En el año **2000** se establece el primero laboratorio de informática forense regional del FBI.

The FBI Laboratory Seal

En **2001**, se realizó el primer taller de investigación forense digital -Digital Forensics Research Work Shop (www.dfrws.org)-, reuniendo a los expertos de universidades, militares y el sector privado para discutir los retos principales y buscar las necesidades de este campo. Este taller también impulsó una idea propuesta muchos años atrás, provocando la creación de la Publicación Internacional de Evidencias Digitales -*International Journal of Digital Evidence* (www.ijde.org)-.

El rápido desarrollo de la tecnología y los crímenes relacionados con computadoras crean la necesidad de especialización:

- "First Responders" (Técnicos de escena de crimen digital): expertos en recogida de datos de una escena del crimen. Deberían tener entrenamiento básico en manejo de evidencias y documentación, así como en reconstrucción básica del crimen para ayudarles a ubicar todas las fuentes posibles de evidencias.

- Analistas de Evidencias Digitales: procesan la evidencia adquirida por los anteriores para extraer todos los datos posibles sobre la investigación.

- Investigadores digitales: analizan todas las evidencias presentadas por los dos anteriores para construir un caso y presentarlo ante los encargados de tomar las decisiones.

Estas especializaciones no estan limitadas solamente a los agentes de la ley y se han desarrollado también en el mundo empresarial. Aún cuando una sola persona sea responsable de recopilar, procesar y analizar las evidencias digitales, es útil considerar estas tareas por separado. Cada área de especialización requiere diferentes habilidades y procedimientos; tratándolos por separado hace más fácil definir el adiestramiento y los estándares en cada área. Entendiendo la necesidad de estandarización, en **2002**, el Scientific Working Group for Digital Evidence (SWGDE) publicó unas lineas generales para el adiestramiento y buenas prácticas. Como resultado de estos esfuerzos, la American Society of Crime Laboratory Directors (ASCLD) propuso requerimientos para los analistas forenses de evidencias digitales en los laboratorios. Hay además algunos intentos de establecer estándares internacionales (ISO 17025; ENFSI 2003).

A finales del año **2003** y respondiendo al creciente interés del análisis forense de intrusiones en computadoras, se propone el primer Reto de Análisis Forense por parte

de Rediris, en el cual se publica la imagen de un disco duro que ha sufrido un incidente de seguridad y se reta a responder a las siguientes preguntas:

- ¿Quién ha realizado el ataque ?, (dirección IP de los equipos implicados en el ataque)

- ¿Comó se realizo el ataque ? (Vulnerabilidad o fallo empleado para acceder al sistema)

- ¿Qué hizo el atacante ? (Que acciones realizo el atacante una vez que accedío al sistema, ¿por qué accedió al sistema ?).

Al final 14 personas enviaron el informe a Rediris de los casi 500 que se presentaron, y los ganadores se llevaron licencias y manuales de software de Análisis Forense (valorados en miles de dólares).

En **2004** los Servicios de Ciencia Forense del Reino Unido planean desarrollar un registro de expertos cualificados, y muchas organizaciones Europeas, incluyendo la Red Europea de Institutos de Ciencia Forense publicaron líneas básicas para investigadores digitales. Además, Elsevier comenzó la publicación de una nueva revista llamada "Digital Investigation: The International Journal of Digital Forensics and Incident Response" (http://www.compseconline.com/digitalinvestigation/).

A comienzos del **2005** se celebra el Reto Rediris v2.0, junto con la Universidad Autónoma de México. Se presentaron casi 1000 participantes y los premios fueron cursos de análisis forense y licencias de software. El segundo premio fue para uno de los ingenieros de la universidad de Granada.

A mediados del **2006** se celebra el III Reto Rediris, en el cual había 3 premios para los mejores de España y 3 para los mejores de Iberoamérica.

1.3 Objetivo y Metodología

Objetivo

Realización de una aplicación "asistente" que sirva para el aprendizaje de las técnicas de análisis forense.

Metodología

Para alcanzar el objetivo del proyecto se van a seguir los siguientes pasos:

- Revisión bibliográfica sobre la metodología del análisis forense de computadores.

- Revisión y evaluación de las herramientas empleadas en los análisis forenses de computadoras.
- Elegir un SSOO y un lenguaje de programación que nos permita realizar interfaces gráficas.
- Seleccionar las herramientas más adecuadas para poder realizar un análisis forense.
- Análisis, diseño e implementación de la aplicación asistente, que mostrará de manera gráfica el funcionamiento de estas herramientas.
- Masterización de un CD autoarrancable que integre el asistente.
- Realización de pruebas.

1.4 Motivación del alumno

Entre las motivaciones que me han llevado a elegir este proyecto las principales son:

- Trabajar sobre el campo de la seguridad informática, tan de moda hoy en día debido a la expansión de Internet.
- Investigar un área novedosa en la seguridad informática, poco documentada en español y sobre la que apenas existen estándares como es el análisis forense de computadoras.
- Contribuir con una aplicación de utilidad para los alumnos de la asignatura Seguridad en Sistemas Informáticos y para el que desee usarla.
- Utilizar un lenguaje de programación de alto nivel que supera en sencillez a Java: Python.
- Expandir mis conocimientos sobre Linux.
- Aumentar mi vocabulario cientifico/tecnico en inglés.

2 APLICANDO LA CIENCIA FORENSE A LAS COMPUTADORAS

El análisis de evidencias digitales es un proceso similar al corte de un diamante. Al quitar el material áspero innecesario, se puede ver el cristal puro debajo. El diamante es esculpido y pulido para permitir a otros apreciar sus facetas. De manera similar, un examinador de evidencias digitales extrae bits relevantes de grandes masas de datos y los presenta de manera que los pueda comprender el que tomará las decisiones. En ambos casos, las imperfecciones en el material subyacente reducen el valor final del producto.

Continuando con la analogía, excavar diamantes en bruto de la tierra requiere un conjunto de habilidades, mientras que un cortador de diamantes requiere otro conjunto completo de habilidades. Un joyero que examina gemas para certificar su pureza y que las combina para obtener una pieza más grande, requiere otro conjunto de habilidades. Los investigadores digitales a menudo realizan todas las tareas requeridas de recolección, documentación y conservación de evidencias digitales para extraer datos útiles y combinarlos para crear una imagen más clara de lo que ocurrió en general. Los investigadores digitales necesitan una metodología para ayudarles a realizar estas tareas correctamente, encontrar la verdad científica, y en último caso, ser admitido como prueba en un juicio.

Aquí es donde la ciencia forense es útil, ofreciendo métodos probados para procesar y analizar evidencias y alcanzar conclusiones. Los conceptos de la ciencia forense pueden ayudar a los investigadores a obtener más y mejores datos que podrían escapar si se examinara todo a simple vista.

Sin embargo, la informática forense es una ciencia relativamente nueva, por lo que si buscamos un método único y aceptado que nos guíe paso a paso en este proceso, no lo encontraremos. Existen en la literatura diferentes aproximaciones a un modelo que nos permita diferenciar las distintas fases por las que pasa un análisis de evidencias digitales. A continuación veremos una recopilación de los distintos modelos presentados en orden cronológico:

2.1 Modelo de Casey (2000)

Eoghan Casey, en el año 2000 presenta un modelo [2] para procesar y examinar evidencias digitales. Este modelo ha ido evolucionando en las siguientes Tiene los siguientes pasos principales:

1. La Identificación
2. La Conservación, la Adquisición, y la documentación
3. La clasificación, la comparación, y la individualización
4. La reconstrucción

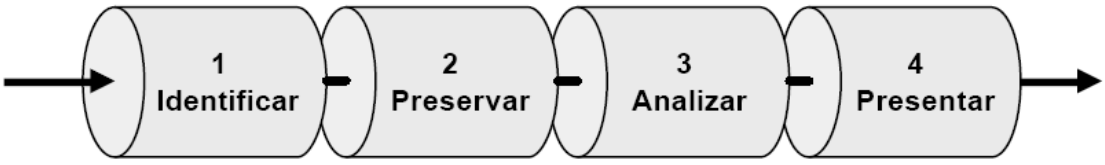
En los últimos dos pasos es cuando la prueba es analizada. Casey señala que éste es un ciclo de procesamiento de prueba, porque al hacer la reconstrucción pueden hallarse pruebas adicionales que provoquen que el ciclo comience. El modelo se replantea primero en términos de sistemas de cómputo sin tener en cuenta la red, y luego ejercido para las distintas capas de red (desde la capa física hasta la capa de aplicación, e incluyendo la infraestructura de la red) para describir investigaciones en redes de computadoras. El modelo de Casey es muy general y se aplica exitosamente para ambos sistemas, las computadoras aisladas y con entornos de red.

2.2 Modelo publicado por el U.S. Dep. of Justice (2001)

Este modelo [3] se publicó en el año 2001 y quizás sea el más sencillo.

Básicamente existen cuatro elementos clave en un análisis forense de computadoras, que son:

1. Identificación
2. Conservación
3. Análisis
4. Presentación



Este modelo supuso una de las grandes bases en este campo ya que a partir de estos conceptos clave, varios autores han desarrollado sus modelos para englobar todos los pasos de una investigación forense de computadoras.

2.3 Modelo de Lee (2001)

Lee et. al [2] propone la investigación como un proceso. Este modelo se ocupa sólo de investigación de la escena de delito, y no del proceso investigador completo. Identifica cuatro pasos dentro del proceso:

- Reconocimiento
- Identificación
- Individualización
- Reconstrucción

El **reconocimiento** es el primer paso, en el cual se buscan ítems o patrones como pruebas potenciales. El investigador debe saber qué partes mirar y dónde puede ser encontrado. El reconocimiento deriva en dos subactividades: La documentación y la adquisición/preservación.

La **identificación** de los tipos diversos de prueba es el siguiente paso. Esto implica la clasificación de la prueba, y una subactividad, la comparación. Físicas, biológicas, químicas, y otras propiedades de los artículos de prueba son comparadas según los estándares conocidos.

La **individualización** se refiere a determinar si los ítems de prueba posible son únicos a fin de que puedan ser conectados con un acontecimiento o individuo particular. Dentro de esto, los ítems deben ser evaluados e interpretados.

La **reconstrucción** implica unificar el significado de las salidas de las anteriores partes del proceso, y cualquier otra información pertinente que los investigadores pudieron haber obtenido, para proveer una detallada relación de los acontecimientos y las acciones en la escena de delito. Esto deriva en las fases de información y la presentación.

Basado en los pasos citados anteriormente, Lee Et Al . describe árboles lógicos para varios tipos diferentes de escenas según el tipo de delito, es decir, una serie de acciones relatadas que el investigador puede usar para asegurar la probabilidad más alta de que toda prueba pertinente será reconocida, identificada e individualizada, conduciendo a una reconstitución útil.

2.4 Modelo del DFRWS (2001)

El primer Forensics Digital Research Workshop (Palmer, 2001) produjo un modelo [4] que muestra los pasos para el análisis forense digital en un proceso lineal. Los pasos son los siguientes:

1. La identificación
2. La preservación
3. La colección
4. El examen
5. El análisis
6. La presentación
7. La decisión

Identification	Preservation	Collection	Examination	Analysis	Presentation	Decision
Event/Crime Detection	Case Management	Preservation	Preservation	Preservation	Documentation	
Resolve Signture	Imaging Technologies	Approved Methods	Traceability	Traceability	Expert Testimony	
Profile Detection	Chain of Custody	Approved Software	Validation Techniques	Statistical	Clarification	
Anomalous Detection	Time Synch.	Approved Hardware	Filtering Techniques	Protocols	Mission Impact Statement	
Complaints		Legal Authority	Pattern Matching	Data Mining	Recommended Countermeasure	
System Monitoring		Lossless Compression	Hidden Data Discovery	Timeline	Statistical Interpretation	
Audit Analysis		Sampling	Hidden Data Extraction	Link		
Etc.		Data Reduction		Spacial		
		Recovery Techniques				

El modelo no pretende ser el definitivo, sino más bien como una base para el trabajo futuro que definirá un modelo completo, y también como una armazón para la investigación de futuro. El modelo DFRWS se replantea como lineal, pero la posibilidad de retroalimentación de un paso para los previos es mencionada. El informe DFRWS no discute los pasos del modelo con todo lujo de detalles sino por cada paso se listan un número de asuntos pendientes.

2.5 Modelo de Reith, Carr y Gunsch (2002)

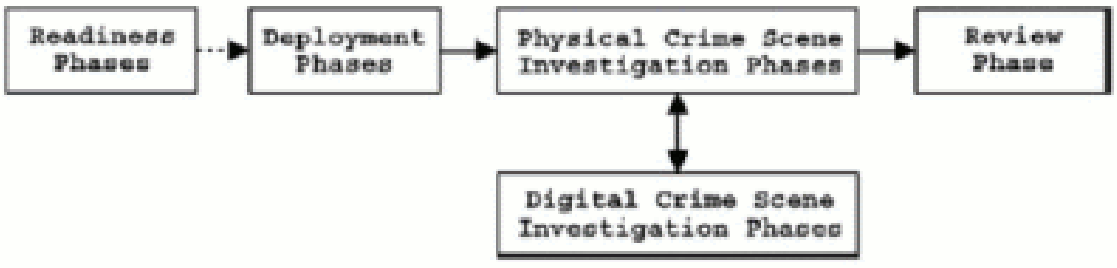
Reith, Carr y Gunsch (2002) [5] describen un modelo que hasta cierto punto deriva del modelo DFRWS. Los pasos en su modelo son:

1. La identificación
2. La preparación
3. La estrategia de acercamiento
4. La preservación
5. La colección
6. El examen
7. El análisis
8. La presentación
9. Devolviendo la evidencia

Este modelo es notable en cuanto a que explícitamente pretende ser un modelo abstracto aplicable para cualquier tecnología o cualquier tipo de ciberdelito. Se pretende que el modelo pueda ser utilizado como base otros métodos más detallados para cada tipo específico de investigación.

2.6 Modelo integrado de Brian Carrier y Eugene Spafford (2003)

Brian Carrier y Eugene Spafford [6] han propuesto otro modelo que organiza el proceso en cinco grupos, cada uno dividido en 17 fases.



Fases de Preparación

El objetivo de esta fase es asegurar que las operaciones e infraestructuras están preparadas para soportar una investigación completa. Incluye dos fases:

- Fase de preparación de operaciones: que asegura que los investigadores están adiestrados y equipados para tratar con un incidente cuando este ocurre.

- Fase de preparación de infraestructuras: que asegura que la infraestructura subyacente es suficiente para tratar con incidentes. Por ejemplo, cámaras fotográficas, material de conservación y transporte de hardware, etc.

Fases de Despliegue

El propósito es proporcionar un mecanismo para que un incidente sea detectado y confirmado. Incluye dos fases:

1. Fase de Detección y Notificación: donde el incidente es detectado y y notificado a las personas apropiadas.

2. Fase de Confirmación y Autorización: en la cual se confirma el incidente y se obtiene la aprovación legal para llevar a cabo la búsqueda.

Fases de Investigación Física de la escena del crimen

La meta de estas fases es recopilar y analizar las evidencias físicas y reconstruir las acciones que ocurrieron durante el incidente. Incluye seis fases:

1. Fase de Conservación: que busca conservar la escena del crimen de modo que la evidencia pueda ser identificada más tarde y recolectada por personal adiestrado en identificación de evidencias digitales.

2. Fase de Inspección: que requiere que un investigador recorra la escena física del delito e identifique elementos de evidencia física.

3. Fase de Documentación: que incluye tomar fotografías y videos de la escena del delito y de la evidencia física. El objetivo es capturar tanta información como sea posible de modo que el esquema y los detalles importantes de la escena del crimen son conservados y grabados.

4. Fase de búsqueda y recolección: que entraña una búsqueda y recolecolección en profundidad de la escena de modo que se identifican evidencias físicas adicionales y se establecen vías para comenzar la investigación digital.

5. Fase de Reconstrucción: que incluye organizar los resultados del análisis hecho usandolos para desarrollar una teoría del incidente.

6. Fase de Presentación: que presenta la evidencia digital y física en un juicio o ante la dirección de una empresa.

Fases de Investigación de la Escena Digital del Delito

El objetivo es recolectar y analizar la evidencia digital que se obtuvo de la fase de investigación física y a través de otras fuentes. Incluye fases similares a las de la investigación física, aunque en este caso el objetivo principal es la evidencia digital. Las seis fases son:

1. Fase de conservación: que conserva la escena digital del delito de modo que la evidencia pueda ser después analizada.

2. Fase de Inspección: por la que el investigador transfiere los datos relevantes de una jurisdicción que está fuera del control físico o administrativo del investigador, a una posición controlada.

3. Fase de Documentación: que incluye documentar la evidencia digital cuando es encontrada. Esta información es útil en la fase de presentación.

4. Fase de Búsqueda y Recolección: se realiza un análisis en profundidad de la evidencia digital. Se usan herramientas software para revelar ficheros ocultos, borrados, corruptos, etc. Además se suelen usar lineas de tiempo para ver la actividad de los ficheros y usuarios en un instante o periodo dado.

5. Fase de reconstrucción: que incluye ubicar las piezas del puzle y desarrollar una hipótesis investigativa..

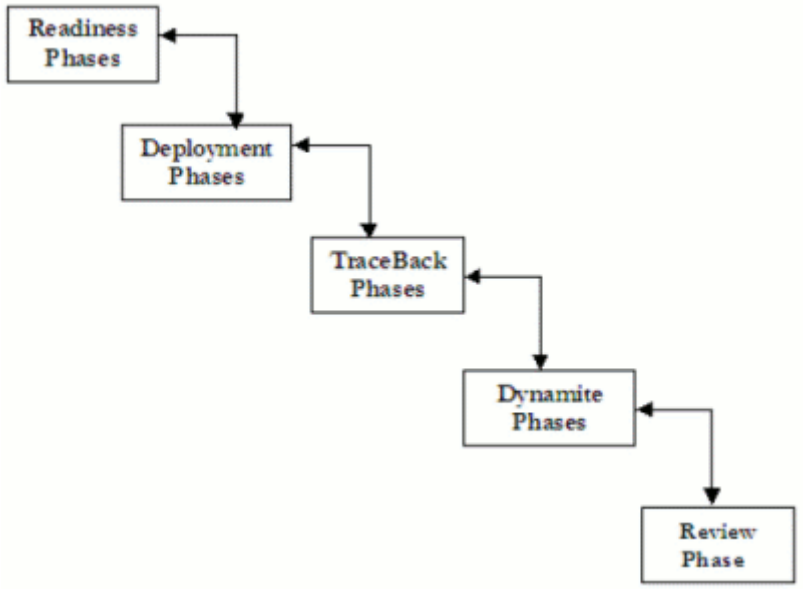
6. Fase de Presentación: que consiste en presentar la evidencia encontrada y unirla a la evidencia física encontrada.

Fase de revisión

Esto conlleva una revisión de la investigación entera e identifica áreas de mejora.

2.7 Modelo mejorado propuesto por Venansius Baryamureeba y Florence Tushabe (2004)

Este modelo [2] se basa en el anterior e intenta mejorar algunos aspectos, aunque básicamente son muy similares. Este modelo consiste en cinco fases principales:



Fases de Preparación

Las mismas que para el modelo anterior

Fases de despliegue

Proporcionan un mecanismo para detectar y confirmar un delito. Se realizan en el mismo lugar donde se detectó el delito y consisten en cinco fasees:

1. Fase de Detección y Notificación: cuando se detecta un incidente y se notifica a las personas apropiadas.

2. Fase de Investigación Física de la escena del delito: cuando se examina la escena física del delito y se identifican evidencias digitales potenciales.

3. Fase de investigación Digital de la escena del delito: cuando se realiza una examinación de la escena y se obtienen evidencias con la consecuente estimación del impacto o daño causado al manipular el sistema en búsqueda de evidencias digitales.

4. Fase de Confirmación: cuando el incidente es confirmado y se obtienen autorización legal para realizar una investigación en profundidad.

5. Fase de informe; que supone presentar las pruebas físicas y digitales a personas jurídicas o a dirección corporativa.

Fases de Hipótesis

Dentro de estas fases se intentan reconstruir los hechos cometidos en la escena física del delito de forma que podamos identificar los dispositivos que se usaron para cometer el acto. Consiste en dos fases:

1. Investigación digital de la escena del delito: se elabora una primera hipótesis con las pistas obtenidas en fases anteriores. Por ejemplo, si tenemos una dirección IP sospechosa en nuesto sistema podemos rastrear su origen buscando por internet.

2. Fase de Autorización: cuando se obtiene autorización de las entidades locales para permitir investigaciones más exaustivas y acceder a más información.

Fases Dinamita

Estas fases investigan la hipótesis elaborada en el paso anterior. El objetivo de recopilar y analizar los elementos que se encontraron en la fase anterior es obtener más evidencias y poder asegurar que el delito ocurrió allí y/o encontrar posibles culpables. Consiste en cuatro fases:

- Fase de Investigación Física de la escena del delito: se examina de nuevo la escena física bajo el punto de vista de la hipótesis inicial buscando nuevas

evidencias digitales.

- Fase de Investigación Digital de la escena del delito: se examina la evidencia digital en busca de pruebas del incidente y permitir una estimación del momento en que ocurrió el incidente.
- Fase de Reconstrucción: reconstruir todas las piezas del puzle digital e identificar las hipótesis más probables.
- Fase de Comunicación: que consiste en elaborar la presentación de las interpretaciones y conclusiones finales sobre la evidencia física y digital que han sido investigadas por un juicio o por una empresa.
- Fase de Revisión

La investigación entera es revisada y se buscan áreas de mejora.

2.8 Modelo extendido de Séamus Ó Ciardhuáin (2004)

En el año 2004, el IJCE (**International Journal of Digital Evidence**) publica un modelo extendido [2] para investigaciones de ciberdelitos, cuyo autor fue Séamus Ó Ciardhuáin. Vamos a profundizar en este modelo para comprender el proceso completo de un análisis forense de computadoras. Según este artículo, los modelos existentes no cubren todos los aspectos de investigación de ciberdelito; enfocan principalmente la atención en el procesamiento de prueba digital. Aunque son valiosos, no son lo suficientemente generales para describir completamente el proceso investigador para ayudar al desarrollo de nuevas técnicas y herramientas investigadoras.

El mayor fallo en los modelos existentes es que explícitamente no identifican la información que fluye en las investigaciones. Por ejemplo, Reith Et Al. (2002) no menciona explícitamente la cadena de custodia en su modelo. Éste es un fallo primoridal cuándo uno considera las diferentes leyes, las prácticas, los lenguajes, etcétera que deben ser correctamente distribuido en investigaciones reales.

Otro asunto con los modelos existentes es que han tendido a concentrarse en la parte central del proceso de investigación, o sea la colección y el examen de la prueba digital. Sin embargo, las etapas anteriores y posteriores deben ser tomadas en consideración si desean lograr un modelo integral, y en particular si todos los flujos pertinentes de información a través de una investigación deben ser identificados.

Las actividades en una investigación se mencionan a continuación:

1. La conciencia
2. Autorización
3. Planificación
4. La notificación
5. Buscar e identificar pruebas
6. La colección de prueba
7. Transporte de prueba
8. El almacenamiento de prueba
9. El examen de prueba
10. La hipótesis
11. La presentación de hipótesis
12. La prueba /defensa de hipótesis
13. La diseminación de información

Este modelo tiene forma de cascada y las actividades se suceden unas a otras. Los flujos de información de una actividad a la siguiente pasan hasta el final del proceso de investigación. Por ejemplo, la **cadena de custodia** se forma por la lista de aquellos que han manipulado una evidencia digital y debe pasar de una etapa a la siguiente agregando los nombres en cada paso.

La conciencia: alerta de incidente

El primer paso en una investigación es la creación de una conciencia de que la investigación es necesaria. Esta conciencia es típicamente creada por acontecimientos externos a la organización que llevará a cabo la investigación, por ejemplo un delito se dice a la policía o un auditor recibe instrucciones de realizar una auditoría. También puede resultar de acontecimientos internos, por ejemplo un sistema de detección de intrusión alerta a un administrador de sistema de que la seguridad de un sistema ha sido comprometida.

La conciencia existe en este modelo porque hace más clara la investigación, estableciendo un punto de partida en el que basarnos. La mayoría de anteriores modelos explícitamente no muestran esta actividad y tampoco incluyen una relación visible para los acontecimientos causativos. Ésta es una debilidad de algunos modelos semejantes porque los acontecimientos que provocan la investigación pueden influenciar el tipo de investigación. Es vital tener en cuenta tales diferencias para asegurar que el planteamiento correcto es llevado a una investigación en un contexto particular.

Autorización

Después de que la necesidad de una investigación es identificada, la siguiente actividad es obtener autorización. Esto puede ser muy complicado y puede requerir interacción con ambas entidades externas e internas para obtener la autorización necesaria. En un extremo, un administrador de sistema puede requerir sólo una aprobación verbal simple de su compañía para que lleve a cabo una investigación detallada de los sistemas de cómputo; en el otro extremo, las instituciones para la aplicación de la ley usualmente requieren autorizaciones legales formales alegando en detalle lo que está permitido en una investigación.

La planificación

La actividad planificadora es fuertemente influenciada por información del interior y de fuera de la organización investigativa. De fuera, los planes serán influenciados por reglas y legislación que establecen el contexto general de la investigación y que no está bajo el control de los investigadores. También habrá información obtenida por los investigadores de otras fuentes externas. Desde dentro de la organización, allí estarán las propias estrategias de la organización, las políticas, y la información acerca de investigaciones previas.

La actividad planificadora puede dar lugar a la necesidad de volver hacia atrás y obtener más autorización, por ejemplo cuando el alcance de la investigación es mayor que la información que se puede obtener.

La notificación

La notificación en este modelo se refiere a informar al objetivo de la investigación o a otras partes concernientes que la investigación está teniendo lugar. Esta actividad puede no ser apropiada en algunas investigaciones, por ejemplo, cuando se requiere la sorpresa para prevenir la destrucción de la evidencia. Sin embargo, puede ser necesaria la notificación, o puede que otras organizaciones deban estar al tanto de la investigación.

La Búsqueda e Identificación De Evidencias

Esta actividad se ocupa de localizar la evidencia e identificar lo que es para la siguiente actividad. En el caso más simple, esto puede implicar encontrar la computadora usada por un sospechoso y confirmar que es de interés para los investigadores. Sin embargo, en más ambientes complicados esta actividad no puede ser franca; por ejemplo puede requerir rastrear computadoras a través de ISPs múltiples y posiblemente en otros países basándonos en el conocimiento de una dirección IP.

La adquisición

La adquisición es la actividad en la cual la organización investigativa toma posesión de la prueba en una forma que puede ser conservada y analizada, por ejemplo las imágenes de de discos duros o la recolección de computadoras enteras. Esta actividad es el foco de la mayoría de debate en la literatura por su importancia para el resto de la investigación. Los errores o las escasa práctica a estas alturas pueden inutilizar la prueba, particularmente en investigaciones que están sujetas a las normas legales estrictas.

Transporte

Después de colección, la prueba debe ser transportada a una ubicación adecuada para su posterior examen. Ésta podría ser simplemente la transferencia física de computadoras a una posición segura; sin embargo, también podría ser la transmisión de datos a través de redes. Es importante para asegurar durante el transporte que la prueba permanece válida para el posterior uso.

El almacenamiento

Las evidencias adquiridas en la mayoría de los casos necesitará ser almacenada porque la inspección no puede tener lugar inmediatamente. El almacenamiento debe tener en cuenta la necesidad de conservar la integridad de la prueba.

El examen

El examen de la prueba implicará el uso de un número potencialmente grande de técnicas para encontrar e interpretar datos significativos. Puede precisar reparación de datos dañados de forma que conservan su integridad. Según los resultados de la búsqueda /identificación y las actividades de adquisición, pueden haber volúmenes muy grandes de datos para ser examinados así es que se requieren las técnicas automatizadas para ayudar al investigador.

La hipótesis

Basados en el examen de la prueba, los investigadores deben construir una hipótesis de qué ocurrido. El grado de formalidad de esta hipótesis depende del tipo de investigación. Por ejemplo, una investigación de policía resultará en la preparación de una hipótesis detallada con material cuidadosamente documentado del examen, adecuado para el uso en los tribunales. Una investigación interna por el administrador de sistemas de una compañía resultará en al menos un informe formal para la gestión. Una vez que se tiene una hipótesis, puede volverse a la fase de examen, de forma que los investigadores desarrollen una comprensión mayor de los acontecimientos.

La presentación

La hipótesis debe presentarse a otras personas aparte de los investigadores. Para una investigación de policía la hipótesis será antepuesta a un jurado, mientras que para una compañía interna, se tomará en cuenta la hipótesis antes de tomar una decisión.

La Prueba /defensa

En general la hipótesis no será indiscutible; una hipótesis contraria y una evidencia comprobatoria serán antepuestas ante un jurado, por ejemplo. Los investigadores tendrán que probar la validez de su hipótesis y defenderla en contra de las críticas de la defensa o el acusación. Los mejores resultados pueden obtenerse si se usa el retroceso para obtener más pruebas y construir una mejor hipótesis.

La difusión

La actividad final en el modelo es la difusión de información de la investigación.

Alguna información puede estar disponible solamente dentro de la organización investigativa, mientras otra información puede ser más ampliamente difundida. La información influenciará investigaciones futuras y también puede influenciar las políticas y los procedimientos. La recopilación y mantenimiento de esta información son, por consiguiente, un aspecto clave para dar soporte al trabajo de investigadores y es propensa a ser un área provechosa para el desarrollo de aplicaciones avanzadas que incorporen técnicas como el "data mining" y los sistemas expertos.

Un ejemplo de la actividad de diseminación está descrito por Hauck Et Al. (2002). Describen un sistema llamado Coplink que provee soporte de tiempo real para investigadores policiales en forma de una herramienta de análisis basada en una gran colección de información de investigaciones previas. Un ejemplo más está descrito por Harrison Et Al. (2002). Su sistema del prototipo no es de tiempo real, pero en lugar de eso provee una función de archivo para la experiencia y el conocimiento de investigadores.

3 MODELO DE CASEY (2004)

Cualquiera de los modelos anteriores es válido para describir el proceso de un análisis forense de evidencias digitales, unos con mayor detalle que otros. Para las intenciones de este proyecto me he basado en el modelo de Casey ya que es el menos abstracto y quizás el más extendido. Como podemos apreciar, con el paso de los años los modelos tienden a tener más etapas para describir el proceso de investigación. El modelo de Casey ha evolucionado desde el primer modelo presentado en el 2002 hasta el modelo publicado en el 2004 en su segunda edición de su libro [7] que recoge los siguientes pasos:

- Autorización y preparación
- Identificación
- Documentación, Adquisición y Conservación
- Extracción de Información y Análisis
- Reconstrucción
- Publicación de conclusiones

1. **Autorización y Preparación:** Lo primero que se debe hacer es ir a la escena del delito a recoger pruebas, pero antes debemos prepararnos con el material y los permisos necesarios para llevarlo a cabo.

2. **Identificación:** Una vez que estamos en la escena del delito debemos identificar todo el hardware y software que encontremos.

3. ****Documentación**:** Esta etapa se realiza durante todo el proceso. Debemos anotar todos los pasos realizados para ayudar a una reconstrucción final de los hechos y con mayor detalle aún si se va a presentar como prueba en un juicio.

4. **Adquisición:** Debemos extraer todo el hardware encontrado que pueda tener pruebas. Generalmente la prueba no es el hardware en sí (huellas digitales, números de serie de CPU), sino el contenido de los mismos. De modos que debemos extraer una imagen de cada dispositivo encontrado.

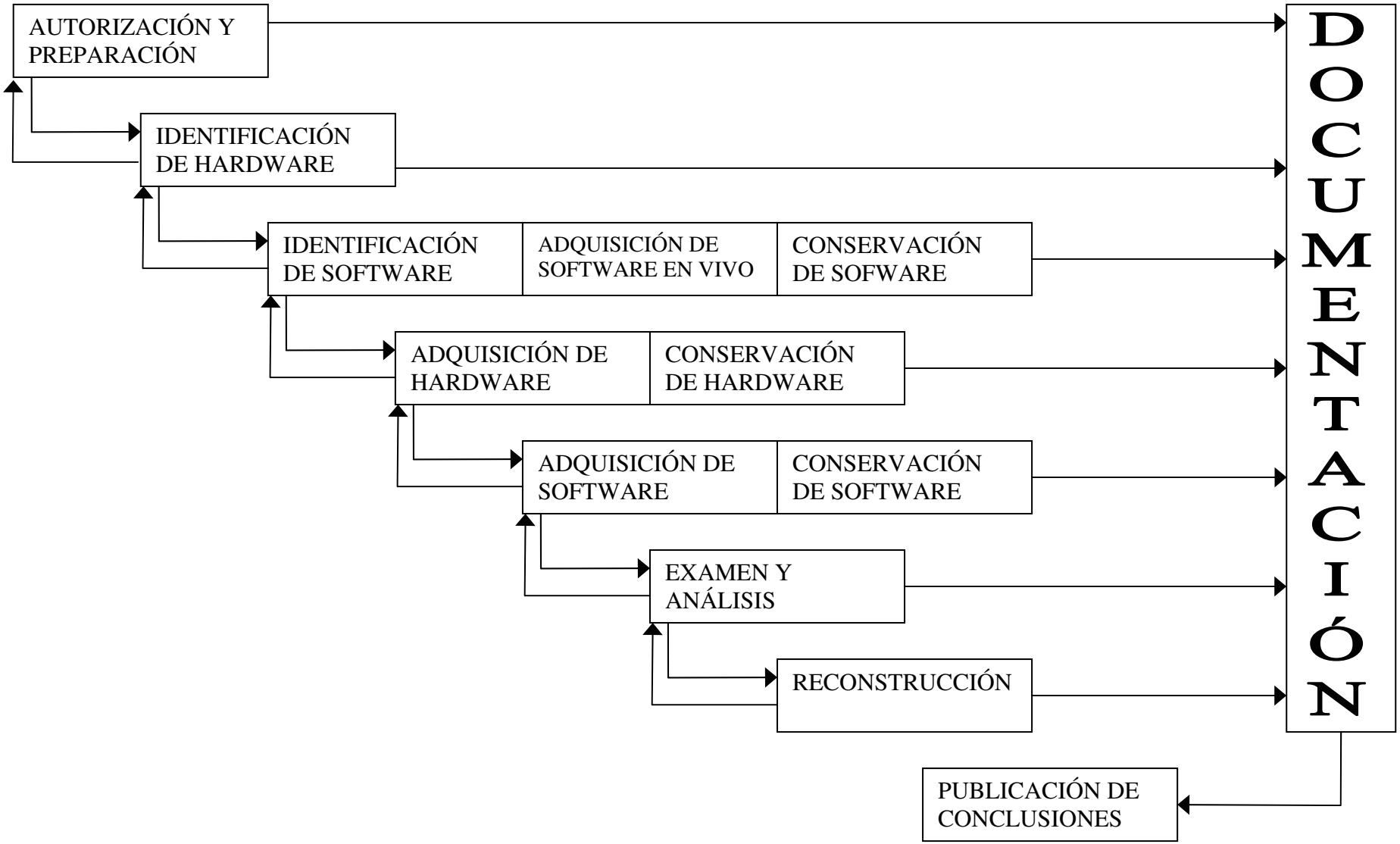
5. **Conservación:** El hardware debe conservarse de forma que no se altere su contenido y es primordial hacer varias copias de la imagen extraída de cada dispositivo y nunca manipular el original.

6. **Examen y Análisis:** Con todos los datos obtenidos en las etapas anteriores podemos tener una idea de donde empezar a buscar, por lo que debemos elaborar una hipótesis y a partir de ella comenzar a recopilar datos que nos ayuden a confirmarla. Existen multitud de métodos para extraer datos de un sistema de ficheros que podemos usar para este fin.

7. **Reconstrucción:** Una vez que tenemos datos suficientes debemos ser capaces de responder a las preguntas ¿que pasó? ¿quien lo hizo?¿cuando?¿donde? y en ultima instancia ¿porque?

8. **Publicación de conclusiones:** Los resultados de los análisis forenses deberían publicarse en la medida de lo posible para incrementar el conocimiento de otros investigadores y en último caso para posibles sistemas expertos que en el futuro puedan ayudar en este campo.

El proceso puede verse como en la siguiente figura: cada flecha indica el flujo de información, de modo que la información que obtenemos en una etapa nos sirve para la siguiente y viceversa. En cualquier momento se puede usar lo que se sabe en una etapa para volver a la etapa anterior y obtener más datos. Toda la información generada se guardará como documentación que nos servirá para la publicación final.


AUTORIZACIÓN Y PREPARACIÓN
IDENTIFICACIÓN DE HARDWARE
IDENTIFICACIÓN DE SOFTWARE
ADQUISICIÓN DE SOFTWARE EN VIVO
CONSERVACIÓN DE SOFWARE
ADQUISICIÓN DE HARDWARE
CONSERVACIÓN DE HARDWARE
ADQUISICIÓN DE SOFTWARE
CONSERVACIÓN DE SOFTWARE
EXAMEN Y ANÁLISIS
RECONSTRUCCIÓN
DOCUMENTACIÓN
PUBLICACIÓN DE CONCLUSIONES

3.1 Autorización y Preparación

Autorización

El objetivo detrás de cualquier investigación realizada por un forense o un equipo de respuesta rápida sobre un sistema de ficheros puede ser de tipo 'legal' o 'casual'. Teniendo en consideración que estos términos no tienen un significado estandarizado para describir los motivos de una investigación y cada uno de ellos se diferencia bastante del otro debemos detallar más.

Investigación Legal: La mayoría de las investigaciones forenses de tipo legal tienen como objetivo asistir a los órganos oficiales a llevar acabo una investigación criminal a fin de llevar ante la justicia al culpable del delito. En investigaciones de este tipo es imprescindible seguir de forma estricta los procedimientos para el tratamiento de pruebas que van a ser presentadas en el juzgado. Por ejemplo, el mero error de sobreescribir cualquier prueba en el sistema de ficheros por información aleatoria (pérdida de datos) es suficiente para considerar el resto de las pruebas de la misma índole como inviables por parte de un juez o fiscal. Investigaciones legales, a menudo, únicamente se limitan a la conservación de datos y esfuerzos de mantener la integridad de información en el sistema de ficheros una vez el hecho del compromiso ha sido probado. Las pruebas tras ser tratadas de forma correcta se transfieren al poder de órganos oficiales para ser analizados por parte de sus recursos. El nivel de participación del forense en la investigación una vez las pruebas han sido transferidas depende del deseo del denunciante y la voluntad de órganos oficiales.

Investigación Casual: Cualquier tipo de investigación casual no tiene como objetivo la persecución legal del individuo responsable del acto criminal. La investigación se realiza por el interés desde el punto de vista forense, por lo tanto las técnicas, herramientas y metodología utilizada puede ser usada de forma más agresiva. La realización de una investigación forense casual requiere más conocimiento y experiencia por parte del investigador, ya que en estos casos no existen requerimientos estrictos de terceros referentes a la cantidad y calidad de pruebas obtenidas.

Antes de manipular una evidencia digital, hay muchas cosas que se deben considerar. Una de ellas es que estemos seguros de que nuestra búsqueda no va a violar ninguna ley o dar lugar a responsabilidades legales.

Los profesionales de la seguridad en computadores deberían obtener instrucciones y autorizaciones escritas de sus abogados antes de realizar cualquier investigación dentro de una organización. Una política de organización determina en gran parte si se pueden buscar en las computadoras de los empleados, analizar los e-mails y otros datos. Sin embargo, una búsqueda justificada normalmente se necesita para acceder a las áreas que un empleado consideraría personales o privadas sin su consentimiento. Hay algunas circunstancias que permiten búsquedas justificadas en un lugar de trabajo, pero los profesionales de la seguridad deben dejar estas decisiones a sus abogados.

Preparación

Antes de empezar un análisis forense se recomienda describir como se va a realizar la recolección de evidencias. Si es posible tener acceso a alguien que esté íntimamente relacionado con la computadora, obtener información general como el tipo de computadora, su sistema operativo, si esta en una red LAN, en Internet, etc. Además puede que necesitemos algunas herramientas como CD's Forenses, contenedores adecuados para transportar el hardware, y otras herramientas como puede ser un destornillador.

3.2 Documentación

La documentación es esencial en todas las fases del manejo y procesamiento de evidencia digital. Documentando quien adquiere y maneja evidencias en un momento dado es algo imprescindible para mantener la **Cadena de Custodia**. Esto no es algo inusual para alguien que maneja una evidencia para posteriormente presentar las conclusiones ante un juicio.

La continuidad de la posesión o Cadena de Custodia debe ser establecida para que la evidencia sea admitida como válida, aunque frecuentemente todas las personas involucradas en la adquisición, transporte y almacenamiento de evidencias son llamados para testificar en un juicio. De modo que, para evitar confusiones y mantener el control completo de la evidencia en cada momento, la Cadena de Custodia debería estar obligada a cumplir un mínimo.

Así que, debería anotarse cuidadosamente cuando se adquiere la evidencia, de donde y por quien. Por ejemplo, si la evidencia se copia en un disquete, deberíamos anotar en la etiqueta del mismo y en la cadena de custodia la fecha y hora actuales, las iniciales de la persona que hizo la copia, como hizo la copia y la información relativa al contenido del disquete. Adicionalmente, los valores MD5 o SHA de los archivos originales deberían ser notados antes de copiarse.

A continuación podemos ver un ejemplo de una Cadena de Custodia con información mínima para un disco duro cuyo número de serie es el 123456.

Chain of Custody Log

Line	Item	Date	Time	Who	Description
1	Hard disk drive, ser #123456	7/15/04	10:15 AM	M. SOLOMON	Seized hard drive from scene, permission provided by business owner
2	Hard disk drive, ser #123456	7/15/04	10:45 AM	M. SOLOMON	Transported HDD to evidence locker in main office
3	Hard disk drive, ser #123456	7/16/04	7:30 AM	M. SOLOMON	Removed HDD to create analysis copy
4	Hard disk drive, ser #123456	7/16/04	9:15 AM	M. SOLOMON	Returned HDD to evidence locker
5					

Si la prueba es pobremente documentada, entonces un abogado puede arrojar dudas más fácilmente sobre las habilidades de los interesados y puede convencer al tribunal de no aceptar la evidencia.

La documentación que muestra que la evidencia se encuentra en su estado original se usa regularmente para demostrar que es auténtica y que está inalterada.

Documentar la posición original de prueba también puede ser útil al tratar de reconstruir un delito. Cuando existen varias computadoras implicadas, asignando letras a cada posición y números para cada fuente de prueba digital ayudarán a seguir la pista a los ítems. Además, los investigadores digitales pueden estar obligados a brindar testimonio años más tarde o, en el caso de muerte o enfermedad, un investigador digital puede ser incapaz brindando testimonio. Entonces, la documentación debería proveer todo lo que alguien más necesitará muchos años más tarde para entender la evidencia. Finalmente, al examinar prueba, se requieren notas detalladas para posibilitar a otro investigador competente a evaluar o reproducir lo que estaba hecho e interpretar los datos.

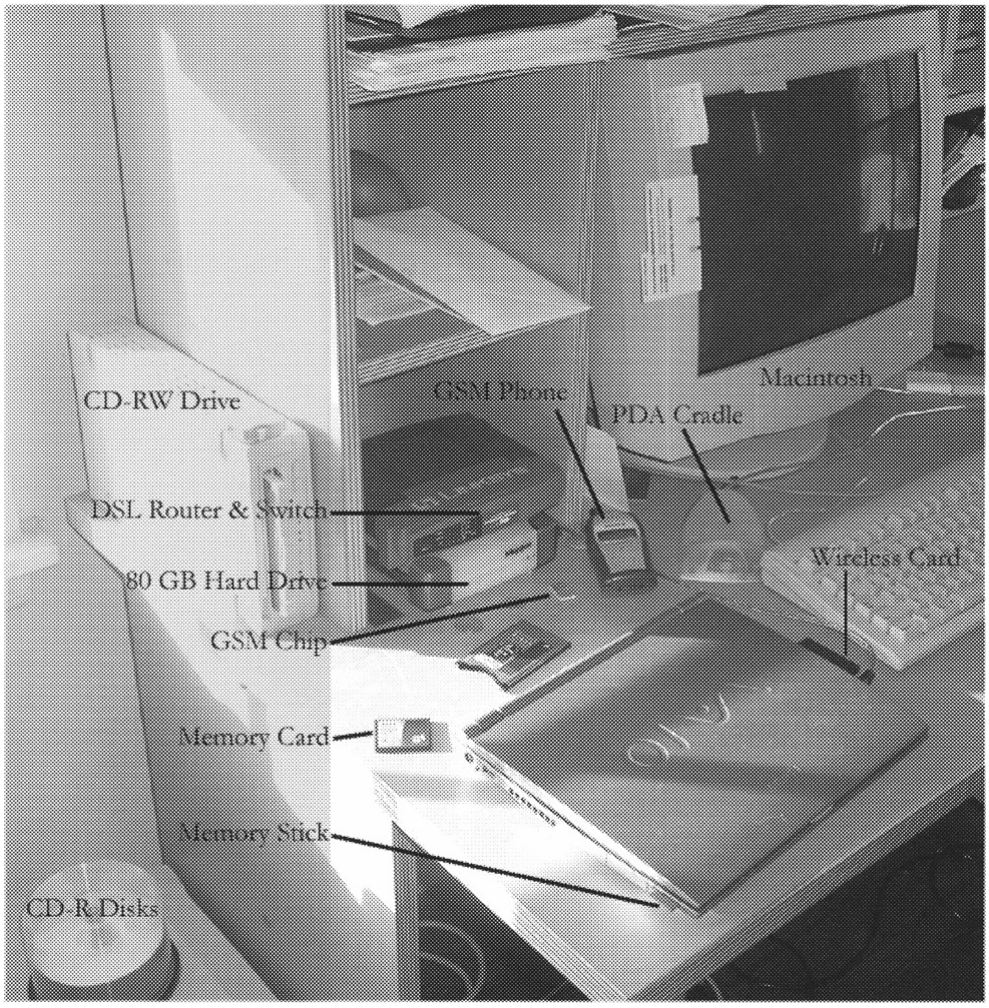
3.3 Identificación

La identificación de las evidencias digitales es un proceso con dos pasos:

- Primero, el investigador debe reconocer el hardware (por ejemplo, ordenadores, disquetes o cables de red) que contienen información digital.
- Segundo, el investigador debe distinguir entre la información relevante y otros datos intrascendentes según lo que estemos buscando.

3.3.1 Identificación de Hardware

Hay muchos productos computerizados que pueden tener evidencias recogidos en [3], como telefonos, dispositivos inalámbricos, PDAs, Routers, Firewalls y otros dispositivos de red. Hay muchas formas de almacenar datos multimedia, como disquetes, cds, cintas magnéticas, pen drives, memory cards, etc.



Una selección de hardware.

Vamos a ver a continuación los diferentes componentes que podemos encontrarnos en una escena del delito y las evidencias que se pueden extraer de ellos:

Dentro de un PC:

Unidad Central de Procesamiento(UCP o CPU)

Descripción: A menudo llamadas el "chip", es un microprocesador alojado dentro de la torre de la computadora, en un circuito electrónico junto con otros componentes. Esta puede ser o no extraíble.

Usos principales: Realiza todas las funciones aritméticas y lógicas en la computadora. Controla el funcionamiento de la computadora.

Evidencia Potencial: El dispositivo en sí mismo pueden ser una prueba de robo, falsificación, o remarcado de numero de serie.

Memoria

Descripción: Circuito electrónico extraíble situado en el interior de la computadora. La información que se almacena aquí normalmente no se mantiene cuando se apaga la computadora.

Usos principales: Almacena los datos y programas del usuario cuanto la computadora está en ejecución.

Evidencia Potencial: Igual que el anterior.

Discos duros

Descripción: Una caja precintada que contiene discos recubiertos con una sustancia capaz de almacenar datos magnéticamente. Puede encontrarse internamente dentro de un PC o como un disco externo. Su contenido puede estar sin formatear o formateado, de forma que estaría organizado en un sistema de ficheros.

Usos primarios: Almacenamiento de información como programas, texto, imágenes, video, multimedia, etc.

Evidencia Potencial: Todas las que se puedan encontrar en un sistema de ficheros (véase el apartado de identificación de software)

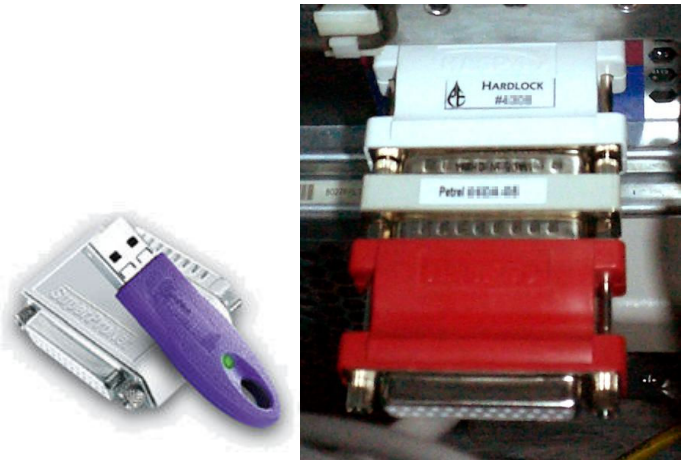
Dispositivos de Control de Acceso

Descripción:

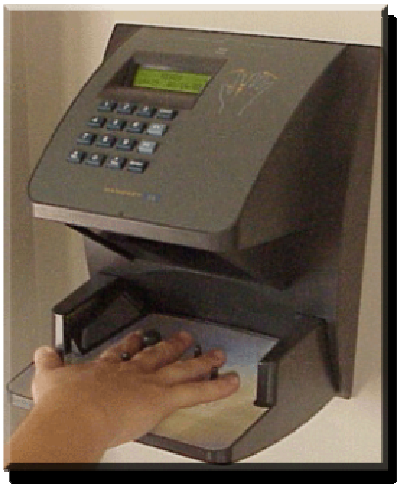
- Una tarjeta inteligente (smart card), o tarjeta con circuito integrado (TCI), es cualquier tarjeta del tamaño de un bolsillo con circuitos integrados incluidos. Aunque existe un diverso rango de aplicaciones, hay dos categorías principales de TCI. Las Tarjetas de memoria contienen sólo componentes de memoria no volátil y posiblemente alguna lógica de seguridad. Las Tarjetas microprocesadoras contienen memoria y microprocesadores.

La percepción estándar de una "tarjeta inteligente" es una tarjeta microprocesadora de las dimensiones de una tarjeta de crédito (o más pequeña, como por ejemplo, tarjetas SIM o GSM) con varias propiedades especiales (ej. un procesador criptográfico seguro, sistema de archivos seguro, características legibles por humanos) y es capaz de poveer servicios de seguridad (ej. confidencialidad de la información en la memoria).

- Un “dongle” es un pequeño dispositivo que se conecta a un puerto (puerto paralelo, USB, etc) del ordenador, y gracias a él se verifica que un programa es original y no una copia. Cuando no está conectado al PC, el software funciona en modo restringido o simplemente no funciona.



- Un escáner biométrico es un dispositivo conectado a un sistema computacional que reconoce características físicas de un individuo (por ej., huellas digitales, la voz o la retina).



Usos principales: Permite el acceso al control de computadoras, programas o a funciones, funcionando como una clave de encriptación.

Evidencia potencial: Información de identificación/autenticación de los usuarios, nivel de acceso, configuraciones, permisos y el dispositivo en sí mismo.

Contestadores automáticos

Descripción: Un dispositivo electrónico que es parte de un teléfono o está conectado entre un teléfono y la conexión a la red. Algunos modelos usan una cinta magnética, mientras que otros usan un sistema de grabación electrónico (digital).

Usos principales: Almacena mensajes de voz de la persona que llama cuando la parte llamada no contesta a la llamada. Normalmente muestra un mensaje de voz de la parte llamada antes de grabar el mensaje.

Evidencia Potencial: Los contestadores automáticos pueden almacenar mensajes de voz y, en algunos casos, información sobre fechas y horas sobre cuando se dejó el mensaje. También pueden contener otras cosas almacenadas:

- -Información de identificación de la persona que llama.
- -Mensajes borrados.
- -El último número llamado
- -Notas recordatorias.
- -Nombres y números de teléfono.
- -Cintas.

Cámaras Digitales

Descripción: Dispositivo de grabación digital para imágnees y video, con un dispositivo de almacenamiento en su interior y hardware de conversión que permite transferir los datos a la computadora.

Usos principales: Captura imágenes y/o video en un formato digital que es fácilmente transferible a una computadoras para visualizar o editar.

Evidencia Potencial

- - Imagenes.
- - Sellos de fecha y hora.
- - Carretes/Tarjetas de memoria.
- - Video.
- - Sonido.

Dispositivos Portátiles (Asistentes Digitales Personales) [PDAs], Agendas Electrónicos)

Descripción: Un asistente digital personal (PDA) es un computador de mano originalmente diseñado como agenda electrónica. Hoy en día se puede usar como una computadora doméstica (ver películas, crear documentos, navegar por Internet...).

Usos Primarios: Computación de mano, almacenamiento y comunicación.

Evidencia Potencial

- - Libreta de direcciones.
- - Información sobre citas
- - Documentos.
- - E-mails.
- - Escritura a mano.
- - Passwords.
- - Libreta de teléfonos.
- - Mensajes de texto.
- - Mensajes de voz.

Tarjetas de Memoria

Descripción: Dispositivos electrónicos de almacenamiento extraíbles, que no pierden la información cuando no se suministra con corriente de la tarjeta. Estas tarjetas suelen tener una memoria de tipo flash, aunque en algunos casos, como en las compactFlash, se le puede incluir un minidisco duro, que aunque almacena más información, es más sensible a los golpes y consume más energía. Se usan en una variedad de dispositivos como cámaras digitales, MP3s, PDAs, ordenadores, etc.

Algunos ejemplos son:

- CompactFlash (**CF**) I y II

- Memory Stick (**MS**)

- MicroSD

- MiniSD

- Multi Media Card (**MMC**)

- Secure Digital (**SD**)

- SmartMedia Card (**SM/SMC**)

- xD-Picture Card

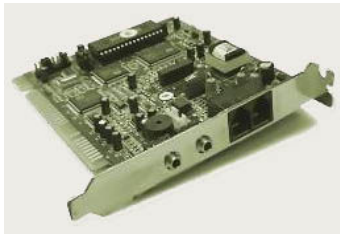
Usos principales: Proporciona métodos adicionales extraíbles para el almacenamiento y transpote de información.

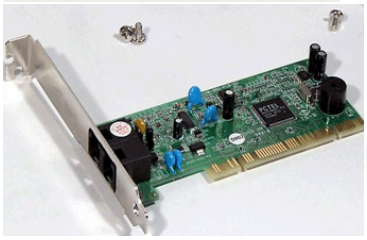
Evidencia Potencial: Todas las que se puedan encontrar en un sistema de ficheros (véase el apartado de identificación de software)

Componentes de redes de ordenadores

Módems

Descripción: Módems, internos y externos (analógicos, DSL, ISDN, cable), inalámbricos, etc.




Usos principales: Un módem se usa para facilitar la comunicación electrónica, permitiendo a la computadora acceder a otras computadoras y/o redes via línea telefónica, inalámbrica u otro medio de comunicación.

Evidencia Potencial: El módem en sí mismo.

Tarjetas de Red de Área Local (LAN) o Tarjetas de Interfaz de Red (NIC)

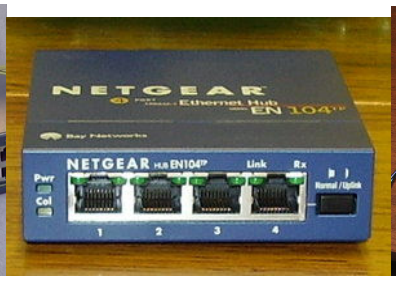
Descripción: Tarjetas de red y cables asociados. Estas tarjetas también pueden ser inalámbricas.

Usos principales: Una tarjeta LAN/NIC se usa para conectar computadoras. Las tarjetas permiten el intercambio de información y de recursos.

Evidencia Potencial: El dispositivo en si mismo, dirección MAC(Media Access Control).

Routers, Hubs, y Switches

Descripción: Estos dispositivos electrónicos se usan en sistemas de redes de computadoras. Los routers, switches y hubs proporcionan un medio para conectar diferentes sistemas de computadoras o redes.

Usos principales: Equipamiento usado para distribuir y facilitar la distribución de datos a través de redes de computadoras.

Evidencia Potencial: Los dispositivos en sí mismos. Además, para los routers, sus ficheros de configuración.

Servidores

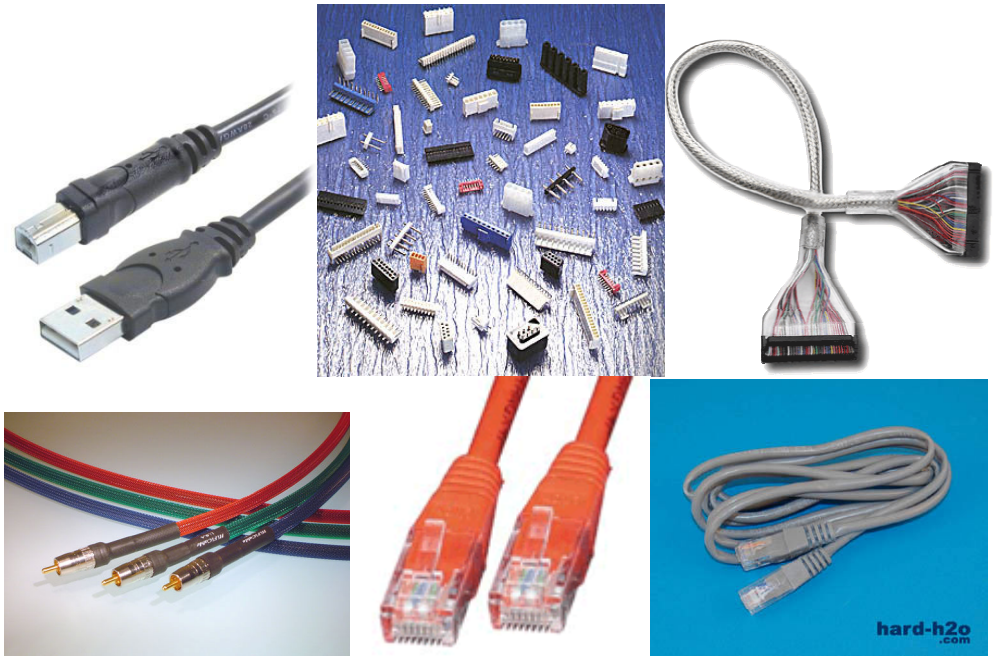
Descripción: Un servidor es una computadora que proporciona algunos servicios a otras computadoras conectadas a él vía red. Cualquier computadora, incluyendo un ordenador portátil puede configurarse para ser un servidor.

Usos primarios: Proporciona recursos compartidos como e-mail, almacenamiento de ficheros, servicios Web y servicios de impresión para una red de ordenadores.

Evidencia Potencial: Todas las que se puedan encontrar en un sistema de ficheros (véase el apartado de identificación de software)

Cables y conectores de Red

Descripción: Los cables de red pueden tener diferentes colores, grosores y formas, dependiendo de los componentes a los que esté conectado.



Usos principales: Conecta componentes de una red de computadoras.

Potential Evidence: La evidencia son ellos mismos.

Dispositivos Buscapersonas (Buscas)

Descripción: Un dispositivo electrónico de mano, que puede contener evidencias volátiles (números de teléfono, correo de voz, e-mail, etc.). Los PDAs y los móviles también pueden usarse como dispositivos buscapersonas.

Usos principales: Para enviar y recibir mensajes electrónicos, numéricos (números de teléfono, etc.), y alfanuméricos (texto, a menudo incluyendo e-mail)

Evidencia Potencial:

- - Información de direcciones
- - Mensajes de texto
- - E-mail.
- - Mensajes de voz
- - Números de teléfono

Impresoras

Descripción: Un sistema de impresión térmico, de láser, de tinta o de impacto, conectado a la computadora via cable (serie, paralelo, USB, firewire, etc.) o por puerto infrarrojos. Algunas impresoras contienen un buffer de memoria, permitiéndoles recibir y almacenar múltiples páginas de documentos mientras imprimen. Algunos modelos incluyen un disco duro.

Usos principales: Imprimir texto, imágenes, etc., de la computadora al papel.

Evidencia Potencial: Las impresoras pueden mantener logs de su uso, información de fechas y horas, y además, si esta conectada a una computadora, puede almacenar información de su identidad de red.

- - Documentos.
- - Disco duro.
- - Cartuchos de tinta.
- - Información/identidad de red.

- - Imágenes sobreimpresas en el rodillo
- - Sellos de fecha y hora
- - Log del uso del usuario

Dispositivos Removibles de Almacenamiento y Multimedia

Descripción: Soporte usado para almacenar información eléctrica, magnética o digital. (por ej., disquetes, CDs, DVDs, cartucho, cinta).

Usos principales: Dispositivos portátiles que pueden almacenar programas, texto, fotos, video, archivos multimedia, etc.

Evidencia Potencial: Todas las que se puedan encontrar en un sistema de ficheros (véase el apartado de identificación de software).

Escáners

Descripción: Un periférico que se utiliza para convertir, mediante el uso de la luz, imágenes impresas a formato digital.

Hay varios tipos. Hoy en día los más extendidos son los planos.

Tipos:

- *De rodillo:* Como el escáner de un fax

- *Planos:* Como el de las fotocopiadoras.

- *De mano:* En su momento muy económicos, pero de muy baja calidad. Prácticamente extintos.

Hoy en día es común incluir en el mismo aparato la impresora y el escáner. Son las llamadas impresoras multifunción.

Usos principales: Convierte documentos, imágnees a ficheros electrónicos que pueden verse, manipularse o transmitirse en una computadora.

Evidencia Potencial: El dispositivo en sí mismo puede ser una evidencia. La posibilidad de poder escanear podría ser una ayuda para actividades ilegales (pornografía infantil, fraude de cheques, falsificación y robo de identidad). Además, existen imperfecciones como las marcas en el cristal que pueden permitir realizar una identificación única de un escaner usado para procesar ciertos documentos.

Teléfonos

Descripción: Un dispositivo de telecomunicación diseñado para transmitir conversación por medio de señales eléctricas. Puede ser individual (teléfono móvil) o dependiente de una estación base remota (inalámbricos), o conectados directamente a la toma de teléfono (fijo).






Usos principales: Comunicación bidireccional desde un dispositivo a otro usando lineas terrestres, transmisión de radio, sistemas celulares o una combinación. Los teléfonos son capaces de almacenar información.

Evidencia Potencial: Muchos teléfonos pueden almacenar nombres, números de teléfono, e información de identificación de las personas que llaman. Además, algunos teléfonos celulares pueden almacenar informaciones sobre citas, e-mail y registros de voz.

- - Calendario/información sobre citas.
- - Passwords.
- - Información sobre personas de contacto
- - Libreta telefónica
- - Número de serie electrónico
- - Mensajes de texto
- - E-mails.
- - Mensajes de voz.
- - Notas recordatorias
- - Navegadores web

Instrumentos electrónicos varios

Hay muchos tipos adicionales de equipamiento electrónico que son muy numerosos para ser listados y que podrían encontrarse en una escena del delito. Sin embargo, hay muchos dispositivos no tradicionales que pueden ser una excelente fuente de información investigativa y/o evidencia. Ejemplos de estos son: equipamiento de clonación de móviles, faxes, fotocopiadoras, grabadoras de audio, grabadores de bandas magnéticas de tarjetas de crédito, etc.

Fotocopiadoras



Algunas fotocopiadoras mantienen registros de acceso de usuario e históricos de las copias hechas.

Evidencia Potencial:

- - Documentos.
- - Logs de acceso de usuarios.
- - Sellos de fecha y hora.

Copiadores de bandas magnéticas



Los Credit card skimmers se usan para leer la información de la banda magnética de tarjetas de plástico.

Evidencia Potencial: La información contenida en la banda magnética incluye:

- - Fecha de caducidad de la tarjeta
- - Dirección del usuario.
- - Número de tarjeta de crédito

- - Nombre de usuario.

Máquinas de FAX

Los faxes pueden almacenar números de teléfono programados y un histórico de documentos transmitidos y recibidos. Además, algunos contienen memoria que permite faxes multipágina. Algunos pueden contener cientos de páginas de faxes entrantes y/o salientes.

Evidencia Potential:

- - Documentos.
- - Números de teléfono.
- - Carrete.
- - Logs de faxes enviados/recibidos

Global Positioning Systems (GPS)

Los GPSs pueden proporcionar información sobre viajes previos mediante información del destino, puntos intermedios y rutas. Algunos almacenan automáticamente los destinos previos e incluyen logs de viajes.

Evidencia Potencial:

- - Dirección de la casa
- - Coordenadas de puntos intermedios.
- - Destinos previos.
- - Nombres de lugares.
- - Logs de viajes

3.3.2 Identificación del software

Generalmente se considera que todo el contenido del hardware identificado contiene potencialmete evidencia digital. Por esto, una vez que se ha retirado el hardware para su análisis en el laboratorio, se debe extraer su contenido. Por esto no podemos identificar las evidencias digitales hasta que no hayamos adquirido el hardware y extraído el software que contiene. Pero existen algunos casos en los que se pueden identificar evidencias digitales en el lugar del delito. Es lo que llamamos una adquisición de datos en vivo, que se realiza cuando el sistema se encuentra encendido o no se puede apagar por diversas razones, como que se trate de un sistema crítico (sistemas informáticos de los hospitales).

En este punto debemos decir que hay dos tipos de datos:

- Datos volátiles: son datos que se pierden si el sistema es apagado. Ejemplos de los mismos puede ser una lista de procesos en ejecución y usuarios activos.
- Datos No Volátiles: son datos que no se pierden cuando apaga el sistema e incluyen el disco duro.

Por tanto sabemos que **si apagamos un sistema encendido perderemos los datos volátiles** y en algunos casos puede ser muy interesante obtenerlos. Para determinar que evidencia recoger primero debemos seguir el **Orden de Volatilidad**: una lista de fuentes de evidencias ordenadas por su volatilidad relativa.

En general puede verse de la siguiente manera:

Registros, memoria de periféricos, cachés, etc.	nanosegundos
Memoria Principal	Nanosegundos

Estado de la Red	Milisegundos
Procesos en ejecución	segundos
Disco	minutos
Disquetes, copias de seguridad, Discos duros, etc.	Años
CD-ROMs, DVDs, etc.	Decenas de años

Un ejemplo de orden de volatilidad podría ser:

1. Registros y cache
2. Tablas de enrutamiento
3. Cache Arp
4. Tabla de procesos en ejecución
5. Estadísticas y módulos Kernel
6. Memoria principal
7. Ficheros temporales del sistema
8. Memoria secundaria
9. Configuración del Router
10. Topología de red

Una vez que hemos obtenido la información volátil debemos pensar en apagar el sistema. Una de las decisiones más difíciles al encontrarse con una computadora sospechosa que esta encendida es cómo apagar el sistema de manera que no se corrompa la integridad de los archivos. En la mayoría de los casos, el tipo de sistema operativo empleado en la computadora será la clave a la hora de tomar esta decisión. Con unos, bastará con tirar del enchufe del ordenador, y en otros, desconectando el PC sin permitir al sistema operativo iniciar sus comando internos de apagado podría resultar desde la pérdida de archivos vitales hasta la rotura del disco duro.

El problema es que si usamos cualquier comando o funcionalidad del sistema para apagar el sistema, corremos el riesgo de que se ejecute código malicioso o de que se modifiquen los logs del sistema. Por ejemplo, los comandos “shutdown” o “sync” podrían haber sido modificados de forma que cuando los ejecutemos el sistema borre ficheros críticos. Por lo tanto es preferible usar nuestros propios ejecutables de forma externa.

El riesgo típico de tirar del cable de la pared es que el sistema estará en un estado inconsistente, y cuando el sistema se encienda de nuevo iniciará un proceso intensivo de reconstrucción.

Generalmente parece que hay una regla aceptada que dice **"si esta encendido, no lo apagues, y si esta apagado, no lo enciendas"**. En caso de que esté encendido lo más común es simplemente fotografiar la pantalla y tirar del cable de la pared. Deberemos anotar que se tiró del cable para tener en cuenta más tarde que el SSOO puede estar en un estado inconsistente. Esto es útil saberlo sobre todo si más tarde se decide arrancar el sistema de nuevo en un entorno seguro.

En el caso de que no se pueda apagar el sistema por ser crítico su funcionamiento, se debe hacer un análisis mínimamente intrusivo intentando recopilar la mayor cantidad de datos posibles relacionados con la investigación. Esto puede verse con mayor detalle en la sección de "Examen y Análisis", donde se pueden ver los archivos que son interesantes según el tipo de delito.

3.4 Adquisición

Una vez identificadas, las evidencias deben ser recogidas y conservadas de modo que puedan ser identificadas después con facilidad. Una buena forma de hacer esta recogida es de forma que no se alteren. Imagínese por un momento una supuesta escena del crimen donde haya una nota suicida escrita en la pantalla. Antes de examinar el contenido digital de la computadora se debería antes fotografiar la pantalla y tomar huellas digitales.

Pero aquí nos topamos con otro problema: ¿que hacemos cuando nos encontramos una computadora encendida? ¿La apagamos directamente? Si manipulamos la computadora en busca de datos podemos alterar la evidencia. Por ejemplo, si encontramos una computadora con un sistema Linux y probamos a hacer un 'ls' para ver el listado actual de un directorio, modificaremos los registros de actividad, el contenido de la memoria ram, etc.

3.4.1 Adquisición del hardware

Aunque este apartado se base en los datos almacenados en las computadoras, vamos a hacer una mención al hardware para asegurarnos de que la evidencia que contiene se conserva correctamente.

Hay dos factores que se deben considerar al recolectar el hardware. En un lado, para no dejar ninguna evidencia atrás, un investigador puede decidir que hay que recoger todas las piezas que se encuentren. Por otro lado, un investigador puede recoger solo lo esencial para ahorran tiempo, esfuerzo y recursos. Algunas computadoras de instituciones en continuo funcionamiento, como hospitales, el hecho de modificar algo puede costar vidas humanas. En algunos casos, simplemente no es factible recoger el hardware por su tamaño o cantidad.

¿Que hacer si una computadora esta conectada a otra? En una era en la que las redes de computadoras son lo habitual, sería absurdo pensar que podemos obtener todas las computadoras que están conectadas a una dada. En una red de área local situada en un piso, edificio o en un campus universitario, un PC puede estar conectado a cientos de computadoras. Este PC puede además estar conectado a internet, por lo que deberíamos tomar muchas computadoras en todo el mundo. En definitiva, esta elección se debe tomar en función del número de pruebas que necesitemos y los recursos para almacenarlas que dispongamos.

Si se decide recoger una computadora entera, deberían considerarse todos sus periféricos, como impresoras o unidades de cinta. Las hojas impresas que estén relacionadas con la computadora pueden contener información que ha sido cambiada o borrada de la computadora, como números de teléfono, direcciones de e-mail, etc. Además se recomienda mirar en la basura en busca de evidencias. Un conocido investigador forense una vez bromeó acerca de que cuando él llegaba a su casa y su

familia estaba ya durmiendo en la cama, no despertaba a su esposa para preguntarle lo que había hecho durante el día, sino que simplemente examinaba la basura.

3.4.2 Adquisición del software

Cuando se trata con evidencias digitales, lo principal es el contenido de la computadora más que el hardware en sí. En este apartado veremos como se adquieren estos contenidos de forma que no se altere la información que contienen y podamos estar seguros de que tenemos una copia exacta.

Hay dos tipos de adquisición de datos: en vivo o post-mortem. La diferencia está basada en el sistema operativo usado durante la copia:

- Una adquisición **en vivo** ocurre cuando los datos son copiados desde un sistema sospechoso usando el sistema operativo sospechoso. Esta se hace normalmente antes de la adquisición de hardware y fue mencionada previamente en el apartado de Identificación de Software.

- Una adquisición **post-mortem** se realiza cuando el dispositivo a analizar no está en ejecución y por tanto los datos son copiados posteriormente en un entorno controlado. Esto ocurre cuando el disco es extraído del sistema sospechoso y ubicado en un sistema controlado, y también cuando el sistema sospechoso es arrancado con un dispositivo autoarrancable, por ejemplo, un CD-Rom.

En general son preferibles las adquisiciones post-mortem sobre las adquisiciones en vivo, ya que no hay peligro de que el sistema operativo nos de información falsa. Algunos casos requieren una adquisición en vivo, por ejemplo:

- Cuando el sistema es un servidor crítico que no puede apagarse por los daños que ocasionaría.
- Cuando los datos necesitan ser adquiridos pero un apagado podría alertar a un atacante de que el sistema ha sido identificado (en el caso de Hackers).
- Cuando los datos se perderán cuando se desconecte de la electricidad. Ejemplos incluyen la memoria y volúmenes encriptados que son montados y la clave es desconocida.

Hay dos opciones cuando se recoge una evidencia digital de una computadora: copiar solamente la información que se necesita o copiarlo todo. Si se va a hacer un examen rápido o si solo una pequeña parte de la evidencia es de interés (por ejemplo, un fichero log), es más práctico buscar inmediatamente en la computadora y obtener la

información requerida. Sin embargo, si hay abundancia de evidencias en la computadora, es recomendable copiar el contenido entero y examinarlo cuidadosamente a posteriori.

La ventaja de tomar solamente lo que se necesite es que resulta más barato, rápido y menos caro que copiar contenidos enteros. En algunos casos es suficiente solamente con tomar los ficheros de actividad y los datos no borrados, en cuyo caso un backup del sistema sería suficiente.

Hay además un riesgo de que el sistema haya sido modificado con el fin de ocultar o destruir las evidencias (por ejemplo, usando un ***rootkit***). Por ejemplo, si un investigador busca ficheros log en una computadora, puede haber ficheros logs borrados en el espacio libre que pueden serle útiles. Cuando se toman solo unos pocos ficheros es necesario documentar el proceso meticulosamente y registrar los ficheros en su estado original. Por ejemplo, obteniendo un listado completo de los ficheros con sus características asociadas como los nombres de ruta, ***sellos de tiempo***, tamaños y valores MD5.

Dados los riesgos y el esfuerzo de tomar solo unos pocos ficheros, en la mayoría de los casos es recomendable adquirir el contenido completo de un disco ya que un investigador raramente sabe a priori lo que contiene una computadora. Antes de copiar datos de un disco es recomendable calcular el valor MD5 del disco original para compararlo luego con sus copias y así demostrar que son idénticas.

Cuando se toma el contenido completo de una computadora se debe hacer una copia bit a bit, en lo que llamaremos una imagen forense, es decir, una copia exacta. Una imagen forense duplica todo lo que contenga un cluster de disco, incluyendo el "slack space" y otras areas de la superficie del disco, mientras que con otros métodos de copia de ficheros solamente se duplica el fichero y se deja el slack space atrás.

En cualquier dato, la evidencia digital se pierde si no se realiza una copia bit a bit. Por supuesto, esto solo nos concierne si el slack space puede contener información importante. Si solamente necesitamos la información que contiene un fichero y no se requiere el slack space, una copia del fichero sería suficiente.

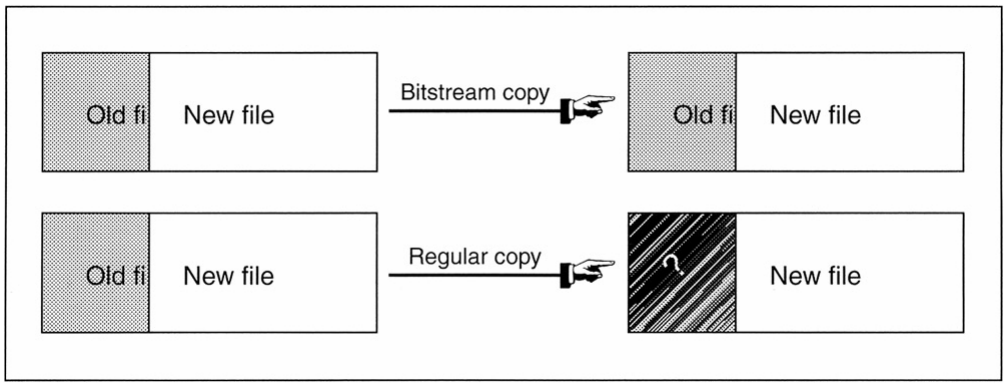


Figura: Comparación entre una copia normal y una bit a bit.

La mayoría de las herramientas pueden interpretar copias bit a bit creadas usando la herramienta EnCase (que se verá más adelante) y con el comando UNIX 'dd', haciéndolos a ambos los estándares 'de facto'. "Safeback" es otro formato de ficheros comun pero solamente se usa en la policía. EnCase y Safeback incluyen información adicional en sus ficheros para comprobar su integridad. En todo caso hay una ley empírica en la recolección de evidencias digitales que siempre se debería recordar:

Ley empírica de Recolección de Evidencias y Conservación: Si solamente haces una copia de la evidencia digital, esa evidencia será dañada o se perderá completamente.

De modo que siempre se deben hacer dos o más copias de la evidencia digital y comprobar que al menos una de las copias se realiza correctamente y puede ser accedida desde otra computadora.

Es muy importante guardar la copia digital en discos completamente limpios. Si se guarda en un disco que ya tenía algunos datos (por ejemplo, un disco duro usado), los datos antiguos pueden quedar en el slack space, contaminando así la evidencia. De modo que es una buena práctica usar un programa que escriba un patrón determinado en el disco (por ejemplo, 00000000) y verifique que este patrón se escribió en todos los sectores.

Como regla general, una computadora que se use para almacenar y analizar evidencias digitales no debería estar conectado a Internet. Existe el riesgo de que alguien gane acceso no autorizado a la evidencia.

Cuando se quiere extraer una imagen de una computadora se debe hacer con la mínima alteración posible para la misma. Una forma de hacerlo es introduciendo un disco boot preparado con las herramientas para extraer la imagen y arrancar la máquina con él. En algunos casos no es posible o deseable arrancar la máquina sospechosa, por lo que la mejor alternativa es quitar el/los disco/s duro/s de la computadora y ubicarlo en otra más segura, o insertarlo en un sistema especial de recolección de evidencias para su procesamiento. Los dispositivos de duplicación de Hardware como los fabricados por Intelligent Computer Solutions y Logicube son útiles para copiar datos de una unidad IDE o SCSI en otra.

Recuerde que a menudo es posible preguntar al dueño o administrador del sistema por ejemplo, si los datos estan encriptados. Estas personas deberían poder facilitarnos el acceso a ellos.

Técnicas de adquisición de datos

Hay tres técnicas principales que podemos usar para copiar datos del sistema sospechoso a un sistema seguro:

- La primera técnica usa la red para copiar datos desde el sistema sospechoso a un servidor seguro. Esta técnica puede usarse para adquisiciones en vivo o post-

mortem. La forma más fácil para llevarlo a cabo es usando la herramienta "netcat", que puede actuar como cliente y como servidor. El servidor seguro ejecuta netcat con la opción "-l" para poner netcat en modo escucha. Un número de puerto es asignado usando la opción "-p". Esto provoca que netcat escuche del puerto especificado y copie cualquier dato recibido en la pantalla. Úsese la redirección para guardar los datos en un fichero. Por ejemplo, para escuchar del puerto 9000 y guardar a fichero.out, usaríamos:

nc –l –p 9000 > fichero.out

Un método alternativo de copiar datos sobre la red es montar una unidad de red. Usando Samba o NFS, una unidad en el servidor seguro puede ser montada y los datos copiados en ella.

- Las otras dos técnicas de adquisición requieren una extracción física de las unidades. En un caso, el disco es extraido desde el sistema sospechoso y ubicado en un sistema seguro. En el otro, un nuevo disco es ubicado en el sistema sospechoso y es arrancado desde un CD-Rom seguro y el disco sospechoso es copiado en forma de imagen al disco nuevo. Esto es útil si hay hardware especial como un disco RAID (Redundant Arrays of Inexpensive Disks).

Guía Básica

Hay algunas ideas clave que deben ser aplicadas en cualquier técnica de adquisición de datos. El objetivo es guardar el estado del sistema de modo que pueda ser analizado en un laboratorio. Cada situación será diferente y requerirá diferentes técnicas. Cuando usemos una herramienta deberemos tener en cuenta que siga las siguientes pautas:

- Minimizar la escritura en el disco del sistema sospechoso: si se nos da la ocasión en que tengamos que elegir entre dos técnicas o herramientas, elegiremos la que escriba el mínimo de datos en el disco sospechoso. Utilizar herramientas que envían la salida a la salida estándar y canalizan los datos sobre la red a un servidor. Cualquier dato escrito en el sistema sospecho podría estar sobre una evidencia.

- No fiarse de nada del sistema sospechoso: Si alguien tuvo permisos de administrador en un sistema pudo modificar desde el núcleo hasta los ejecutables. Deberemos usar nuestras propias versiones certificadas de los ejecutables cuando sea posible.

- No instalar herramientas de adquisición de datos en el sistema: Úsese mejor un CD-Rom de herramientas certificadas cuando sea posible.

- Mantener el Orden de volatilidad: El OOV (Order Of Volatility) fué documentado por Dan Farmer y Wietse Venema y se compone de las ubicaciones donde se almacenan los datos, ordenadas por la frecuencia en la que

cambian sus valores. Podemos querer adquirir el que cambie primero más rápido, aunque algunos cambian demasiado rápido y pueden no sernos útiles (por ejemplo, los registros). La siguiente lista muestra el orden de volatilidad comenzando por el más volátil:
    - Conexiones de Red y Ficheros Abiertos
    - Procesos
    - Usuarios activos
    - Disco duro

- Calcular un valor hash fuerte para los datos adquiridos: Ejemplos de hashes fuertes son MD5, SHA-1 y SHA-2.

- Documentar qué herramientas ejecutamos y las modificaciones que hacemos en el sistema: Podemos usar el comando "script" en UNIX para esto.

3.5 Examen y Análisis

3.5.1 Filtrado/Reducción de los datos para análisis

Antes de profundizar en los detalles del análisis de una evidencia digital, es necesaria una breve discusión sobre la reducción de los datos a analizar. Con el decremento del coste del almacenamiento de datos y el incremento del volumen de ficheros comerciales en sistemas operativos y aplicaciones software, los investigadores digitales pueden sentirse abrumados fácilmente por la inmensa cantidad de ficheros contenidos en un disco duro. Por consiguiente, los examinadores necesitan procedimientos para centrarse en los datos potencialmente útiles. El proceso de filtrar los datos irrelevantes, confidenciales o privilegiados incluye:

- Identificar ficheros válidos del SSOO y otras entidades que no tienen relevancia para la investigación.
- Enfocar la investigación en los datos más probablemente creados por el usuario.
- Gestionar ficheros redundantes, que es particularmente útil cuando se trata con cintas de respaldo.

Otras técnicas menos metódicas de reducción de datos como búsqueda de cadenas específicas de texto o extraer solo ciertos tipos de ficheros, puede no solo hacernos perder pistas importantes, sino que puede dejar al investigador en un mar de datos superfluos. En resumen, una reducción de datos cuidadosa generalmente permite un análisis más eficiente y minucioso.

3.5.2 Búsqueda y recopilación de información

La siguiente guía esta basada en [3] y sirve para ayudar a los investigadores a identificar lás búsquedas más comunes de un análisis forense según su categoría específica de delito. La guía también ayudará a determinar el alcance del análisis que será realizado.

3.5.2.1 DELITOS CONTRA PERSONAS

Investigación de Muerte

- Libretas de direcciones

- Diarios
- E-mail/notas/cartas
- Registros de bienes/financieros
- Imágenes
- Logs de actividad en internet
- Documentos legales y testamentos
- Registros médicos
- Registros telefónicos

Violencia Domestica

- Libretas de direcciones
- Diarios
- E-mail/notas/cartas
- Registros de bienes/financieros
- Registros médicos
- Registros telefónicos

Amenazas/Acoso por E-mail

- Libretas de direcciones
- Diarios
- E-mail/notas/cartas
- Registros de bienes/financieros
- Imágenes
- Logs de actividad en internet
- Documentos legales
- Registros telefónicos
- Investigación de fondo sobre la víctima (posibles motivos, amigos, enemigos, etc)

3.5.2.2 DELITOS SEXUALES

Abuso/Explotación Infantil (Pornografía infantil)

- Logs de chat
- Marcas de tiempo en ficheros
- Software de camara digital
- E-mail/notas/cartas
- Juegos
- Software de edición y visionado de imágenes
- Imágenes/Videos
- Logs de actividad en internet
- Directorios creados por el usuario y nombres de fichero que clasifican imágenes

Prostitución

- Libretas de direcciones
- Biografías
- Agendas
- Bases de datos de clientes/registros
- E-mail/notas/cartas
- Identificación falsa
- Registros de bienes/financieros
- Logs de actividad en internet
- Registros médicos
- Publicidad en páginas de la web

3.5.2.3 FRAUDES/OTROS DELITOS FINANCIEROS

Fraude en subastas (Online)

- Datos de cuentas relativos a sitios de subastas
- Software de contabilidad y ficheros de datos asociados
- Libretas de direcciones
- Calendario
- Logs de chat
- Logs de actividad en internet
- Ficheros históricos/caché de explorador de internet
- Información de comprador/Datos de tarjeta de crédito
- Bases de datos
- Software de cámara digital
- Ficheros de imagen
- E-mail/notas/cartas
- Registros de bienes/financieros
- Software de acceso a instituciones financieras online
- Registros/Documentos de “testimonios”
- Registros telefónicos

Intrusión en un computador

- Libretas de direcciones
- Ficheros de configuración
- E-mail/notas/cartas
- Programas ejecutables
- Logs de actividad en internet
- Dirección de Internet (IP) y nombre de usuario
- Logs de IRC
- Código fuente
- Ficheros de texto (nombres de usuario y password)

Fraude económico (Incluyendo Fraude Online y Falsificación de dinero)

- Libretas de direcciones
- Agendas
- Imágenes de monedas
- Imágenes de cheques, monedas/billetes o giros postales
- Copiadoras de bandas magnéticas
- Información de comprador/Datos de tarjeta de crédito
- Bases de datos
- E-mail/notas/cartas
- Formularios de transacciones financieras falsos
- Identificación falsa
- Registros de bienes/financieros
- Imágenes de firmas
- Logs de actividad en internet
- Software de acceso a instituciones financieras online

Extorsión

- Marcas de tiempo de ficheros
- E-mail/notas/cartas
- Logs históricos
- Logs de actividad en internet
- Ficheros temporales de internet
- Nombres de usuario

Apuestas ilegales/Juego

- Libretas de direcciones
- Agendas
- Bases de datos del cliente y registros de jugador
- Información de cliente/Datos de tarjeta de crédito
- Dinero electrónico
- E-mail/notas/cartas
- Registros de bienes/financieros
- Logs de actividad en internet
- Imágenes de los jugadores
- Software de acceso a instituciones financieras online
- Estadísticas de apuestas en deportes

Robo de identidad

- Herramientas de hardware y software
    - Backdrops
    - Generadores de tarjetas de crédito
    - Lectores/Grabadores de tarjetas

    - Cámaras digitales
    - Escaners
- Plantillas de identificación
    - Certificados de nacimiento
    - Cheques
    - Imágenes digitales para identificación por fotografía
    - Permisos de conducir
    - Firmas electrónicas
    - Registros de vehículos ficticios
    - Documentos sobre seguros de automóviles
    - Firmas escaneadas
    - Tarjetas de seguridad social
- Actividad en internet relacionada con el robo de Identidad
    - E-mails y envíos a grupos de noticias
    - Documentos borrados
    - Pedidos online
    - Información de comercio online
    - Ficheros del sistema y el slack space de ficheros
    - Actividad en la web en sitios de falsificación
- Instrumentos negociables
    - Cheques comerciales
    - Cheques bancarios
    - Cheques individuales
    - Cheques de viajero
    - Dinero falsificado
    - Números de tarjeta de crédito
    - Documentos ficticios de juicios
    - Vales de regalo ficticios
    - Documentos ficticios de préstamos
    - Recibos ficticios de ventas
    - Giros postales
    - Documentos de transferencia de mercancías
    - Documentación de transferencia de vehículos

Narcoticos

- Libretas de direcciones
- Agendas
- Bases de datos
- Fórmulas/Recetas de drogas
- E-mail/notas/cartas
- Identificación falsa
- Registros de bienes/financieros
- Logs de actividad en internet
- Imágenes/Plantillas de formularios de recetas médicas

Piratería de Software

- Logs de chat
- E-mail/notas/cartas
- Ficheros de imagen de certificados software
- Números de serie
- Utilidades e información sobre crackeo de software
- Directorios creados por el usuario y nombres de ficheros del software clasificado con copyright.

Fraude de Telecomunicaciones

- Software de clonación
- Bases de datos de clientes/registros
- Registros con el par: Número de serie electrónicos/Número de identificación de móvil
- E-mail/notas/cartas
- Registros de bienes/financieros
- Manuales de "how to phreak" (phreak: prenombre que se le da una persona que penetra de manera ilegal a la red de teléfonos o de computadoras)
- Actividad de internet
- Registros telefónicos

En una escena física, se debe buscar material de duplicación y empaquetado. La siguiente información debería ser documentada cuando esté disponible:

- Sumario del caso
- Direcciones de internet(IP)
- Listas de palabras clave
- Apodos de internet (Nicknames)
- Passwords
- Puntos de contacto
- Documentos de soporte
- Tipo de delito

3.5.3 Cuadro Resumen

A	Pornografía Infantil	**F**	Fraude subastas online	**K**	Robo de identidad
B	Prostitución	**G**	Intrusión	**L**	Narcóticos
C	Muerte	**H**	Fraude económico	**M**	Piratería
D	Violencia doméstica	**I**	Extorsión	**N**	Fraude de telecomunicaciones
E	Amenazas/Acoso	**J**	Apuestas ilegales		

	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**
Información General:														
Bases de datos		✓				✓		✓		✓		✓		
Email/Notas/Cartas	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Registros Financieros		✓	✓	✓	✓	✓		✓		✓		✓		✓
Registros Medicos		✓	✓	✓										
Registros Telefónicos			✓	✓	✓	✓								✓
Información Específica:														
Datos sobre cuentas						✓								
Software de contabilidad						✓								
Libreta de direcciones		✓	✓	✓	✓	✓	✓	✓		✓		✓		
Backdrops											✓			
Biografias			✓											
Certificados de nacimiento											✓			
Agendas		✓				✓		✓		✓		✓		
Logs de Chat	✓					✓							✓	
Imagenes de cheque, moneda o giro postal								✓			✓			
Tarjetas de comprobación de disponible											✓			
Software de clonación														✓
Ficheros de configuración							✓							
Dinero falso											✓			
Generadores de tarjetas de crédito											✓			
Números de tarjetas de crédito											✓			
Lectores/Grabadores de tarjetas de crédito														

											✓			
Copiadores de bandas magnéticas								✓						
Bases de datos de clientes		✓								✓				✓
Información de clientes/Tarjetas de crédito						✓		✓		✓				
Marcas de tiempo	✓								✓					
Diarios			✓	✓	✓									
Software/Imágenes de cámaras digitales	✓					✓					✓			
Licencia de conducir											✓			
Recetas de drogas												✓		
Dinero electrónico										✓				
Firmas electrónicas											✓			
Documentos borrados de internet											✓			
Registros con el par ESN/MIN de móviles														✓
Programas ejecutables							✓							
Formularios falsos de transacciones financieras								✓						
Identificación falsa		✓						✓				✓		
Documentos falsos de juicios											✓			
Vales de regalo falsos											✓			
Documentos falsos de préstamos											✓			
Recibos falsos											✓			
Registros de vehículos falsos											✓			
Juegos		✓												
Edición y Visionado de imágenes	✓													
Logs históricos										✓				

Manuales “phreak”														✓
Imágenes	✓		✓		✓	✓								
Imágenes de firmas								✓						
Ficheros de certificados software													✓	
Reproductores de imagenes										✓				
Logs de actividad de internet	✓	✓	✓		✓	✓	✓	✓	✓	✓		✓	✓	✓
Históricos/Caché de los navegadores						✓								
Nombre Usuario / IP							✓							
Logs de IRC							✓							
Documentos legales y testamentos			✓		✓									
Videos	✓													
Software de acceso a bancos online						✓		✓		✓				
Pedidos online e info. de comercio											✓			
Imágenes de recetas médicas												✓		
Documentos de “testimonios”						✓								
Firmas escaneadas											✓			
Números de serie													✓	
Tarjetas de Seguridad Social											✓			
Utilidades/Información de crackeo de software													✓	
Código fuente							✓							
Estadísticas de apuestas deportivas										✓				
Documentos de transferencia de mercacías											✓			
Fichreos del sistema y slack space de ficheros											✓			

Ficheros temporales de internet									✓					
Nombres de usuario							✓		✓					
Directorios creados por el usuario y nombres de ficheros clasificados como software con copyright													✓	
Directorios creados por el usuario y nombres de ficheros de imágenes	✓													
Docuementación de seguros de vehículos											✓			
Búsqueda de fondo de la víctima					✓									
Actividad Web en sitios de falsificación											✓			
Publicidad de páginas web		✓												

3.5.4 Técnicas de extracción de información

La presente información se ha extraido de [1] y [7]

3.5.4.1 ORGANIZACIÓN DE UN SISTEMA DE FICHEROS

La motivación de un sistema de ficheros es medianamente simple: Las computadoras necesitan un método para el almacenamiento a largo plazo y la recuperación de datos. Los sistemas de ficheros proveen un mecanismo para que usuarios almacenen datos en una jerarquía de archivos y directorios. Un sistema de ficheros consta de datos de su estructura y de los datos de usuario, que están organizados de tal manera que la computadora sepa donde encontrarlos. En la mayoría de los casos, el sistema de ficheros es independiente de cualquier computadora específica.

Un sistema de ficheros se estructura en varias capas o categorías que pasamos a enumerar a continuación:

La categoría de **sistema de ficheros** contiene la información general del mismo. Todos los sistemas de ficheros tienen una estructura general para ellos, pero cada instancia de un sistema de ficheros en única, ya que tiene un tamaño único y puede ser modificada y adaptada para su uso específico.

La categoría **contenido** contiene los datos de los que se compone el contenido de un fichero. La mayoría de los datos de un sistema de ficheros atienden a esta categoría, y se organizan en una colección de contenedores de tamaño estándar. Cada sistema de ficheros asigna un nombre diferente a estos contenedores, como clusters o bloques, aunque nosotros usaremos *Unidad de datos* para generalizar el término.

La categoría **meta-datos** contiene los datos que describen un fichero. Esta categoría contiene información de donde se almacenan los ficheros, como de grandes son, las últimas fechas de lectura o escritura sobre él, e información de control de acceso. Nótese que esta categoría no tiene el contenido de un fichero o su nombre. Ejemplos de estructuras de datos en esta categoría incluyen las entradas de directorio FAT, las entradas en la MFT (Master File Table) de un sistema NTFS, y las estructuras de i-nodos de UFS y Ext3.

La categoría **nombre de fichero**, o categoría interfaz humana, contiene los datos que relacionan un nombre a cada fichero. En la mayoría de los sistemas de ficheros, esos datos están almacenados en el contenido de un directorio y hay una lista de nombres de ficheros con su correspondiente dirección de meta-datos. Esta categoría es similar a un nombre de host en una red. Los dispositivos de red se comunican entre ellos usando una dirección IP, que es difícil de recordar para la gente. Cuando un usuario introduce el nombre de host de una computadora remota, la computadora local debe traducir el nombre a una dirección IP antes de que comience la comunicación.

La categoría **aplicación** contiene datos que proporcionan características especiales. Esos datos no se necesitan durante el proceso de lectura o escritura de un fichero y, en la mayoría de los casos, no necesitan ser incluidos en la especificación del sistema de ficheros. Esos datos están incluidos en la especificación porque puede ser más eficiente implementarlos en el sistema de ficheros en lugar de en un fichero normal. Ejemplos de datos en esta categoría incluyen las estadísticas de cuota del usuario y la bitácora de un sistema de ficheros. Estos datos pueden ser útiles durante una investigación, pero pueden ser falsificados más fácilmente que otros datos, ya que no se necesitan para escribir y leer un fichero.

3.5.4.2 CATEGORÍA DE SISTEMA DE FICHEROS

La categoría de sistema de ficheros contiene los datos generales que nos permiten identificar como de único es el sistema de ficheros y donde se encuentran otros datos importantes. En muchos casos, la mayoría de estos datos están situados en una estructura de datos estándar en los primeros sectores del sistema de ficheros, de forma similar a tener un mapa de un edificio en el recibidor del mismo. Con esta información, los datos pueden ser localizados fácilmente.

El análisis de datos en la categoría de sistema de ficheros es necesario para todos los tipos de análisis de un sistema de ficheros, ya que es durante esta fase cuando se encuentra la localización de las estructuras de datos de otras categorías. Por lo tanto, si alguno de estos datos se corrompe o se pierde, se complica el análisis en otras categorías porque deberíamos encontrar una copia de seguridad o adivinar donde se encuentran estas estructuras. Además de la información general, el análisis de esta categoría puede mostrar la versión de un sistema de ficheros, su etiqueta (nombre), la aplicación que lo creó y la fecha de creación. Hay pocos datos en esta categoría de forma que un usuario debería poder cambiar o verla sin la ayuda de un editor hexadecimal. En muchos casos, los datos no generales que se encuentran en esta categoría son considerados como intrascendentes y podrían no ser exactos.

Técnicas de análisis

Los datos de esta categoría son normalmente valores individuales e independientes. Por lo tanto no hay mucho que hacer con ellos, salvo mostrarlos o usarlos en una herramienta. Si se están recuperando datos a mano, la información que contiene puede ser útil. Si se trata de determinar en que computadora fue creado el sistema de ficheros, un ID de volumen o su versión puede sernos de utilidad. Las estructuras de datos de esta categoría frecuentemente tienen valores no usados y direcciones de almacenamiento que podrían esconder pequeñas cantidades de datos. Un chequeo de la consistencia en esta categoría consiste en comparar el tamaño del sistema de ficheros con el tamaño del volumen en el que se encuentra. Si el volumen es más grande, los sectores que se

encuentran después del sistema de ficheros son llamados “volume slack” y puede ser usado para ocultar datos.

3.5.4.3 CATEGORÍA CONTENIDO

La categoría contenido incluye las direcciones de almacenamiento donde se alojan los ficheros y directorios de forma que puedan guardar datos. Los datos en esta categoría están organizados normalmente dentro de grupos del mismo tamaño, que llamaremos unidades de datos, que son por ejemplo los clusters o bloques. Una unidad de datos tiene un estado: asignado o no asignado. Normalmente hay algunos tipos de estructuras de datos que mantienen el estado de cada unidad de datos.

Cuando se crea un nuevo fichero o un fichero existente se hace mas grande, el Sistema Operativo busca una unidad de datos no asignada y la asigna a un fichero. Cuando se borra un fichero, las unidades de datos asignadas al fichero se ponen con estado no asignado y pueden asignarse a nuevos ficheros. La mayoría de los SSOO no limpian el contenido de la unidad de datos cuando se borra un fichero, sino que simplemente cambian su estado a no asignado. Este “borrado seguro” solo puede hacerse con herramientas especializadas o con SSOO que provean esta habilidad.

El análisis de esta categoría de contenido esta pues enfocada a recuperar datos perdidos y hacer búsquedas de datos a bajo nivel. Debido a la inmensa cantidad de datos que se pueden encontrar en esta categoría, normalmente no se analiza a mano. Como referencia, si un investigador examinara un sector de 512 bytes en cinco segundos, para analizar 40 GB necesitaría 388 días trabajando durante 12 horas diarias.

Información General

Veamos a continuación como se direccionan las unidades de datos, como se asignan y como se manejan las unidades de datos dañadas.

Direccionamiento lógico del sistema de ficheros

Un volumen es una colección de sectores direccionables que un SSOO o una aplicación pueden usar para almacenar datos. Los sectores en un volumen no necesitan ser consecutivos en un dispositivo de almacenamiento físico. En lugar de eso, necesitan sólo dar la impresión que lo están. Un disco duro es un ejemplo de un volumen que se encuentra organizado en sectores consecutivos. Un volumen también puede ser el resultado de ensamblar y la combinación de volúmenes más pequeños.

Un sector puede tener múltiples direcciones, cada una desde una perspectiva diferente. Cada sector tiene una dirección relativa al inicio del dispositivo de almacenamiento, que es lo que llamamos una dirección física. Los sistemas de volumen crean volúmenes y asignan direcciones lógicas de volumen que son relativas al inicio del volumen.

Los sistemas de ficheros usan las direcciones lógicas de volumen, pero además asignan direcciones lógicas de sistemas de ficheros, ya que agrupan varios sectores consecutivos para formar una unidad de datos. En la mayoría de los sistemas de ficheros, cada sector en el volumen es asignado a una dirección lógica de sistema de ficheros. Un ejemplo de un sistema de ficheros que no asigna una dirección lógica de sistema de ficheros a cada sector es FAT.

Estrategias de asignación

Un SSOO puede usar diferentes estrategias para asignar unidades de datos. Normalmente un SSOO asigna unidades de datos consecutivas, pero esto no es siempre posible. Cuando un fichero no tiene unidades de datos consecutivas se dice que está fragmentado.

Una primera estrategia busca una unidad de datos disponible empezando por la primera unidad de datos del sistema de ficheros. Después de que una unidad de datos ha sido asignada usando esta estrategia y se necesita una segunda unidad de datos, la búsqueda comienza de nuevo en el inicio del sistema de ficheros. Este tipo de estrategia puede fácilmente producir ficheros fragmentados ya que el fichero no es asignado de una pieza. Un SSOO que usa esta estrategia suele sobrescribir más a menudo los datos de ficheros borrados al inicio del sistema de ficheros. Por tanto tendremos más suerte recuperando contenidos borrados del final del sistema de ficheros.

Una estrategia similar está disponible; ésta inicia su búsqueda con la unidad de datos que fue más recientemente asignada en lugar de al comienzo. Este algoritmo es más balanceado para recuperar datos ya que las unidades de datos en el inicio del sistema de ficheros no son reasignadas hasta que las unidades de datos de final hayan sido reasignadas. Esto es así porque no se buscan unidades de datos libres desde el inicio hasta que no se haya llegado al final del sistema de ficheros.

Otra estrategia es la del mejor ajuste, que busca unidades de datos consecutivas que puedan alojar la cantidad de datos necesaria. Esta estrategia trabaja bien si se conocen cuantas unidades de datos necesitará un fichero, pero cuando el fichero crece, las nuevas unidades de datos que se necesitan pueden no ser consecutivas y tendríamos un fichero fragmentado.

Cada SSOO puede elegir una estrategia de asignación para un sistema de ficheros. Algunos sistemas de ficheros especifican que estrategia debería usarse, pero no existe ninguna manera para forzarlo. Debería pues probar la implementación de un sistema de ficheros antes de asumir que se esta usando la estrategia de la especificación.

Además de probar el sistema operativo para determinar su estrategia de asignación, se debería considerar la aplicación que crea el contenido. Por ejemplo, cuando se actualiza un fichero existente, algunas aplicaciones abren el fichero original, lo actualizan y guardan los nuevos datos sobre los originales. Otras aplicaciones pueden hacer una segunda copia del fichero original, actualizar esta copia y luego renombrar la

copia de forma que se sobrescribe el fichero original. En este caso, el fichero se almacena en nuevas unidades de datos, ya que es parte de un nuevo fichero.

Unidades de datos dañadas

Muchos sistemas de ficheros tienen la habilidad de marcar una unidad de datos como dañada. Esto era necesario en los discos duros más antiguos, que no tenían la capacidad de manejar errores. El sistema operativo debería detectar que una unidad de datos era mala y marcarla de forma que no se asignara a un fichero. En la actualidad los modernos discos duros pueden detectar un sector erróneo y reemplazarlo por uno de repuesto, de modo que no se necesita la funcionalidad del sistema de ficheros.

Es fácil esconder datos usando esta funcionalidad del sistema de ficheros, si existe (solo está en discos duros antiguos). Muchas herramientas que prueban la consistencia de un sistema de ficheros no verifican que una unidad de datos que está marcada como dañada esté actualmente dañada. Por lo tanto, un usuario podría añadir manualmente una unidad de datos a la lista de dañadas y así esconder datos.

Técnicas de análisis

Ahora que hemos visto los conceptos básicos de la categoría contenido, vamos a ver como analizar los datos. Esta sección cubre diferentes técnicas de análisis que pueden ser usados cuando se buscan evidencias.

Visualizando las unidades de datos

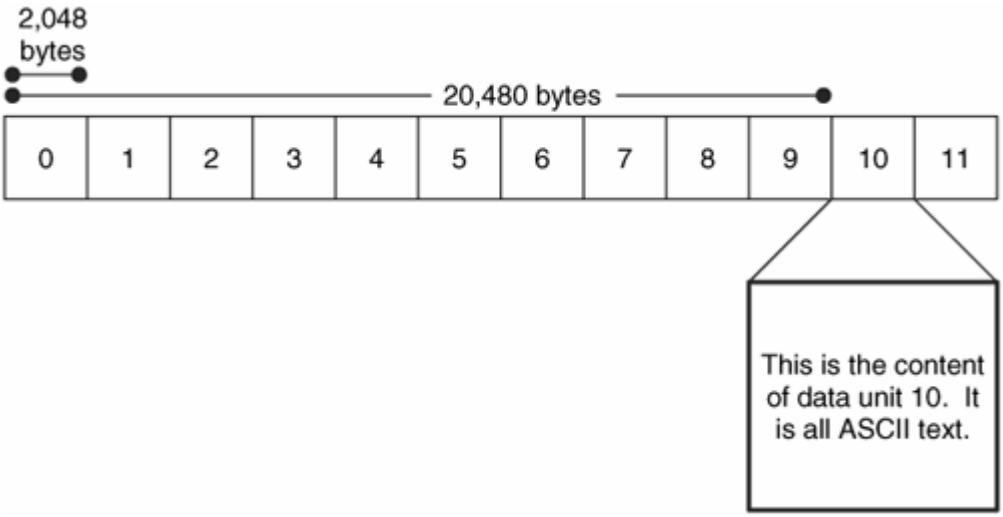
Esta es una técnica usada cuando el investigador conoce la dirección donde puede estar la evidencia, tal como una asignada a un fichero especifico o una que tiene un especial significado. Por ejemplo, en muchos sistemas de ficheros FAT32, el sector 3 no es usado por el sistema de ficheros y está lleno de ceros. Es fácil esconder datos en este sector, y por tanto, visualizando el contenido del sector 3 podemos ver si ha sido modificado si no esta lleno de ceros.

La teoría que encierra este tipo de análisis es simple. El investigador introduce la dirección lógica del sistema de ficheros de una unidad de datos y una herramienta calcula la dirección del byte o sector de la unidad de datos. La herramienta busca la localización y lee los datos. Por ejemplo considere un sistema de ficheros donde la unidad de datos 0 empieza en el byte con offset 0, y cada unidad de datos tiene 2 kb (2048 bytes). El offset de byte de cada unidad de la unidad de datos 10 está en el kb 20 (20480 bytes).

La teoría que encierra esto es simple. El investigador introduce la dirección lógica del sistema de ficheros de la unidad de datos y una herramienta calcula su dirección de byte o sector. La herramienta busca en esa ubicación y lee los datos. Por ejemplo, considérese un sistema de ficheros donde la unidad de datos 0 comienza en el desplazamiento de byte 0, y cada unidad de datos es de 2.048 bytes (2 kb). El

desplazamiento de byte de la unidad de datos 10 será de 20.480 bytes (20 kb). Podemos ver esto en la siguiente figura:

Contenido de la unidad de datos 10.



Hay muchas herramientas, como editores hexadecimales y herramientas de investigación que proveen esta función.

Búsqueda de cadenas

En la técnica anterior, conocíamos donde podía estar la evidencia. En esta técnica sabemos que el contenido que debería tener la evidencia, pero no sabemos donde está. Una búsqueda lógica del sistema de ficheros busca en cada unidad de datos un valor o frase específicos. Por ejemplo, podríamos buscar la frase "forense" o un valor específico de una cabecera de fichero. Ésta técnica suele usarse en la búsqueda de datos en la memoria de Intercambio (Swap), que suele ser un conjunto de datos en bruto sin metadatos ni nombres de fichero apuntando a ellos.

Esta técnica de búsqueda se ha llamado históricamente una búsqueda física ya que usa la ordenación física de los sectores. Esta búsqueda es precisa cuando se analiza un disco simple, pero en caso de sistemas que usen "disk spanning" o discos RAID, el orden de los sectores no es el orden físico.

Desafortunadamente, los ficheros no siempre alojan unidades de datos consecutivas y si el valor que estamos buscando se encuentra en dos unidades de datos no consecutivas de un fichero fragmentado, una búsqueda lógica en el sistema de ficheros no lo encontrará.

Estado de Asignación de Unidades de Datos

Si no conocemos la localización exacta de la evidencia, pero sabemos que no está asignada, podemos enfocar nuestra atención ahí. Algunas herramientas pueden extraer todas las unidades de datos no asignadas de la imagen de un sistema de ficheros a un

fichero separado, y otras pueden restringir su análisis a solo las áreas no asignadas. Si extraemos solo los datos no asignados, la salida será una colección de datos en bruto sin estructura de sistema de ficheros, de modo que no se puede usar con una herramienta de análisis de sistema de ficheros.

Orden de asignación de las unidades de datos

Previamente hemos visto algunas estrategias que un sistema operativo puede usar cuando asigna unidades de datos. La estrategia que se use depende generalmente del SSOO; por consiguiente se encuentra en el área del análisis a nivel de aplicación. Por ejemplo, Windows ME puede usar una estrategia de asignación diferente para un sistema de ficheros FAT que Windows 2000, pero ambos producen un sistema de ficheros FAT válido.

Si el orden de asignación relativo de dos o más unidades de datos es importante, podemos considerar la estrategia de asignación del SSOO para ayudarnos a determinarlo. Esto es muy difícil, ya que requiere determinar la estrategia que usa el SSOO y necesitaremos examinar cada escenario que podríamos tener según el estado de las unidades de datos en un momento dado. Esto implica conocer información a nivel de aplicación. Esta técnica se usa durante la reconstrucción de eventos, lo cual ocurre después de que hayamos reconocida las unidades de datos como evidencias.

Pruebas de consistencia

Esta es una importante técnica de análisis para cada categoría de datos. Nos permite determinar si el sistema de ficheros esta en un estado sospechoso. Una prueba de consistencia en la categoría contenido usa datos de la categoría meta-datos y verifica que cada unidad de datos asignada tiene exactamente una entrada de meta-datos apuntando a ella. Esto se hace para prevenir que un usuario manipule manualmente el estado de asignación de una unidad de datos sin que esta tenga un nombre. Las unidades de datos asignadas que no tienen una correspondiente estructura de meta-datos se llaman huérfanas.

Otras pruebas examinan cada unidad de datos que se lista como dañada. Si tenemos una imagen de un disco duro que contiene sectores defectuosos, muchas de las herramientas de adquisición de datos llenarán los datos dañados con ceros. En este caso, las unidades de datos ubicadas en la lista de defectuosos deberían tener ceros en su interior.

Técnicas de borrado seguro

Ahora que sabemos como analizar datos en esta categoría, vamos a pasar a ver como un usuario puede hacernos la vida mas dura. La mayoría de las herramientas de borrado seguro operan en la categoría contenido y escriben ceros o datos aleatorios en las unidades de datos de un fichero asignado o de todas las unidades de datos.

El borrado seguro está siendo cada vez más común y una característica estándar en algunos sistemas operativos. Las que se construyen en los sistemas operativos son más efectivas a la hora de limpiar todos los datos (poner todos los bits a 0). Las aplicaciones externas frecuentemente dependen del SSOO para actuar de cierto modo; por tanto, no pueden ser tan efectivos. Por ejemplo, hace muchos años había una herramienta basada en Linux que escribía ceros en una unidad de datos antes de que se estableciera como no asignada, pero el SSOO no escribía inmediatamente ceros en el disco. Más tarde el SSOO vería que la unidad estaba no asignada y por tanto no se tomaría la molestia de escribir ceros en ella. De manera semejante, muchas herramientas asumen que cuando escriben datos en un fichero, el SSOO usará las mismas unidades de datos. Un SSOO puede elegir asignarle otras unidades de datos y, en tal caso, el contenido del fichero aun existirá.

La detección del uso de herramientas de borrado seguro en esta categoría puede ser difícil. Obviamente, si una unidad de datos no asignada contiene ceros o valores aleatorios, podemos sospechar de una herramienta de este tipo. Si la herramienta escribe valores aleatorios o hace copias de otras unidades de datos existentes, la detección es virtualmente imposible sin una evidencia a nivel de aplicación de que se usó una de estas herramientas. Por supuesto, si se encuentra una herramienta de borrado seguro en el sistema, podemos hacer pruebas para ver si se usó cual fue su último tiempo de acceso. También se pueden encontrar copias temporales de los archivos si estos fueron borrados explícitamente.

3.5.4.4 CATEGORÍA META-DATOS

La categoría meta-datos es donde residen los datos descriptivos. Aquí podemos encontrar, por ejemplo, el último tiempo de acceso y las direcciones de las unidades de datos que un fichero tiene asignadas. Hay pocas herramientas que se identifiquen como de análisis de meta-datos. En su lugar, vienen combinadas con el análisis de la categoría de nombre de fichero.

Muchas estructuras de meta-datos son almacenadas en una tabla estática o dinámica, y cada entrada tiene una dirección. Cuando un fichero es borrado, la entrada de metadatos se modifica al estado no asignado y el SSOO puede limpiar algunos valores de la entrada.

El análisis en esta categoría está enfocado a determinar más detalles sobre un fichero específico o buscar un fichero que cumple ciertos requerimientos. Esta categoría tiende a tener más datos intrascendentes que otras categorías. Por ejemplo, la fecha del último acceso o el número de escrituras pueden no ser precisos. Además, un investigador no puede concluir que un usuario tuvo o no permisos de lectura de un fichero sin otras evidencias de otras categorías.

Información General

En esta sección miraremos los conceptos básicos de la categoría meta-datos. Veremos otro esquema de direccionamiento, “slack space”, recuperación de ficheros borrados, ficheros comprimidos y encriptados.

Dirección lógica de fichero

Previamente hemos visto como una unidad de datos tiene una dirección lógica de sistema de ficheros. Una dirección lógica de fichero de una unidad de datos es relativa al inicio del fichero al cual está asignado. Por ejemplo, si un fichero tiene asignadas dos unidades de datos, la primera unidad de datos debería tener una dirección lógica de fichero de 0, y el segundo una dirección de 1. El nombre o dirección de metadatos para el fichero es necesaria para hacer una única dirección lógica de fichero.

Slack Space

El “Slack space” es una de las palabras de moda en el análisis forense que mucha gente ha oído alguna vez. El Slack space ocurre cuando el tamaño de un fichero no es múltiplo del tamaño de la unidad de datos. Un fichero debe tener asignada una unidad de datos completa aunque muchas veces solo use una pequeña parte. Los bytes no usados al final de la unidad de datos es lo que se llama Slack space. Por ejemplo, si un fichero tiene 100 bytes, necesita tener asignada una unidad completa de 2048 bytes. Los 1948 bytes que sobran sería slack space.

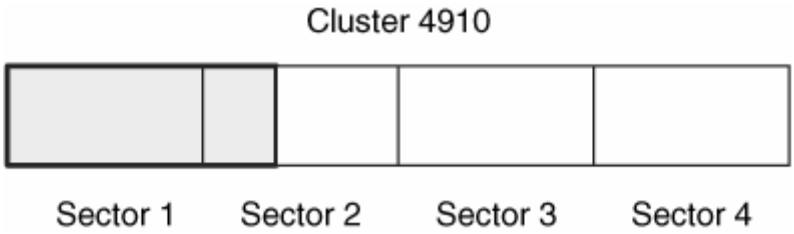
Este espacio es interesante porque las computadoras son perezosas. Algunas de ellas no limpian los bytes no usados, de modo que el slack space contiene datos de ficheros anteriores o de la memoria. Debido al diseño de la mayoría de las computadoras, hay dos áreas interesantes en el Slack space. La primera área se ubica entre el final del fichero y el final del sector en el que el fichero termina. La segunda área se encuentra en los sectores que no contienen contenido del fichero. Hay dos áreas distintas porque los discos duros están basados en bloques y solo pueden ser escritos en sectores de 512 bytes. Siguiendo el ejemplo anterior, el SSOO no puede escribir solo 100 bytes en el disco, sino que debe escribir 512. Por lo tanto, necesita rellenar los 100 bytes con 412 bytes de datos. Esto se puede comparar al envío por correos de un objeto en una caja. El espacio sobrante hay que rellenarlo con algo hasta completar la caja.

El primer área del slack space es interesante porque el SSOO determina con que rellenar el contenido. El método obvio es rellenar el sector con ceros y esto es lo que la mayoría de SSOO hacen. Esto es como rellenar la caja anterior con papel de periódico. Algunos SSOO antiguos llamados DOS y más tarde Windows, rellenaban el sector con datos de la memoria. Esto es como rellenar la caja con copias de tu declaración de hacienda. Este área de slack space se llamó RAM slack, y ahora normalmente es rellenada con ceros. El RAM slack podía revelar passwords y otros datos que se supone que no deberían estar en el disco.

La segunda área del slack space se compone de los sectores en desuso de una unidad de datos. Esta área es interesante porque los SSOO limpian los sectores y otros los ignoran. Si se ignora, los sectores contendrán datos del fichero al que pertenecían previamente.

Consideremos un sistema de ficheros NTFS con clusters de 2048 bytes y sectores de 512 bytes, con lo que cada clúster se compone de 4 sectores. Nuestro fichero tiene 612 bytes, de modo que usa el primer sector entero y 100 bytes más del segundo sector. Los 412 bytes que sobran del segundo sector son rellenados con los datos que elija el SSOO. El tercer y cuarto sectores pueden ser limpiados con ceros por el SSOO, o pueden no tocarse conservar los datos de un fichero borrado. Podemos ver esto en la siguiente figura donde las áreas en gris representan el contenido del fichero y el espacio blanco es el slack space.

Slack space de un fichero de 612 bytes en un cluster de 2048 bytes donde cada sector tiene 512 bytes.



Una analogía común para el slack space es el video VHS. Supongamos una cinta VHS de 60 minutos. Una noche alguien graba 60 minutos de un episodio de una serie de TV. En otra ocasión decide ver de nuevo el episodio y al final rebobina la cinta. Otra noche graba 30 minutos de otro programa de TV. En ese punto la cinta queda "asignada" a este programa de TV, ya que el contenido anterior se borró, pero aún quedan 30 minutos de la serie de TV en el espacio sobrante de la cinta.

Recuperación de ficheros basada en los Meta-datos

En algunos casos, se puede querer buscar evidencias en los ficheros borrados. Hay dos métodos principales para recuperar ficheros borrados: basados en meta-datos y basados en aplicación. Las segundas se mencionan en apartados posteriores y por tanto vamos a hablar de la primera aquí. La recuperación basada en meta-datos trabaja cuando los meta-datos del fichero borrado aun existen. Si el meta-dato fue borrado o si la estructura de meta-datos se reasignó a un nuevo fichero, se necesitará el apoyo de las técnicas basadas en aplicación.

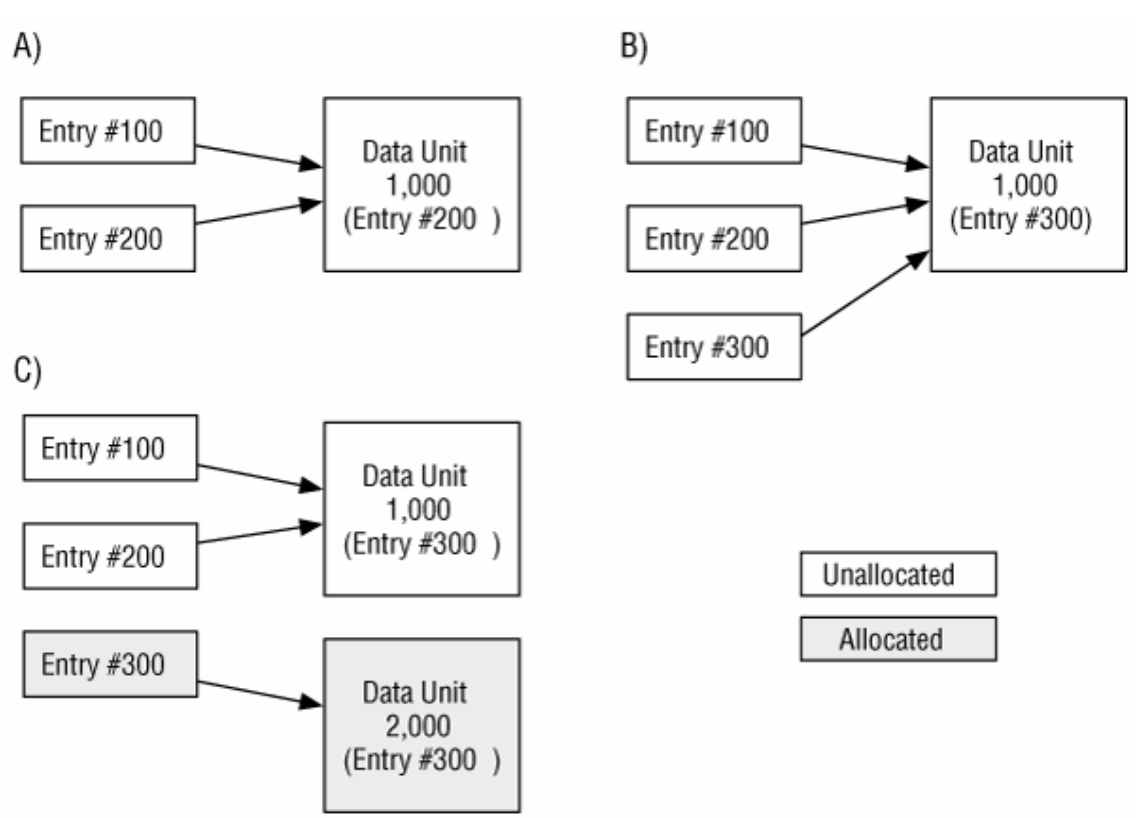
Después de encontrar la estructura de meta-datos del fichero, recuperarlo es fácil. No es diferente de leer los contenidos de un fichero asignado. Se necesita ser cuidadosos a la hora de hacer recuperación basada en meta-datos ya que las estructuras

de meta-datos y las unidades de datos podrían no estar sincronizadas, de modo que la unidad de datos esté asignada a nuevos ficheros.

Esto es similar a enlazar a una persona con un hotel en el cual haya estado. Después de que la persona registre la salida, todavía puede haber un registro de que él estuvo en el cuarto 427, pero la condición del cuarto de ese punto de adelante puede no tener nada que ver con él, ya que otros clientes la han podido usar.

Cuando se recuperan archivos borrados, puede ser difícil detectar cuando una unidad de datos ha sido reasignada. Vamos a considerar una secuencia de asignaciones y borrados. La estructura de meta-datos 100 tiene asignada la unidad de datos 1000 y guarda los datos en ella. El fichero cuya entrada de meta-datos es la 100 es borrado y por tanto esta entrada y la unidad de datos 1000 pasan al estado de no asignadas. Un nuevo fichero es creado en la estrada de meta-datos 200, y se le asigna la unidad de datos 1000. Más tarde, ese fichero es también borrado. Si analizamos este sistema, encontraremos dos entradas de meta-datos no asignadas que tienen la misma dirección de unidad de datos.

La secuencia de estados donde los ficheros son asignados y borrados y en C no está claro de donde vienen los datos de la unidad de datos 1000.



Necesitamos determinar cual de las entradas se asignó al fichero más recientemente. Un método para hacer esto es usar la información temporal de cada entrada (u otros

datos externos), pero puede que no seamos capaces de asegurarlo. Otro método es usar el tipo de fichero, si el meta-dato almacena esa información. Por ejemplo, la entrada de meta-datos 200 podría haber pertenecido a un directorio, de modo que podríamos analizar el contenido de la unidad 1000 para ver si tiene el formato de un directorio.

Aunque podamos determinar que la entrada 200 tuvo asignada la unidad de datos después que la 100, no sabemos si la entrada 200 fue la ultima entrada que lo tuvo asignado. Para ver esto, considere una entrada 300 asignada a la unidad de datos 1000 después de que se estableciera la entrada 200 a no asignada. Ese fichero es luego borrado y podemos ver el resultado en la figura B, donde hay tres entradas no asignadas que tienen la misma dirección de unidad de datos.

A continuación, un nuevo fichero fue creado y la entrada 300 fue reasignada a una nueva unidad de datos 2000. Si analizáramos el sistema en este estado no encontraríamos ninguna evidencia en la entrada 300, sino en la 100 y la 200, lo cual se muestra en la figura C.

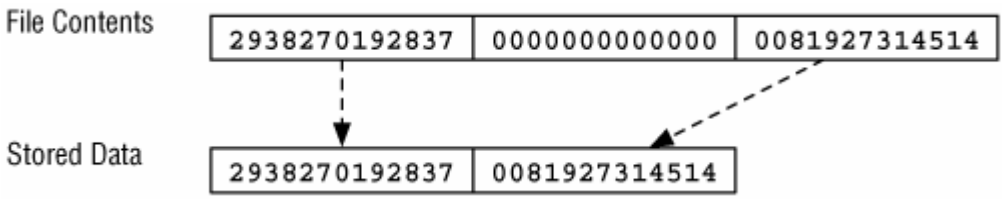
El objetivo de este ejemplo es mostrar que aunque una estructura de meta datos no asignada aun contenga las direcciones de unidades de datos, es muy difícil determinar si el contenido de la unidad de datos corresponde a ese fichero o un fichero fue creado después de que se estableciera la estructura de meta-datos como no-asignada. Se puede verificar que una recuperación de fichero fue precisa intentando abrirla en la aplicación que pensemos que la creó. Por ejemplo, si recuperamos un fichero “datos.txt” lo podemos abrir con un editor de textos Si a un nuevo fichero se le asigna la unidad de datos de un fichero borrado y escribe datos en ella, la estructura interna del fichero puede estar corrupta y un visor podría no abrirla.

Ficheros comprimidos y Sparse

Algunos sistemas de ficheros permiten que los datos se almacenen en un formato comprimido de forma que ocupen menos unidades de datos en el disco. Para lo ficheros, la compresión puede ocurrir en al menos tres niveles. En el nivel más alto es cuando los datos de un formato de fichero son comprimidos. Por ejemplo, un fichero JPEG es un ejemplo de esto donde los datos que almacenan la información de la imagen son comprimidos, pero no así la cabecera. El siguiente nivel es cuando un programa externo (winzip, winrar, gzip,...) comprime un fichero entero y crea un nuevo fichero. El fichero comprimido debe ser descomprimido a otro fichero antes de poder ser usado.

El último y más bajo nivel de compresión es cuando el sistema de ficheros comprime los datos. En este caso, una aplicación que escribe en el fichero no conoce que el fichero esta siendo comprimido. Hay dos técnicas de compresión básicas usadas por los sistemas de ficheros. La más intuitiva es usar las mismas técnicas de compresión que se usan sobre ficheros y se aplican a las unidades de datos de los ficheros. La segunda técnica es no asignar una unidad de datos física si va a estar llena de ceros. Los ficheros que saltan unidades de datos llenas con ceros son llamados “sparse files”, y un ejemplo puede verse en la figura 8.12. hay muchas maneras de implementar esto, por ejemplo, en el Unix File System (UFS), se escribe un 0 a cada campo que usualmente almacena la dirección de un bloque. Un fichero no puede tener asignado el bloque 0, por lo que el sistema al leer el campo dirección sabe que ese bloque esta lleno de ceros.

Un fichero almacenado en formato "sparse" donde las unidades de datos con ceros no se escriben.



Los ficheros comprimidos pueden presentar un reto para un investigador porque la herramienta de investigación debe soportar los algoritmos de compresión. Además, algunas formas de búsqueda de cadenas y recuperación de ficheros son inefectivas debido a que examinan los datos comprimidos sin saber que lo están.

Ficheros encriptados

El contenido de un fichero puede ser almacenado en una forma encriptada para protegerlo contra accesos no autorizados. La encriptación puede ser aplicada por una aplicación externa (por ejemplo PGP), por la misma aplicación que crea el fichero o por el SSOO cuando crea el fichero. Antes de que un fichero se escriba en disco, el SSOO encripta el fichero y guarda el texto cifrado en la unidad de datos. Datos como el nombre del fichero y el último tiempo de acceso, normalmente no son encriptados. La aplicación que escribió los datos no conoce que el fichero está encriptado en el disco. Otro método de encriptación de contenidos de fichero es encriptar un volumen entero (por ejemplo con PGP Disk, Macintosh encrypted disk images y Linux AES encrypted loopback images). En este caso, todos los datos del sistema de ficheros son encriptados y no solo el contenido. En general, el volumen que contiene el SSOO no es encriptado completamente.

Los datos encriptados pueden presentar un reto a un investigador ya que la mayoría de los ficheros son inaccesibles si no se conoce la clave de encriptación o password. En el peor de los casos, si no se conoce la técnica de encriptación. Algunas herramientas existen para probar cada combinación posible de claves o passwords, llamados ataques de fuerza bruta, pero estos no son útiles si no se conoce el algoritmo. En cualquier caso, si solo se encriptaron algunos ficheros y directorios en lugar del volumen entero, pueden encontrarse copias de los datos desencriptados en los ficheros temporales o en el espacio no asignado, ya que probablemente el fichero fue borrado.

Técnicas de análisis

Vamos a ver como se analizan datos en la categoría meta-datos. Usaremos los meta-datos para ver el contenido de los ficheros, buscar valores y localizar ficheros borrados.

Búsqueda de Metadatos

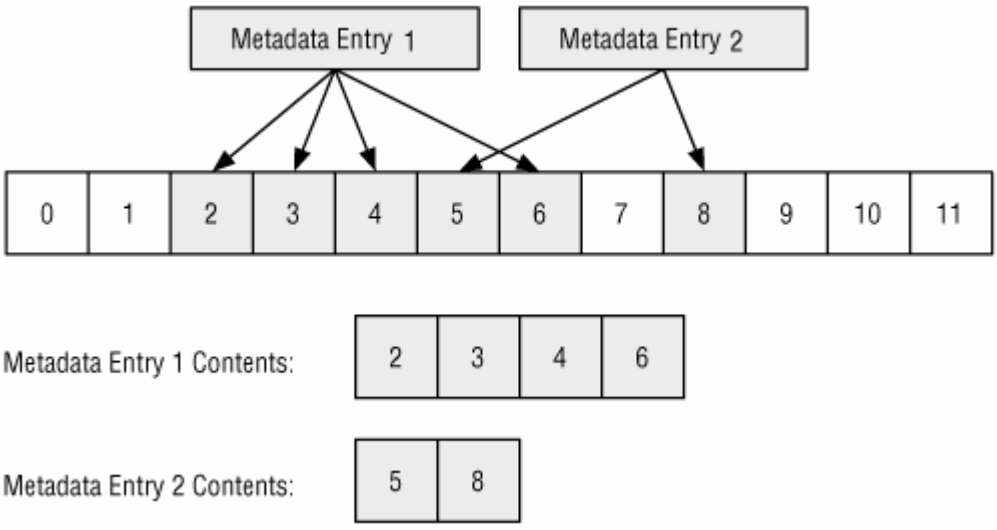
En muchos casos, analizamos los metadatos porque encontramos el nombre de un fichero que apunta a una estructura de metadatos específica y queremos aprender más sobre el fichero. Por lo tanto, necesitamos ubicar los metadatos y procesar su estructura de datos. Por ejemplo, si buscamos a través de los contenidos de un directorio y encontramos un fichero llamado "secretos.txt", podemos querer saber su contenido y cuando fue creado. La mayoría de las herramientas automáticamente realizan esta búsqueda cuando se listan los nombres de fichero en un directorio y permiten ordenar la salida basándose en los valores de meta-datos.

Los procedimientos exactos para esta técnica dependen del sistema de ficheros porque los meta-datos podrían estar en varios lugares del sistema de ficheros.

Después de buscar los meta-datos de un fichero, podemos ver los contenidos del fichero leyendo las unidades de datos asignadas al fichero. Haremos esto cuando estemos buscando evidencias en el contenido de un fichero.

Este proceso ocurre en las categorías de meta-datos y contenido. Sumaos la técnica de búsqueda de meta-datos para encontrar las unidades de datos asignadas al fichero y luego usar la técnica de visionado de contenido para encontrar el contenido actual. Podemos verlo en la figura 8.14 donde las unidades de datos asignadas a las entradas de meta-datos 1 y 2 son mostradas. Muchas herramientas gráficas combinan este proceso con el listado de nombres de ficheros. Cuando se selecciona un fichero, la herramienta busca las unidades de datos que se listan en los meta-datos.

Combinamos la información de las entradas de metadatos y las unidades de datos para ver el contenido de un fichero.



Durante este proceso, necesitamos mantener en mente el slack space porque el fichero puede no estar usando completamente el final de la unidad de datos.

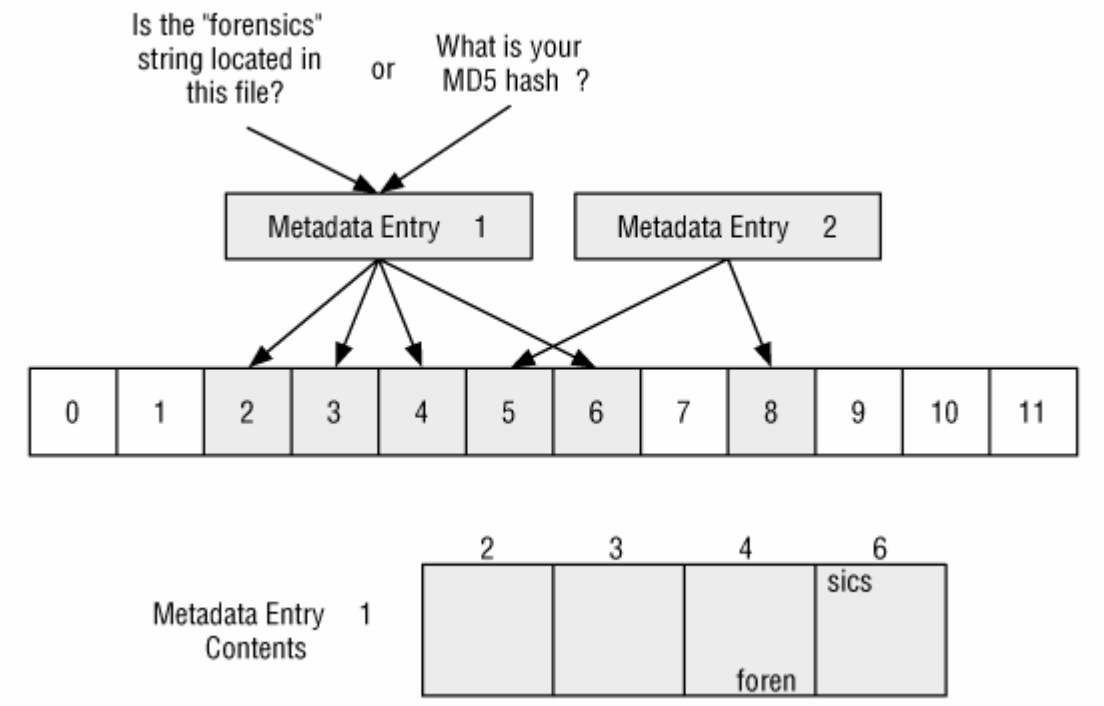
Calcularemos cuanto espacio se está usando al final dividiendo el tamaño del fichero entre el tamaño de una unidad de datos.

Búsqueda lógica de ficheros

La técnica anterior asumió que tenemos los meta-datos para encontrar el contenido de un fichero y poder ver su contenido. Muchas veces, este no es el caso y tendremos que buscar un fichero basándonos en su contenido. Por ejemplo, si queremos todos los ficheros con la palabra "virus" dentro de él. Esto es lo que llamamos una búsqueda lógica de ficheros. Esta búsqueda usa las mismas técnicas que vimos para el visionado de ficheros, salvo que ahora buscamos datos con un valor específico en vez de visionarlos.

Este proceso puede sonar muy similar a la búsqueda lógica en el sistema de ficheros. Lo es, excepto que ahora buscamos las unidades de datos en el orden que fueron usadas por ficheros y no por su ordenación en el volumen. Podemos verlo en la figura 8.15 donde tenemos dos entradas de meta-datos y las unidades de datos que tienen asignadas. En este caso, buscamos las unidades de datos 2, 3, 4 y 6 como un conjunto (un fichero). El beneficio de esta búsqueda sobre la búsqueda lógica del sistema de ficheros es que los valores que las unidades de datos o sectores fragmentados serán encontrados. Por ejemplo, en este caso estamos buscando el término "forense" comenzando desde la unidad de datos 4 y terminando en la 6. No lo encontraríamos en la búsqueda lógica del sistema de ficheros porque no está contenido en unidades de datos consecutivas. Una variación de esta búsqueda es buscar un fichero con un valor hash específico MD5 o SHA-1.

Una búsqueda lógica en las unidades de datos asignadas a una entrada de metadatos.



Teniendo en cuenta que solo las unidades de datos asignadas tienen dirección lógica de fichero, deberíamos realizar una búsqueda lógica de volumen de las unidades de datos no asignadas para el mismo valor. Por ejemplo, una búsqueda lógica de fichero con la configuración de la figura anterior, no se hubiera buscado en las unidades de datos 0 y 1, de modo que deberíamos hacer una segunda búsqueda que incluya 0, 1, 7, 9, 10, 11 y así sucesivamente.

Análisis de Meta-datos No Asignados

Si estamos buscando contenido borrado, no deberíamos limitarnos solo a los nombres de ficheros borrados que son mostrados en un listado de directorio. Veremos algunos ejemplos en la Categoría de Nombre de Ficheros, pero es posible que se reuse un nombre de fichero antes que lo sea la estructura de meta-datos. Por tanto, nuestra evidencia podría estar en una entrada de meta-datos no asignada y no podríamos verla porque ya no tiene un nombre.

Búsqueda y Ordenación de Atributos de Meta-Datos

Es bastante común buscar ficheros basándose en uno de sus valores de meta-datos. Por ejemplo, podría ser que encontramos una alerta en el log de un IDS (Intrusion Detection System) y queremos encontrar todos los ficheros que fueron creados dos minutos después de que se iniciara la alarma. O puede ser que estemos investigando un usuario y queramos encontrar todos los ficheros que en los que escribió en un determinado momento.

Los tiempos de fichero pueden cambiarse fácilmente en algunos sistemas, pero también pueden proporcionarnos muchas pistas. Por ejemplo, si tenemos una hipótesis de que un atacante ganó acceso a una computadora a las 8:13 p.m. e instaló herramientas de ataque, podemos probar la hipótesis buscando todos los ficheros creados entre las 8:13 p.m. y las 8:23 p.m. Si no encontramos ningún fichero de interés en este intervalo de tiempo, pero encontramos herramientas de ataque que fueron creadas en un tiempo diferente, podemos sospechar que los tiempos han sido manipulados, que nuestra hipótesis es incorrecta o ambas cosas.

Los datos temporales pueden también ser usados cuando nos encontramos con una computadora que conocemos un poco. Los datos temporales muestran qué ficheros fueron recientemente accedidos y creados. Esa información puede darnos sugerencias sobre como se usó una computadora.

Algunas herramientas crean líneas de tiempo de la actividad del fichero. En muchas líneas de tiempo, cada fichero tiene tantas entradas en la línea de tiempo como valores temporales. Por ejemplo, si tiene el último acceso, la última escritura y la última modificación, tendremos tres entradas en la línea de tiempo.

En TSK, la herramienta `mactime` se usa para hacer lineas de tiempo. Un ejemplo de la salida de mactime para el directorio `C:\Windows` podría ser:

```
Wed Aug 11 2004 19:31:58     34528 .a. /system32/ntio804.sys

                             35392 .a. /system32/ntio412.sys

[REMOVED]

Wed Aug 11 2004 19:33:27     2048 mac /bootstat.dat

                             1024 mac /system32/config/default.LOG

                             1024 mac /system32/config/software.LOG

Wed Aug 11 2004 19:33:28   262144 ma. /system32/config/SECURITY

                           262144 ma. /system32/config/default
```

En la salida anterior, podemos ver la actividad del fichero en cada segundo. La primera columna tiene la marca de fecha, la segunda es el tamaño del fichero, y la tercera muestra si esta entrada contiene tiempo de modificación (m-time), acceso al contenido (a-time), o cambio en los meta-datos (c-time). La última columna muestra el nombre del fichero. Hay mucha más información que no se muestra ya que esto es solo un ejemplo.

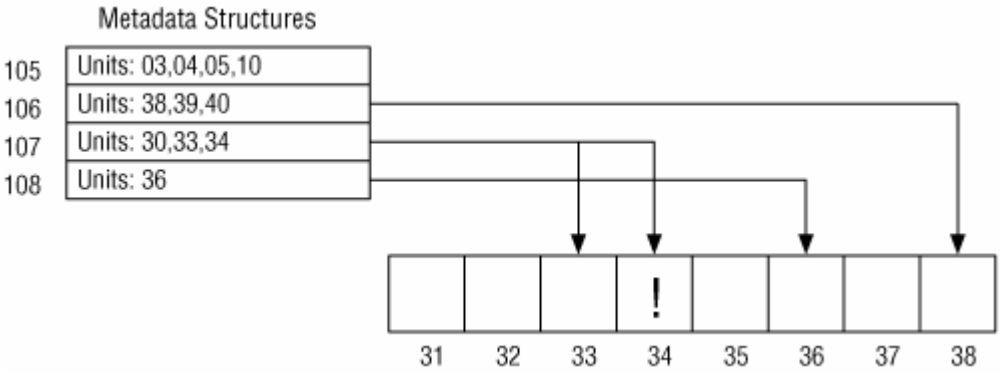
Nótese que se debe entender como un sistema de ficheros almacena sus marcas de tiempo antes de intentar correlacionar tiempos de ficheros y entradas de log de varias computadoras. Algunas marcas de tiempo se almacenan en UTC, que significa que se necesita saber el desplazamiento de la zona de tiempo donde se ubicaba la computadora para determinar el valor actual de tiempo. Por ejemplo, si alguien accede a un fichero a las 2:00 p.m. en Boston, el SSOO grabará que accedí a las 7:00 p.m. UTC, ya que Boston se encuentra cinco horas detrás de la UTC. Cuando un investigador analiza el fichero, necesita convertir las 7:00 p.m. a las 2:00 p.m., hora de Boston (lugar donde ocurrieron los hechos) o a su hora local (si el análisis no se realiza en Boston). Otros sistemas de ficheros almacenan el tiempo con respecto a la zona de tiempo local, y almacenaría 2:00 p.m. en el ejemplo previo.

Podemos además querer buscar ficheros para los cuales un determinado usuario ha tenido acceso de escritura. Esto muestra qué ficheros podría haber creado un usuario, si asumimos que el SSOO impone permisos y el sospechoso no tenía permisos de administrador. También podemos buscar por el ID del propietario del fichero, si existe. Este método se usa cuando investigamos un usuario específico.

Si previamente hemos realizado una búsqueda lógica del sistema de ficheros y encontramos datos interesantes en una de las unidades de datos, podemos querer buscar las entradas de meta-datos para esa dirección de unidad de datos. Esto puede mostrar qué ficheros tienen asignada la unidad de datos y a continuación podemos encontrar las otras unidades de datos que son parte del mismo fichero. Un ejemplo de esto podemos encontrarlo en la figura 8.16, donde tenemos una evidencia en la unidad de datos 34. Buscamos los meta-datos y encontramos que la estructura de meta-datos 107 apunta a esa unidad de datos, a la vez que a la 33 y la 36. Si el SSOO no limpia los valores

dirección cuando un fichero es borrado, este proceso puede identificar estructuras de meta-datos no asignadas.

Puede ser útil buscar entre las estructuras de meta-datos para encontrar una que tenga unidades de datos asignadas.



Orden de asignación de las estructuras de datos

Si necesitamos saber los tiempos relativos de asignación entre dos entradas, debemos ser capaces de comprender la estrategia de asignación del SSOO para determinarlo. Esto es muy dependiente de cada SSOO y bastante difícil. Las entradas de meta-datos son normalmente asignadas usando la estrategia de la primera disponible o la siguiente disponible.

Pruebas de consistencia

Una prueba de consistencia con los meta-datos puede revelar intentos de esconder datos o puede mostrarnos que la imagen del sistema de ficheros tiene algunos errores internos que nos impedirán ver información precisa. Los únicos datos con los que podemos obtener conclusiones en una prueba de consistencia son con los datos esenciales, que incluyen las direcciones de unidad de datos, el tamaño y el estado de asignación de cada entrada de meta-datos.

Una prueba que puede realizarse consiste en examinar cada entrada asignada y verificar que las unidades de datos que tiene asignadas se encuentran en estado asignado. Esto puede verificar que el número de unidades de datos asignadas es consistente con el tamaño del fichero. La mayoría de los sistemas de ficheros no asignan más unidades de datos de las necesarias.

Además podemos verificar que las entradas para los tipos especiales de ficheros no tienen unidades de datos asignadas a ellos. Por ejemplo, algunos sistemas de ficheros tienen ficheros especiales llamados sockets que son usados para procesar comunicaciones con otro usuario y no pueden alojar unidades de datos.

Otra prueba de consistencia usa información de los datos de la categoría de nombre de fichero y verifica que cada entrada de directorio asignada tiene un nombre asignado que apunta a él.

Técnicas de limpieza

Los meta-datos pueden ser limpiados cuando un fichero es borrado para hacer más difícil recuperar ficheros. Los tiempos, tamaño y direcciones de unidades de datos pueden limpiarse con ceros o datos aleatorios. Un investigador puede detectar una limpieza encontrado una entrada llena de ceros u otros datos inválidos si comparamos con una entrada válida. Una herramienta de limpieza más inteligente llenaría los valores con datos válidos pero que no tuvieran correlación con el fichero original.

3.5.4.5 CATEGORÍA NOMBRE DE FICHERO

Esta categoría incluye los nombres de ficheros, que permiten al usuario referirse a un fichero por su nombre en vez de por su dirección de meta-datos. En esencia, esta categoría de datos incluye sólo el nombre de un archivo y su dirección de metadatos. Algunos sistemas de archivo también pueden incluir información de tipo del archivo o información temporal, pero eso no es estándar

Una parte importante del análisis de nombre de fichero es determinar donde se encuentra alojado el directorio raíz, ya que lo necesitamos para encontrar un fichero si nos dan la ruta completa. El directorio raíz es el directorio base del cual cuelgan todos los demás directorios. Por ejemplo, en Windows “C:\” es el directorio raíz de la unidad C:. Cada sistema de ficheros tiene su propia forma de definir la localización del directorio raíz.

Información General

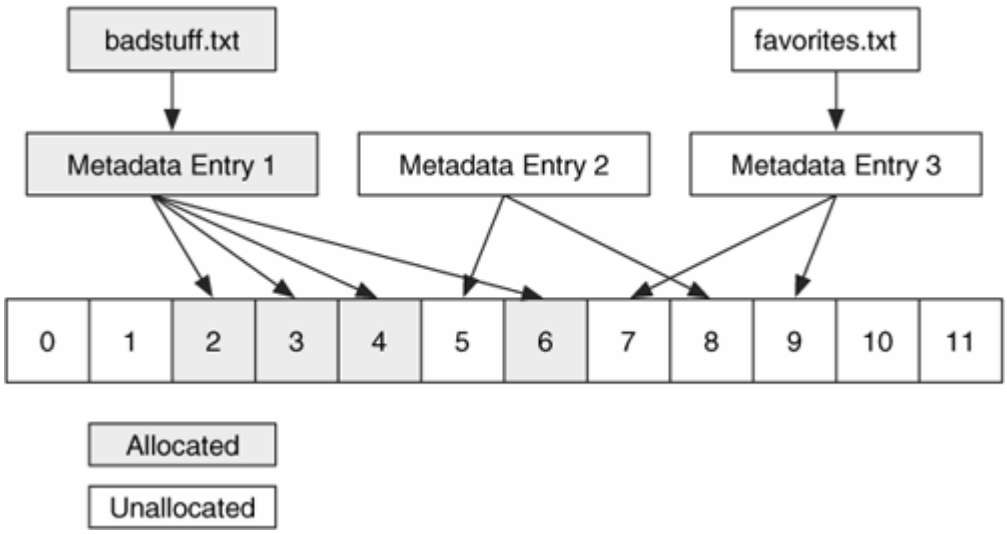
En esta sección, vamos a ver los conceptos generales de la categoría de nombre de fichero. Esta categoría es relativamente simple y necesitamos ver solamente la recuperación de ficheros basada en el nombre de fichero.

Recuperación de ficheros basada en el Nombre del Fichero

Anteriormente vimos en la categoría de Meta-datos que los ficheros borrados pueden recuperarse usando sus meta-datos. Ahora usaremos el nombre del fichero borrado y sus correspondientes direcciones de meta-datos para recuperar el contenido del fichero usando recuperación basada en meta-datos. En otras palabras, la parte difícil se hace en la capa de meta-datos, y todo lo que tenemos que hacer en esta capa es identificar las entradas de meta-datos en las cuales enfocar nuestra atención.

Podemos verlo en la figura 8.17 conde tenemos dos nombres de ficheros y tres entradas de meta-datos. El fichero favorites.txt esta borrado, y su nombre apunta a una entrada de meta-datos no asignada. Podemos intentar recuperar el contenido de los meta-datos usando técnicas de recuperación basadas en meta-datos. Nótese que el contenido de la entrada de meta-datos 2 también puede ser recuperado, aunque no tenga un nombre.

Podemos recuperar ficheros basándonos en su nombre, pero aun así se usarán técnicas de meta-datos para la recuperación.



Para finalizar, si examinamos ficheros borrados desde la perspectiva de nombre de fichero, debemos tener en cuenta que los meta-datos y las unidades de datos podrían haber sido reasignadas a otro fichero. Además recordar que se necesita examinar las estructuras de meta-datos no asignadas para encontrar las que no tienen nombres apuntando hacia ellas.

Técnicas de Análisis

En esta sección veremos las técnicas de análisis que pueden realizarse con los datos de la categoría de nombre de fichero.

Listado de nombre de fichero

El propósito de la categoría de nombre de fichero es asignar nombres a los ficheros. Por tanto, no debe sorprendernos que una de las técnicas de investigación más comunes sea listar los nombres de los ficheros y directorios. Haremos esto cuando busquemos evidencias basándonos en el nombre, la ruta o la extensión de un fichero. Después de que un fichero ha sido reconocido, podemos usar su dirección de metadatos para obtener más información. Variaciones en esta técnica ordenarán los ficheros basándose en sus extensiones de modo que los ficheros del mismo tipo son agrupados.

Muchos sistemas de ficheros no limpian el nombre del fichero de un fichero borrado, de modo que los nombres de ficheros borrados pueden ser mostrados en el listado. Sin embargo, en algunos casos, la dirección de meta-datos es borrada cuando un fichero es borrado, y puede que no seamos capaces de obtener más información.

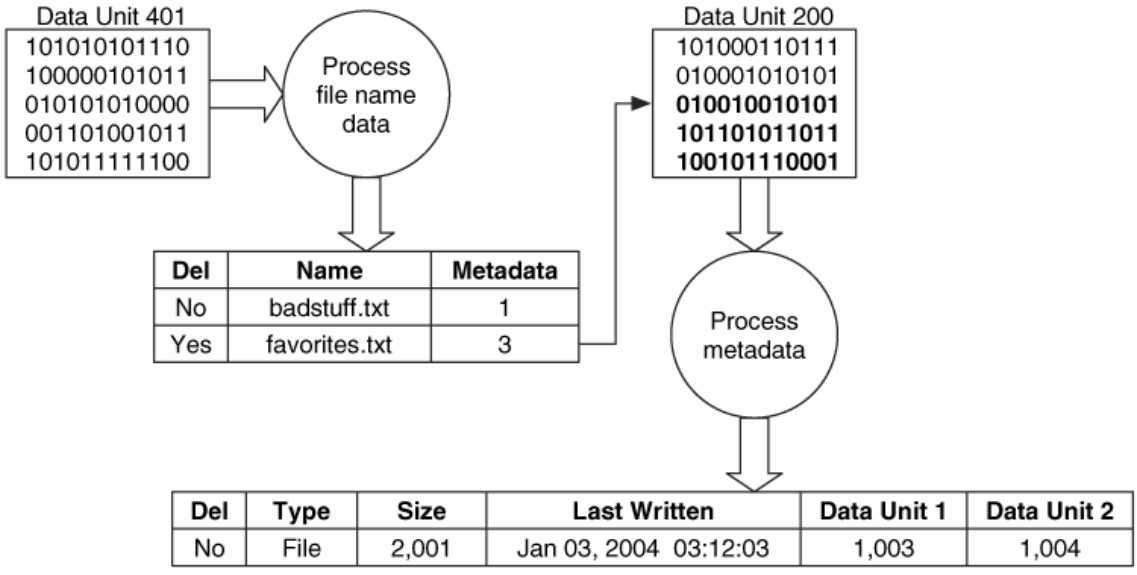
El principio de esta técnica es buscar el directorio raíz del sistema de ficheros. Este proceso es normalmente el mismo que el que se vio para la técnica de visionado lógico

de ficheros en la categoría meta-datos. El diseño del directorio raíz se almacena en una entrada de meta-datos, y necesitamos encontrar la entrada y las unidades de datos que el directorio tiene asignadas.

Después de localizar los contenidos del directorio, los procesaremos y obtendremos una lista de ficheros y sus correspondientes direcciones de meta-datos. Si un usuario quiere ver el contenido de un fichero de los que se lista, puede usar la técnica del visionado lógico de ficheros usando la dirección de meta-datos. Si un usuario quiere listar los contenidos de un directorio diferente, deberá cargar y procesar los contenidos del directorio. En otro caso, este proceso se basa en el visionado lógico de ficheros.

La mayoría de las herramientas de análisis ofrecen esta técnica, y muchas combinan los datos de la categoría de nombre de fichero con los datos de la categoría meta-datos, de modo que podamos ver, por ejemplo, las fechas y horas asociadas con el nombre del fichero en una vista. La figura 8.20 muestra un ejemplo de este proceso de análisis donde procesamos la unidad de datos 401 y encontramos dos nombres. Nosotros estamos interesados en el fichero "favorites.txt" y nos percatamos de que su meta-datos es la entrada 3. Nuestro sistema de ficheros almacena que la estructura de meta-datos 3 apunta a la unidad de datos 200, de modo que procesaremos los datos relevantes de la unidad de datos y obtendremos el tamaño y las direcciones del contenido del fichero.

Relación entre nombres de fichero y metadatos.



Búsqueda de nombre de fichero

El listado de nombres de ficheros trabaja bien si sabemos el fichero que estamos buscando, pero ese no es siempre el caso. Si no sabemos el nombre completo del fichero, podemos buscar las partes que conocemos. Por ejemplo, podemos saber la extensión, o podemos saber el nombre del fichero, pero no la ruta completa. Una búsqueda mostrará una serie de ficheros que cumplen con un patrón de búsqueda. En la

figura 8.20, si hiciéramos una búsqueda por un fichero con extensión ".txt", la herramienta examinaría cada entrada y devolvería "badstuff.txt" y "favorites.txt". Nótese que buscando por la extensión no necesariamente se devuelven ficheros de un cierto tipo ya que la extensión puede haber sido cambiada a propósito para esconder el fichero. Las técnicas de análisis a nivel de aplicación que dependen del nombre de la estructura del fichero pueden usarse para encontrar todos los ficheros de un cierto tipo.

El proceso requerido para buscar un nombre es igual que el que vimos para el listado de nombres de fichero: cargar y procesar los contenidos de un directorio. Comparamos cada entrada del directorio con el patrón objetivo. Cuando encontramos un directorio, debemos buscar dentro de él si hacemos una búsqueda recursiva.

Otra búsqueda de esta categoría es buscar el nombre del fichero que tiene asignada una cierta entrada de meta-datos. Esto es necesario cuando encontramos evidencias en una unidad de datos y luego buscamos la estructura de meta-datos que tiene asociada.

Pruebas de consistencia

Las pruebas de consistencia para esta categoría verifican que todos los nombres asignados apuntan a una estructura de metadatos con estado asignado. Esto es válido para algunos sistemas de ficheros que tienen múltiples nombres de ficheros para el mismo fichero, y muchos de ellos implementan esta funcionalidad teniendo más de una entrada de nombre de fichero con la misma dirección de meta-datos.

Técnicas de borrado seguro

Una herramienta de borrado seguro en esta categoría limpia los nombres y direcciones de meta-datos de la estructura. Una técnica de borrado seguro debe escribir sobre los valores en la estructura de nombre de fichero, de modo que un análisis mostrará que existió una entrada pero los datos ya no son válidos. Por ejemplo, el nombre de fichero "setplog.txt" podría ser reemplazado por "abcdefgh.123". Con algunos SSOO, esto es difícil, ya que el SSOO ubicará el nuevo nombre al final de una lista, usando una estrategia del siguiente disponible.

Otra técnica de limpieza de nombres de fichero es reorganizar la lista de nombres de modo que uno de los nombres de fichero existentes sobrescribe el nombre de fichero borrado. Esto es mucho más complejo que el primer método y mucho más efectivo que una técnica de ocultamiento porque el investigador nunca sabrá que hay algo fuera de lo normal en ese directorio.

3.5.4.6 CATEGORÍA APLICACIÓN

Algunos sistemas de ficheros contienen datos que pertenecen a la categoría aplicación. Estos datos no son esenciales para el sistema de ficheros, y normalmente existen como datos especiales del sistema de ficheros en lugar de un fichero normal ya

que es más eficiente. Esta sección cubre una de las características más comunes de la categoría de aplicación, llamada “journaling” o bitácora.

Técnicamente, cualquier fichero que un SSOO o aplicación crea, podría ser designado como una característica en un sistema de ficheros. Por ejemplo, Acme Software podría decidir que sus SSOO deberían ser más rápidos si un área del sistema de ficheros es reservada como un libro para anotar direcciones. En lugar de salvar nombres y direcciones en un fichero, deberían salvarse en una sección especial del volumen. Esto puede producir una mejora en el rendimiento, pero no es esencial para el sistema de ficheros.

Journals del sistema de ficheros

Como cualquier usuario de computadoras sabe, no es inusual para una computadora pararse y quedarse colgada. Si el SSOO estaba escribiendo datos en el disco o si estaba esperando para escribir algunos datos al disco cuando la computadora se colgó, es probable que el sistema de ficheros esté en un estado inconsistente. Podría haber una estructura de meta-datos con unidades de datos asignadas, pero sin punteros entre ellos ni nombre de fichero apuntando a la estructura de meta-datos.

Para encontrar las inconsistencias, un SSOO arranca un programa que escanea el sistema de ficheros y busca punteros perdidos y otros signos de corrupción. Esto puede tomar un buen rato para sistemas de ficheros grandes. Para hacer el trabajo del programa de escaneo más fácil, algunos sistemas de ficheros implementan un journal. Antes de que ninguna estructura de meta-datos cambie en el sistema de ficheros, se hace una entrada en el journal que describe el cambio que ocurrirá. Después de que se hagan los cambios, se incluye otra entrada en el journal para indicar que los cambios se han producido con éxito. Si el sistema se cuelga, el programa de escaneo lee el journal y localiza las entradas que no fueron completadas. Más tarde el programa completa los cambios o los deshace a su estado original.

Muchos sistemas de ficheros ahora soportan journaling ya que ahorran tiempo son sistemas grandes. El journal está en la categoría de aplicación porque no es necesario para el sistema operativo para funcionar. Existe para hacer más rápidas las pruebas de consistencia.

Los journals de sistemas de ficheros pueden sernos útiles en investigaciones, aunque hasta hoy no son completamente utilizados. Un journal muestra qué eventos del sistema de ficheros han ocurrido recientemente, y esto podría ayudar con la reconstrucción de eventos de un incidente reciente. La mayoría de las herramientas forenses no procesan los contenidos del journal de un sistema de ficheros.

3.5.4.7 TÉCNICAS DE BÚSQUEDA A NIVEL DE APLICACIÓN

En esta sección se discutirán las distintas técnicas que nos permitirán recuperar ficheros borrados y organizar ficheros no borrados para su análisis. Estas técnicas son independientes del sistema de ficheros.

Ambas de esas técnicas se apoyan en el hecho de que muchos ficheros tienen estructura para ellos, incluyendo un valor "firma" que es único para ese tipo de fichero. La firma puede usarse para determinar el tipo de un fichero desconocido.

Recuperación de ficheros basada en aplicación (Data Carving)

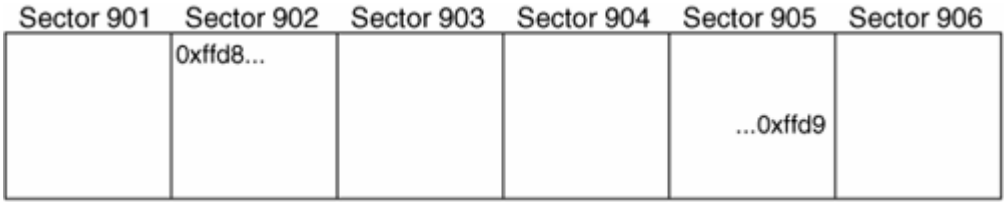
Este es un proceso donde una porción de datos es examinada para buscar firmas que correspondan al inicio o final de tipos de ficheros conocidos. El resultado de este proceso de análisis es una colección de ficheros que contienen una de las firmas. Esto se realiza normalmente en el espacio no asignado de un sistema de ficheros y permite al investigador recuperar que no tienen estructura de meta-datos apuntando a ellos. Por ejemplo, una imagen JPEG tiene unos valores estándar para la cabecera y la cola. Un investigador puede querer recuperar imágenes borradas, por lo que debería usar una herramienta que busque las cabeceras JPEG en el espacio no asignado y que extraiga el contenido que comprende desde la cabecera hasta la cola del fichero encontrado.

Una herramienta de ejemplo que realiza esto es FOREMOST [22] que ha sido desarrollada por los agentes especiales Kris Kendall y Jesse Kornblum de la Oficina de Investigaciones especiales de las fuerzas aéreas de los Estados Unidos. Foremost analiza un sistema de ficheros en bruto o una imagen de disco basada en los contenidos de un fichero de configuración, que tiene una entrada para cada firma. La firma contiene el valor de cabecera conocido, el tamaño máximo del fichero, la extensión típica del fichero, si existe sensibilidad a las mayúsculas y minúsculas y un valor opcional de cola. Un ejemplo puede verse aquí para un JPEG:

```
jpg     y       200000  \xff\xd8            \xff\xd9
```

Esto muestra que la extensión típica es "jpg", que la cabecera y la cola son sensibles a las mayúsculas y minúsculas, que la cabecera es 0xffd8 (valor hexadecimal) y que la cola es 0xffd9. El tamaño máximo del fichero es 200.000 bytes Si no se encuentra la cabecera tras leer todos los datos, se parará para ese fichero. En la figura 8.21 podemos ver un conjunto de datos de ejemplo donde la cabecera JPEG es encontrada en los dos primeros bytes del sector 902 y el valor cola es encontrado en el medio del sector 905. Los contenidos de los sectores 902, 903, 904 y el inicio del sector 905 serían extraídos como una imagen JPEG.

Bloques de datos en bruto en los que encontramos una imagen JPEG mediante su cabecera y su cola



Una herramienta similar es LAZARUS (disponible en The Coroner's Toolkit [31] en realizada por Dan Farmer, la cual examina cada sector de una imagen de datos en bruto y ejecuta el comando file sobre ella. Se crean grupos de sectores consecutivos que tienen el mismo tipo. El resultado final es una lista con una entrada para cada sector y su tipo. Esto es básicamente un método de ordenación de las unidades de datos usando sus contenidos. Este es un concepto interesante, pero la implementación es en Perl y puede ser lenta.

Ordenación de tipos de fichero

Los tipos de fichero pueden usarse para organizar los ficheros en un sistema de ficheros. Si la investigación esta buscando un tipo específico de datos, un investigador puede ordenar los ficheros basándose en su estructura. Una técnica de ejemplo podría ser ejecutar el comando file sobre cada fichero y agrupar tipos similares de ficheros. Esto agruparía todas las imágenes y todos los ejecutables en grupos diferentes, por ejemplo. Muchas herramientas forenses tienen esta característica, pero no siempre esta claro si la ordenación está basada en la extensión o en la firma del fichero.

SISTEMAS DE FICHEROS ESPECÍFICOS

Pare tener una referencia, la tabla 8.1 contiene los nombres de las estructuras de datos en cada categoría para los distintos sistemas de ficheros.

Las estructuras de datos en cada categoría de datos

	Sistema de ficheros	**Contenido**	**Metadatos**	**Nombre de fichero**	**Application**
Ext2, Ext3	Superbloque, grupo, descriptor	Bloques, Mapa de bits de bloques	Inodos, Mapa de bits de inodos, atributos extendidos	Entradas de directorio	Journal
FAT	Sector de arranque, FSINFO	Clusters, FAT	Entradas de directorio, FAT	Entradas de directorio	N/A

Las estructuras de datos en cada categoría de datos

	Sistema de ficheros	**Contenido**	**Metadatos**	**Nombre de fichero**	**Applicación**
NTFS	$Boot, $Volume, $AttrDef	Clusters, $Bitmap	$MFT, $MFTMirr, $STANDARD_ INFORMATION, $DATA, $ATTRIBUTE_LI ST, $SECURITY_DE SCRIPTOR	$FILE_NAME, $IDX_ROOT, $IDX_ALLOC ATION, $BITMAP	Cuota de disco, Journal, Change Journal
UFS	Superbloque, grupo, descriptor	Bloques, fragmentos, mapa de bits de bloques, mapa de bits de fragmentos	Inodos, mapa de bits de inodos, atributos extendidos	Entradas de directorio	N/A

Hay otros sistemas de ficheros que no se mencionan, pero que podemos encontrarnos. HFS+ es el sistema de ficheros por defecto de las computadoras Apple. Apple ha publicado las estructuras de datos y detalles del sistema de ficheros. ReiserFS es uno de los sistemas de ficheros Linux y es el que toman por defecto algunas distribuciones como SUSE. El Journaled File System (JFS) para Linux es de IBM, y es otro sistema de ficheros que se usa en los sistemas Linux. Nótese que es diferente del sistema de ficheros JFS que IBM desarrolló para sus sistemas AIX.

3.5.4.9 RESÚMEN

El análisis de un sistema de ficheros se usa para generar la mayoría de las evidencias en las investigaciones digitales actuales. Hemos visto las categorías de datos en un sistema de ficheros, de forma que tengamos una buena base a la hora de investigar un sistema de ficheros. Hemos visto también técnicas de análisis para cada categoría de datos desde un punto de vista académico. La tabla 8.2 muestra un resumen de las técnicas de análisis que pueden usarse basadas en los tipos de datos que estamos buscando.

Los métodos y lugares de búsqueda, dependiendo del tipo de evidencia que se busque

Necesidades del análisis	Categoría de datos	Técnica de búsqueda
Un fichero basado en su nombre, extensión o directorio	Nombre de fichero	Búsqueda de nombre de fichero o listado de contenidos de directorio
Un fichero asignado o no, basado en sus valores de tiempo	Nombre de fichero y meta-datos	Búsqueda de atributos de meta-datos
Un fichero asignado basado en un valor de su contenido	Nombre de fichero(usando meta-datos y contenido)	Búsqueda lógica de sistema de ficheros
Un fichero asignado basado en su valor HASH (MD5,SHA-1)	Nombre de fichero (usando meta-datos y contenido)	Búsqueda lógica de sistema de ficheros con valores hash
Un fichero asignado o una unidad de datos no asignada basados en un valor de su contenido	Nombre de fichero (usando meta-datos y contenido)	Búsqueda lógica de sistema de ficheros con recuperación basada en meta-datos y búsqueda lógica de fichero
Un fichero no asignado basado en su tipo de aplicación	Aplicación y contenido	Recuperación basada en aplicación de unidades de datos no asignadas
Datos no asignados basados en su contenido(y no su tipo de aplicación)	Contenido	Búsqueda lógica de sistema de ficheros

3.5.5 Reconstrucción

Una reconstrucción investigativa nos ayuda a obtener una imagen más completa del delito: que ha pasado, quien causó los eventos, cuando, donde, como y porqué. La evidencia digital es una fuente de información rica y a menudo inexplorada. Puede establecer acciones, posiciones, orígenes, asociaciones, funciones, secuencias y más datos necesarios para una investigación. Los ficheros Log son una fuente particularmente rica de fuente de información sobre conductas, ya que graba muchas acciones. Interpretando correctamente la información de varios ficheros log, es a menudo posible determinar lo que hizo un individuo con un alto grado de detalle.

Las piezas individuales de datos digitales pueden no ser útiles por sí mismas, pero pueden revelar patrones cuando las combinamos. Si una víctima lee su correo a una hora específica o frecuenta una zona particular de internet, una ruptura en este patrón puede ser el indicativo de un evento inusual. Un delincuente puede solo trabajar los fines de semana, en un cierto lugar, o de una única manera. Teniendo esto en cuenta podemos decir que existen tres formas de reconstrucción que deberían realizarse cuando se analizan evidencias para desarrollar una imagen más clara de un delito y ver discrepancias o brechas.

- Análisis Temporal (cuando): ayuda a identificar secuencias y patrones de tiempo en los eventos.
- Análisis Relacional (quien, que y donde): los componentes de un delito, su posicion e interacción.
- Análisis Funcional (como): qué fue posible e imposible.

3.5.5.1 Análisis Temporal

Cuando se investiga un delito, es normalmente deseable conocer la fecha, la hora y la secuencia de eventos. Afortunadamente, además de almacenar, recuperar, manipular y transmitir datos, los ordenadores mantienen muchos registros de tiempo. Por ejemplo, la mayoría de los sistemas operativos están al tanto de la creación, modificación y acceso de ficheros y directorios. Estos “sellos de tiempo” pueden ser muy útiles a la hora de determinar qué ocurrió en la computadora. En una investigación de robo de propiedad intelectual, los sellos de tiempo de los ficheros pueden mostrar cuanto tardó el intruso en localizar la información deseada en un sistema y a qué ficheros accedió. Una mínima cantidad de búsqueda (ficheros accedidos por el intruso), indica que conocía bien el sistema atacado y una gran búsqueda indica menos conocimiento del sistema. En una investigación de pornografía infantil, el sospechoso declara que su esposa puso pornografía en su ordenador personal sin su conocimiento durante su amarga separación para que repercutiera negativamente en la batalla por la custodia de sus hijos. Sin embargo, los sellos de tiempo de los ficheros indican que fueron ubicados en el sistema mientras su enemistada esposa estaba fuera del país visitando a su familia.

También, el ordenador del sospechoso contenía restos de e-mails y otras actividades online, indicando que había usado la computadora en ese tiempo.

Además de los sellos de tiempo, algunas aplicaciones incrustan información de fecha-tiempo dentro de los ficheros o logs creados, así como bases de datos, que pueden mostrar por ejemplo, las páginas Web visitadas.

Cuando se investiga un delito que implica ordenadores, es importante poner especial atención a la fecha/hora en que se cometió, a cualquier discrepancia entre la hora actual y la hora del sistema, la zona horaria del reloj de la computadora y los sellos de tiempo en objetos digitales.

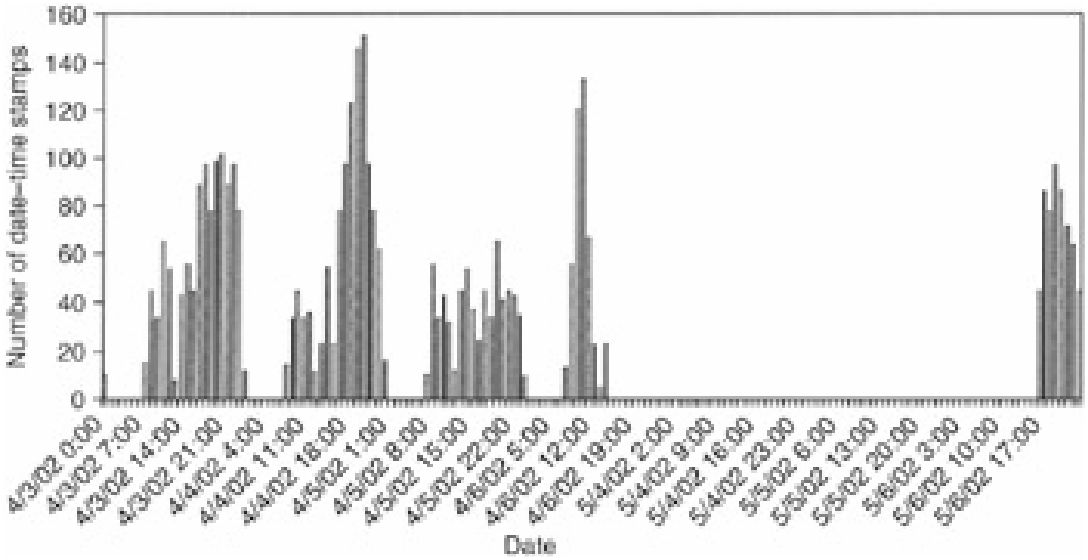
El simple acto de creación de una linea de tiempo de cuando los ficheros implicados fueron creados, modificados y accedidos puede resultar en una sorprendente cantidad de información. Creando una línea de tiempo de eventos puede ayudar al investigador a identificar patrones y brechas, arrojando luz sobre un delito y llevándonos a otras fuentes de evidencia. Por ejemplo, la siguiente figura muestra una linea de tiempo de las actividades de una mujer perdida en los días anteriores a su desaparición como se reconstruye de su computadora. La secuencia cronológica de eventos ayuda al investigador a determinar que la víctima había viajado a Virginia a tener un encuentro masoquista con un hombre que conoció online. Cuando el investigador buscó la casa del hombre, encontraron el cuerpo de la mujer.

Linea de tiempo de actividades sobre la computadora de la víctima, mostrando correspondencia de e-mail, sesiones de chat online, ficheros borrados, búsqueda de mapas en la Web y planos online para viajar.

DATE	**ACTIVITY**
Dia 1	Webs sadomasoquistas (BDSM) visitadas
Dia 2	Correspondencia de Hotmail de naturaleza BDSM con origen desconocido, IP de virginia. Alrededor de la misma hora se ingresa en Hotmail para revisar el correo. Se visitan más páginas BDSM.
Dia 3	Logs de conversaciones de sesiones de chat online muestran una conversación de naturaleza sexual/BDSM con origen desconocido e IP de Virginia.
Dia 4	Direcciones obtenidas desde MapQuest, en dirección a virginia.
Dia 4	Fichero borrado
Dia 4	No hay actividad despues de las 8 P.M.

Representando la información temporal en deferentes formas se pueden extraer patrones. Por ejemplo, la siguiente figura muestra un histograma de sellos de tiempo de una computadora usada por los trabajadores del turno de noche de una compañía. Un empleado es sospechoso de ver material obsceno y posiblemente ilegal durante su turno desde la media noche hasta las 8 A.M., pero las marcas de tiempo de los ficheros

muestran que la actividad se realizó entre las 4 P.M. y la media noche, implicando así a su compañero.



Histograma de las marcas de tiempo (creados y modificados) mostrando una brecha entre los turnos sopechosos (las fechas están en inglés: Mes/Dia/Año)

Las brechas en la figura anterior sugieren que la computadora no fue usada durante el turno del sospechoso, pero es sabido que el empleado tuvo que hacer uso de los recursos de red para su trabajo (web, email), lo que indica que el sospechoso cambiaba regularmente el reloj del sistema al inicio de su turno a 8 horas antes. Resulta interesante que en una de esas veces cambió accidentalmente el mes además de la hora, por lo que se produjo actividad que data del mes de Mayo, despues de 1600 horas. Esto apoya la hipótesis de que cambió la hora durante sus turnos. Además, se realizaba una copia de seguridad por la central todas las noches a las 2 A.M., que aparecían en el Windows NT Application Event Log como 8 horas antes, apoyando de nuevo la teoría de que el reloj había sido alterado.

Otra aproximación al análisis de la información de las marcas de tiempo es usando una tabla para acentuar patrones de cuando ocurrió un evento. La siguiente tabla muestra los e-mails enviados por el jefe de un grupo criminal durante muchos meses a otros miembros del grupo. Las comunicaciones sobre un plan criminal comenzaron a mediados de Junio, cesando a comienzos de Julio y retomándolas como fecha tope el 11 de Septiembre.

Tabla que muestra los mensjaes de e-mail enviados por un sospechoso durante muchos meses a otros miembros de un grupo criminal.

Direcciones de email	**Dom, 16 Jun**	**Vie, 21 Jun**	**Dom, 23 Jun**	**Mie, 26 Jun**	**Sab, 29 Jun**	**Dom, 30 Jun**	**Jue, 11 Jul**	**Vie, 26 Jul**	**Lun, 29 Jul**	**Vie, 2 Ago**	**Mie, 14 Ago**	**Jue, 15 Ago**	**Jue, 29 Ago**	**Dom, 8 Sep**	**Mie, 11 Sep**
Miembro1	Xx				x	x							xxx	xx	X
Miembro2	Xx		x	x			x		x		x	x	x	x	x
Miembro3	Xx	x	x	x			x	X		xxx			x	x	x

3.5.5.2 Análisis Relacional

En un esfuerzo para identificar relaciones entre sospechosos, la victima y la escena del crimen, puede ser útil crear un grafo con nodos que representan lugares en los que se ha estado o acciones que se han realizado, como IPs, e-mails, transacciones financieras, números de teléfono marcados, etc., y determinar si hay conexiones destacables entre esos nodos. Por ejemplo, en una investigación de fraude a gran escala, representando transferencias de fondos dibujando linas entre individuos y organizaciones se puede revelar la mayor parte de la actividad en el fraude. Igualmente, trazando los mensajes de e-mail enviados y recibidos por un sospechoso podría ayudar a desvelar a supuestos cómplices por el gran número de mensajes intercambiados.

Es posible que con tanta información parezca que nada esta conectado. Los investigadores deben decidir cuanto peso asignar a las relaciones que encuentren. Estas reconstrucciones dan mejores resultados en diagramas con pocas entidades. A medida que se incrementan las entidades y relaciones se incrementa la dificultad de identificar las conexiones importantes. Para facilitar esta tarea existen herramientas que ofrecen la posibilidad de realizar diagramas y asignar pesos a cada conexión. Además se están desarrollando otras herramientas que permiten trabajar con muchas entidades usando algoritmos sofisticados.

3.5.5.3 Análisis Funcional

Cuando se reconstruye un delito, a menudo es útil considerar qué condiciones fueron necesarias para hacer que ciertos aspectos del delito fueran posibles. Por ejemplo, a menudo es útil testear el hardware original para asegurarnos de que el sistema fue capaz de realizar algunas acciones básicas, como chequear la capacidad de una unidad de disquetes para leer/escribir si tenemos un disquete con evidencias.

En una investigación hay varios propósitos para evaluar como funcionaba un sistema computacional:

- Para determinar si el individuo o la computadora tenían capacidad para cometer el delito.
- Para ganar un mejor entendimiento de una parte de la evidencia digital o del delito en general.
- Para probar que la evidencia digital fue manipulada indebidamente.
- Para comprender los motivos e intenciones del agresor. Por ejemplo, si fue algo accidental o premeditado.
- Para determinar el funcionamiento del sistema durante el lapso de tiempo pertinente.

Debemos tener en mente que el propósito de la reconstrucción funcional es considerar todas las posibles explicaciones para un determinado conjunto de circunstancias, y no simplemente responder a la cuestión que se plantea.

Puede ser necesario determinar como un programa o computador estaba configurado para ganar un mejor entendimiento de un delito o una parte de una evidencia digital. Por ejemplo, si se requiere un password para acceder a cierta computadora o programa, este detalle funcional debería ser anotado. Conociendo que un cliente de e-mail estaba configurado para chequear automáticamente el correo en busca de mensajes nuevos cada 15 minutos, puede ayudar a los investigadores a diferenciar actos humanos de actos automáticos.

3.5.6 Publicación de conclusiones

La última fase de un análisis de evidencias digitales es integrar todo el conocimiento y conclusiones en un informe final que de a conocer los descubrimientos a otros y que el examinador puede tener que presentar en un juicio. La escritura de un informe es una de las fases más importantes del proceso, ya que es la única presentación visual que otros tendrán sobre el proceso entero. A menos que los descubrimientos sean comunicados claramente en el informe, es improbable que otros aprecien su significancia. Un buen informe que describe claramente los descubrimientos del examinador puede convencer a la oposición a llegar a un acuerdo en un juicio, mientras que un informe pobre puede animar a la oposición para ir a juicio. Las suposiciones y la falta de fundamentos resultan en un informe débil. Por tanto, es importante construir argumentos sólidos suministrando todas las evidencias encontradas y demostrando que la explicación proporcionada es la más razonable.

Mientras sea posible, respaldar las suposiciones con múltiples fuentes independientes de evidencia e incluyendo todas las pruebas relevantes junto con el informe ya que puede ser necesario en un juicio hacer referencia a las mismas cuando se explican los descubrimientos en el informe. Establecer claramente como y donde se encontró toda la evidencia para ayudar a los que tomarán las decisiones a interpretar el informe y permitir a otros examinadores competentes a verificar resultados. Presentando escenarios alternativos y demostrando porqué son menos razonables y menos consistentes con respecto a la evidencia puede ayudar a reforzar las conclusiones clave. Explicando porqué otras explicaciones son improbables o imposibles demuestra que el método científico fúe aplicado, es decir, que se hizo un esfuerzo para desmentir la conclusión alcanzada por el examinador, pero que esta resistió un escrutinio crítico. Si la evidencia digital fue alterada después de recopilarla, es crucial mencionar esto en el informe, explicando la causa de las alteraciones y sopesando su impacto en el caso (por ejemplo, insignificante, servero).

A continuación se muestra una estructura simple para un informe:

- Introducción: Número de caso, quien requirió el informe y qué se buscaba, quien escribió el informe, cuando y que se encontró.

- Resumen de la evidencia: resumir qué evidencias se examinaron y cuando, valores MD5, cuando y donde se obtuvo la evidencia, de quién y su condición (anotar signos de daño o sabotaje)

- Resumen del análisis: resumir las herramientas usadas para realizar el análisis, cómo se recuperaron los datos importantes (por ej: si se desencriptaron, o se recuperaron ficheros borrados) y como se descartaron ficheros irrelevantes.

- Análisis del sistema de ficheros: inventario de ficheros importantes, directorios y datos recuperados que son relevantes para la investigación con características importantes como nombres de ruta, marcas de tiempo, valores MD5, y

localización física de los sectores en el disco. Nótese cualquier ausencia inusual de datos.

- Análisis/Reconstrucción: describir e interpretar el proceso de análisis temporal, relacional y funcional.

- Conclusiones: el resumen de conclusiones debería seguir a las secciones previas en el informe y debería hacer referencia a la evidencia hallada y a la imagen reconstruida a partir de ellas.

- Glosario de Términos: explicaciones de términos técnicos usados en el informe

- Apéndice de soporte: la evidencia digital usada para alcanzar las conclusiones, claramente numerada para su referencia.

Además de presentar lo hechos en un caso, los investigadores digitales generalmente interpretan la evidencia digital en el informe final. La interpretación implica opinión, y cada opinión suministrada por un investigador tiene una base estadística. Por tanto en un informe el investigador debe indicar claramente el nivel de certeza que tiene cada conclusión y cada parte de la evidencia para ayudar en un juicio a darles un peso a cada una. La Escala de Certeza de Casey(C-Scale, Casey Certainly Scale) proporciona un método para transmitir certeza cuando nos refereimos a una evidencia digital en un contexto dado y cualificar las conclusiones apropiadamente.

NIVEL DE CERTEZA	**INDICADORES/ DESCRIPCIÓN**	**CALIFICACIÓN**	**EJEMPLOS**
C0	La evidencia contradice los hechos conocidos	Erróneo/incorrecto	El examinador encuentra una vulnerabilidad en Internet Explorer que permitió a una Web particular el crear ficheros, accesos directos y favoritos. El sospechoso no creó esos elementos en el sistema a propósito.
C1	La evidencia es altamente cuestionable	Altamente incierto	Entradas perdidas en un log o signos de manipulación
C2	Solo una fuente de evidencia que no esta protegida contra manipulaciones	Algo de incerteza	Cabeceras de E-mail, entradas de logs del sistema sin otra evidencia que los apoye
C3	La/s fuente/s de evidencia son más difíciles de manipular pero no hay evidencias suficientes para apoyar una conclusión firme o hay inconsistencias	Posible	Una intrusión viene de Polonia sugiriendo que el intruso puede ser de ese área. Sin embargo, mas tarde se detecta una intrusión de Korea del

NIVEL DE CERTEZA	INDICADORES/ DESCRIPCIÓN	CALIFICACIÓN	EJEMPLOS
	inexplicables en la evidencia disponible		Sur, sugiriendo que el intruso puede ser de otra parte o que haya más de un intruso
C4	(a) La evidencia está protegida contra manipulación o (b) la evidencia no está protegida contra manipulación pero hay múltiples fuentes independientes de evidencias coincidentes	Probable	En el ejemplo anterior, encontramos varios ficheros log borrados que indican que existen dos intrusos.
C5	Concondancia entre evidencias de múltiples fuentes independientes que están protegidas contra manipulación. Sin embargo, existen pequeñas incertidumbres (por ej: error temporal, datos perdidos)	Algo de certeza	Direcciones IP y cuentas de usuario apuntan a la casa del sospechoso. Monitorizando el tráfico de internet indica que la actividad criminal proviene de esa casa.
C6	La evidencia es resistente a manipulaciones e incuestionable	Certeza	Aunque esto es inconcebible por el momento, tales fuentes de evidencia digital pueden existir en el futuro

Algunos investigadores usan un sistema menos formal de grados de probablidad que pueden ser usados en sentido afirmativo o negativo:

1) Casi definitivo

2) Muy probable

3) Probable

4) Muy posible

5) Posible

Cuando se determina el nivel de certeza de una cierta parte de evidencia digital puede ser importante considerar el contexto. Por ejemplo, muchos ordenadores Macintosh no requieren contraseña y permiten a cualquier usuario cambiar el reloj del sistema, haciendo más difícil para un investigador tener confianza en las marcas de tiempo y atribuir actividades a un individuo.

Por último, además de presentar un buen informe técnico final, se puede exigir a un investigador que elabore un informe para un responsable con menos conocimientos técnicos. Por ejemplo, el director de una empresa puede necesitar saber lo que ha pasado en pocas frases para determinar las acciones a tomar. El departamento de relaciones públicas puede necesitar detalles para transmitirlos a los accionistas. Los abogados pueden necesitar un informe resumen para ayudarles a concentrarse en los aspectos clave del caso. Se necesita creatividad y trabajo duro para crear representaciones claras y no técnicas de aspectos importantes en un caso como lineas de tiempo, reconstrucciones relacionales y análisis funcionales. Sin embargo, el esfuerzo requerido para generar esas representaciones es necesario para dar a los abogados, directivos, etc., la mejor visión posible de los hechos.

4. ASPECTOS LEGALES

4.1 Legislación internacional

En el contexto internacional, son pocos los países que cuentan con una legislación apropiada. Entre ellos, se destacan, Estados Unidos, Alemania, Austria, Gran Bretaña, Holanda, Francia, España, Argentina y Chile.

Dado lo anterior a continuación se mencionan algunos aspectos relacionados con la ley en los diferentes países, así como con los delitos informáticos que persigue.

» Estados Unidos.

Este país adoptó en 1994 el Acta Federal de Abuso Computacional que modificó al Acta de Fraude y Abuso Computacional de 1986.

Con la finalidad de eliminar los argumentos hipertécnicos acerca de qué es y que no es un virus, un gusano, un caballo de Troya y en que difieren de los virus, la nueva acta proscribe la transmisión de un programa, información, códigos o comandos que causan daños a la computadora, a los sistemas informáticos, a las redes, información, datos o programas. La nueva ley es un adelanto porque está directamente en contra de los actos de transmisión de virus.

Asimismo, en materia de estafas electrónicas, defraudaciones y otros actos dolosos relacionados con los dispositivos de acceso a sistemas informáticos, la legislación estadounidense sanciona con pena de prisión y multa, a la persona que defraude a otro mediante la utilización de una computadora o red informática.

En el mes de Julio del año 2000, el Senado y la Cámara de Representantes de este país -tras un año largo de deliberaciones- establece el Acta de Firmas Electrónicas en el Comercio Global y Nacional. La ley sobre la firma digital res ponde a la necesidad de dar validez a documentos informáticos -mensajes electrónicos y contratos establecidos mediante Internet- entre empresas (para el B2B) y entre empresas y consumidores (para el B2C).

» Alemania.

Este país sancionó en 1986 la Ley contra la Criminalidad Económica, que contempla los siguientes delitos:

- * Espionaje de datos.
- * Estafa informática.
- * Alteración de datos.
- * Sabotaje informático.

» Austria.

La Ley de reforma del Código Penal, sancionada el 22 de Diciembre de 1987, sanciona a aquellos que con dolo causen un perjuicio patrimonial a un tercero influyendo en el resultado de una elaboración de datos automática a través de la confección del programa, por la introducción, cancelación o alteración de datos o por actuar sobre el curso del procesamiento de datos. Además contempla sanciones para quienes comenten este hecho utilizando su profesión de especialistas en sistemas.

» Gran Bretaña.

Debido a un caso de hacking en 1991, comenzó a regir en este país la Computer Misuse Act (Ley de Abusos Informáticos). Mediante esta ley el intento, exitoso o no, de alterar datos informáticos es penado con hasta cinco años de prisión o multas. Esta ley tiene un apartado que específica la modificación de datos sin autorización.

» Holanda.

El 1º de Marzo de 1993 entró en vigencia la Ley de Delitos Informáticos, en la cual se penaliza los siguientes delitos:

- El hacking.
- El preacking (utilización de servicios de telecomunicaciones evitando el pago total o parcial de dicho servicio).
- La ingeniería social (arte de convencer a la gente de entregar información que en circunstancias normales no entregaría).
- La distribución de virus.

» Francia.

En enero de 1988, este país dictó la Ley relativa al fraude informático, en la que se consideran aspectos como:

- Intromisión fraudulenta que suprima o modifique datos.
- Conducta intencional en la violación de derechos a terceros que haya impedido o alterado el funcionamiento de un sistema de procesamiento automatizado de datos.
- Conducta intencional en la violación de derechos a terceros, en forma directa o indirecta, en la introducción de datos en un sistema de procesamiento automatizado o la supresión o modificación de los datos que éste contiene, o sus modos de procesamiento o de transmisión.
- Supresión o modificación de datos contenidos en el sistema, o bien en la alteración del funcionamiento del sistema (sabotaje).

» **Chile**.

Chile fue el primer país latinoamericano en sancionar una Ley contra delitos informáticos, la cual entró en vigencia el 7 de junio de 1993. Esta ley se refiere a los siguientes delitos:

- La destrucción o inutilización de los de los datos contenidos dentro de una computadora es castigada con penas de prisión. Asimismo, dentro de esas consideraciones se encuentran los virus.

- Conducta maliciosa tendiente a la destrucción o inutilización de un sistema de tratamiento de información o de sus partes componentes o que dicha conducta impida, obstaculice o modifique su funcionamiento.

- Conducta maliciosa que altere, dañe o destruya los datos contenidos en un sistema de tratamiento de información.

4.2 Legislación nacional (España)

En [8] podemos ver lo que en España se entendiende por Delito Informático:

Ataques que se producen contra el derecho a la intimidad.
Delito de descubrimiento y revelación de secretos mediante el apoderamiento y difusión de datos reservados registrados en ficheros o soportes informáticos. (Artículos del 197 al 201 del Código Penal)

Infracciones a la Propiedad Intelectual a través de la protección de los derechos de autor.
Especialmente la copia y distribución no autorizada de programas de ordenador y tenencia de medios para suprimir los dispositivos utilizados para proteger dichos programas. (Artículos 270 y otros del Código Penal)

Falsedades.
Concepto de documento como todo soporte material que exprese o incorpore datos. Extensión de la falsificación de moneda a las tarjetas de débito y crédito. Fabricación o tenencia de programas de ordenador para la comisión de delitos de falsedad.(Artículos 386 y ss. del Código Penal)

Sabotajes informáticos.
Delito de daños mediante la destrucción o alteración de datos, programas o documentos electrónicos contenidos en redes o sistemas informáticos. (Artículo 263 y otros del Código Penal)

Fraudes informáticos.
Delitos de estafa a través de la manipulación de datos o programas para la obtención de un lucro ilícito. (Artículos 248 y ss. del Código Penal)

Amenazas.
Realizadas por cualquier medio de comunicación. (Artículos 169 y ss. del Código Penal)

Calumnias e injurias.
Cuando se propaguen por cualquier medio de eficacia semejante a la imprenta o la radiodifusión. (Artículos 205 y ss. del Código Penal)

Pornografía infantil.
Entre los delitos relativos a la prostitución al utilizar a menores o incapaces con fines exhibicionistas o pornográficos.

- **La inducción, promoción, favorecimiento o facilitamiento de la prostitución de una persona menor de edad o incapaz. (art 187)**

- **La producción, venta, distribución, exhibición, por cualquier medio, de material pornográfico en cuya elaboración hayan sido utilizados menores de edad o incapaces, aunque el material tuviere su origen en el extranjero o fuere desconocido. (art 189)**

- **El facilitamiento de las conductas anteriores (*El que facilitare la producción, venta, distribución, exhibición...*). (art 189)**

- **La posesión de dicho material para la realización de dichas conductas.(art 189)**

Legislación en relación con las telecomunicaciones:

CONSTITUCIÓN ESPAÑOLA.

Artículo 149.1.- El Estado tiene la competencia exclusiva sobre:21ª "Régimen General de comunicaciones..., correos y telecomunicaciones, cable aéreos, submarinos y radiocomunicación. Las telecomunicaciones son servicios de interés general y se prestan en régimen de competencia".

Artículo 51.- "Los poderes públicos garantizarán la defensa de los consumidores y usuarios, protegiendo, mediante procedimientos eficaces, la seguridad, la salud y los legítimos intereses económicos de los mismos.

CÓDIGO PENAL.

- Defraudación por valor superior a 50.000 pts, utilizando energía eléctrica, gas, agua, **telecomunicaciones** u otro elemento, energía o fluido ajeno, por alguno de los medios siguientes:

1.- Valiéndose de mecanismo instalados para realizar la defraudación.

2.- Alterando maliciosamente las indicaciones o los aparatos contadores.

3.- Empleando cualesquiera otros medios clandestinos.

(Art 255)

- Hacer uso de cualquier equipo terminal de telecomunicaciones, sin consentimiento de su titular, ocasionando a este un perjuicio superior a 50.000 pesetas.

(Art 256)

5 HERRAMIENTAS

5.1 Evolución de las Herramientas de Investigación

Según [7], en el comienzo de la investigación criminal de computadoras, era común para los investigadores usar el mismo ordenador que estaban examinando para hacer con este la investigación. Un riesgo de esta estrategia era que operando la computadora que contiene las evidencias se podía alterar la evidencia de modo que era indetectable. Aunque programas como dd en UNIX existían en los años 80 y podía usarse para capturar datos borrados almacenados en el disco duro, estas herramientas no eran comúnmente usadas y la mayoría de los análisis en aquel tiempo se realizaban en el nivel de sistema de ficheros, descuidando los datos borrados.

No fue sino hasta el comienzo de los 90, cuando se desarrollaron herramientas como SafeBack y DIBS, que permitían a los investigadores recopilar todos los datos de un disco, sin alterar detalles importantes. Alrededor de esas fechas, se desarrollaron herramientas como aquellas para el IRS de los Estados Unidos (Hacienda), por parte de Maresware y NTI. Éstas ayudaban a los investigadores a procesar datos en un disco de ordenador. La Real Policía Montada de Canadá también desarrolló herramientas especializadas para examinar computadores. Por aquel entonces mucha gente temía por el valor probatorio de las computadoras, y se impuso la necesidad de herramientas más avanzadas. Para satisfacer esta necesidad, se desarrollaron herramientas integradas como FTK para más fácil el trabajo del investigador. Estas herramientas permitían hacer el análisis más eficiente, automatizando rutinas y mostrando datos en una interfaz gráfica para ayudar a encontrar los detalles importantes. Recientemente, se ha renovado el interés de Linux como plataforma de análisis y herramientas como The Sleuth Kit o SMART han sido desarrolladas para proveer de una interfaz amigable al usuario. Herramientas más sofisticadas usan poderosos microscopios y están disponibles para recuperar datos sobrescritos de los discos duros, pero son demasiado caras para la mayoría de los bolsillos

Desafortunadamente, muchas personas aún no son conscientes de la necesidad de estas herramientas. A pesar de que los jueces han sido indulgentes con los investigadores que maltratan las evidencias digitales, esto está cambiando a medida que aumenta el conocimiento sobre esta materia.

Ha habido una progresión similar en la evolución de herramientas para recopilar evidencias en los sistemas de comunicación. A finales de los 80, Clifford Stoll describió como hacía impresiones del tráfico de red en un esfuerzo de preservarlo como evidencia. Las herramientas de monitorización de tráfico como tcpdump y Ethereal pueden usarse para capturar tráfico de red, pero no están diseñadas específicamente para recopilar evidencias digitales. Herramientas comerciales como Carnivore, NetIntercept, NFR Security, NetWitness, y SilentRunner se han desarrollado con búsqueda integrada, visualización y características de análisis para ayudar a los investigadores a extraer información del tráfico de red. Más que apoyarse en las herramientas, las redes requieren el ingenio del investigador para recopilar y analizar las evidencias.

También se ha producido una evolución parecida en las herramientas de recopilación de evidencias de sistemas embebidos. Es común para los investigadores leer datos de teléfonos móviles, buscas, agendas electrónicas, y otros asistentes

personales digitales directamente desde los dispositivos. Sin embargo, esta estrategia no permite el acceso a datos borrados y puede que el análisis no sea posible si el dispositivo está protegido por contraseña o no tiene una forma de mostrar todos los datos que contiene. Por lo tanto, herramientas como ZERT, TULP y Cards4Labs han sido desarrolladas para acceder a datos protegidos por contraseña y a los datos borrados. Otras técnicas más sofisticadas están disponibles implicando a microscopios electrónicos para recuperar datos encriptados de sistemas embebidos, pero son demasiado caras.

Con el paso de los años se han encontrado fallos en varias herramientas de procesamiento de evidencias digitales, que provocan que la evidencia se pierda o se malinterprete. Para evitar los errores judiciales que pueden resultar de estos fallos, es deseable evaluar la fiabilidad de las herramientas usadas comúnmente. El Instituto Nacional de Estándares y Testeo (NIST) está haciendo un esfuerzo para testear algunas de éstas herramientas. Sin embargo, ésta es una tarea ardua, debido a que puede no ser posible crear tests estándar para características avanzadas de varias herramientas, puesto que cada herramienta tiene diferentes características.

En 2002 Brian Carrier propuso reducir la complejidad de los test permitiendo a la gente ver el código fuente de componentes cruciales del software. Facilitando a los programadores de todo el mundo el código fuente, se permitiría a los testeadores de herramientas ganar un mejor entendimiento del programa e incrementaría la probabilidad de encontrar fallos. Es algo reconocido que los desarrolladores de herramientas comerciales querrán mantener algunas partes de sus programas de forma privada para proteger su ventaja en la competencia. Sin embargo, ciertas operaciones como copiar datos de un disco duro son suficientemente comunes y cruciales para requerir un estándar abierto. Últimamente, dada la complejidad de los sistemas de computadoras y las herramientas usadas para examinarlos, no es posible eliminar o cuantificar los errores, la incertidumbre y las pérdidas, de modo que los investigadores deben validar sus propios resultados usando varias herramientas.

5.2 ANÁLISIS DE DISCOS

Herramientas basadas en Linux

5.2.1 LINReS, de NII Consulting Pvt. Ltd.

LINReS [9] es una herramienta de Primera Respuesta diseñada para ejecutarse en sistemas Linux sospechosos/comprometidos, con un mínimo impacto en el mismo para satisfacer varios requerimientos estándar forenses. Esta herramienta ha sido probada con éxito en sistemas RedHat Linux.

LINReS consiste en su mayor parte en binarios compilados estáticamente e incluye las librerías compartidas que pueden requerir los binarios que no están compilados estáticamente. En definitiva, no se usa ningún binario del sistema comprometido, mitigando el riesgo de recopilar información de un sistema con un rootkit, troyano, etc.

Este herramienta sigue un modelo simple cliente-servidor en el que el sistema sospechoso actúa como servidor y la estación de trabajo del investigador (bajo el sistema operativo MS-Windows) actúa como cliente y recibe todos los datos de primera respuesta del sistema sospechoso. La elección de MS-Windows como sistema cliente se debe principalmente a la facilidad del uso de la opción persistente (-L) del comando Netcat.

LINReS contiene tres herramientas diferentes que recogen datos volátiles y no volátiles del sistema sospechoso que satisfacen los requerimientos de la fase “Respuesta Inicial”

Características principales

- Recoge información volátil y no volátil del sistema sospechoso
- Recoge meta-datos de los ficheros del sistema sospechoso
- Calcula los hashes de todos los ficheros del sistema sospechoso
- Transfiere datos a través de la red usando conexiones de Netcat persistentes

- Interacción mínima sobre el sistema sospechoso

- Usa en su mayoría binarios compilados estáticamente

5.2.2 SMART, by ASR Data

SMART [10] es una herramienta software que ha sido diseñada y optimizada para dar soporte a los investigadores forenses y al personal de seguridad informática en la consecución de sus respectivas tareas y metas. El software y metodología de SMART han sido desarrollados con la intención de integrar los requerimientos técnicos, legales y de usuario final en un paquete completo que permite al usuario realizar su trabajo de manera más efectiva y eficiente.

SMART es más que un simple programa forense. Las características de SMART le permiten ser usado en multitud de escenarios, incluyendo:

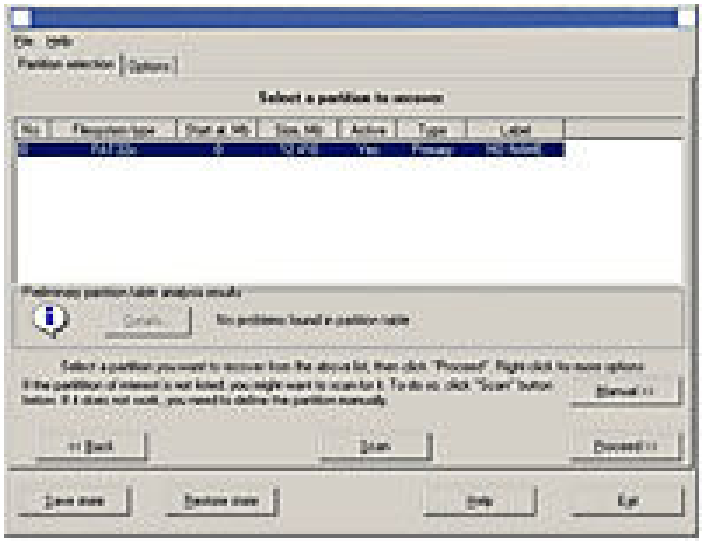
- Investigaciones de "Knock-and-talk"
- Vista previa remota o in-situ de un sistema objetivo
- Análisis post mortem
- Testing y verificación de otros programas forenses
- Conversión de ficheros entre distintos formatos forenses

SMART es usado actualmente por:

- Agentes de la ley Federales, estatales y locales en U.S.A.
- Organizaciones militares y de inteligencia en U.S.A.
- Empresas de contabilidad
- Investigadores forenses
- Especialistas en recuperación de datos
- Profesionales en recuperación de desastres
- Profesionales de la seguridad informática
- Profesionales de la privacidad en la asistencia sanitaria
- Auditores internos
- Administradores de sistemas

Herramientas basadas en Macintosh

5.2.3 Macintosh Forensic Software, de BlackBag Technologies, Inc.



The BlackBag Macintosh Forensic Software (BlackBag MFS) [11] es un conjunto de herramientas independientes que dan al examinador el nivel más alto de flexibilidad permitido en el campo forense. Los examinadores pueden lanzar una o más aplicaciones durante un análisis para obtener la mayor cantidad posible de evidencias de forma eficiente y segura de un sistema de ficheros Macintosh (HFS o HFS+).

- Adquisición de imágenes

- BlackBag MFS soporta varios métodos de adquisición de imagen. Se recomienda usar "dd" dada su flexibilidad y fiabilidad. Los métodos soportados son: dd, iLook, Disk Copy y SafeBack.

- Analizar la imagen

- Un análisis se realiza mejor usando la misma plataforma en la que se encuentra la evidencia original. Por ejemplo, los formatos Macintosh, HFS y HFS+, contienen información ilegible por otros sistemas operativos. Mientras se realiza un análisis sobre un sistema operativos Macintosh, un examinador pueden usar características de seguridad inherentes, como el apagado del automontaje de discos, cerrar un dispositivo, etc., que mantendrán la integridad de los ficheros sospechosos a lo largo del análisis, aunque el examinador ejecute aplicaciones del sistema de ficheros montado.

- BlackBag Macintosh Forensic Software

- La siguiente lista representa las características principales dentro del conjunto de herramientas independientes de BlackBag MFS.

- *Breakup* simplifica la gestión de los directorios que contienen miles de imágenes, reduciendo un gran directorio a tamaños más pequeños y manejables. Reduciendo un gran directorio se puede ayudar en el visionado de ficheros para su análisis o en la selección de ritas de ficheros para crear un CD de evidencias, para copias de seguridad, etc.

- *CommentHunter* proporciona una rápida instantánea de la actividad de un sospechoso, introduciendo todos los comentarios sobre ficheros conocidos en un fichero fácil de leer. Viendo esta instantánea, un examinador se evita la necesidad de abrir miles de ficheros individualmente.

- *DirectoryScan* muestra un listado de directorio de un volumen o directorio seleccionado. El listado puede usarse posteriormente como una guía para llevar a cabo el análisis o en muchos casos, puede contener la información incriminatoria ya que los ficheros visibles e invisibles se muestran en la lista.

- *FileSearcher* permite al examinador la búsqueda en el sistema de ficheros completo de una variedad de características diferentes, incluyendo nombres de fichero (extensiones), tipos de fichero, y codificación del creador.

5.2.4 MacForensicLab, de Subrosasoft

MacForensicsLab™ [12] es un conjunto completo de herramientas forenses y de análisis en un paquete consistente.

- **La seguridad lo primero** - MacForensicsLab tiene mucho cuidado a la hora de asegurar la integridad de la evidencia. DiskArbitration puede deshabilitarse con el clic de un botón para asegurarnos que el SSOO Mac. no intentará montar y por tanto, alterar, el disco duro sospechoso.

- **Logs detallados** – Cada acción tomada mientras que se usa el software es almacenada en logs altamente detallados para proporcionar al investigador la mayor cantidad posible de información. En cualquier fase se pueden crear notas para almacenar las impresiones del investigador durante el proceso.

- **Informes del caso en HTML** – Una combinación de datos del gestor del caso y de los ficheros logs (cronología, recuperaciones, análisis, adquisición, catálogos, favoritos, notas) puede ser exportada en un informe HTML para su visionado en cualquier navegador Web.

- **Hashing flexible** – Los procesos de adquisición de datos incluyen la utilización de hashes MD5, SHA1 y SHA256. Los hashes pueden crearse desde un fichero o directorio con tan solo hacer clic en un botón.

- **Recuperar evidencias después de que un disco o dispositivo ha sido formateado** - MacForensicsLab recuperará ficheros, hará búsquedas de cadenas y permitirá el análisis de las unidades formateadas recientemente.

- **Recupera evidencias de medios corruptos** – Se procesará cualquier dato intacto en el disco, y recuperará cadenas y ficheros enteros o parciales donde quiera que se encuentren.

- **Trabaja con datos de otros sistemas operativos** - MacForensicsLab está preparado para realizar adquisición de datos y análisis de unidades con Windows, Linux, y otros sistemas operativos.

- **Proporciona métodos muy rápidos y fáciles para encontrar y marcar evidencias** – con la herramienta "Browse", MacForensicsLab permite al investigador el visionado de ficheros en vista nativa a la vez que se recorre una estructura completa de directorios.

Herramientas basadas en Windows

5.2.5 BringBack de Tech Assist, Inc.

BringBack™ [13] proporciona recuperación de datos de sistemas operativos Windows™ & Linux (ext2), además de imágenes digitales almacenadas en tarjetas de memoria, etc.

Características:

- Disk Viewer – está diseñado para asistir a una persona experta en la valoración de las condiciones de un volumen. Proporciona búsqueda, navegación y visionado de los formatos más comunes.
- Los sistemas de ficheros soportados son FAT16, FAT32 y NTFS (todas las versiones)
- Proporciona soporte limitado para ext2 (sistema de ficheros Linux)
- Recupera hardware RAID0 y RAID5
- Motor de validación – comprueba ficheros en disco antes de recuperarlos para ver los que estar rotos y los que no.

  La siguiente es la lista de formatos de fichero conocidos por la versión actual del motor de validación de datos (BringBack™ 2.1), en ningún orden en particular:

  - Almacenamiento con estructura OLE
    - Microsoft Word **.doc**
    - Microsoft Excel **.xls**

        - Microsoft Power Point **.ppt**
        - Windows Installer package **.msi**
    - **.exe, .dll, .cpl** Módulo ejecutable Windows (Win32/PE format)
    - **.ace** archivo comprimido
    - **.arj** archivo comprimido
    - **.asf** video
    - **.dbf** formato de base de datos
    - **.gif** imagen
    - **.gz** archivo comprimido gzip
    - **.ico** fichero icono Windows
    - **.inf** fichero INF Windows
    - **.jpg, .jpeg** imagen JPEG
    - **.mid** sonido MIDI
    - **.mp3** sonido
    - **.pdf** documento Adobe Acrobat
    - **.png** imagen
    - **.qt** video QuickTime
    - **.rar** archivo comprimido
    - **.rm** real Media
    - **.rmid** fichero de sonido
    - **.tga** imagen
    - **.tiff** imagen
    - **.url** fichero URL de Internet Explorer
    - **.wav** fichero de sonido
    - **.zip** archivo comprimido

- Recuperación de ficheros de imagen digital desde cámaras digitales
- Funciona sobre Windows NT/2000/XP/2003
- Puede ejecutarse desde un CD, por ejemplo, y no necesita escribir en el disco excepto por dos cosas:
  1. Ficheros log opcionales (configurable)
  2. Ficheros recuperados

5.2.6 EnCase, by Guidance Software

EnCase [14], desarrollada por Guidance Software Inc., permite asistir al especialista forense durante el análisis de un crimen digital.

Se trata del software líder en el mercado, el producto más ampliamente difundido y de mayor uso en el campo del análisis forense.

Algunas de las características más importantes de EnCase se relacionan a continuación:

- Copiado Comprimido de Discos Fuente. Encase emplea un estándar sin pérdida (loss-less) para crear copias comprimidas de los discos origen. Los archivos comprimidos resultantes, pueden ser analizados, buscados y verificados, de manera semejante a los normales (originales). Esta característica ahorra cantidades importantes de espacio en el disco del computador del laboratorio forense, permitiendo trabajar en una gran diversidad de casos al mismo tiempo, examinando la evidencia y buscando en paralelo.

- Búsqueda y Análisis de Múltiples partes de archivos adquiridos. EnCase permite al examinador buscar y analizar múltiples partes de la evidencia. Muchos investigadores involucran una gran cantidad de discos duros, discos extraíbles, discos "zip" y otros tipos de dispositivos de almacenamiento de la información. Con Encase, el examinador puede buscar todos los datos involucrados en un caso en un solo paso. La evidencia se clasifica, si esta comprimida o no, y puede ser colocada en un disco duro y ser examinada en paralelo por el especialista. En varios casos la evidencia puede ser ensamblada en un disco duro grande o un servidor de red y también buscada mediante EnCase en un solo paso.

- Diferente capacidad de Almacenamiento. Los datos pueden ser colocados en diferentes unidades, como Discos duros IDE o SCSI, drives ZIP, y Jazz. Los archivos pertenecientes a la evidencia pueden ser comprimidos o guardados en CD-ROM manteniendo su integridad forense intacta, estos archivos pueden ser utilizados directamente desde el CD-ROM evitando costos, recursos y tiempo de los especialistas.

- Varios Campos de Ordenamiento, Incluyendo Estampillas de tiempo. EnCase permite al especialista ordenar los archivos de la evidencia de acuerdo a diferentes campos, incluyendo campos como las tres estampillas de tiempo (cuando se creó, último acceso, última escritura), nombres de los archivos, firma de los archivos y extensiones.

- Análisis Compuesto del Documento. EnCase permite la recuperación de archivos internos y meta-datos con la opción de montar directorios como un sistema virtual para la visualización de la estructura de estos directorios y sus archivos, incluyendo el slack interno y los datos del espacio libre.

- Búsqueda Automática y Análisis de archivos de tipo Zip y Attachments de E-Mail. Firmas de archivos, Identificación y Análisis. La mayoría de las graficas y de los archivos de texto comunes contiene una pequeña cantidad de bytes en el comienzo del sector los cuales constituyen una firma del archivo. EnCase verifica esta firma para cada archivo contra una lista de firmas conocida de extensiones de archivos. Si existe alguna discrepancia, como en el caso de que un sospechoso haya escondido un archivo o simplemente lo haya renombrado, EnCase detecta automáticamente la identidad del archivo, e incluye en sus resultados un nuevo ítem con la bandera de firma descubierta, permitiendo al investigador darse cuenta de este detalle.

- Análisis Electrónico Del Rastro De Intervención. Sellos de fecha, sellos de hora, registro de accesos y la actividad de comportamiento reciclado son a menudo puntos críticos de una investigación por computador. EnCase

proporciona los únicos medios prácticos de recuperar y de documentar esta información de una manera no invasora y eficiente. Con la característica de ordenamiento, el análisis del contenido de archivos y la interfaz de EnCase, virtualmente toda la información necesitada para un análisis de rastros se puede proporcionar en segundos.

- Soporte de Múltiples Sistemas de Archivo. EnCase reconstruye los sistemas de archivos forenses en DOS, Windows (todas las versiones), Macintosh (MFS, HFS, HFS+), Linux, UNIX (Sun, Open BSD), CD-ROM, y los sistemas de archivos DVDR. Con EnCase un investigador va a ser capaz de ver, buscar y ordenar archivos desde estos discos concurridos con otros formatos en la misma investigación de una manera totalmente limpia y clara.

- Vista de archivos y otros datos en el espacio Libre. EnCase provee una interfaz tipo Explorador de Windows y una vista del Disco Duro de origen, también permite ver los archivos borrados y todos los datos en el espacio Libre. También muestra el Slack File con un color rojo después de terminar el espacio ocupado por el archivo dentro del cluster, permitiendo al investigador examinar inmediatamente y determinar cuándo el archivo reescrito fue creado. Los archivos Swap y Print Spooler son mostrados con sus estampillas de datos para ordenar y revisar.

- Integración de Reportes. EnCase genera el reporte del proceso de la investigación forense como un estimado. En este documento realiza un análisis y una búsqueda de resultados, en donde se muestra el caso incluido, la evidencia relevante, los comentarios del investigador, favoritos, imágenes recuperadas, criterios de búsqueda y tiempo en el que se realizaron las búsquedas.

- Visualizador Integrado de imágenes con Galería. EnCase ofrece una vista completamente integrada que localiza automáticamente, extrae y despliega muchos archivos de imágenes como .gif y .jpg del disco. Seleccionando la "Vista de Galería" se despliega muchos formatos de imágenes conocidas, incluyendo imágenes eliminadas, en el caso de una vista pequeña. El examinador puede después escoger las imágenes relevantes al caso e inmediatamente integrar todas las imágenes en el reporte de EnCase. No es necesario ver los archivos gráficos usando software de terceros, a menos que el formato de archivo no sea muy conocido y todavía no sea soportado por EnCase.

EnCase es un software costoso, y en Estados Unidos los costos se dividen así:

- Gobierno y Educación US$1,995

  Sector Privado US$2,495

5.2.7 FBI, by Nuix Pty Ltd

La empresa Nuix desarrolló FBI [15] con la asistencia de cuatro grandes agencias gubernamentales de U.S.A. Se desarrolla desde el año 2000 y sus clientes incluyen departamentos de gobierno, agencias de regulación, policía y cuerpos anticorrupción, bancos, y un número creciente de empresas e instituciones australianas.

Algunas aplicaciones de FBI incluyen:

- Búsqueda rápida de material inapropiado
- Identificación de filtraciones no autorizadas de documentos
- Recopilación de información para investigaciones de fraude
- Investigaciones de textos en Chino, Coreano, Árabe y Japonés
- Auditorias de alto riesgo de empleados

Formatos de datos de entrada soportados

Formatos de almacenamiento de emails:

- EDB,STM (Microsoft Exchange)
- PST,OST (Microsoft Outlook)
- MSG (Microsoft Outlook – ficheros con un solo correo)
- NSF (Lotus Notes / Domino)
- DBX,MBX (Microsoft Outlook Express)
- MBOX (Estándar)
- EML (Estándar, un email por archivo)
- EMLX (Apple Mac OS X Mail.app)
- BOX (Foxmail)
- Hotmail y Yahoo! Mail HTML (extraídos de la caché del explorador)

Protocolos de servidores de email:

- IMAP
- POP
- Novell GroupWise (vía IMAP como una "aplicación de confianza")

Formatos de imagen de disco:

- E01 (EnCase)
- Imágenes en bruto “dd” en un fichero individual

Tipos de sistemas de ficheros:

- NTFS (Microsoft Windows NT)
- FAT32 (MS-DOS, Microsoft Windows)
- Ext2 (Linux)

Formatos de documentos:

- HTML

- Texto plano
- PDF
- DOC,DOT (Microsoft Word)
- XLS,XLT (Microsoft Excel)
- PPT,POT,PPS (Microsoft PowerPoint)
- RTF
- WPS,WKS,XLR (Microsoft Works)
- WPD (Corel WordPerfect)
- CPR,SHW (Presentaciones Corel, Corel SlideShow)
- WK4 (Lotus 1-2-3) *

Formatos de imágenes:

- PNG (Portable Network Graphics)
- JPEG (Joint Photographic Experts Group)
- TIFF (Tagged Image File Format)
- GIF (Graphics Interchange Format)
- BMP (Windows bitmap)
- PBM,PPM,PGM (Portable bitmaps, pixelmaps, greymaps)
- RAW (Imágenes en bruto de cámaras digitales)
- WBMP (Wireless bitmaps)

Formatos de archivos comprimidos:

- ZIP
- GZIP

Otros formatos a comentar:

- Caché del explorador Internet Explorer
- Caché del explorador Mozilla

Características de Búsqueda

La sintaxis te búsqueda soporta:

- Consultas con comodines
- Consultas difusas
- Operadores booleanos (AND, OR, NOT)
- Consultas de frases
- Consultas de proximidad
- Consultas de campos específicos de metadatos

Los filtros disponibles para las búsquedas son:

- Tipo de fichero (tipo MIME)
- comentarios
- clasificaciones
- fecha de comunicación
- listas de palabras

Formatos de salida soportados

Los informes pueden ser exportados en los siguientes formatos:

- XHTML
- PDF
- TIFF
- CSV

Integración con otros sistemas de gestión de evidencias:

- Ringtail CaseBook

Los gráficos pueden exportarse a los siguientes formatos:

- SVG
- PNG

5.2.8 Forensic Toolkit (FTK), de AccessData

AccessData Forensic Toolkit® (FTK™) [16] ofrece la posibilidad de realizar un análisis forense completo. Proporciona poderosas búsquedas y filtrados de ficheros, permitiendo ordenar miles de ficheros para encontrar más rápidamente la evidencia. FTK está reconocido como la herramienta líder en el análisis de emails.

Características:

Fácil de usar

- Visionado de 270 formatos de fichero diferentes.
- FTK Explorer permite navegar a través de los ficheros de una imagen adquirida.
- Genera logs e informes.
- Compatible con Password Recovery ToolkitTM y Distributed Network Attack®.

Búsqueda avanzada

- El indexado de textos completos de dtSearch® proporciona resultados de búsqueda instantáneos.
- Búsquedas avanzadas para imágenes JPEG y texto de Internet.
- Localiza patrones de binarios usando Live Search.
- Recupera automáticamente ficheros y particiones borradas.

Ficheros soportados y Formatos de Adquisición

- Formatos de fichero: NTFS, NTFS comprimido, FAT 12/16/32, y Linux ext2/ext3.
- Formatos de imagen adquirida: Encase, SMART, Snapback, Safeback (no incluye la version 3), y Linux DD.

Análisis de Emails y ficheros Zip

- Soporta: Outlook, Outlook Express, AOL, Netscape, Yahoo, Earthlink, Eudora, Hotmail, y correo MSN.
- Visionado, búsqueda, impresión y exportación de mensajes de email y sus adjuntos.
- Recupera emails borrados o parcialmente borrados.
- Automáticamente extrae datos desde ficheros comprimidos PKZIP, WinZip, WinRAR, GZIP, y TAR.

Known File Filter™ (KFF™) (Filtro de ficheros conocidos)

- Identifica y marca ficheros de programas y sistemas operativos estándar.
- Identifica y marca pornografía infantil conocida y otros ficheros con evidencias potenciales.
- Incluye las bases de datos de hashes conocidos: NIST y Hashkeeper

Registry Viewer™ (Visor del registro)

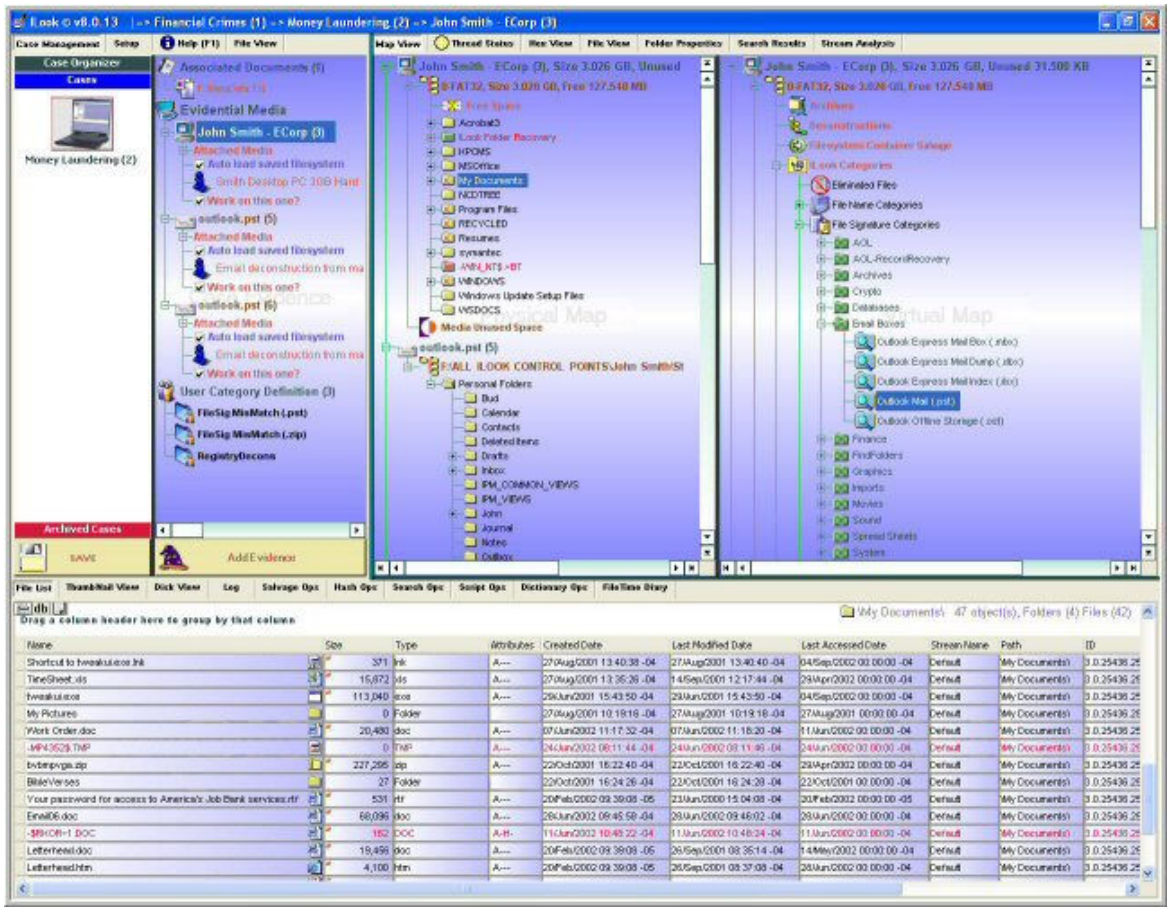
- Accede a datos protegidos
- Visionado de ficheros de registro independientes
- Generación de informes

5.2.9 ILook Investigator,

Realizado por Elliot Spencer y el U.S. Dept of Treasury, Internal Revenue Service - Criminal Investigation (IRS)

Ilook [17] es una herramienta forense multihilo, compatible con Unicode, diseñada para analizar una imagen de un sistema de ficheros. Funciona en las siguientes plataformas de 32 bits: Win2K o WinXP, y la siguiente de 64 bits: Windows Server 2003 64bit. Puede usarse también para examinar imágenes obtenidas desde otras herramientas de adquisición de imágenes que producen una imagen en bruto (raw).

El hardware recomendado para la versión actual, **ILookv8**, es un procesador Pentium 4 a 2Ghz con 1GB de RAM. ILookv8 contiene una ayuda Online que está disponible una vez instalado.



Características de ILook v8

1.Adquisición de la imagen de un sistema de ficheros.

2.Identificación y soporte para los siguientes sistemas de ficheros: FAT12, FAT16, FAT32, FAT32x, VFAT, NTFS, HFS, HFS+, Ext2FS, Ext3FS, SysV AFS, SysV EAFS, SysV HTFS, CDFS, Netware NWFS, Reiser FS, ISO9660

3.Una interfaz tipo Explorer que permite al investigador el visionado y navegación a través del sistema de ficheros.

4.Posibilidad de extracción granular que permite extraer un fichero o una parte del mismo desde una imagen.

5.Motor de búsqueda de expresiones regulares.

7.Visionado de ficheros en múltiples formatos.

8.Generador de diccionarios de passwords.

9.Visor hexadecimal.

10. Posibilidad de recuperación de ficheros.

11. Rutinas de verificación de hashes.

12.Recuperación de directorios huérfanos FAT.

14.Características de Etiquetado y Categorización de datos.

15.Posibilidad de hacer informes.

16.Características de gestión de casos y evidencias, y manejo de elementos multi-evidencia.

17.Funciones para el tratamiento de la caché de Internet y bandejas de correo.

18.Características de investigación directamente en los dispositivos.

20.Funciones de filtrado y eliminación de ficheros.

21.Una base de datos de resultados de búsqueda almacena los resultados de todas las búsquedas hechas en un caso.

22.La vista de Mapa de bits de un volumen permite un visionado general de la capa física de cualquier volumen seleccionado..

23.Métodos sofisticados de automatización de procesos.

24.Acceso completo a la arquitectura DotNet de Microsoft's, y los compiladores C# y VB.Net.

26.Detecciones de ficheros protegidos para los tipos de ficheros comúnmente protegidos por password.

27.Carpetas virtuales de categorización de datos.

28.Visor integrado de thumbnails.

29.Diario FileTime.

30.Análisis de un stream de datos.

Bases de datos de Hash soportadas

ILookv8 usa la Hashkeeper Database diseñada y mantenida por el U.S. DOJ National Drug Intelligence Center (NDIC) http://www.usdoj.gov/ndic/about.htm

ILookv8 usa el National Institute of Standards and Technogy (NIST), National Software Reference Library (NSRL) http://www.nsrl.nist.gov/downloads.htm.

5.2.10 Safeback de NTI & Armor Forensics

- **SafeBack [18] – Usos Primarios:**

- Usado para crear copias de respaldo de discos duros en sistemas basados en Intel.
- Usado para recuperar imágenes de tipo SafeBack en otra unidad de disco duro de igual o mayor capacidad.
- Usado como una herramienta de conservación de evidencias.
- Usado como herramienta de inteligencia en agencias militares..

- **SafeBack – Características del programa y Beneficios:**

- **Basado en MS-DOS** por facilidad de operación, velocidad y para eliminar problemas creados por Windows concerniente a la alteración potencial de datos.
- Incorpora dos implementaciones separadas del algoritmo testado por el NIST, SHA256, para asegurar la integridad de todos los datos contenidos en el dispositivo de almacenamiento a usar.
- Proporciona un seguimiento de auditoria del proceso de adquisición de imágenes para documentar evidencias. Por defecto se muestra un valor hash SHA256 para cada imagen extraída que puede ser comparado con el original.
- Copia con precisión todas las áreas del disco duro.
- Soporta otros discos duros no-DOS y no-Windows, por ejemplo UNIX, siempre que esté en un sistema basado en Intel.
- Permite extraer imágenes de un disco duro a través del puerto de la impresora.
- Permite duplicar un disco duro en otro de modo directo.
- Los archivos imagen de tipo SafeBack pueden ser almacenado en un fichero grande o en ficheros separados de tamaño fijo. Esta característica es útil a la hora de hacer copias en CDs o DVDs.
- Permite realizar copias lógicas o físicas a elección del usuario.
- Copia y recupera múltiples particiones conteniendo uno o más sistemas operativos.
- Se usan combinaciones matemáticas del algoritmo CRC para garantizar la precisión de la copia, por ejemplo, el RSA MD5.
- Escribe en unidades SCSI de cintas backup o en discos duros.
- La versión actual de SafeBack comprime las secciones no usadas o no formateadas del disco duro para incrementar la velocidad del proceso y ahorrar espacio de almacenamiento en el archivo imagen SafeBack.

5.2.11 X-Ways Forensics, de X-Ways AG

X-Ways Forensics [19] es un entorno de trabajo avanzado para examinadores forenses. Funciona bajo Windows 98/Me/2000/XP/2003 (la funcionalidad completa

solo es soportada bajo Windows 2000/XP/2003). Está estrechamente integrada con la herramienta WinHex y puede comprarse como una licencia forense para WinHex. X-Ways Forensics abarca todas las características conocidas de WinHex, que son:

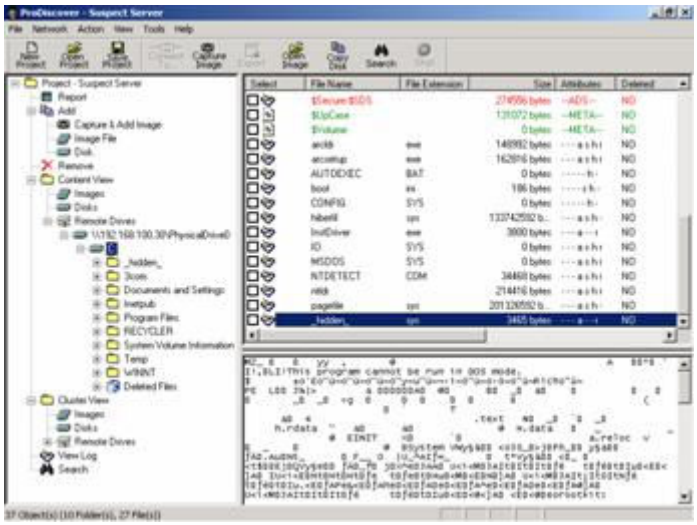
- Clonado y extracción de imágenes con X-Ways Replica
- Examen de la estructura de directorios almacenada en archivos imagen en bruto
- Soporte nativo para FAT, NTFS, Ext2/3, CDFS, UDF
- Interpretación integrada de sistemas RAID 0 y RAID 5, y discos dinámicos.
- Visionado y volcado de la memoria RAM y de la memoria virtual de los procesos en ejecución.
- Varias técnicas de recuperación de datos y ficheros.
- Borrado seguro de discos duros para producir medios estériles desde el punto de vista forense.
- Recopilación del Slack Space de los ficheros, del espacio libre, el espacio entre particiones y texto genérico de unidades de disco o imágenes.
- Fácil detección y acceso al Alternate Data Streams (ADS) del sistema de ficheros NTFS.
- Cálculo en masa del hash de ficheros (CRC32, MD5, SHA-1, SHA-256, ...)
- No depende exclusivamente de MD5 debido a que produce colisiones.
- Motor de búsqueda para realizar varias búsquedas lógicas y/o físicas al mismo tiempo.
- Favoritos/anotaciones

Y además las características que aporta X-Ways:

- Soporte para sistemas de fichero HFS, HFS+, ReiserFS, Reiser4, UFS, UFS2
- Vista de galería de imágenes
- Vista de calendario
- Vista previa con visores integrados de más de 400 tipos.
- Protección contra escritura para asegurar la autenticidad de los datos.
- Gestión completa de casos
- Actividad de logging automática
- Informes automatizados que pueden importarse a varios formatos
- Habilidad para asociar comentarios sobre ficheros para incluirlos en el informe o para filtrarlos
- Habilidad para etiquetar ficheros y añadirlos a tablas resumen de elementos notables
- Árbol de directorios a la izquierda, con la capacidad de explorar y etiquetar directorios incluyendo todos sus subdirectorios. También se ven los ficheros y directorios borrados.
- Filtros dinámicos basados en tipos de ficheros conocidos, Categoría de hash, marcas de tiempo, tamaño, comentarios, etc.
- Identificación automática de documentos encriptados MS Office y PDF
- Encuentra imágenes embebidas dentro de documentos, por ejemplo de MS Word, PDF o PowerPoint automáticamente
- Visor interno del registro de Windows
- Detección del color de la piel en una imagen. Por ejemplo, en una vista de galería ordenada por el porcentaje de color de piel se pueden encontrar más fácilmente pornografía infantil

- Detección de Host-Protected Areas (HPA), conocidas como áreas protegidas ATA
- Capacidad para importar los conjuntos de hashses NSRL RDS 2.x, HashKeeper, y ILook para buscar ficheros conocidos.
- Permite crear un conjunto propio de hashes.

5.2.12 Prodiscover, de Techpathways



ProDiscover® ***Forensics*** [20] es una herramienta de seguridad en computadores que permite encontrar datos en un disco protegiendo la evidencia y crear informes de calidad usados en procedimientos legales.

- **Características y Beneficios:**

- Crea una copia Bit a Bit del disco a analizar, incluyendo las secciones HPA.
- Busca en ficheros o discos enteros, incluyendo el Slack Space, la sección HPA y los ADS de los sistemas NTFS.
- Vista previa de todos los ficheros, incluyendo los borrados y los ocultos y sus metadatos.
- Mantiene la compatibilidad entre herramientas leyendo y escribiendo las imágenes forenses en el formato “dd” de UNIX.
- Examina y realiza referencias cruzadas a los datos a nivel de fichero o cluster para asegurar que nada está oculto, incluso en el Slack Space.
- Genera y almacena automáticamente los valores hash MD5, SHA1 o SHA256 para probar la integridad de los datos.
- Utiliza bases de datos de hashes definidas por el usuario o la del National Drug Intelligence Center Hashkeeper para identificar ficheros positivamente.
- Examina los sistemas de ficheros FAT12, FAT16, FAT 32 y todos los NTFS, incluyendo Dynamic Disk y Software RAID.

- Examina el sistema de ficheros Sun Solaris UFS y el Linux ext2 / ext3.
- Visor integrado de thumbnails y visor de registro.
- Extrae el histórico de Internet.
- Utiliza scripts de Perl para automatizar tareas de investigación.
- Extrae información EXIF de ficheros JPEG para identificar a los creadores del fichero.
- Generación automática de informes en formato XML.

Herramientas de código abierto

5.2.13 AFFLIB

Es una librería para trabajar con imágenes de disco. Actualmente AFFLIB soporta imágenes en los formatos: dd, AFF, AFD y EnCase.

La Librería AFF viene con las siguientes utilidades:

Aconvert - Convierte una o más imágenes dd a formato AFF.

Acompare - Compara un fichero dd con sus ficheros AFF.

Ainfo - Informa sobre un fichero AFF, incluyendo todos los segmentos y sus contenidos. Valida códigos MD5 Y SHA1.

Acat - Copia un fichero AFF a un fichero dd (o a la salida estándar)

5.2.14 Autopsy

- **Descripción**

El Autopsy Forensic Browser [21] es una interfaz gráfica para la herramienta de análisis digital The Sleuth Kit. Ambos pueden analizar sistemas de ficheros Windows y UNIX (NTFS, FAT, UFS1/2, Ext2/3).

The Sleuth Kit y Autopsy son de Código Abierto y funcionan sobre plataformas UNIX. Ya que Autopsy está basado en HTML, podemos conectarnos al servidor Autopsy desde cualquier plataforma usando un navegador HTML. Proporciona una interfaz de tipo gestor de ficheros y muestra detalles sobre datos borrados y estructuras de sistema de ficheros.

- **Modos de Análisis**

- Un **dead analysis** (análisis post-mortem) ocurre cuando un sistema de análisis dedicado se usa para examinar los datos de un sistema sospechoso. En este caso, Autopsy y The Sleuth Kit se ejecutan en un entorno controlado, normalmente en un laboratorio. Autopsy y T.S.K. soportan los formatos dd, AFF y Expert Witness.

- Un **live analysis** (análisis en vivo) ocurre cuando el sistema sospechoso está siendo analizado mientras está en ejecución. En este caso, Autopsy y The Sleuth Kit se ejecutan desde un CD en un entorno no fiable. Esto se usa frecuentemente durante una respuesta a un incidente mientras el incidente está siendo confirmado. Después de confirmarlo. Después de confirmarlo, el sistema puede ser adquirido y realizarse posteriormente un **dead análisis**.

- **Técnicas de búsqueda de evidencias**

- **Listado de Ficheros**: Analiza los ficheros y directorios incluyendo los nombres de ficheros borrados y ficheros con nombres basados en Unicode.

- **Contenido del fichero**: El contenido de los ficheros puede verse en formato raw (en bruto), en ASCII o en hexadecimal. Cuando se interpretan los datos Autopsy los limpia para evitar daños en el sistema local de análisis. Autopsy no requiere ningún lenguaje de scripts por parte del cliente.

- **Bases de datos de Hashes**: Busca ficheros conocidos/desconocidos comparando su valor hash con una base de datos con hashes verificados. Autopsy usa el NIST National Software Reference Library (NSRL) y bases de datos creadas por el usuario

- **Ordenación por tipo de fichero**: Ordena los ficheros basándose en su estructura interna para identificar tipos de ficheros conocidos. Autopsy puede también extraer solamente las imágenes gráficas (incluyendo thumbnails). La extensión del fichero se comparará también con el tipo de fichero para verificar si coincide con el que describe su estructura interna (alguien ha cambiado la extensión del fichero a propósito)

- **Línea de tiempo de la actividad de los ficheros**: En algunos casos, teniendo una línea de tiempo de la actividad de los ficheros, puede ayudar a identificar áreas de un sistema de ficheros que pueden contener evidencias. Autopsy permite crear líneas de tiempo que contienen entradas para los tiempos de Modificado, Accedido o Cambiado (MAC) de los ficheros, borrados y no borrados.

- **Búsqueda de palabras**: Se buscan cadenas de caracteres en el sistema de ficheros usando cadenas ASCII y expresiones regulares. Las búsquedas pueden realizarse en el sistema de ficheros completo o solo en el espacio libre. Se puede crear un fichero índice para búsquedas más rápidas.

- **Análisis de MetaDatos**: Las estructuras de metadatos contienen los detalles sobre los ficheros y directorios. Autopsy permite ver los detalles de cualquier estructura de metadatos en el sistema de ficheros. Esto es útil para recuperar

contenidos borrados. Autopsy buscará los directorios para identificar la ruta completa que tiene asignada la estructura.

- **Análisis de Unidades de Datos**: Las Unidades de Datos es donde se almacena el contenido de fichero (bloques, clusters, etc.). Autopsy permite ver los contenidos en una variedad de formatos, incluyendo ASCII, hexadecimal y en cadenas. Además se pueden buscar estructuras de metadatos que tengan asignadas ciertas unidades de datos.

- **Detalles de Imagen**: Pueden verse los detalles de una imagen, incluyendo un esquema de la estructura y tiempos de actividad. Este modo proporciona información que es útil durante la recuperación de datos.

- **Gestión de casos**

- **Gestor de casos**: Las investigaciones están organizadas en "casos", que pueden contener uno o más "hosts". Cada host está configurado para tener su propia zona horaria y reloj, de modo que el tiempo muestra como lo veía el usuario original. Cada host puede contener una o más imágenes de sistemas de ficheros a analizar.

- **Secuenciador de eventos**: Eventos basados en el tiempo pueden añadirse desde la actividad de los ficheros o desde logs de IDS o firewall. Autopsy ordena los eventos de modo que la secuencia de eventos puede determinarse más fácilmente.

- **Notas**: Las notas pueden guardarse basadas en el host o en el investigador. Esto permite hacer notas rápidamente sobre ficheros y estructuras. Todas las notas se almacenan en formato ASCII.

- **Integridad de la Imagen**: Es crucial el asegurar que los ficheros no han sido modificados durante el análisis. Autopsy, por defecto, generará un valor MD5 para todos los ficheros que se importen o creados. La integridad de cualquier fichero usa se puede validar en cualquier momento.

- **Informe**: Autopsy puede crear informes ASCII para los ficheros y otras estructuras del sistema.

- **Logging**: Se crean logs de auditoria a nivel de caso, host e investigador, de modo que podemos recordar nuestras acciones. Se anotan también los comandos exactos de Sleuth Kit que son ejecutados.

- **Diseño Abierto**: El código de Autopsy es código abierto y todos los ficheros que usa están en formato raw (en bruto). Todos los ficheros de configuración están en texto ASCII y los casos se organizan en directorios. Esto hace fácil exportar datos y archivarlos. Esto además no restringe al investigador para usar otras herramientas que pueden resolver el problema específico más apropiadamente.

- **Modelo Cliente Servidor**: Autopsy está basado en HTML y por tanto no es necesario estar en el mismo sistema en el que están las imágenes a analizar. Esto

permite a múltiples investigadores a usar el mismo servidor y conectarse desde sus ordenadores personales.

Autopsy está escrito en Perl y funciona en las mismas plataformas de UNIX que The Sleuth Kit:

- Linux
- Mac OS X
- Open & FreeBSD
- Solaris

5.2.15 FOREMOST

Foremost [22] es un programa de consola para recuperar ficheros basándose en:

- Su cabecera: Los datos al inicio de un fichero
- Su cola: Los datos al final de un fichero
- Sus estructuras de datos internas

Este proceso se denomina comúnmente como Data Carving. Foremost puede trabajar sobre archivos imagen, como los generados por dd, Safeback, Encase, etc. o directamente sobre una unidad. Se pueden especificar las cabeceras y colas en un ficheros de configuración o usar opciones de línea de comandos para especificar tipos de ficheros integrados. Estos tipos de datos conocidos buscan las estructuras de un formato especificado permitiendo una recuperación más fiable y rápida:

Desarrollado originalmente por la Air Force Office of Special Investigations y The Center for Information Systems Security Studies and Research de los Estados Unidos, foremost fue abierto al público. Inicialmente fue diseñado para imitar la funcionalidad de “CarvThis”, un programa de MS-DOS escrito por el Defense Computer Forensics Lab en 1999.

FORMATOS CONOCIDOS

jpg, gif, png, bmp, avi, mpg, exe, rar, wav, riff, wmv, mov, pdf, ole (incluye PowerPoint, Word, Excel, Access, y Star-Writer), doc (más eficiente que *ole* para buscar documentos de Word), zip (incluye los ficheros .jar y OpenOffice), htm, cpp

FICHERO DE CONFIGURACIÓN

El fichero de configuración se usa para controlar los tipos de ficheros que busca foremost. Se incluye una configuración simple de fichero, foremost.conf. Para cada tipo de fichero, el fichero de configuración debe describir su extensión, si la cabecera y la cola son sensibles a las mayúsculas, el tamaño máximo del fichero, la cabecera y por último la cola, que es opcional.

Cada línea que comience por una almohadilla "#" se consideran comentarios, por lo que basta con colocar una delante de cada tipo de fichero que no queramos que aparezca en el resultado.

Las cabeceras y las colas se decodifican antes de usarse. Para especificar valores en hexadecimal (\x[0-f][0-f]) o en octal (\[1-9][1-9][1-9]). Los espacios se representan por \s.

Ejemplo: "\x4F\123\I\sCCI" se decodifica a "OSI CCI"

```
# ficheros GIF y JPG (muy comunes)
#
# extensión  mayúsculas  max-tam   cabecera                cola (opcional)

      gif      y        155000      \x47\x49\x46\x38\x37\x61   \x00\x3b
      gif      y        155000      \x47\x49\x46\x38\x39\x61   \x00\x00\x3b
      jpg      y        200000      \xff\xd8\xff               \xff\xd9
```

EJEMPLOS

Ejecutar el caso por defecto

foremost imagen.dd

Generar un fichero de auditoria y mostrar los resultados por pantalla

foremost -av imagen.dd

Buscar todos los tipos de fichero definidos

foremost -t all -i imagen.dd

Buscar ficheros gif y pdf

foremost -t gif,pdf -i imagen.dd

Buscar ficheros jpeg saltando los 100 primeros bloques de la imagen
foremost -s 100 -t jpg -i imagen.dd

5.2.16 FTimes

FTimes [23] es una herramienta de recolección de evidencias. El primer propósito de FTimes es recopilar y desarrollar información topográfica y atributos sobre directorios y ficheros especificados para contribuir en un análisis forense o una intrusión.

FTimes es una herramienta ligera en el sentido que no necesita ser instalada en un sistema dado para trabajar sobre el. Cabe en un disquete y proporciona solamente una interfaz de línea de comandos.

FTimes se diseñó para almacenar 4 tipos de información en ficheros log: ficheros de configuración, indicadores de progreso, métricas y errores. La salida producida es de tipo texto delimitado y por tanto es fácilmente asimilada por una variedad de herramientas.

FTimes básicamente implementa dos capacidades generales: topografía de fichero y búsqueda de cadenas. La topografía de fichero es el proceso de mapear atributos clave de directorios sobre un sistema de ficheros. La búsqueda de cadenas es el proceso de ahondamiento en los directorios y ficheros de un sistema de ficheros dado, para buscar una secuencia específica de bytes. Respectivamente, esto se refiere al modo “map” y al modo “dig”.

FTimes soporta dos entornos operativos:

- Estación Individual: El operador usa FTimes para hacer cosas como examinar evidencias (por ej: una imagen de disco de un sistema comprometido), busca ficheros con atributos específicos, verifica la integridad, etc.

- Cliente-Servidor: El operador puede gestionar varias estaciones a la vez desde un Servidor de Integridad de manera segura y autenticada. Un Servidor de Integridad es un sistema blindado que maneja FTimes, GET, PING y peticiones PUT HTTP/S.

FTimes ha sido escrito en lenguaje C y portado a muchos SSOO populares como AIX, BSDi, FreeBSD, HP-UX, Linux, Solaris, y Windows 98/ME/NT/2K/XP. FTimes no requiere intérpretes ni máquinas virtuales.

FTimes detecta y procesa los Alternate Data Streams (ADS) cuando se ejecuta en sistemas Windows NT/2K/XP. Esto es útil en el caso que se haya usado el ADS para esconder herramientas y/o información.

5.2.17 Gfzip

Generic Forensic Zip [24] es un conjunto de herramientas y librerías para la creación y acceso aleatorio de ficheros forenses comprimidos. Estos ficheros que usan un formato abierto (gfzip), permiten que una imagen de disco "dd" sea almacenada de forma comprimida y que sea accesible aleatoriamente mediante la librería **libgfz**. Una segunda librería, **libgfzcreate** está disponible para la creación de ficheros gfz desde programas usados para adquirir datos de imágenes de disco. Finalmente el proyecto incluye un conjunto de herramientas básicas de líneas de comandos para la creación y verificación de ficheros gfzip y para restaurar imágenes dd desde ficheros gfz.

Junto con la compresión, los ficheros gfzip son seguros para el uso forense ya que se usan certificados x509 y firmas multi-nivel (sha256). El certificado x509 se usa para marcar dentro del fichero gfz con la información de la persona que adquirió la imagen.

5.2.18 Gpart

Gpart [25] es una herramienta que trata de adivinar la tabla primaria de la partición de un disco duro de tipo PC en caso de que la tabla primaria de la partición en el sector 0 esté dañada, incorrecta o suprimida. La tabla supuesta o adivinada debe escribirse a un fichero o dispositivo. Los sistemas de ficheros o tipos de particiones soportados son:

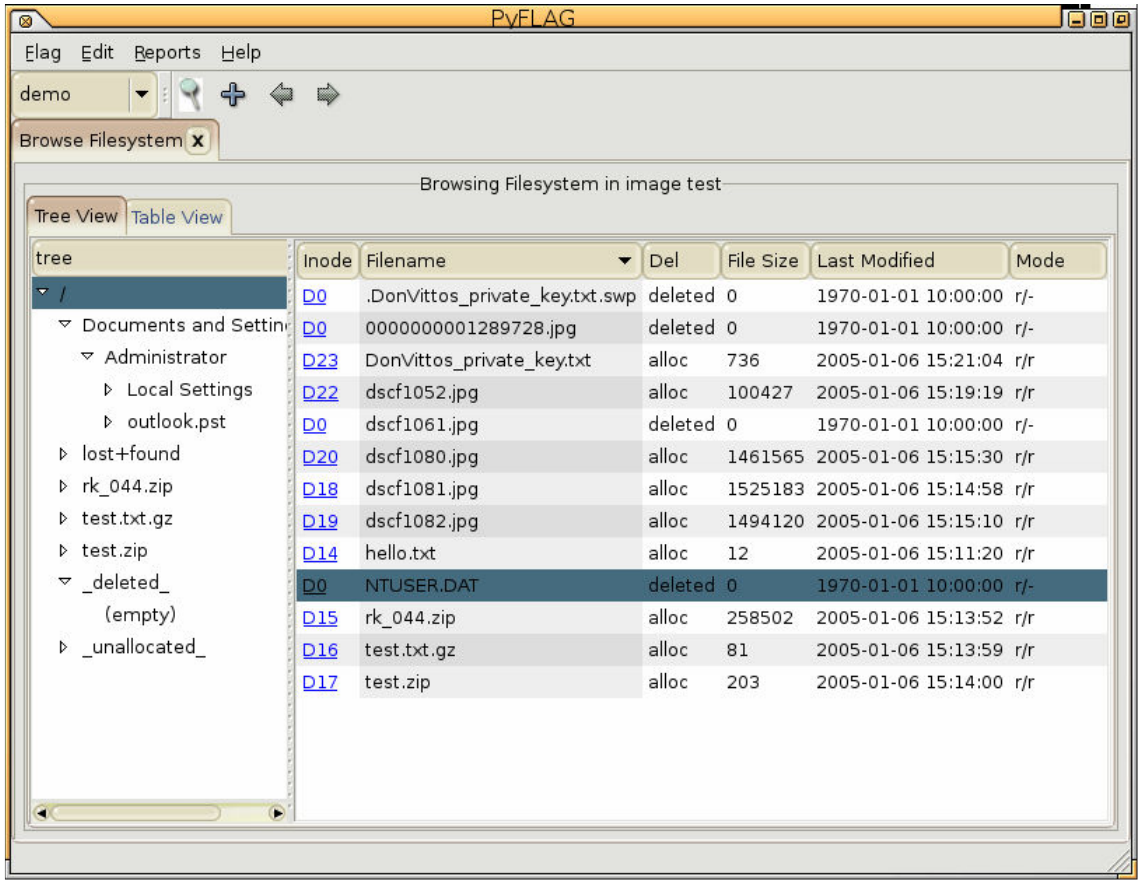
- DOS/Windows FAT (FAT 12/16/32)
- Linux ext2
- Particiones Linux Swap, versiones 0 y 1 (Linux >= v2.2.X)
- OS/2 HPFS
- Windows NT/2000 FS
- *BSD disklabels
- Solaris/x86 disklabels
- Minix FS

- Reiser FS
- Modelo de volumen físico Linux LVM (LVM de Heinz Mauelshagen)
- SGI XFS sobre Linux
- Sistema de ficheros BeOS
- Sistema de ficheros QNX 4.x

5.2.19 Magic Rescue

Magic Rescue [26] escanea un dispositivo de bloques en busca de tipos de ficheros que pueda recuperar y llama a un programa externo que los extrae. Busca "bytes mágicos" en los contenidos de los ficheros, de modo que puede usarse como utilidad de recuperación de ficheros borrados o para recuperar una unidad o partición corruptas. Trabaja sobre cualquier sistema de ficheros, aunque en los sistemas de ficheros muy fragmentados solamente puede recuperar el primer trozo de cada fichero.

5.2.20 PyFlag

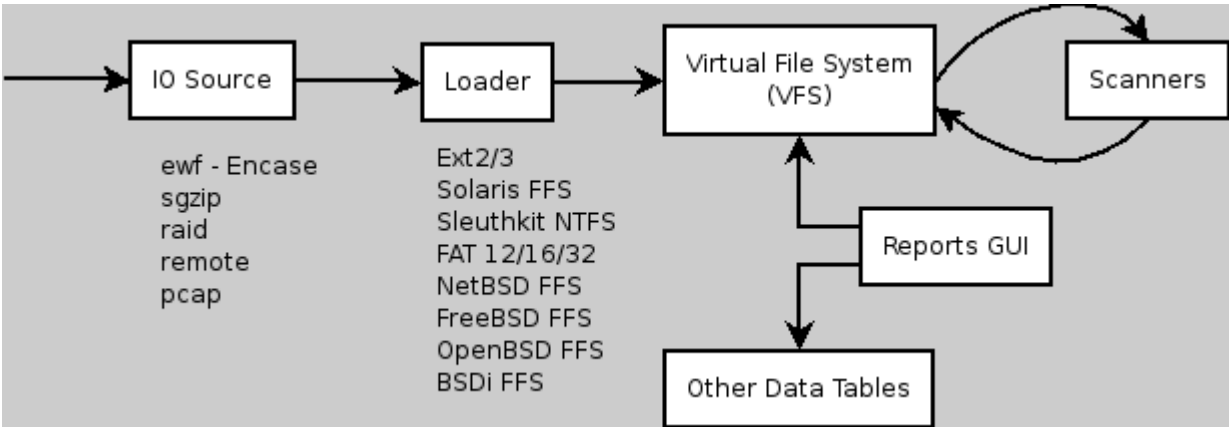


FLAG (Forensic and Log Analysis GUI) fue diseñado para simplificar el proceso de análisis de ficheros log y en investigaciones forenses. Comenzó como un proyecto en el Departamento Australiano de Defensa y más tarde se publicó. A menudo, cuando se investiga un caso grande, se necesita analizar y correlacionar una gran cantidad de datos. PyFlag [27] usa una base de datos como programa de respaldo para asistir en la gestión de grandes volúmenes de datos.

PyFLAG tiene una interfaz basada en Web, y está preparado para desplegarse en un servidor central y ser usado por varios usuarios al mismo tiempo. Los datos se almacenan en casos que mantienen la información separada.

- **Visión General**

La arquitectura general de PyFlag se muestra a continuación:



Estos son los componentes principales de PyFlag:

- IO Sources Los datos forenses a menudo están en una variedad de formatos. Esta es una abstracción que permite a PyFlag gestionar varios tipos de archivos imágenes usando diferentes drivers para presentar una visión de los datos lógica y consistente.

- The FileSystem Loader: Cada imagen puede contener una variedad de sistemas de ficheros diferentes. Este es un driver que permite a PyFlag soportar diferentes formatos de sistemas de ficheros. Este driver es responsable de cargar el VFS inicialmente con un listado de ficheros encontrados en la imagen a investigar.

- The Virtual File System PyFlag usa la idea original de Unix de que “todo es un fichero”. El VFS es la presentación principal al usuario. No tiene que existir necesariamente en la imagen, pero representa información que ha sido deducida sobre el sistema de ficheros.

- Scanners El escaneo es un proceso que pasa todos los ficheros de un cierto directorio a través de uno o más escaners. Un Scanner es un componente que estudia los ficheros que están siendo escaneados y recoge información sobre esos ficheros según ciertos requisitos definidos por el usuario.

- The GUI and table widget La interfaz gráfica proporciona un mecanismo para examinar los resultados de los escaners y navegar por el VFS. Un Informe es un conjunto limitado de funcionalidades que proporcionan acceso a datos especializados recopilados por los escaners.

- Scripting and automation A menudo desearemos poder automatizar algunas tareas de modo que las hagamos más eficientemente.

- Network Forensics: Varios módulos destinados a capturar información de la red y procesarla.

5.2.21 Scalpel

Scalpel [28] es un recuperador de ficheros que lee una base de datos de definiciones de cabeceras y colas correspondientes a cada tipo de fichero, y a continuación extrae los ficheros que coincidan de un conjunto de ficheros imagen o ficheros de dispositivo. Scalpel es independiente del sistema de ficheros y recuperará ficheros de particiones FATx, NTFS, ext2/3 o particiones RAW.

Scalpel es el resultado de reescribir completamente el código de Foremost 0.69, otro recuperador de ficheros, para aumentar el rendimiento y disminuir el uso de memoria, aunque su sintaxis es muy parecida.

5.2.22 Scrounge-Ntfs

Scrounge-Ntfs [29] es una herramienta de línea de comandos que permite la recuperación de datos de particiones NTFS corruptas. Lee cada bloque del disco duro y recupera un árbol del sistema de ficheros reconstruido en otra partición.

Aunque este software se ha usado en multitud de casos, no está probado a fondo y presenta las siguientes limitaciones:

- Muy lento en unidades extremadamente fragmentadas.

- No soporta ficheros comprimidos o encriptados.

- No está implementada la búsqueda de particiones.

5.2.23 The Sleuth Kit

Esta es una colección de herramientas de análisis forense de sistemas de ficheros para sistemas UNIX y Windows. Las herramientas permiten al investigador ver los datos borrados y las estructuras de bajo nivel en el sistema de ficheros.

Diseño: The Sleuth Kit [20] esta basado en el diseño original The Coroner's Toolkit (TCT) y contiene muchas herramientas de línea de comandos destinadas a hacer un trabajo específico. Las herramientas están organizadas en las capas de un sistema de ficheros básico, y la primera letra de cada herramienta corresponde a la capa a la que pertenece.

En este sistema de ficheros básico hay cuatro capas:

Capa del sistema de ficheros: Esta capa contiene datos que describen la estructura del sistema de ficheros. Esto incluye fragmentos y tamaños de bloque, el número de estructuras de meta-datos, así como otra información general. Los datos en esta capa se almacenan en estructuras llamadas superbloques y el registro maestro de arranque. Todas las herramientas en esta capa comienzan con "fs".

Capa de unidad de datos: Esta capa contiene las unidades de disco que son usadas para almacenar el contenido de los ficheros. En un sistema de ficheros Linux EXT3, esta capa contiene los bloques y fragmentos, y en NTFS contiene los clusters. En esta capa encontraremos trozos de un fichero que se encuentra distribuido por el disco. Todas las herramientas de esta capa comienzan con "d".

Capa de meta-datos: Esta capa contiene las estructuras de meta-datos que describen un fichero. Estas estructuras contienen los punteros a las unidades de datos e información descriptiva como tiempos de acceso/modificación/borrado del fichero y permisos. Un sistema EXT3 usa inodos y estructuras, un sistema FAT usa estructuras de datos de entradas de directorio, y un NTFS usa entradas de la Tabla Maestra de Ficheros (Master File Table, MFT). Las estructuras en esta capa trabajan como guardianes del

registro de los ficheros, ya que manejan todas las unidades de datos donde se almacena el contenido y guardan la información descriptiva. Todas las herramientas en esta capa comienzan con “i”.

Capa de Nombre de Fichero: Esta capa contiene las estructuras de nombre de fichero que permiten a los humanos interactuar con los ficheros. En muchos casos, la capa de meta-datos contiene la información descriptiva sobre los ficheros, pero ésta información viene dada en direcciones numéricas. La mayoría de los humanos le dan a los ficheros un nombre en lugar de una dirección numérica, por lo que esta capa enlaza los ficheros y directorios con su dirección de meta-datos.

El resto del nombre de la herramienta corresponde a la función que realiza en esa capa. Estas incluyen:

- Herramientas que listan detalles sobre múltiples estructuras en una capa (acaban en “ls”).
- Herramientas que listan detalles sobre una estructura específica en una capa (acaban en “stat”).
- Herramientas que sirven de enlace entre capas (acaban en “find”).
- Herramientas que muestran contenidos en una capa (acaban en “cat”).

Sintaxis:

Cada herramienta tiene un único conjunto de argumentos de línea de comandos, pero hay algunos principios básicos que aplican la mayoría de las herramientas:

- Todos las herramientas de sistema de ficheros requieren que el tipo de sistema de ficheros sea especificado con la opción “-f”. Los tipos de sistemas de ficheros incluyen Linux-ext3, ntfs o fat32. Por ejemplo, un Linux EXT3FS debería tener “-f Linux-ext3” como argumento.
- El nombre de la imagen es necesario en todos los comandos. Cuando se ejecutan en un sistema funcionando, se puede introducir el nombre de un dispositivo, como /dev/sda1.

5.2.24 The Coroner's Toolkit (TCT)

The Coroner's Toolkit (TCT) [31] es una colección de herramientas de línea de comandos que están orientadas a recopilar y analizar datos forenses en un sistema UNIX.

Algunos componentes notables de TCT son:

- file: Decide el tipo de un fichero dado analizando su interior, sin mirar el nombre o la extensión.
- grave-robber: Una utilidad para capturar información sobre i-nodos, para luego pueda ser procesada por el programa mactime del mismo toolkit.
- icat: Copia un fichero dentro de una imagen a la salida estándar, dado su i-nodo.
- ils: Lista la información de un i-nodo.
- lastcomm: Muestra información detallada sobre los últimos comandos ejecutados en el sistema.
- mactime: Crea una línea de tiempo ASCII de la actividad de un fichero. La entrada debe estar en el formato creado por grave-robber o ils.
- md5: Calcula el valor hash MD5 de uno o varios ficheros.
- pcat: Copia la memoria usada por un proceso a la salida estándar.
- unrm y lazarus: Herramientas para la recuperación de archivos borrados (logs, RAM, swap, etc.). Estas aplicaciones identifican y recuperan la información oculta en los sectores del disco duro
- memdump: este programa vuelca el contenido de la memoria del sistema a la salida estándar, saltando los agujeros en los mapas de memoria. La salida está en formato raw (en bruto) y se puede enviar por la red para evitar que se cambie el contenido de la caché de memoria. Se puede usar netcat, stunnel o openssl según las necesidades del usuario.

5.2.25 Zeitline

Zeitline [32][33]: un editor de líneas de tiempo

En la informática forense, el área de recuperación de datos está bien cubierta con técnicas y herramientas apropiadas. Pero se ha hecho poco sobre como analizar y evaluar los datos. Solo algunas herramientas como “mactimes” o analizadores de logs son los que existen en la actualidad. Una comprensiva reconstrucción de eventos en un

sistema que guarde marcas de tiempo, logs del sistema, logs de firewall y datos de aplicación, normalmente es realizada a mano por el investigador.

Esta herramienta proporciona una ayuda a la hora de procesar grandes cantidades de datos, con una interfaz gráfica cuyo elemento principal es el "evento". Un evento consiste en un intervalo de tiempo en el que tuvo lugar, una fuente que denote el origen del evento, y una descripción del evento. Un evento puede contener una lista de sub-eventos y puede ser parte de un super-evento que describa a varios. Por ejemplo, los eventos "access program gcc", "access file x", y "access file y" pueden englobarse dentro del super-evento "install rootkit z".

Una interfaz gráfica permite al investigador gestionar los eventos. Los supereventos pueden crearse a partir de otros eventos seleccionados. Los eventos pueden moverse mediante "arrastrar y soltar" o directamente asignados a una jerarquía. La jerarquía de eventos se muestra con forma de árbol para tener una visión global o concreta según lo necesite el investigador.

5.3 EXTRACCIÓN DE META-DATOS

5.3.1 Antiword

Microsoft Word solamente existe en plataformas soportadas por el mismo Microsoft. En otras plataformas, leer un archivo Word es normalmente difícil, algunas veces caro y a menudo imposible.

Antiword [34] es un lector gratuito de MS Word para Linux y RISC OS. Hay otras versiones para FreeBSD, BeOS, OS/2, Mac OS X, Amiga, VMS, NetWare, Plan9, EPOC, Zaurus PDA, MorphOS, Tru64/OSF y DOS. Antiword convierte archivos binarios de Word 2, 7, 97, 2000, 2002 y 2003 a texto plano y a PostScript.

El nombre viene de: "ANTIdoto contra las personas que envían ficheros Word a todo el mundo, ya que ellos creen que todo el mundo usa Windows y por tanto utiliza WORD"

5.3.2 Catdoc y XLS2CSV

Catdoc [35] es un programa que lee uno mas ficheros Word y muestra el texto que contiene en la salida estándar. Por tanto, hace el mismo trabajo con los ficheros .doc como el comando **cat** para los ficheros de texto plano ASCII.

Se acompaña de **xls2csv,** programa que convierte hojas de cálculo Excel en ficheros con valores separados por comas y la herramienta **catppt,** utilidad para extraer el texto de ficheros PowerPoint.

xls2csv no extrae ningún formato ni fórmulas. El concepto es que podamos ver los datos y no la forma en que fueron creados.

- **Plataformas Soportadas**

- Unix. Catdoc se desarrolló inicialmente para Linux y Sparc Solaris.

- MS-DOS. Catdoc también funciona sobre MS-DOS, incluso en máquinas XT. Para MS-DOS se proporcionan ejecutables compilados en modo 16 bits. No existe soporte para Windows.

5.3.3 Jhead

Este programa [36] muestra o modifica datos Exif en ficheros JPEG.

Cosas que jhead puede extraer del Exif:

- Fecha y hora en la que se creó la fotografía
- Marca y modelo de la cámara
- Thumbnails integradas de baja resolución
- Velocidad de disparo de fotos
- Flash usado (si/no)
- Distancia a la que se enfocó la cámara
- Longitud focal y calcula la longitud focal equivalente en 35mm.
- Resolución de la imagen
- Información GPS, si está almacenada en la imagen.

Cosas que jhead puede modificar del Exif:

- Establecimiento de marcas de tiempo
- Trasferir cabeceras Exif entre imágenes
- Reemplazar los thumbnails dentro de las cabeceras Exif
- Editar comentarios jpegs (pero no los comentarios Exif)
- Borrar la información Exif
- Crear una nueva cabecera Exif que contenga fechas y thumbnails

5.3.4 VINETTO

Esta herramienta [37] examina los ficheros Thumbs.db. Consiste en un script en Python para línea de comandos que trabaja en Linux, Mac OS X y Cygwin(win32).

- **Un intento de tipología**

Esta tipología constituye un intento de clasificar los thumbnails de acuerdo a la forma en que los sistemas operativos Microsoft los almacenan en ficheros Thumbs.db.

Formato de Thumbnails dentro del fichero Thumbs.db	*Observaciones y comentarios*	*Thumbnail visto desde Windows (captura de pantalla)*	*Thumbnail recuperado por vinetto*
Tipo 2	a) Estos thumbnails de Tipo 2 se almacenan como fichero estándar JFIF(*): tienen cabecera, tabla Huffman y tabla de cuantización. b) Uno puede encontrarlos en sistemas Windows XP (Home y Pro) y 2003 Server. c) Estos thumbnails están asociados a nombres de fichero simples, y no a la ruta completa con la letra de la unidad como en el Tipo 1a Aquí, es un trabajo fácil para vinetto: solo se preocupa de escribir lo indicado en la cabecera en un fichero con extensión .jpg		

Tipo 1b	a) Los thumbnails de Tipo 1b consisten en un stream de datos en bruto RGBA JPEG: no tienen cabecera estándar, ni tabla Huffman ni tabla de cuantización. b) Lo mismo que el Tipo 2 (no verificado) c) Lo mismo que el Tipo 2 Aquí vinetto usa la Librería de Imágenes de Python para dividir la imagen en sus componentes R, G, B y A. El componente "A" parece no ser útil en este tipo.		
Tipo 1a	a) Los thumbnails de Tipo 1b consisten en un stream de datos en bruto RGBA JPEG: no tienen cabecera estándar, ni tabla Huffman ni tabla de cuantización. Sin embargo, los tipos 1a y 1b no son idénticos. b) Uno puede encontrarlos en sistemas Windows 9x, ME y 2000. c) Estos thumbnails se asocian a la ruta completa incluyendo la letra de la unidad. Vinetto procesa la imagen de la misma manera que 1b, salvo que en este caso, el componente "A" parece que sería útil.		

Nota : (*) Un fichero estándar JFIF comenzará siempre con los cuatro bytes hexadecimales FF D8 FF E0, seguidos de dos bytes variables (a menudo 00 10), seguidos de 'JFIF'.

5.3.5 Word2x

Word2x [38] es un programa con licencia pública y gratuito para convertir documentos Word en texto sin usar software Microsoft. Actualmente soporta los formatos de salida: texto plano, LaTeX y HTML. El programa convierte Word a un formato central y posteriormente se pasa al formato deseado.

Los entornos en los que trabaja incluyen:

- RS6000 con el sistema AIX (Unix)
- cygwin32 (Plataformas win 32 de Microsoft)
- DEC Alpha AXP bajo OSF/1 (Unix)
- IBM SP2 (Unix)
- Linux (Unix)
- SunOS (Unix)
- gcc sobre Solaris (unix)
- FreeBSD (Unix)
- SGI (Unix)
- OS/2 y EMX.

Los entornos en los que NO trabaja incluyen:

- acc sobre Solaris
- Microsoft Visual C++
- Borland C++

5.3.6 WvWare

Este software [39] extrae metadatos de varios ficheros Microsoft Word (.doc) a otros formatos como texto plano o HTML.

Consiste en una librería que permite el acceso a ficheros con formato Word 2000, 97, 95, 9, 8, 7 y 6. Compila en la mayoría de los sistemas operativos, incluyendo Linux, BSD, Solarios OS/2, AIX, OSF1 y en Amiga VMS. También hay soporte para Windows en 32 bits.

Podemos convertir un fichero Word a los formatos: HTML 4.0, LaTeX, DVI, PS, PDF, texto plano, WML, RTF, e incluso podemos ver un resumen de los metadatos incluidos en documento.

5.3.7 XPDF

Xpdf [40] es un visor de código abierto para ficheros PDF (Portable Document Format), también llamados documentos "Acrobat" por el nombre del software PDF de Adobe. El proyecto Xpdf también incluye un extractor de texto PDF, un conversor de

PDF a PS (PostScript) y otras utilidades como **pdfinfo** que muestra los metadatos de un fichero PDF.

Xpdf funciona bajo el Sistema Windows X sobre UNIX, VMS, y OS/2. Xpdf está diseñado para ser pequeño y eficiente y puede usar fuentes Tipo 1, TrueType o fuentes estándar X.

5.3.8 Metadata Assistant

[41] Extrae metadatos de ficheros Word/Excel/PowerPoint (97 y mayor). Se integra con Outlook 2000, GroupWise y Lotus Notes así como con sistemas de gestión de documentos. Además limpia y convierte los ficheros a formato PDF para una protección adicional.

5.4 ANÁLISIS DE FICHEROS

Herramientas de codigo abierto

5.4.1 File

El comando *file* determina el tipo de un fichero dado, dependiendo de su contenido y no de su extensión o nombre de fichero. Para hacer esto, utiliza un fichero de configuración "mágico" que identifica los tipos de fichero.

5.4.2 Ldd

El comando *ldd* muestra las librerías compartidas que se indiquen o las que dependan de un programa.

5.4.3 Ltrace

El comando *ltrace* es un programa que simplemente ejecuta un comando especificado si existe. Intercepta y anota las llamadas a librerías dinámicas que son llamadas por el proceso ejecutado y las señales que recibe el proceso. Además puede interceptar y mostrar las llamadas al sistema ejecutadas por el programa. Es un comando muy similar a **strace**

5.4.4 Strace

En el caso más simple, *strace* ejecuta el comando especificado si existe. Intercepta y anota las llamadas al sistema realizadas por el proceso ejecutado y las señales que recibe el proceso. En la salida de error estándar se muestra el nombre de cada llamada al sistema, sus argumentos y el valor de retorno.

Un ejemplo de la ejecución de strace podría ser el siguiente si lo ejecutamos sobre el comando "cat /dev/null":

open("/dev/null", O_RDONLY) = 3

5.4.5 Strings

Strings mostrará las cadenas de caracteres imprimibles en ficheros. Permite elegir diferentes conjuntos de caracteres (ASCII, UNICODE). Es una forma rápida de hojear ficheros, particiones, etc., con el propósito de buscar palabras, nombres de fichero, palabras clave (contraseñas), etc. Es un comando principalmente útil para determinar el contenido de los ficheros que no son de texto.

5.4.6 Galleta

La herramienta de línea de comandos Galleta [42]Analiza gramaticalmente los ficheros cookie.

Actualmente hay una carencia de métodos y herramientas de código abierto que los analistas forenses puedan usar para examinar datos encontrados en ficheros Microsoft. Muchas investigaciones de delitos relacionados con computadoras requieren la reconstrucción de ficheros Cookies del navegador Internet Explorer.

Esta herramienta analiza los ficheros Cookie. De ahí su nombre: Galleta, palabra española que en inglés se dice “cookie”. Galleta analizará gramaticalmente la información dentro de un fichero Cookie y muestra los resultados delimitados por campos, de forma que pueden exportarse fácilmente a una hoja de cálculo.

Galleta funciona en múltiples plataformas incluyendo Windows (a través de Cygwin), Macintosh, Linux y plataformas BSD.

Uso:

galleta [opciones] <nombre de fichero>

opciones:

-t Campo delimitador (Por defecto, el tabulador)

5.4.7 Pasco

[43] Analiza gramaticalmente ficheros '*index.dat'*.

Esta herramienta de línea de comandos permite la reconstrucción de la actividad de Internet de un individuo. Pasco proviene de la palabra en latín que significa "browse" en ingles y "hojear" o "echar un vistazo" en español. Analiza gramaticalmente la información contenida en "index.dat" que es el fichero de actividad del navegador Internet Explorer, y muestra los resultados al igual que el anterior, delimitado por campos para que pueda importarse su contenido desde una hoja de cálculo.

Al igual que Galleta, Pasco funciona en múltiples plataformas incluyendo Windows (a través de Cygwin), Macintosh, Linux y plataformas BSD.

Uso: pasco [opciones] <nombre de fichero>

Opciones:

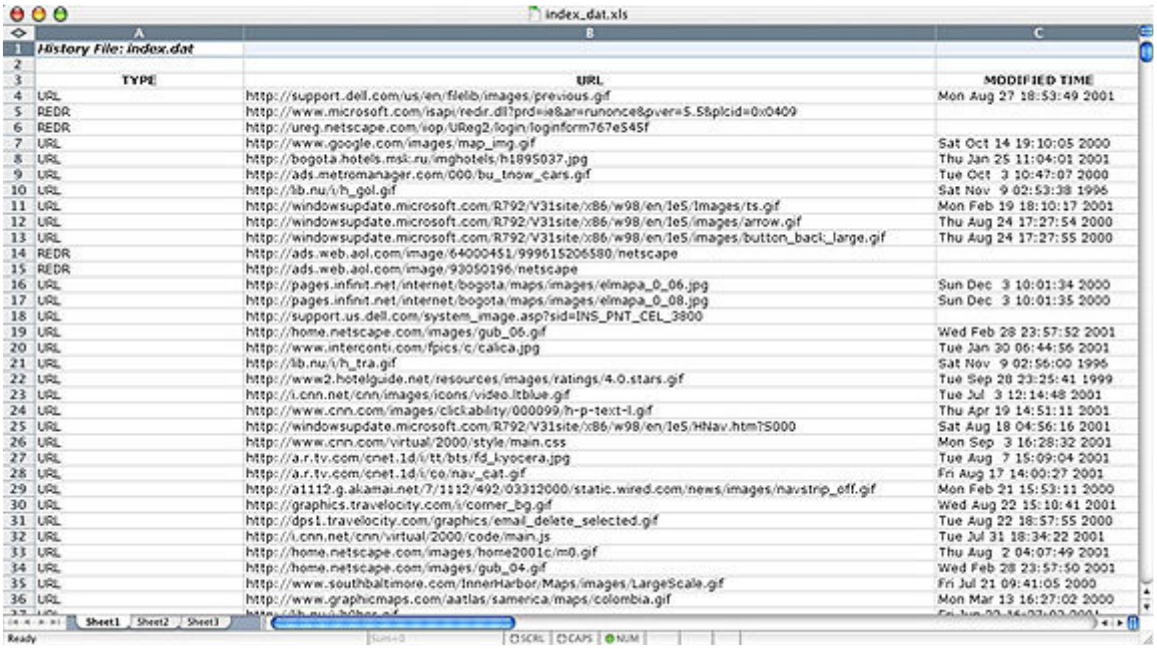
-d Recupera registros de actividad borrados

-t Delimitador de Campos (por defecto, el Tabulador)

Ejemplo de uso:

./pasco index.dat > index.txt

Si abrimos index.txt en el programa MS Excel como un fichero delimitado por Tabuladores, podemos ordenar la información de varias maneras. El resultado podría ser el siguiente:



index_dat.xls

	A	B	C
1	History File: index.dat		
2			
3	TYPE	URL	MODIFIED TIME
4	URL	http://support.dell.com/us/en/filelib/images/previous.gif	Mon Aug 27 18:53:49 2001
5	REDR	http://www.microsoft.com/isapi/redir.dll?prd=ie&ar=runonce&pver=5.5&plcid=0x0409	
6	REDR	http://ureg.netscape.com/sop/URegZ/login/loginform767e545f	
7	URL	http://www.google.com/images/map_img.gif	Sat Oct 14 19:10:05 2000
8	URL	http://bogota.hotels.msk.ru/imghotels/h1895037.jpg	Thu Jan 25 11:04:01 2001
9	URL	http://ads.metromanager.com/000/bu_tnow_cars.gif	Tue Oct 3 10:47:07 2000
10	URL	http://lib.nu/i/h_gol.gif	Sat Nov 9 02:53:38 1996
11	URL	http://windowsupdate.microsoft.com/R792/V31site/x86/w98/en/1e5/Images/ts.gif	Mon Feb 19 18:10:17 2001
12	URL	http://windowsupdate.microsoft.com/R792/V31site/x86/w98/en/1e5/images/arrow.gif	Thu Aug 24 17:27:54 2000
13	URL	http://windowsupdate.microsoft.com/R792/V31site/x86/w98/en/1e5/images/button_back_large.gif	Thu Aug 24 17:27:55 2000
14	REDR	http://ads.web.aol.com/image/64000451/999615206580/netscape	
15	REDR	http://ads.web.aol.com/image/93050196/netscape	
16	URL	http://pages.infinit.net/internet/bogota/maps/images/elmapa_0_06.jpg	Sun Dec 3 10:01:34 2000
17	URL	http://pages.infinit.net/internet/bogota/maps/images/elmapa_0_08.jpg	Sun Dec 3 10:01:35 2000
18	URL	http://support.us.dell.com/system_image.asp?sid=INS_PNT_CEL_3800	
19	URL	http://home.netscape.com/images/gub_06.gif	Wed Feb 28 23:57:52 2001
20	URL	http://www.interconti.com/fpics/c/calca.jpg	Tue Jan 30 06:44:56 2001
21	URL	http://lib.nu/i/h_tra.gif	Sat Nov 9 02:56:00 1996
22	URL	http://www2.hotelguide.net/resources/images/ratings/4.0.stars.gif	Tue Sep 28 23:25:41 1999
23	URL	http://i.cnn.net/cnn/images/icons/video.ltblue.gif	Tue Jul 3 12:14:48 2001
24	URL	http://www.cnn.com/images/clickability/000099/h-p-text-l.gif	Thu Apr 19 14:51:11 2001
25	URL	http://windowsupdate.microsoft.com/R792/V31site/x86/w98/en/1e5/HNav.htm?S000	Sat Aug 18 04:56:16 2001
26	URL	http://www.cnn.com/virtual/2000/style/main.css	Mon Sep 3 16:28:32 2001
27	URL	http://a.r.tv.com/cnet.1d/i/tt/bts/fd_kyocera.jpg	Tue Aug 7 15:09:04 2001
28	URL	http://a.r.tv.com/cnet.1d/i/co/nav_cat.gif	Fri Aug 17 14:00:27 2001
29	URL	http://a1112.g.akamai.net/7/1112/492/03312000/static.wired.com/news/images/navstrip_off.gif	Mon Feb 21 15:53:11 2000
30	URL	http://graphics.travelocity.com/i/corner_bg.gif	Wed Aug 22 15:10:41 2001
31	URL	http://dps1.travelocity.com/graphics/email_delete_selected.gif	Tue Aug 22 18:57:55 2000
32	URL	http://i.cnn.net/cnn/virtual/2000/code/main.js	Tue Jul 31 18:34:22 2001
33	URL	http://home.netscape.com/images/home2001c/m0.gif	Thu Aug 2 04:07:49 2001
34	URL	http://home.netscape.com/images/gub_04.gif	Wed Feb 28 23:57:50 2001
35	URL	http://www.southbaltimore.com/InnerHarbor/Maps/images/LargeScale.gif	Fri Jul 21 09:41:05 2000
36	URL	http://www.graphicmaps.com/aatlas/samerica/maps/colombia.gif	Mon Mar 13 16:27:02 2000

Sheet1 Sheet2 Sheet3

Ready

5.4.8 Rifiuti

[44] Examina el fichero INFO2 en la papelera de reciclaje

"Rifiuti" es la palabra italiana que significa "basura". Este programa analizará la información de los ficheros INFO2 en la papelera de reciclaje y la mostrará en un formato con campos delimitados para importarlo a una hoja de cálculo. Al igual que los anteriores, Rifiuti funciona en Windows (a través de Cygwin), Macintosh, Linux, y plataformas BSD.

Uso: rifiuti [opciones] <nombre de fichero>

Opciones:

-t Campo delimitador (por defecto, el tabulador)

5.4.9 Yim2text

[45] Esta pequeño script escrito en Python extrae la información 'encriptada' de los ficheros log del Yahoo Instant Messenger. Esta información se encripta usando un simple "Xor" sobre los datos.

5.5 ANÁLISIS DE INTERNET

5.5.1 Chkrootkit

El **chkrootkit (Check Rootkit)** es un programa de computador en modo consola y un programa común de Unix que ayuda a los administradores de sistema a localizar rootkits conocidos, realizando múltiples pruebas en las que busca entre los binarios ficheros modificados por dicho rootkit. Este shell script usa herramientas comunes de UNIX/Linux como los comandos strings y grep para buscar las bases de las firmas de los programas del sistema y comparar un transversal del archivo del sistema /proc con la salida del comando ps (process status o estados de los procesos) para buscar discrepancias. Básicamente chkrootkit hace múltiples comprobaciones para detectar todo tipo de rootkits y ficheros maliciosos.

Puede ser utilizado desde un "disco de rescate" (típicamente un LiveCD) o puede utilizar opcionalmente un directorio alternativo del cual ejecutar todos sus comandos. Estas técnicas permiten que chkrootkit confíe en los comandos de los cuales depende un bit más.

Hay limitaciones inherentes en la confiabilidad de cualquier programa que procure detectar compromisos (tales como rootkits y virus de computadora). Los nuevos rootkits pueden específicamente intentar detectar y comprometer copias del programa chkrootkit o tomar otras medidas para evadir la detección por ellas.

5.5.2 Cryptcat

Cryptcat es el comando netcat estándar mejorado con encriptación “twofish” y que funciona para Windows NT, BSD y Linux.

Cryptcat es una utilidad Unix muy simple que lee y escribe datos usando el protocolo TCP o UDP mientras encripta los datos que están siendo transmitidos.

5.5.3 Netcat

La utilidad **nc** o **netcat** permite abrir conexiones TCP, enviar paquetes UDP, escuchar en puertos arbitrarios TCP y UDP, escanear puertos y tratar con los protocolos IPv4 e IPv6.

Los usos más comunes son:

- Proxys TCP simples
- Clientes y servidores de shell basados en HTTP
- Testeo de demonios de red
- Un comando proxy HTTP o SOCKS para **ssh**.
- Etc.

5.5.4 NetIntercept

NetIntercept [46] captura paquetes enteros y reensambla hasta 999.999 conexiones TCP directamente, reconstruyendo los ficheros que circularon por la red y creando una base de datos sobre sus descubrimientos. Reconoce alrededor de 100 tipos de protocolos de red y tipos de fichero, incluyendo tráfico web, multimedia, email, y mensajería instantánea.

5.5.5 Rkhunter

rkhunter (o **Rootkit Hunter**) es una herramienta de Unix que escanea los rootkits, los backdoors y los exploit locales. Esto lo hace comparando los hashes MD5 de ficheros importantes con *buenos conocidos* en una base de datos en línea, buscando los directorios por defecto (de rootkits), los permisos incorrectos, los archivos ocultos, las cadenas sospechosas en los módulos del kernel, y las pruebas especiales para Linux y FreeBSD.

5.5.6 Sguil

Sguil [47] es una herramienta destinada a los analistas de la seguridad de red. El componente principal de Sguil es una interfaz gráfica que proporciona eventos en tiempo real para Snort/Barnyard. Además incluye otros componentes que facilitan la monitorización y eventos dirigidos al análisis de alertas IDS. El cliente Sguil está escrito en tcl/tk y puede ejecutarse sobre cualquier sistema operativo que soporte tcl/tk (incluyendo Linux, BSD, Solaris, Macintosh y Win32).

5.5.7 Snort

Snort [48] es un sniffer de paquetes y un detector de intrusos basado en red (se monitoriza todo un dominio de colisión). Es un software muy flexible que ofrece capacidades de almacenamiento de sus bitácoras tanto en archivos de texto como en bases de datos abiertas como lo es MySQL. Implementa un motor de detección de ataques y barrido de puertos que permite registrar, alertar y responder ante cualquier anomalía previamente definida. Así mismo existen herramientas de terceros para mostrar informes en tiempo real (ACID) o para convertirlo en un Sistema Detector y Preventor de Intrusos.

Este IDS implementa un lenguaje de creación de reglas flexible, potente y sencillo. Durante su instalación ya nos provee de cientos de filtros o reglas para backdoor, DDoS, finger, FTP, ataques web, CGI, Nmap...

Puede funcionar como sniffer (podemos ver en consola y en tiempo real qué ocurre en nuestra red, todo nuestro tráfico), registro de paquetes (permite guardar en un archivo los logs para su posterior análisis, un análisis offline) o como un IDS normal (en este caso NIDS). Cuando un paquete coincide con algún patrón establecido en las reglas de configuración, se logea. Así se sabe cuando, de donde y cómo se produjo el ataque.

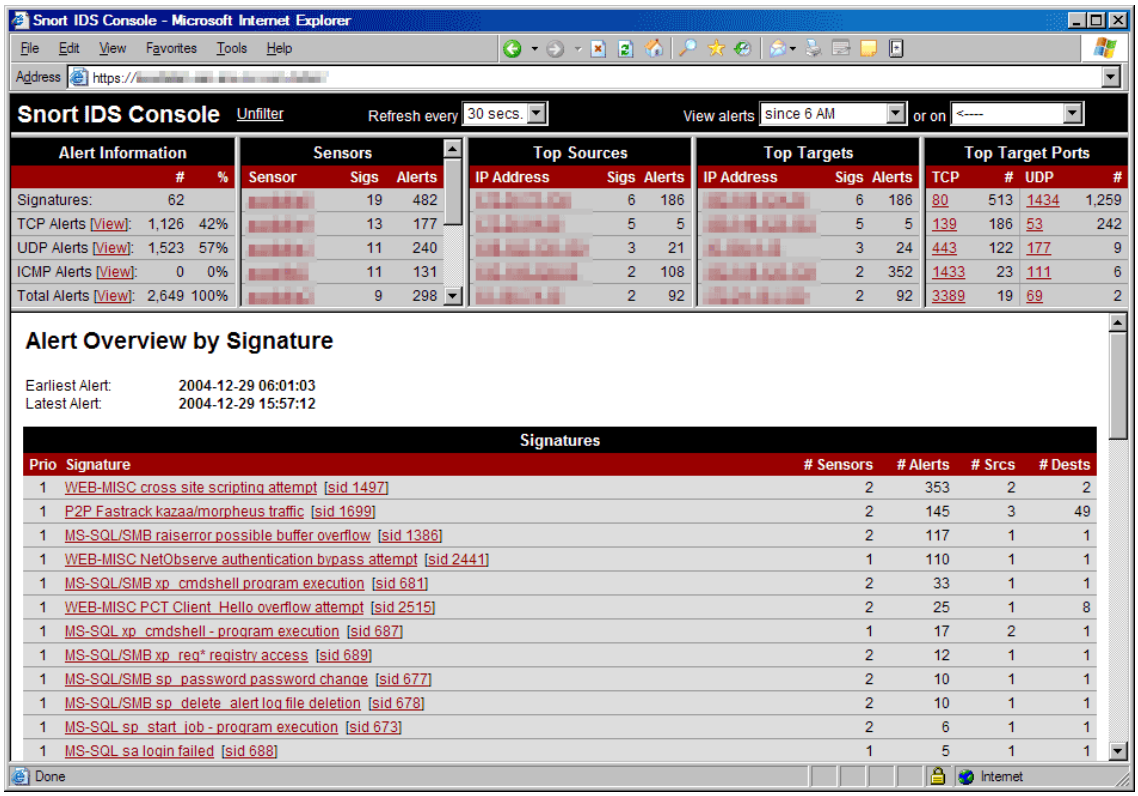
Aún cuando tcpdump es considerada una herramienta de auditoria muy útil, no se considera un verdadero IDS puesto que no analiza ni señala paquetes por anomalías. tcpdump imprime toda la información de paquetes a la salida en pantalla o a un archivo de registro sin ningún tipo de análisis. Un verdadero IDS analiza los paquetes, marca las transmisiones que sean potencialmente maliciosas y las almacena en un registro formateado, así, Snort utiliza la librería estándar libcap y tcpdump como registro de paquetes en el fondo.

Snort está disponible bajo licencia GPL, gratuito y funciona bajo plataformas Windows y UNIX/Linux. Dispone de una gran cantidad de filtros o patrones ya predefinidos, así

como actualizaciones constantes ante casos de ataques, barridos o vulnerabilidades que vayan siendo detectadas a través de los distintos boletines de seguridad.

La característica más apreciada de Snort, además de su funcionalidad, es su subsistema flexible de firmas de ataques. Snort tiene una base de datos de ataques que se está actualizando constantemente y a la cual se puede añadir o actualizar a través de la Internet. Los usuarios pueden crear 'firmas' basadas en las características de los nuevos ataques de red y enviarlas a la lista de correo de firmas de Snort, para que así todos los usuarios de Snort se puedan beneficiar. Esta ética de comunidad y compartir ha convertido a Snort en uno de los IDSes basados en red más populares, actualizados y robustos.

Captura de la consola del sistema:



5.5.8 Tcpdump

tcpdump [49] es un herramienta en línea de comandos cuya utilidad principal es analizar el tráfico que circula por la red.

Permite al usuario capturar y mostrar a tiempo real los paquetes transmitidos y recibidos en la red a la cual el ordenador está atado. Está escrito por Van Jacobson, Craig Leres, y Steven McCanne que trabajaban en este tiempo en el Grupo de

Investigación de Red del Laboratorio Lawrence Berkeley. Más tarde el programa fue ampliado por Andrew Tridgell.

tcpdump funciona en la mayoría de los sistemas operativos UNIX: Linux, Solaris, BSD, Mac OS X, HP-UX and AIX entre otros. En esos sistemas, tcpdump hace uso de la librería libpcap para capturar los paquetes que circulan por la red.

Existe una adaptación de tcpdump para los sistemas Windows que se llama WinDump y que hace uso de la librería Winpcap.

En UNIX y otros sistemas operativos, es necesario tener los privilegios del root para utilizar tcpdump.

El usuario puede aplicar varios filtros para que sea más depurada la salida. Un filtro es una expresión que va detrás de las opciones y que nos permite seleccionar los paquetes que estamos buscando. En ausencia de ésta, el tcpdump volcará todo el tráfico que vea el adaptador de red seleccionado.

Utilización frecuente de tcpdump:

- Para depurar aplicaciones que utilizan la red para comunicar.
- Para depurar la red misma.
- Para capturar y leer datos enviados por otros usuarios o ordenadores. Algunos protocolos como telnet y HTTP no cifran los datos que envían en la red. Un usuario que tiene el control de un router a través del cual circula tráfico no cifrado puede usar tcpdump para lograr contraseñas u otras informaciones.

5.5.9 Tcpextract

tcpxtract [50] es una herramienta para la extracción de ficheros del tráfico de red basándose en su estructura interna. Usando la técnica “data carving” se extraen ficheros basándose en tipos conocidos de cabeceras o colas de fichero. Al igual que la herramienta Foremost, emplea esta técnica, salvo que en este caso Tcpxtract intercepta ficheros transmitidos a través de la red. Otras herramientas con funcionalidad similar son EtherPEG y driftnet, que monitorizan y extraen ficheros gráficos en una red. Estas son comúnmente por administradores de red para vigilar la actividad de sus usuarios. La mayor limitación de estos programas es que tan solo soportan tres tipos de fichero y no existe una forma fácil de añadir más. La técnica de búsqueda que usan no es escalable y no busca en los límites de los paquetes.

Las características de tcpxtract son las siguientes:

- Soporta 26 formatos de fichero populares.
- Pueden añadirse nuevos formatos simplemente editando su fichero de configuración.
- Con una rápida conversión, se puede usar el fichero de configuración de Foremost con tcpextract.
- Algoritmos de búsqueda eficientes y muy escalables.
- Los algoritmos buscan a través de los límites de los paquetes para una total cobertura y calidad forense.
- Usa libpcap, una librería popular, portable y estable para la captura de datos de red.
- Puede usarse en una red o con un fichero de capturas de red de tcpdump.

5.5.10 Tcpflow

tcpflow [51] es un programa que captura datos transmitidos como parte de una conexión TCP (flujo - flow), y almacena los datos de forma conveniente para su depuración a análisis. Un programa como tcpdump muestra un resumen de paquetes que pasan vistos en el cable, pero normalmente no almacena los datos que actualmente están siendo transmitidos. Por contrario, tcpflow reconstruye los streams de datos actuales y almacena cada flujo en un fichero separado para su posterior análisis.

tcpflow entiende números de secuencia y reconstruirá correctamente streams de datos a pesar de las retransmisiones o error en la conexión. Sin embargo, actualmente no entiende los fragmentos IP, es decir, que los flujos con fragmentos IP no se almacenarán correctamente.

tcpflow está basado en la librería LBL Packet Capture Library (captura de paquetes) y por tanto soporta las mismas expresiones de filtrado que programas como tcpdump. Debería compilar en la mayoría de las versiones de UNIX.

tcpflow almacena todos los datos capturados en ficheros que tienen nombres con la forma:

```
128.129.130.131.02345-010.011.012.013.45103
```

donde los contenidos del fichero serán los datos transmitidos desde el host 128.129.131.131, por el puerto 2345, al host 10.11.12.13 por el puerto 45103.

El programa se escribió originalmente para capturar los datos que se envían varios programas que usan protocolos de red no documentados, en un intento de aplicarles la Ingeniería Inversa y descifrar su funcionamiento; por ejemplo: RealPlayer, MSN Messenger, etc.

5.5.11 TrueWitness

Linux, código abierto. Basado en India.

TrueWitness [52] es un comprensivo analizador forense digital creado por NatureSoft en asociación con Laser 5, un editor de distribuciones Linux japonés, y la consultoría Embedded Linux Consultancy.

Consiste en una herramienta de vigilancia de red que ayuda a monitorizar una red completa desde un puesto.

TrueWitness – Características clave

- Identifica amenazas internas y externas para la red
- Captura de paquetes en tiempo real
- Reconstruye sesiones de red convirtiendo los paquetes capturados en streams completos de datos
- Mantiene detalles para cada conexión TCP/IP
- No produce impacto en la velocidad de la red

Interfaz gráfica para el visionado y manejo de los datos

5.5.12 Etherpeek

[53] Las empresas cada vez usan más aplicaciones críticas a través de la red que necesitan la transmisión de video, voz, datos, etc., y todas suelen usar el protocolo IP y Ethernet.

La familia de productos EtherPeek de WildPackets ha sido diseñada específicamente para acelerar la localización de problemas en una red en tiempo real, proporcionando un sistema heurístico de detección y capacidad de diagnóstico específico de Ethernet.

Este analizador de red cubre la mayoría de los requerimientos para gestionar una red LAN, incluyendo monitorización, localización de problemas, análisis LAN remoto, análisis de VoIP y análisis del protocolo de la capa de aplicación, proporcionando un sistema de análisis en tiempo real.

EtherPeek VX

EtherPeek VX, es un analizador de paquetes de VoIP y Ethernet.

EtherPeek NX

EtherPeek NX, es un analizador de red Ethernet.

EtherPeek SE

EtherPeek SE, es un analizador del protocolo Ethernet

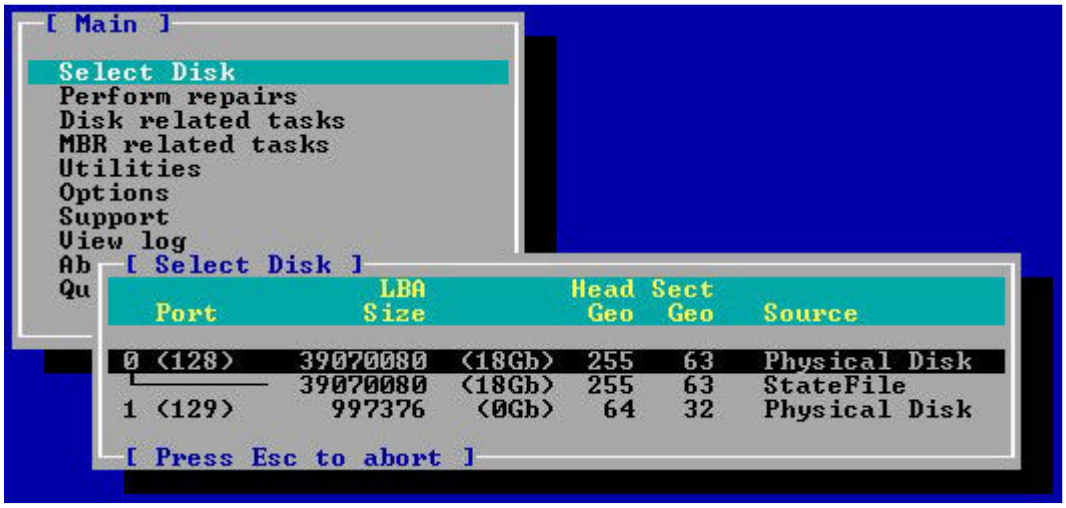
5.6 RECUPERACIÓN DE DATOS

5.6.1 BringBack

Esta aplicación [54] ha sido descrita previamente en el apartado de "Análisis de discos".

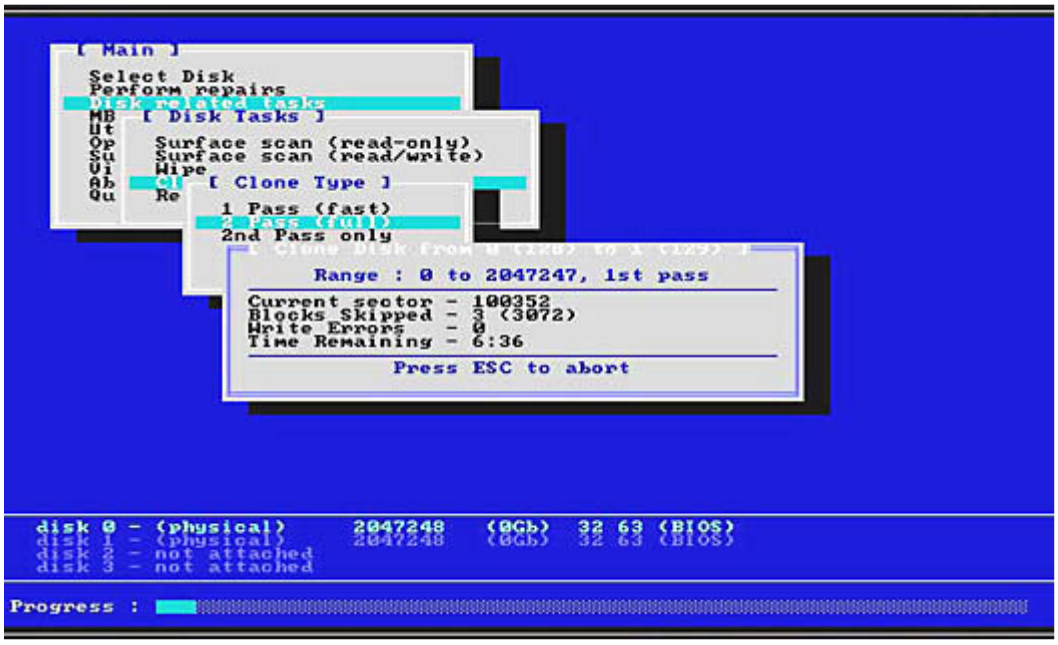
5.6.2 ByteBack Data Recovery Investigative Suite v4.0

ByteBack D.R.I.S.™ [55] es un programa que funciona bajo MS-Dos con interfaz de menú, diseñado para ayudar a resolver muchos problemas relacionados con los discos duros.



Resumen de características de ByteBack D.R.I.S.:

- Repara el MBR (Master Boot Record)
- Reparación de tablas de particiones lógicas.
- Reparación automática de los sectores del Boot (FAT, FAT32, NTFS)
- Copia de respaldo del MBR y la Tabla de Particiones
- Editor de disco (hexadecimal, ASCII, Tabla de particiones y boot del sistema)
- Gestor de la tabla de particiones: activar partición, ocultar partición, etc.
- Opción de deshacer
- Escaneo de la superficie del disco
- Clonado de discos y verificación
- Modo Forense
- Formateo a bajo nivel (fuerza el re-mapeo de sectores en busca de sectores erróneos)
- Borrado seguro de discos (estándar -con ceros-, con patrones o el llamado DOD)
- Logs de actividad
- Constructor de discos autorarrancables (disquete o CD)



Requerimientos del sistema:

- ByteBack D.R.I.S.™ solo opera sobre discos duros que puedan ser accedidos usando la interfaz extendida int13h, e incluyen: IDE, (x)ATA(x) y SCSI. Además puede operar sobre discos RAID.

- Sistema operativo: cualquier versión de MS-Dos 5.0 o superior. Es también compatible con el sistema operativo FreeDOS.

- Disquetera de 1.44 Mb o CD-Rom

- Microprocesador 80386 (compatible) o mayor

5.6.3 RAID Reconstructor

[56] Reconstruye discos RAID de nivel 0 y nivel 5 en tiempo real, a pesar de estar estropeados. Se trata de una aplicación para Windows escrita en el lenguaje Pascal que proporciona la capacidad de examinar discos RAID clasificados como:

- **RAID Level 5 Array** compuesto de 3 a 14 discos

- **RAID Level 0 Array** compuesto de 2 discos

Si no conocemos los parámetros RAID como orden de disco, tamaño de bloque y dirección de rotación, esta aplicación analizará las unidades de disco para determinar los valores correctos y poder extraer una imagen en otra unidad física.

5.6.4 Salvation Data

[57] Consiste en un conjunto de scripts de bajo nivel que leen y reparan los sectores erróneos de los discos duros Maxtor, Seagate, Fujitsu, etc., con comandos propietarios.

5.7 RECUPERACIÓN DE PARTICIONES

5.7.1 Partition Table Doctor

Partition Table Doctor [58] es una aplicación para recuperar particiones dañadas de forma lógica y no física. Comprueba y repara el Master Boot Record (MBR), la tabla de particiones y el sector de arranque de la partición con el error, de modo que se puedan recuperar las particiones borradas, perdidas o borradas. Recupera particiones FAT16, FAT32, NTFS, NTFS5, EXT2, EXT3 y SWAP de las unidades de disco duro IDE/ATA/SATA/SCSI.

Además permite crear un disquete o CD de emergencia para recuperar particiones si nuestro sistema operativo falla al arrancar. Está disponible para MS-DOS, FreeDOS, Windows 95/98/Me, Windows NT 4.0, Windows 2000, Windows XP y Windows 2003 Server.

Características principales:

- Recupera particiones borradas o perdidas (FAT16/ FAT32/ NTFS/ NTFS5/ EXT2/ EXT3/ SWAP).
- Muestra información lógica y física de la unidad de disco duro.
- Arregla el Sector de Arranque de particiones FAT y NTFS.

- Se pueden previsualizar los ficheros y directorios de arranque de cada partición antes de recuperarlos.

- Crea copias de respaldo y permite recuperar el MBR, la tabla de particiones y los sectores de arranque.

- Soporta unidades de disco duro IDE / ATA / SATA / SCSI.

5.7.2 Parted

Es la herramienta de Linux para la gestión de particiones. Está diseñado para crear, destruir, cambiar tamaño, copiar, y chequear las particiones y los sistemas de ficheros. Puede ser útil para crear espacio para otros sistemas operativos, reorganizar la utilización del espacio en disco, copiar datos entre discos duros y la extracción de imágenes de discos duros.

Consiste en una librería llamada "libparted" y una interfaz de línea de comandos, y sirve además como referencia de implementación.

Actualmente solo funciona bajo GNU/Linux, GNU/Hurd y BeOS.

Versiones Gráficas

GParted y QtParted son aplicaciones que poseen una interfaz gráfica y que usan las librerías de *parted.* Están adaptados para GTK+ y Qt, y a menudo se incluyen en los CDs autoarrancables (Live CDs) para hacer más sencillo el particionamiento; por ejemplo, Knoppix y SystemRescueCD.

5.7.3 Active Partition Recovery

Active@ Partition Recovery for DOS es un programa muy pequeño y sencillo basado en MS-DOS (solo 150k de tamaño) con el que podemos:

- Recuperar particiones borradas (FAT, FAT32 y NTFS)

- Restaurar unidades lógicas FAT y NTFS borradas.
- Escanear discos duros y detectar particiones borradas FAT y NTFS y/o unidades lógicas.
- Previsualización de ficheros y directorios en unidades o particiones borradas para recuperar datos.
- Realiza copias de respaldo y permite recuperar el MBR, la tabla de particiones y el Sector de Arranque.
- Muestra información física y lógica de los discos duros.
- Usar esta aplicación desde un disquete de arranque

Active@ Partition Recovery for Windows es una aplicación destinada a cuando se pierde una partición no arrancable; por ejemplo, si podemos arrancar Windows y ejecutar la aplicación podremos recuperar otros discos duros, unidades USB externas, tarjetas de memoria, etc. El fichero BOOT.INI se corrige automáticamente (en caso necesario) para mantener el sistema arrancable, y además se corrige los sectores de arranque y el MBR.

5.7.4 Testdisk

TestDisk [59] es un software gratuito diseñado para ayudar a recuperar particiones perdidas y/o a hacer de nuevo arrancables los discos no arrancables.

TestDisk consulta la BIOS del sistema operativo para encontrar los discos duros y sus características (tamaño LBA y geometría CHS). Realiza un chequeo rápido de la estructura del disco y la compara con la tabla de particiones para buscar entradas erróneas y repararlas. Si tenemos particiones perdidas o una tabla de particiones completamente vacía, TestDisk puede buscar particiones y crear una nueva tabla de particiones o incluso un nuevo MBR si fuera necesario.

TestDisk puede ejecutarse bajo:

- MS-DOS o FreeDOS,

- Windows (NT4, 2000, XP, 2003),
- Linux,
- FreeBSD, NetBSD, OpenBSD,
- Sun y
- Macintosh

TestDisk puede encontrar particiones perdidas de los siguientes sistemas de ficheros:

- BeFS (BeOS)
- BSD (FreeBSD/OpenBSD/NetBSD)
- CramFS, Compressed File System
- DOS/Windows FAT12, FAT16 y FAT32
- HFS y HFS+, Hierarchical File System
- JFS, IBM's Journaled File System
- Linux Ext2 y Ext3
- Linux Raid: RAID 1, RAID 4, RAID 5 y RAID 6
- Linux Swap (versiones 1 y 2)
- LVM y LVM2, Linux Logical Volume Manager
- Mapa de particiones Mac
- Novell Storage Services NSS
- NTFS (Windows NT/2K/XP/2003)
- ReiserFS 3.5, 3.6 y 4
- Sun Solaris i386
- Unix File System UFS y UFS2 (Sun/BSD/...)
- XFS, SGI's – Sistemas de ficheros con Journal

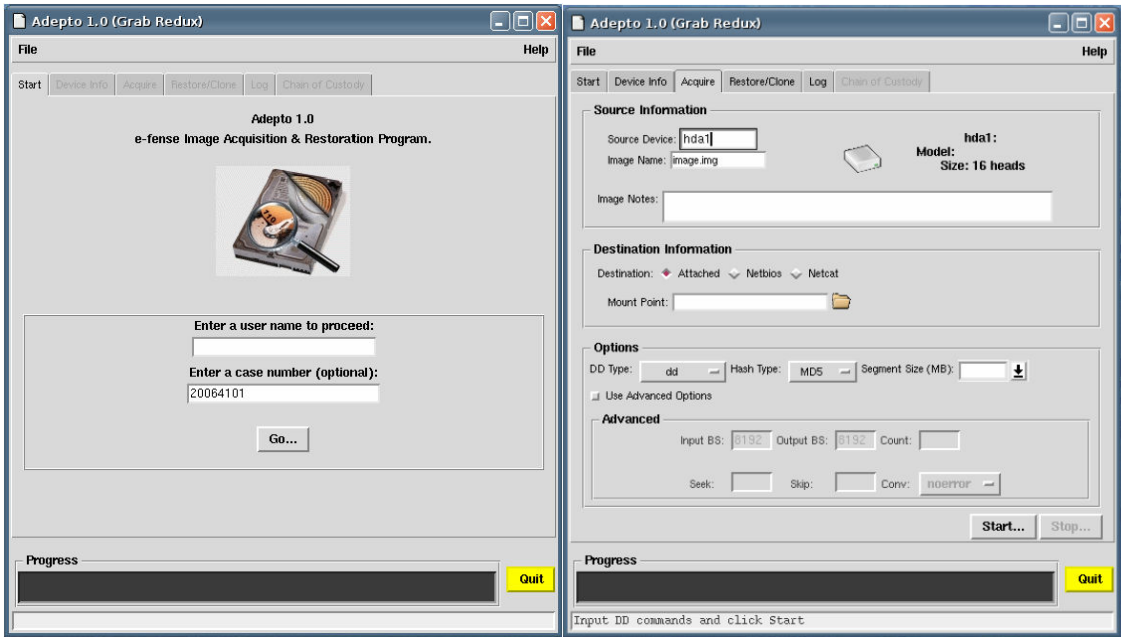
5.8 ADQUISICIÓN DE IMAGENES

Basados en Unix

5.8.1 ewfacquire

[60] Se trata de una parte de la librería Libewf y permite crear ficheros imagen en los formatos de EnCase y FTK (EWF-E01). Además permite crear imágenes con el formato SMART (EWF-S01). Esta librería permite comprimir los datos y calcular sus valores de CRC y MD5 y almacenarlos en el fichero imagen. Posee además un sistema de recuperación de errores inteligente.

5.8.2 Adepto (Grab)



Adepto [61] es una interfaz gráfica para dd/dcfldd/sdd y fue diseñado para simplificar la creación de imágenes forenses y automatizar la cadena de custodia. Fue desarrollado para ejecutarse en el CD de análisis forense Helix.

Características:

- Autodetección de unidades IDE y SCSI, CD-Roms y unidades de cinta.
- Posibilidad de elegir entre dd, dcfldd o sdd
- Verificación de imagen con MD5 o SHA1
- Compresión/Descompresión de imágenes con gzip/gzip2
- Extracción de imágenes a través de una red TCP/IP via Netcat/Cryptcat, o SAMBA (NetBIOS)
- Soporta unidades de cinta SCSI
- Limpia unidades o particiones (llenándola con ceros)
- Divide imágenes en múltiples segmentos
- Logs detallados con fecha/hora y comandos usados.

5.8.3 aimage

Es una parte del sistema AFF y permite crear ficheros en formato Raw (en bruto), AFF, AFD o AFM. Los formatos AFF y AFD pueden ser comprimidos o descomprimidos. Opcionalmente, se puede además calcular los valores MD5 o SHA-1 mientras se copian los datos. Posee una recuperación inteligente de errores de manera similar a *ddrescue*.

5.8.4 dcfldd

Es una versión de *dd* creada por el Digital Computer Forensics Laboratory. Dcfldd es contiene opciones adicionales forenses y de seguridad como el cálculo de valores hash MD5 o SHA1 "al vuelo", es decir, mientras se extraen los datos, y un borrado seguro de discos más rápido.

En concreto, dcfldd tiene las siguientes características adicionales:

- Cálculo de valores Hash "al vuelo".
- Se muestra el estado del proceso por la salida: cantidad de datos transmitida y tiempo restante.

- Borrado seguro de disquetes: borrado rápido o con patrones conocidos.

- Verificación de la imagen: Se puede verificar que la imagen es una copia exacta bit a bit del dispositivo a copiar.

- Verificación de borrado seguro: Se puede verificar que la superficie del disco solo contiene cero o el patrón introducido.

- Salidas múltiples: se puede enfocar la salida a varios ficheros o discos al mismo tiempo.

- Salida partida: permite dividir el fichero de salida en varios ficheros con mayor configurabilidad que con el comando *split*.

- Permite canalizar el fichero de salida y los datos de log hacia otros comandos (pipe - tubería), así como a otros ficheros.

5.8.5 dd

El comando dd (duplicate disk) es una aplicación muy común de UNIX cuyo objetivo principal es la copia a bajo nivel (bit a bit) de un fichero Unix hacia otro fichero Unix, incluyendo ficheros raw (en bruto). Este comando es bastante útil para transferir datos desde un dispositivo/archivo hacia un dispositivo/archivo/etc. Dd tienen un conjunto de opciones de línea de comandos diferente al comando cp de Unix (para copiar ficheros). Además, una copia con el comando cp no copiaría los ficheros borrados u ocultos, y tampoco copiaría el Slack Space (espacio sobrante a nivel de sector) de los ficheros.

La sintaxis básica del comando es la siguiente:

dd if=origen of=destino

donde **if** significa "*input file*", es decir, lo que queremos copiar y **of** significa "*output file*", o sea, el archivo destino (donde se van a copiar los datos); **origen** y **destino** pueden ser dispositivos (lectora de CD, disquetera, etc.), archivos, etc.

Es diferente a la tradicional filosofía de UNIX de usar “-opción valor”. En lugar de eso utiliza “opción=valor”.

Un ejemplo de uso podría ser el siguiente:

Clonando Diskettes:

Primero insertamos el diskette origen y escribimos lo siguiente en una consola:

dd if=/dev/fd0 of=~/diskette.img

Después insertamos el diskette destino (en blanco) y escribimos lo siguiente:

dd if=~/diskette.img of=/dev/fd0

Solo nos queda eliminar la "imagen" que creamos y listo...

rm -f ~/diskette.img

NOTA: El **~** significa "el *directorio home del usuario*", es similar a escribir **$HOME**

5.8.6 EnCase LinEn

Versión basada en Linux de la herramienta de extracción de imágenes de la versión basada en DOS de EnCase. Además de realizar las mismas funciones que la versión DOS, supone algunas mejoras, como trabajar con sistemas operativos no-Windows, discos duros extremadamente grandes y mayor velocidad de adquisición.

5.8.7 GNU ddrescue

GNU ddrescue [62] es una herramienta de recuperación de datos. Copia los datos de un fichero o un dispositivo de bloques (disco duro, cdrom, etc.) a otro, intentando recuperar los datos en caso de producirse errores de lectura.

Ddrescue no trunca el fichero de salida si no se le pide. De este modo cada vez que es ejecutado sobre el mismo fichero de salida, intenta rellenar los huecos.

La operación básica de ddrescue es totalmente automática. Es decir, no es necesario esperar a que se produzca un error, parar el programa, leer las anotaciones, ejecutarlo en modo inverso, etc.

Si se usa el fichero de anotaciones (logfile) de ddrescue, los datos son recuperados muy eficientemente. Además se puede interrumpir el rescate en cualquier momento y reanudarlo después en el mismo punto.

Combinación automática de copias de seguridad: Si se tienen dos o más copias dañadas de un mismo fichero, cdrom, etc., y se ejecuta ddrescue en todas ellas, una cada vez, sobre el mismo fichero de salida, se obtendrá probablemente un fichero completo y libre de errores. Esto es así porque la probabilidad de que existan áreas dañadas en los mismos lugares de diferentes ficheros de entrada es muy baja. Usando el fichero de anotaciones (logfile), sólo se intentan leer los bloques que se necesiten de la segunda copia y sucesivas.

El fichero log es salvado periódicamente en disco. De modo que en caso de bloqueo puede reanudarse el rescate sin apenas recopiado.

También el mismo logfile puede ser usado por múltiples comandos que copian diferentes áreas del fichero, y por múltiples intentos de rescate sobre diferentes subconjuntos de una misma área.

Ddrescue alinea su búfer de E/S al tamaño del sector de forma que pueda ser usado para leer de dispositivos en bruto (raw devices). Por razones de eficiencia, también lo alinea al tamaño de página de memoria si el tamaño de página es un múltiplo del tamaño de sector.

5.8.8 dd_rescue

[63] Una herramienta similar a dd, con la diferencia de que dd no continua leyendo el siguiente sector si se tropieza con sectores erróneos que no puede leer.

Diferencias:

- `dd_rescue` no proporciona conversión de caracteres.
- Diferente sintaxis de comandos.
- `dd_rescue` no finaliza cuando se produce un error en el fichero de entrada. Se especifica un número máximo de errores tras el cual finaliza el proceso.

- `dd_rescue` no trunca el fichero de salida a no ser que se indique.

- Se puede hacer el proceso de copiado desde el final de un fichero hasta el principio, en orden inverso.

- Utiliza dos tamaños de fichero, uno grande y otro pequeño.

5.8.9 iLook IXimager

Es la herramienta primaria de adquisición de imágenes de iLook. Está basada en Linux y produce archivos imagen autenticables con el formato de iLook. En concreto existen tres formatos destinados a cumplir tres objetivos:

1. El formato por defecto: **ILook Default Image Format (IDIF)** – Consiste en una cabecera con mecanismos para proteger la integridad de la imagen y una carga útil de datos comprimidos.

2. El formato sin comprimir: **ILook Raw Bitstream Format (IRBF)** – Se diferencia del IDIF en que la carga útil no está comprimida.

3. El formato encriptado: **ILook Encrypted Image Format (IEIF)** – La carga útil se encuentra encriptada

5.8.10 MacQuisition Boot CD/DVD

Proporciona una extracción segura de imágenes para unidades de disco duro Macintosh. Se trata de una aplicación con una interfaz de usuario intuitiva para usuarios novatos y con una herramienta de línea de comandos para usuarios expertos.

5.8.11 rdd

Rdd [64] es una herramienta basada en *dd* que proporciona robustez y otras funciones respecto a la lectura de errores, hashing MD5 y SHA-1, CRC, transferencia por red, particionamiento de la salida, etc.

Rdd fue desarrollado en el Instituto Forense Holandés (Netherlands Forensic Institute, NFI).

5.8.12 sdd

Se trata de una herramienta basada en *dd,* que mejora la velocidad en ciertas situaciones, cuando el tamaño de bloque de entrada es diferente del tamaño del bloque de salida. Muestra estadísticas más entendibles y durante el proceso muestra un indicador del tiempo restante. Además soporta el protocolo RMT (Remote Tape Server) para hacer copias remotas más rápida y fácilmente.

5.8.13 Otros

AccessData

Su ultima herramienta permite "LEER, ADQUIRIR, DESCIFRAR, ANALIZAR y EMITIR INFORMES"

ASR (SMART Acquisition)

Herramienta que forma parte de la aplicación SMART para adquirir imágenes y analizar discos.

EnCase

Encase puede extraer imágenes de discos solo en formato E0*.

FTK Imager by AccessData

Herramienta de la aplicación FTK que puede extraer imágenes y convertir formatos de fichero. Permite manejar los formatos E0* y dd.

iLook

Actualmente iLook V8 puede extraer imágenes solo en Windows.

Paraben

Un conjunto completo de herramientas para Windows.

ProDiscovery

Adquisición de imagines y búsqueda sobre sistemas de ficheros FAT12, FAT16, FAT32 y todos los NTFS.

X-Ways Forensics

Tiene algunas capacidades limitadas de adquisición de imágenes. La salida está en formato dd, es decir, en bruto.

X-Ways Replica

Realiza clonaciones y adquisiciones de imágenes. La salida está en formato dd, es decir, en bruto (copia bit a bit). Detecta automáticamente las areas protegidas ATA (HPA).

La mayoría de los entornos de Windows podrían acceder al disco original sin preguntar, alterando de esta manera los tiempos de acceso de los datos. Para solucionar esto, X-Ways Replica está basado en MS-Dos.

Este software actualmente no se encuentra disponible como un producto separado, sino que se incluye en el paquete Evidor.

5.9 OTRAS HERRAMIENTAS

Virtualizadores de sistemas

5.9.1 QEMU

QEMU [65] es un programa que ejecuta máquinas virtuales dentro de un sistema operativo, ya sea Linux, Windows, etc. Esta máquina virtual puede ejecutarse en cualquier tipo de Microprocesador o arquitectura (x86, x86-64, PowerPC, MIPS, SPARC, etc.). Está licenciado en parte con la LGPL y la GPL de GNU.

En principio, se trata de emular un sistema operativo dentro de otro sin tener que hacer reparticionamiento del disco duro, empleando para su ubicación cualquier directorio dentro de éste.

El programa no dispone de GUI, pero existe otro programa llamado QEMU manager que hace las veces de interfaz gráfica si se utiliza QEMU desde Windows. También existe una versión para Linux llamado qemu-launcher.

QEMU posee dos modos de operación:

- **Modo de usuario emulado:**

  Puede ejecutar procesos compilados para un tipo de CPU en otro CPU. Las llamadas al sistema son pensadas para endianness y desarreglos en 32/64 bits. Wine y Dosemu son los principales objetivos de QEMU.

- **Modo de emulación de Sistema computador completo:**
  Emula el sistema computador completo, incluyendo el procesador y varios periféricos. Puede ser usado para proveer varios almacenamientos de web virtual, en una sola computadora.

La Mayoría del programa está amparado bajo licencia LGPL, con el modo de usuario emulado bajo GPL.

5.9.2 VMware

[66] VMware Inc., filial de EMC Corporation que proporciona la mayor parte del software de virtualización disponible para ordenadores compatibles X86. Entre este software se incluyen **VMware Workstation**, y los gratuitos **VMware Server** y **VMware Player**. El software de VMware puede correr en Windows, Linux, y en breve debut lo hará en Mac OS X. El nombre corporativo de la compañía es un juego de palabras usando la interpretación tradicional de las siglas "VM" en los ambientes de computación, como *máquinas virtuales* (Virtual Machines).

VMware es un sistema de virtualización por software. Un sistema virtual por software es un programa que simula un sistema físico (un ordenador) con unas características hardware determinadas. Cuando se ejecuta el programa (simulador), proporciona un *ambiente de ejecución* similar a todos los efectos a un ordenador físico (excepto en el *puro acceso físico* al hardware simulado), con CPU (puede ser más de una), BIOS, tarjeta gráfica, memoria RAM, tarjeta de red, sistema de sonido, conexión USB, disco duro (pueden ser más de uno), etc...

Un virtualizador por software permite ejecutar (simular) varios ordenadores (sistemas operativos) dentro de un mismo hardware de manera simultánea, permitiendo así el mayor aprovechamiento de recursos. No obstante, y al ser una capa intermedia entre el sistema físico y el sistema operativo que funciona en el hardware emulado, la velocidad de ejecución de este último es menor, pero en la mayoría de los casos suficiente para usarse en entornos de producción.

VMware es similar a su homólogo **Virtual PC**, aunque existen diferencias entre ambos que afectan a la forma en la que el software interactúa con el sistema físico. El rendimiento del sistema virtual varía dependiendo de las carácterísticas del sistema físico en el que se ejecute, y de los recursos virtuales (CPU, RAM, etc...) asignados al sistema virtual.

Mientras que VirtualPC emula una plataforma x86, Vmware la virtualiza, de forma que la mayor parte de las instrucciones en VMware se ejecutan directamente sobre el hardware físico, mientras que en el caso de Virtual PC se *traducen* en llamadas al sistema operativo que se ejecuta en el sistema físico.

Productos gratuitos

VMware Placer

Es un producto gratuito que permite correr máquinas virtuales creadas con otros productos de VMware, pero no permite crearlas él mismo. Las máquinas virtuales se pueden crear con productos más avanzados como VMware Workstation.

Desde la liberación de VMware Player, han surgido páginas web donde es posible crear las máquinas virtuales, como [VMX Builder].

También es posible crear y redimensionar discos duros virtuales usando [qemu]. Por ejemplo, con la orden siguiente se creará una imagen de disco de 2Gb que puede ser usado con VMware.

qemu-img create -f vmdk mi-disco-duro-1.vmdk 2G

VMware ha establecido una comunidad alrededor de sus productos gratuítos, donde proporciona acceso a una creciente lista de máquinas virtuales gratuítas, y de libre disposición, con multitud de sistemas operativos y aplicaciones específicas preconfiguradas y listas para ejecutar.

También existen herramientas gratuitas para crear VMx, montar, manipular y convertir discos y disquetes VMware, para que los usuarios de VMware Player pueden crear y mantener VMs de manera gratuita, ***incluso para uso comercial***.

VMware Server (antes GSX)

En un principio era una versión de pago, hace unos meses fue liberada para ser descargada y utilizada de forma gratuita. Esta versión, a diferencia de la anterior, tiene un mejor manejo y administración de recursos; también corre dentro de un sistema operativo (host), está pensada para responder a una demanda mayor que el Workstation.

Productos Comerciales

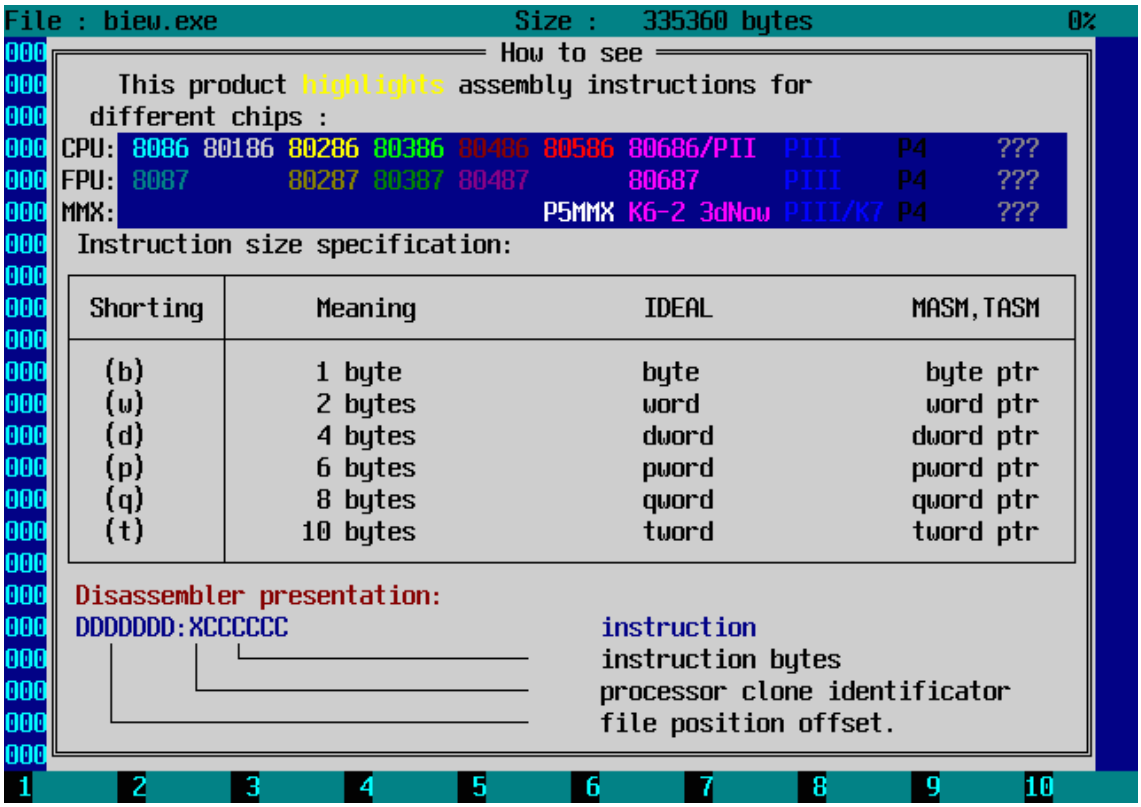
VMware Workstation

Es uno de los más utilizados pues permite la emulación en plataformas PC X86, esto permite que cualquier usuario con una computadora de escritorio o laptop pueda emular tantas máquinas virtuales como los recursos de hardware lo permitan. Esta versión es una aplicación que se instala dentro de un sistema operativo (host) como un programa estándar, de tal forma que las máquinas virtuales corren dentro de esta aplicación, existiendo un aprovechamiento restringido de recursos.

(i) VMware ESX Server

Esta versión es un sistema complejo de virtualización, pues corre como sistema operativo dedicado al manejo y administración de máquinas virtuales dado que no necesita un sistema operativo host sobre el cual sea necesario instalarlo. Pensado para la centralización y virtualización de servidores, esta versión no es compatible con una gran lista de hardware doméstico.

5.9.3 biew



[67] Se trata de un visor de ficheros binarios multiplataforma con un editor incorporado en binario, hexadecimal y en modo desensamblado, usando para esto último la sintaxis nativa de Intel.

5.9.4 hexdump

Esta herramienta de línea de comandos es un filtro que muestra el fichero especificado o la entrada estándar, en un formato especificado por el usuario: ascii, decimal, hexadecimal, octal.

5.9.5 Hex Workshop, de BreakPoint Software, Inc.

Este editor hexadedimal [68] para Windows que permite editar, cortar, copiar, pegar, insertar, borrar e imprimir con ficheros, así como exportar en formato RTF o HTML. Además se pueden realizar búsquedas, reemplazos, comparaciones, cálculos de resúmenes (hash), añadir favoritos, cambiar colores, etc.

Hex Workshop está integrado con el sistema operativo Windows de tal manera que soporta el "drag and drop", es decir, arrastrar y soltar, y se puede usar pinchando con el botón derecho en un fichero y eligiendo el programa en el menú contextual.

Contiene además un Visor Integrado de Estructuras para localizar estructuras de datos conocidas en C/C++. Este visor soporta las siguientes estructuras: char, byte, ubyte, word, uword, long, ulong, longlong, float, double, Fecha/Hora OLE, DOSTIME, DOSDATE, FILETIME, y time_t.

5.9.6 khexedit

KHexEdit [69] es un editor de ficheros binarios en formato RAW. Incluye funciones de búsqueda/reemplazo, favoritos, opciones de configuración, soporta "arrastrar y soltar", etc.

5.9.7 WinHex

WinHex [70] es un Editor Hexadecimal de Archivos, Discos y RAM, con otras opciones de recuperación de archivos, informática forense, y como herramienta de seguridad informática. Actualmente forma parte de la herramienta X-Ways.

- Editor de disco para FAT, NTFS, Ext2/3, ReiserFS, Reiser4, UFS, CDFS y UDF.

- Interpretación de sistemas RAID y discos dinámicos.
- Varias técnicas de recuperación de datos en sistemas FAT12, FAT16, FAT32, and NTFS. Se pueden recuperar archivos de tipo: jpg, png, gif, tif, bmp, dwg, psd, rtf, xml, html, eml, dbx, xls/doc, mdb, wpd, eps/ps, pdf, qdf, pwl, zip, rar, wav, avi, ram, rm, mpg, mpg, mov, asf y mid.
- Editor de RAM, una manera de editar RAM y la memoria virtual de otros procesos.
- Edición de estructuras de datos mediante plantillas
- Concatenar, partir, unir, analizar y comparar archivos
- Funciones de búsqueda y reemplazo especialmente flexibles
- Clonado de discos, con licencia especialista también sobre DOS
- Imágenes y backups de discos (comprimibles o divisibles en archivos de 650 MB)
- Programming interface (API) y scripts
- Encriptación AES de 256 bits, checksums, CRC32, digests (MD5, SHA-1, ...)
- Borrado seguro de datos confidenciales/privados
- Importación de todos los formatos de portapapeles
- Formatos de conversión: Binario, Hex ASCII, Intel Hex y Motorola S
- Juego de caracteres: ANSI ASCII, IBM ASCII, EBCDIC
- Salto instantáneo entre ventanas
- Convierte entre formato binario, hexadecimal, ASCII, IBM ASCII y Motorola S.
- Soporta ficheros mayores de 4 GB.
- Proporciona una extensa documentación y ayuda online.

5.10 ANTI-FORENSES

Las técnicas anti-forenses intentan frustrar las investigaciones forenses y sus técnicas. La informática forense es una ciencia relativamente nueva y por tanto estas técnicas son aún más recientes, por lo que no existe una categorización de referencia

para clasificarlas. Sin embargo, podemos afirmar que actualmente existen los siguientes tipos de herramientas anti-forenses.

Herramientas de Borrado Seguro

Cuando se borra algo de un sistema de ficheros, es posible recuperarlo mediante diversas técnicas que buscan en el espacio no asignado o en el slack space en caso de que se haya sobrescrito un fichero. Sin embargo, existen técnicas para hacer que un borrado sea definitivo. Al borrar un fichero simplemente cambian sus metadatos de forma que el fichero pasa a estado no asignado. De este modo los datos siguen aun en nuestro sistema de ficheros (formando parte del espacio libre) aunque el sistema operativo no haga referencia a esos datos. Éstas técnicas escriben un patrón conocido en la superficie del disco ocupada por el fichero para que de este modo sea imposible recuperar el fichero. Habitualmente los ficheros suelen llenarse de ceros o de datos aleatorios.

5.10.1 Declasfy

Se trata de una herramienta comercial para sistemas Windows. El programa está diseñado para hacer un borrado seguro de discos duros usando un método bastante fiable, usando estándares del Departamento de Defensa de los EEUU (DOD), aunque resulta bastante lento:

Primero, rellena el disco duro usando ceros hexadecimales

A continuación, usando unos hexadecimales (0xff)

Por último, usando caracteres o símbolos aleatorios.

Los estándares del DOD especifican un mínimo de 5 sobreescrituras, aunque esta herramienta realiza por defecto 3 sobreescrituras. Al finalizar el borrado, se buscarán sectores usando el direccionamiento LBA, que hace posible encontrar sectores que no sería posible ver de otro modo. De modo que a menudo suelen encontrarse sectores extra que son borrados y añadidos al espacio del disco duro.

5.10.2 Diskzapper

La herramienta comercial Diskzapper Dangerous, borra automáticamente todos los discos tan pronto como se produce el arranque de la computadora. No requiere ninguna acción del ususario. Esta herramienta está pensada para usarse sobre computadoras en las que no es conveniente o posible conectar un teclado y un monitor.

Diskzapper Extreme genera una secuencia aleatoria de bits y escribe cada sector con una secuencia aleatoria diferente. Este proceso se repite hasta 10 veces. De esta forma se previenen las dos técnicas forenses principales, que consisten en buscar datos residuales o en observar los estados magnéticos de los sectores del disco duro. Escribiendo muchas secuencias aleatorias, cualquier dato recuperado será confuso debido a que sus bits se han sobreescrito muchas veces con bits aleatorios.

Para asegurar la compatibilidad con programas de particionamiento de discos, una vez que Diskzapper Extreme ha escrito todos los sectores con datos aleatorios, escribe los primeros sectores con ceros, de modo que la tabla de particiones aparecerá vacía, en lugar de estar llena con bits aleatorios.

5.10.3 Bcwipe

Se trata de una herramienta comercial de borrado seguro de datos para sistemas Windows y Linux.

Soporta el estándar US DoD 5200.28-STD del Departamento de defensa de los EEUU, y el método de borrado seguro de Peter Gutmann.

Esta herramienta tiene varios modos de eliminar de manera segura los contenidos de un disco:

a. Eliminación con borrado seguro: Usando los menús contextuales se ofrece la opción al usuario de borrado seguro sobre un fichero o directorio.

b. Borrado seguro sobre el espacio libre del disco: Si previamente se han borrado archivos privados usando un comando estándar del sistema operativo, se puede limpiar además el espacio libre del disco.

c. Borrado seguro del fichero Swap de intercambio: Este borrado se realiza automáticamente cuando se elige el borrado seguro del espacio libre.

d. Borrado seguro de la papelera de reciclaje: Una opción en el menú contextual aparecerá cuando pinchemos con el botón derecho en la papelera, y nos dará la posibilidad de limpiar la papelera de reciclaje.

e. Borrado seguro de ficheros específicos de Windows ME: permite limpiar el contenido de algunos ficheros que crea este SSOO para su restauración.

5.10.4 Srm

srm es una pequeña aplicación con licencia GPL para usarlo por línea de comandos, similar al comando Unix "rm" pero con borrado seguro (Secure rm). A diferencia de rm, sobrescribe las unidades de datos de un fichero antes de cambiar su estado a no asignado.

5.10.5 Darik's Boot and Nuke (DBAN)

[71] Consiste en una aplicación gratuita de código abierto, compuesta de un disquete de arranque que realiza borrados seguros de los discos duros en la mayoría de las computadoras. Borrará completamente y de forma automática cualquier disco duro que detecte.

Caracterísitcas:

- Métodos de borrado:
  - Quick Erase
  - Canadian RCMP TSSIT OPS-II Standard Wipe
  - American DoD 5220-22.M Standard Wipe
  - Gutmann Wipe
  - PRNG Stream Wipe
- Plataformas soportadas:
  - Hardware:
    - Discos SCSI
    - Discos IDE, SATA y PATA
    - Funciona sobre cualquier computadora x86 de 32 bits con al menos 8 megas de memoria RAM.
  - Software
    - Soporta todas las plataformas de Microsoft y realiza un borrado seguro sobre sistemas de ficheros FAT, VFAT y NTFS:
      - MS-DOS, Windows 3.1
      - Windows 95, Windows 98, Windows ME

    - Windows NT 3.0, Windows NT 3.1, Windows NT 3.5, Windows NT 4.0
    - Windows 2000, Windows XP
- DBAN soporta todas las plataformas UNIX y realiza un borrado seguro sobre sistemas de ficheros ReiserFS, EXT, y UFS:
    - FreeBSD, NetBSD, OpenBSD
    - Linux
    - BeOS
    - QNX

5.10.6 DataEraser de OnTrack

Características del Producto

Arrancar y configurar

- DataEraser™ crea un disquete de autoarranque. El disquete se ejecuta independientemente del sistema operativo, por lo que se puede utilizar en cualquier PC compatible con IBM, sea cual sea su sistema operativo.

- Inicialice el programa con las sencillas opciones de menú. Una vez realizadas las selecciones de menú podrá ejecutarse el proceso de sobreescritura sin que usted tenga que intervenir. Alternativamente, puede monitorizar el avance de la acción mediante mensajes de "Tiempo restante".

Elegir la unidad de disco

- Sobreescriba determinadas particiones del disco o sobreescríbalo todo.

- DataEraser™ Professional Version puede también sobreescribir discos ATA/IDE y SCSI.

Seleccione su modalidad de sobreescritura

- DataEraser™ Personal Version puede sobreescribir un disco de una sola pasada.

- Con la versión DataEraser Professional puede elegir entre cuatro opciones de secuencia de sobreescritura:

    - Sobreescritura de una sola pasada.

    - Sobreescritura de triple pasada, que cumple las especificaciones del Departamento de Defensa estadounidense.

    - Sobreescritura de siete pasadas, que cumple la normativa alemana.

- Número de pasadas de sobreescritura definidas por el usuario (de una a noventa y nueve), que supera la normativa alemana y del Departamento de Defensa estadounidense. Elija su modalidad. Defina la modalidad de sobreescritura que desee o elija la predeterminada. ¡Y ya está! DataEraser alcanzará la velocidad máxima de sobreescritura.

5.10.7 shred

Herramienta UNIX de línea de comandos que es parte de las utilidades del nucleo GNU. Sobreescribe un fichero para ocultar sus contenidos y opcionalmente, elimina el fichero. Esto es así debido a que normalmente este comando se utiliza para limpiar ficheros del tipo /dev/hda, que no resulta conveniente borrar. Hay que decir que este comando no es efectivo en muchos sistemas de ficheros, por lo que podría producir errores e incoherencias en algunos discos.

5.10.8 Lenovo SDD

[72] Secure Data Disposal (SDD) es una herramienta para Windows usada para construir disquetes de arranque IBM PC-DOS, con el propósito de limpiar unidades de disco duro. Cada disquete de arranque utiliza el ejecutable Scrub3 DOS para facilitar el limpiado de unidades de disco duro.

Alteración/Eliminación de evidencias

5.10.9 Timestomp

[73] Una herramienta de línea de comandos que permite modificar los cuatro valores de marcas de tiempo NTFS: Modificado, Accedido, Creado, Borrado (MACE)

5.10.10 Evidence Eliminator

Evidence Eliminator es una herramienta comercial para sistemas Windows que permite eliminar:

- El fichero SWAP de intercambio
- Logs de aplicación
- Ficheros temporales
- Papelera de reciclaje
- Las copias de seguridad del registro
- Los datos del portapapeles
- El historial de los documentos abiertos recientemente
- El historial de las aplicaciones ejecutadas recientemente
- El historial de las búsquedas
- Las URLs escritas en internet explorer (i.e.) y netscape
- Los ficheros índice de i.e. (index.dat)
- La caché de i.e. y netscape
- El historial de i.e. netscape
- Contraseñas y autocompletado de i.e.
- Las cookies de i.e. y netscape
- Favoritos de i.e.
- Archivos temporales de i.e.
- Base de datos de Microsoft Outlook
- El historial de Windows Media Player
- Otros ficheros a elección del usuario

5.10.11 Tracks Eraser Pro

Tracks Eraser Pro es un programa comercial para Windows, diseñado para limpiar los rastros de la actividad de Internet en una computadora. Permite eliminar la caché, cookies, historial, URLs escritas, la memoria de autocompletado, el fichero index.dat del navegador, los ficheros temporales, el historial, el historial de búsqueda, el historial de ficheros abiertos/guardados, documentos recientes, etc.

Con los plug-ins gratuítos se puede eliminar facilmente el rastro de hasta 100 aplicaciones populares que usan internet; por ejemplo, las listas de RealPlayer, MediaPlayer y QuickTime, así como los ficheros recientes de Office, Acrobat, WinZip, y otros. También permite limpiar el registro y realizar borrado seguro sobre ficheros escribiendo datos aleatorios un numero de veces sobre ellos.

Ocultación de datos

Ésta técnica consiste en ocultar datos de manera que sea muy difícil o imposible detectarlos con métodos forenses, por ejemplo, usando la esteganografía.

5.10.12 Slacker

Es una herramienta [74] de línea de comandos que permite esconder un fichero dentro del Slack Space del sistema de ficheros NTFS.

5.10.13 Hiderman

Con Hiderman, se pueden esconder uno más ficheros dentro de otro. Por ejemplo, se pueden ocultar cualquier tipo de documento en muchos tipos de fichero, como imágenes, programas ejecutables, música, documentos de Word y Excel y archivos Winrar. El nuevo fichero creado será un fichero normal con un espacio extra ocupado por el documento oculto. Solo se podrá recuperar el fichero usando el mismo programa.

5.10.14 Cloak

Cloak es una herramienta de esteganografía, comercial y para sistemas windows, usada para encriptar y esconder ficheros dentro de imágenes de mapa de bits. Cloak protege los archivos usando algoritmos de encriptación (Cloak-128, Blowfish, Mercury), certificados de seguridad, compresión optimizada y contraseña de protección. Se puede asegurar cualquier tipo de fichero incluyendo los ficheros ejecutables (.exe). Las imágenes creadas que contengan ficheros ocultos, son completamente funcionales y son idénticas a la imagen original.

5.10.15 Runefs

Esta pequeña aplicación permite ocultar datos en los bloques asignados en los bloques dañados de una partición ext2. El primer inodo que puede ser reservado en un sistema de ficheros ext2 es de hecho el inode de bloques dañados (inodo 1) y no el inodo de raíz (inodo 2). Por razones de fallos en implementación es posible almacenar información en bloques marcados como bloques dañados, es decir referenciados por el inodo de bloques dañados.

Parte II: IMPLEMENTACIÓN

ANÁLISIS

Análisis de requerimientos

Nuestro objetivo es la realización de una aplicación que sirva como asistente para la realización de un análisis forense de una computadora.

La aplicación debe ayudar durante el proceso a seguir en un análisis forense post-mortem de un sistema de ficheros. Presuponemos que el usuario ha identificado el dispositivo del que tiene que extraer una imagen. Por lo tanto las etapas del proceso que debe cubrir son:

- Adquisición del Software: La aplicación debe proveer la capacidad de extraer el contenido de un dispositivo a un fichero imagen. Además debe permitir la verificación de la exactitud de la copia.

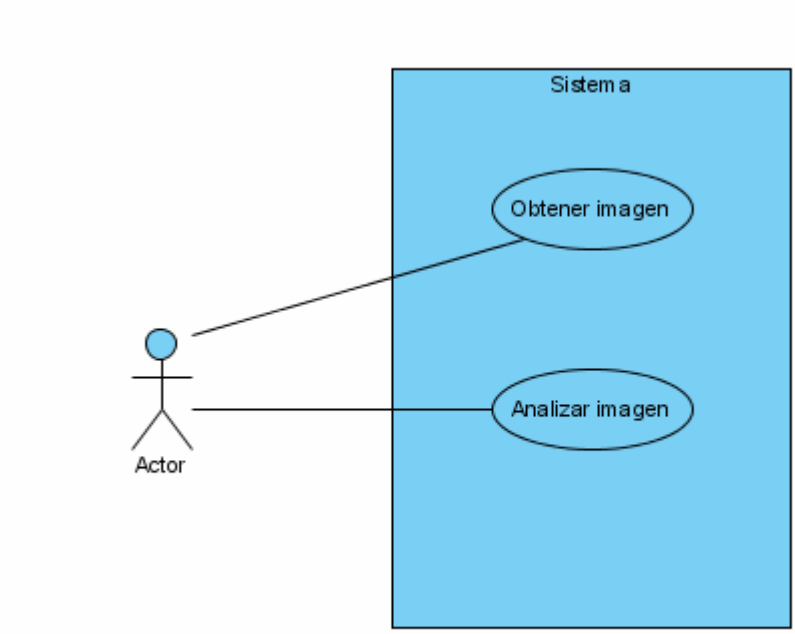
- Documentación: Durante todo el proceso se deberá registrar en un fichero las acciones del usuario para que, de manera didáctica, pueda después ver lo que ha hecho.

- Examen y Análisis
    - Extracción de Información: La aplicación debe proveer al usuario de las técnicas de extracción de datos de un sistema de ficheros, que incluyen las siguientes:
        - Sistema de Ficheros
            - Visualizar datos generales sobre el sistema de ficheros.
        - Contenido
            - Visualizar el contenido de un fichero dada su dirección lógica.
        - Meta-Datos
            - Búsqueda de ficheros basada en su dirección de meta-datos (Número de Inodo)
            - Visualizar el slack-space
            - Mostrar los meta-datos de un fichero dado su nombre
            - Búsqueda de ficheros basada en su tiempo MAC (Modificiación/Acceso/Creación)
        - Nombre de fichero (Interfaz Humana)
            - Búsqueda de ficheros según su nombre.

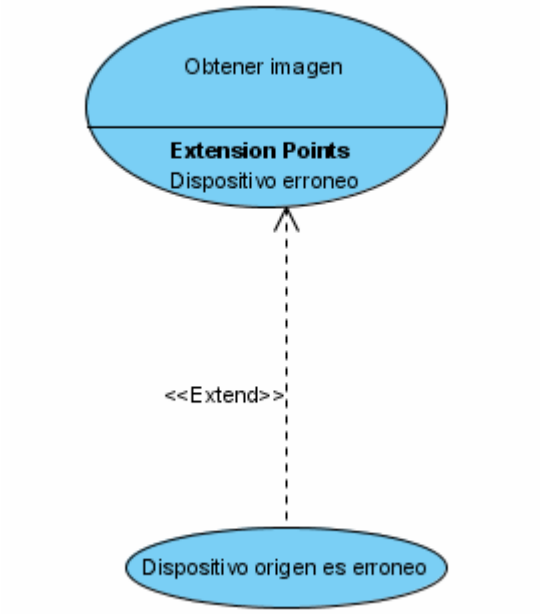
- Reconstrucción y Emitir los resultados: La aplicación deberá incluir un “bloc de notas” para anotar los descubrimientos del usuario y ayudar a la reconstrucción final.

Casos de Uso

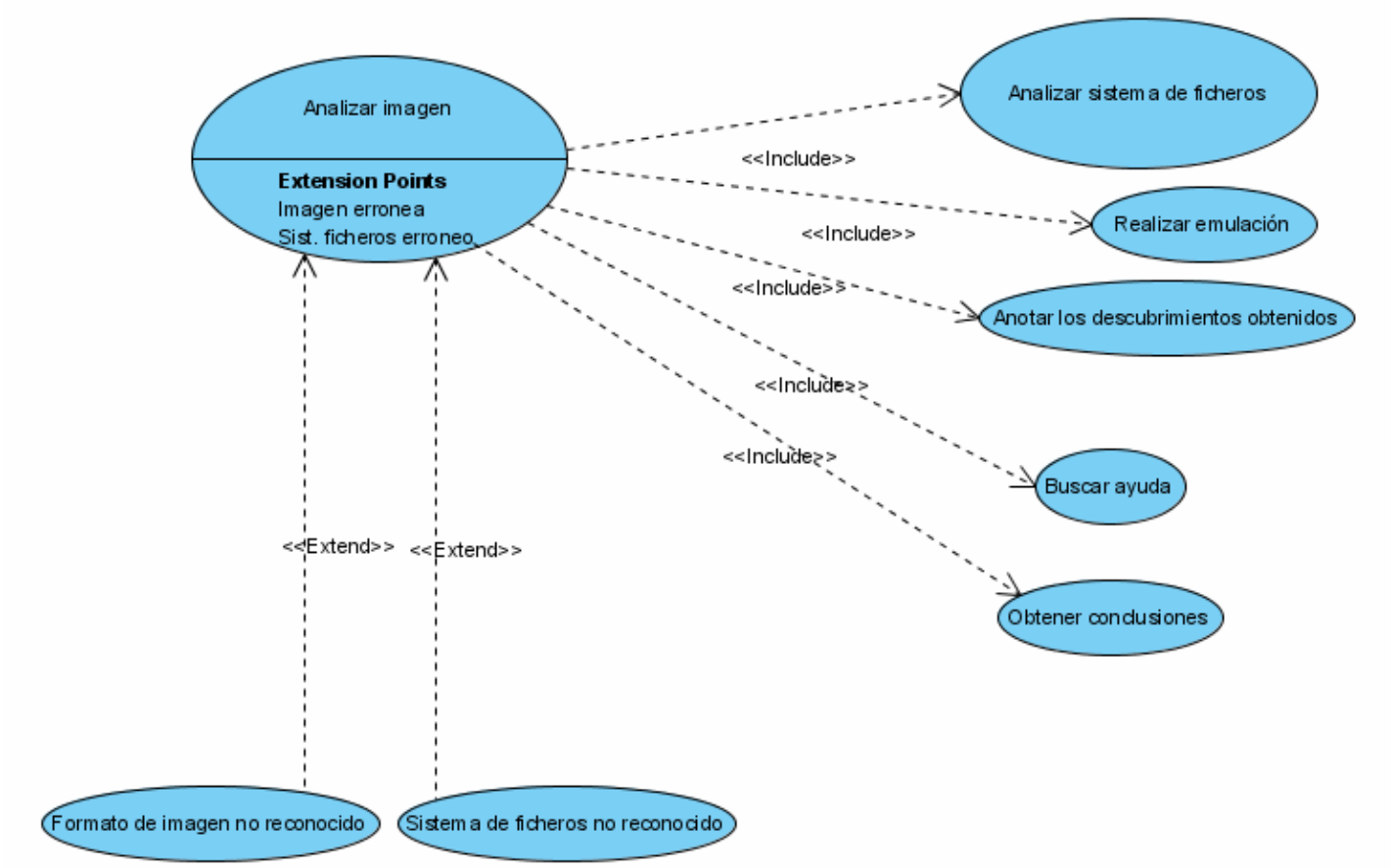
Diagrama frontera de la aplicación:



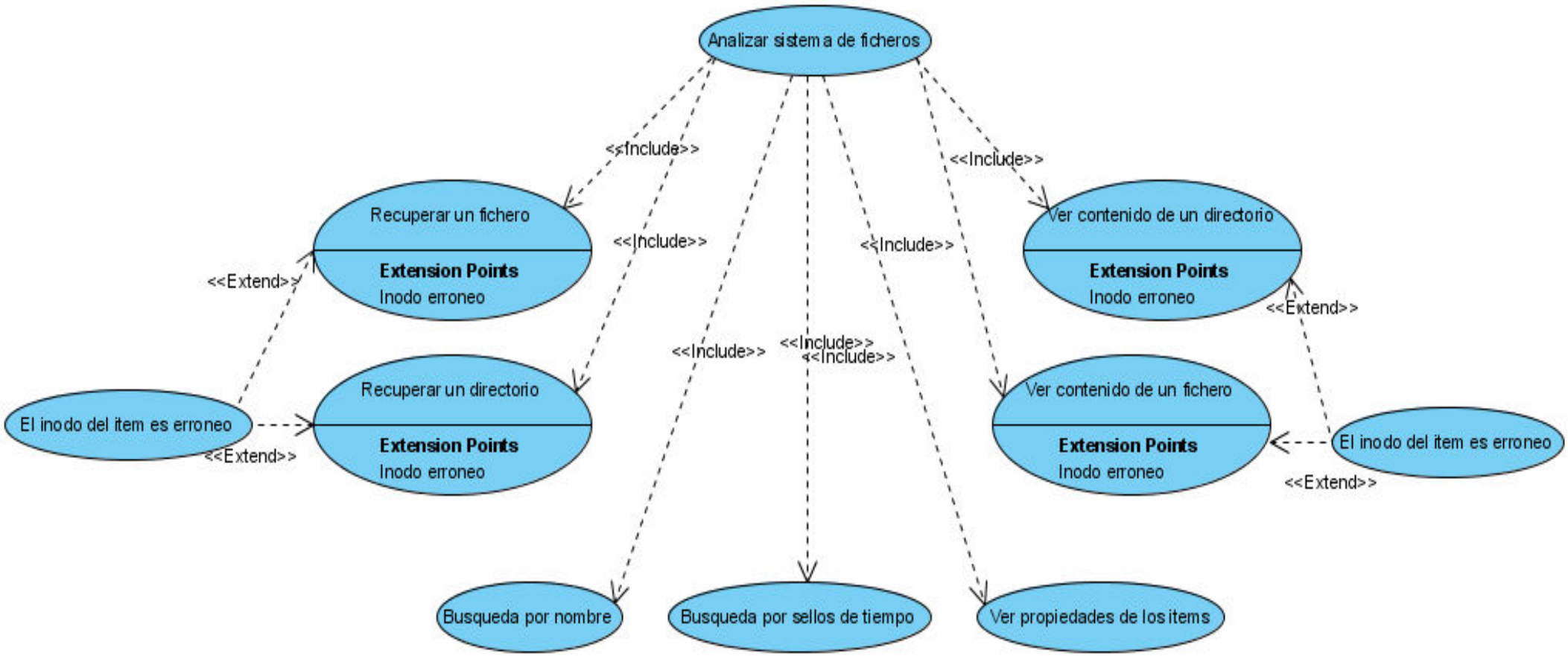
Caso de uso "Obtener Imagen":



Caso de uso “Analizar Imagen”



Caso de uso “Analizar Sistema de Ficheros”


Analizar sistema de ficheros
<<Include>>
<<Include>>
<<Include>>
<<Include>>
<<Include>>
<<Include>>
<<Include>>
Recuperar un fichero
Extension Points
Inodo erroneo
Recuperar un directorio
Extension Points
Inodo erroneo
Ver contenido de un directorio
Extension Points
Inodo erroneo
Ver contenido de un fichero
Extension Points
Inodo erroneo
<<Extend>>
<<Extend>>
<<Extend>>
<<Extend>>
El inodo del item es erroneo
El inodo del item es erroneo
Busqueda por nombre
Busqueda por sellos de tiempo
Ver propiedades de los items

Descripción de los casos de uso

Obtener imagen

- Actor participante: Usuario

- Condiciones de entrada:
  - El usuario dispone de un dispositivo de almacenamiento con espacio suficiente para guardar la imagen.

- Condiciones de salida
  - El usuario obtiene un fichero imagen con el contenido de un dispositivo.

- Flujo de eventos
  - El usuario elije el dispositivo origen del que se va a copiar la información.
  - El usuario elije la ruta y el nombre del archivo en el que se va a copiar la información del dispositivo.
  - El usuario acepta la transmisión de datos.

- **Dispositivo origen es erróneo**

  - Actor participante: Usuario

  - Condiciones de entrada:
    - El usuario dispone de un dispositivo de almacenamiento con espacio suficiente para guardar la imagen.

  - Condiciones de salida
    - Se muestra un mensaje indicando que el dispositivo elegido no es válido.

  - Flujo de eventos
    - El usuario ha elegido el dispositivo origen del que se va a copiar la información y la ruta del fichero destino.
    - El usuario ha aceptado la transmisión de datos.
    - El dispositivo elegido no existe o no es apto para extraer información.

- la misma.

- **Analizar imagen**

  - Actor participante: Usuario

  - Condiciones de entrada:
    - El usuario dispone de un dispositivo de almacenamiento con espacio suficiente para guardar los datos temporales y los ficheros recuperados de la aplicación.
    - El usuario conoce información acerca del caso en el que esta involucrado el fichero imagen.
    - El usuario dispone de un fichero imagen apto para su análisis

  - Condiciones de salida
    - El usuario obtiene información sobre el contenido del fichero imagen.
    - El usuario puede obtener conclusiones acerca de la información extraída.
  - Flujo de eventos
    - El usuario elije un fichero imagen para su análisis.
    - Se muestra el contenido del sistema de ficheros que contiene la imagen.
    - El usuario examina el sistema de ficheros en base a los datos que conoce previamente de la imagen.
    - El usuario obtiene datos relevantes y demostrables para sacar conclusiones acerca del caso al que pertenece la imagen.

- **Formato de imagen no reconocido**

  - Actor participante: Usuario

  - Condiciones de entrada:
    - El usuario dispone de un dispositivo de almacenamiento con espacio suficiente para guardar los datos temporales y los ficheros recuperados de la aplicación.
    - El usuario conoce información acerca del caso en el que esta involucrado el fichero imagen.

        - El usuario dispone de un fichero imagen cuyo formato no es reconocido por la aplicación.

    - Condiciones de salida
        - Se muestra un mensaje indicando que la imagen no se puede analizar.

    - Flujo de eventos
        - El usuario elije un fichero imagen para su análisis.
        - Se comprueba el formato del fichero y se indica que es erróneo.

- **Sistema de ficheros no reconocido**

    - Actor participante: Usuario

    - Condiciones de entrada:
        - El usuario dispone de un dispositivo de almacenamiento con espacio suficiente para guardar los datos temporales y los ficheros recuperados de la aplicación.
        - El usuario conoce información acerca del caso en el que esta involucrado el fichero imagen.
        - El usuario dispone de un fichero imagen cuyo formato es reconocido por la aplicación.
        - El fichero imagen contiene un sistema de ficheros que no es reconocido por la aplicación.

    - Condiciones de salida
        - Se muestra un mensaje indicando que la imagen no se puede analizar ya que contiene un sistema de ficheros erróneo.

    - Flujo de eventos
        - El usuario elije un fichero imagen para su análisis.
        - Se comprueba el formato del sistema de ficheros y se indica que es erróneo.

- **Emular imagen**

    - Actor participante: Usuario

    - Condiciones de entrada:
        - El usuario esta analizando el fichero imagen.
        - La imagen es arrancable (bootable)

- Condiciones de salida
    - Se muestra una pantalla con el resultado de la emulación del arranque de la imagen.

- Flujo de eventos
    - El usuario elije el tipo de dispositivo de la imagen.
    - El usuario elije la memoria RAM que se usará en la emulación.
    - El usuario elije otros parámetros opcionales.
    - El usuario activa la emulación de la imagen.

- **Anotar los descubrimientos obtenidos**

    - Actor participante: Usuario

    - Condiciones de entrada:
        - El usuario esta analizando el fichero imagen y ha descubierto algo que puede ser relevante para el caso en el que esta involucrado el fichero.

    - Condiciones de salida
        - Se guarda el descubrimiento en un fichero.

    - Flujo de eventos
        - El usuario esta analizando el fichero imagen.
        - El usuario abre un editor de textos y anota el descubrimiento.

- **Buscar ayuda**

    - Actor participante: Usuario

    - Condiciones de entrada:
        - El usuario esta analizando la imagen y precisa ayuda.

    - Condiciones de salida
        - El usuario obtiene la ayuda que necesita.

    - Flujo de eventos
        - El usuario esta analizando el fichero imagen.
        - El usuario necesita ayuda y abre una pagina web con enlaces a Internet y a ficheros locales con los que se puede documentar.

- **Obtener conclusiones**

- Actor participante: Usuario

- Condiciones de entrada:
    - El usuario ha terminado de examinar el fichero imagen y quiere sacar conclusiones acerca del mismo.

- Condiciones de salida
    - El usuario puede emitir conclusiones acerca del fichero imagen en relación al caso en el que esta involucrado.

- Flujo de eventos
    - El usuario ha analizado el fichero imagen para su análisis.
    - El usuario revisa las anotaciones realizadas durante el transcurso de la aplicación.
    - El usuario revisa los pasos realizados durante el análisis de la aplicación.

- **Ver contenido de un fichero**

    - Actor participante: Usuario

    - Condiciones de entrada:
        - El usuario esta analizando el fichero imagen y quiere ver el contenido de un fichero.

    - Condiciones de salida
        - Se muestra el contenido del fichero en el formato elegido por el usuario.

    - Flujo de eventos
        - El usuario elije un fichero dentro del sistema de ficheros.
        - El usuario elije la forma en la que quiere ver el fichero.
        - Se muestra el fichero en el formato elegido.

- **Ver contenido de un directorio**

    - Actor participante: Usuario

    - Condiciones de entrada:
        - El usuario esta analizando el fichero imagen y quiere ver el contenido de un directorio.

- Condiciones de salida
    - Se muestra el contenido del directorio.

- Flujo de eventos
    - El usuario elije un directorio dentro del sistema de ficheros.
    - Se muestran los ficheros y directorios que contiene el directorio.

- **Ver contenido de un directorio**

    - Actor participante: Usuario

    - Condiciones de entrada:
        - El usuario esta analizando el fichero imagen y quiere ver el contenido de un directorio.

    - Condiciones de salida
        - Se muestra el contenido del directorio.

    - Flujo de eventos
        - El usuario elije un directorio dentro del sistema de ficheros.
        - Se muestran los ficheros y directorios que contiene el directorio.

- **Recuperar un fichero**

    - Actor participante: Usuario

    - Condiciones de entrada:
        - El usuario esta analizando el fichero imagen y quiere guardar una copia exacta de un fichero.

    - Condiciones de salida
        - Se guarda el fichero elegido.

    - Flujo de eventos
        - El usuario elije un fichero dentro del sistema de ficheros.
        - El usuario elije la ruta y el nombre del fichero a guardar.
        - Se guarda el fichero en la ruta elegida.

- **Recuperar un directorio**

    - Actor participante: Usuario

- Condiciones de entrada:
    - El usuario esta analizando el fichero imagen y quiere guardar una copia exacta de un directorio junto con su contenido.

- Condiciones de salida
    - Se guarda el directorio elegido y su contenido.

- Flujo de eventos
    - El usuario elije un directorio dentro del sistema de ficheros.
    - El usuario elije la ruta y el nombre del directorio a guardar.
    - Se crea un directorio en la ruta elegida.
    - Se guarda el contenido del directorio de manera recursiva.

- **Búsqueda por nombre**

    - Actor participante: Usuario

    - Condiciones de entrada:
        - El usuario esta analizando el fichero imagen y quiere buscar un fichero o directorio por su nombre.

    - Condiciones de salida
        - Se muestra un listado con los ficheros que el usuario busca.

    - Flujo de eventos
        - El usuario introduce un nombre que será el patrón de búsqueda.
        - Se buscan los ficheros que coincidan en parte o totalmente con el patrón de búsqueda.

- **Búsqueda por sellos de tiempo**

    - Actor participante: Usuario

    - Condiciones de entrada:
        - El usuario esta analizando el fichero imagen y quiere buscar un fichero o directorio mediante sus sellos de tiempo o tiempos MAC.

    - Condiciones de salida
        - Se muestra un listado con los ficheros que el usuario busca.

- Flujo de eventos
    - El usuario introduce la fecha inicial y final entra las que se realizará la búsqueda.
    - Se busca en el sistema de ficheros los que estén entre las dos fechas introducidas.

- **Ver propiedades de los items**

    - Actor participante: Usuario

    - Condiciones de entrada:
        - El usuario esta analizando el fichero imagen y quiere visualizar las propiedades de un fichero o directorio.

    - Condiciones de salida
        - Se muestra un listado con las propiedades del ítem seleccionado.

    - Flujo de eventos
        - El usuario elije un fichero o directorio dentro del sistema de ficheros.
        - Se recopilan las propiedades del fichero.

- **El inodo del item es erroneo**

    - Actor participante: Usuario

    - Condiciones de entrada:
        - El usuario esta analizando el fichero imagen y quiere realizar una acción sobre un fichero o directorio cuyo i-nodo es erróneo.

    - Condiciones de salida
        - Se muestra un mensaje mostrando el error correspondiente.

    - Flujo de eventos
        - El usuario elije un fichero o directorio.
        - El usuario desea realizar una acción sobre el ítem.
        - El i-nodo del ítem es menor que 1, es decir, que al borrarse se han sobrescrito sus metadatos.

DISEÑO

Justificación de las herramientas elegidas:

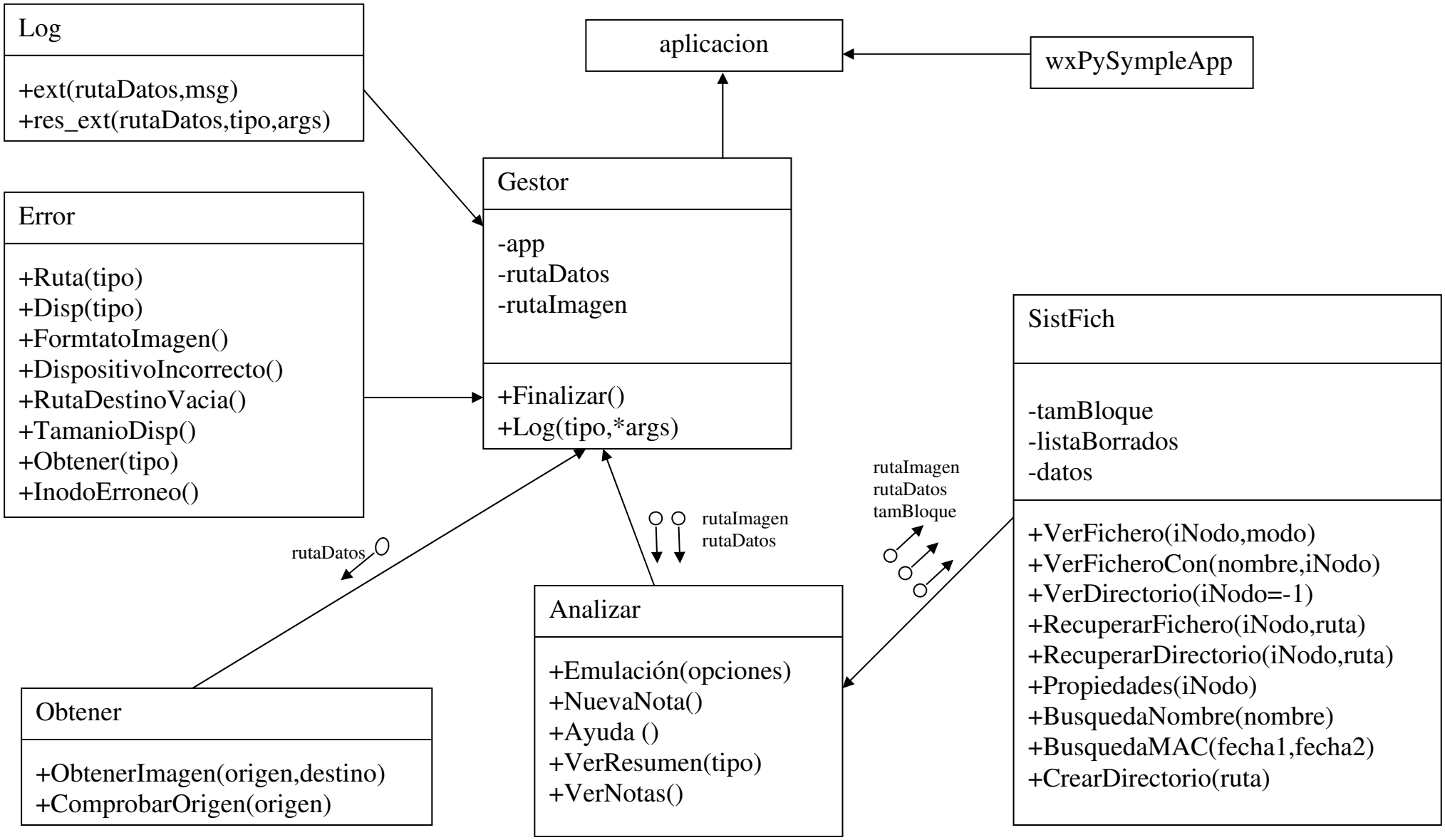
Una vez revisadas las herramientas software disponibles para el análisis forense, se han tomado las siguientes decisiones de diseño:

- Como base de la aplicación, se utilizará el lenguaje de programación Python.
- Para la interfaz gráfica se utilizará el lenguaje wxPython, que forma parte de wxWidgets.
- Para realizar la interfaz gráfica se utilizará la aplicación wxGlade.
- Para la adquisición de imágenes y su verificación se usará la aplicación "dcfldd".
- Para montar la imagen se utilizará el comando "mount" de UNIX. Algunas aplicaciones funcionarán sobre la imagen y otras sobre el sistema de ficheros montado en solo lectura.
- Para la gestión de la imagen y como centro de la aplicación se utilizara "The Sleuth Kit" que nos permitirá hacer lo siguiente:
    - Ver las propiedades del fichero imagen.
    - Ver las propiedades del sistema de ficheros que contiene.
    - Ver el contenido del sistema de ficheros en ASCII y Hexadecimal.
    - Ver los ficheros borrados.
    - Ver el contenido de los ficheros, incluyendo el slack space.
    - Ver los metadatos de los ficheros.
    - Buscar ficheros según su nombre y según sus sellos de tiempo (MAC Times)
- Se usará además el emulador de unidades virtuales QEMU para inicializar la imagen extraída en caso de que esta sea autoarrancable.

- Además la aplicación irá acompañada de software que haga posible visualizar el contenido de los ficheros y/o abrirlos según su formato. Estos son:

    - OpenOffice: Para abrir la mayoría de los documentos.
    - Evince: Para abrir los documentos PDF y PS.
    - Gthumb: Para abrir las imágenes.
    - File-Roller: Para abrir los ficheros comprimidos.
    - Xmms: Para abrir los archivos de música
    - VLC: Para abrir los archivos de video
    - XChm: Para abrir los ficheros con extensión Chm
    - Scite: Editor de textos para programadores usado para abrir el resto de ficheros.

Diagrama de Clases


Log
+ext(rutaDatos,msg)
+res_ext(rutaDatos,tipo,args)
aplicacion
wxPySympleApp
Gestor
-app
-rutaDatos
-rutaImagen
+Finalizar()
+Log(tipo,*args)
Error
+Ruta(tipo)
+Disp(tipo)
+FormtatoImagen()
+DispositivoIncorrecto()
+RutaDestinoVacia()
+TamanioDisp()
+Obtener(tipo)
+InodoErroneo()
SistFich
-tamBloque
-listaBorrados
-datos
+VerFichero(iNodo,modo)
+VerFicheroCon(nombre,iNodo)
+VerDirectorio(iNodo=-1)
+RecuperarFichero(iNodo,ruta)
+RecuperarDirectorio(iNodo,ruta)
+Propiedades(iNodo)
+BusquedaNombre(nombre)
+BusquedaMAC(fecha1,fecha2)
+CrearDirectorio(ruta)
rutaImagen
rutaDatos
rutaDatos
rutaImagen
rutaDatos
tamBloque
Analizar
+Emulación(opciones)
+NuevaNota()
+Ayuda ()
+VerResumen(tipo)
+VerNotas()
Obtener
+ObtenerImagen(origen,destino)
+ComprobarOrigen(origen)

PRUEBAS

Creación del CD-Live

Para crear el Cd de arranque se ha tomado el proceso de customización de un CD-Live de Ubuntu [75], concretamente la versión Dapper, "ubuntu-6.06.1-desktop-i386.iso"

Según [76], para poder ejecutar Ubuntu en nuestra computadora es necesario que ésta cumpla unos requisitos mínimos:

- Procesador Intel x86 o compatible a 200Mhz
- 256 MB de memoria RAM
- Unidad de CD-Rom
- Las BIOS del sistema debe ser capaz de arrancar desde CD-Rom.
- Tarjeta de vídeo estándar SVGA-compatible.

El proceso se ha dividido en 3 scripts, ya que al cambiar de usuario (chroot) el script aborta su ejecución. Para la ejecución correcta de los scripts necesitamos:

- Los 3 scripts situados en el directorio base
- El fichero "ubuntu-6.06.1-desktop-i386.iso" situado en el directorio base
- Conexión a Internet (para apt-get)

Para comenzar la ejecución debemos situarnos en el directorio base:

cd ~

y a continuación llamar al script 1 con permisos de root:

sudo ./script1

Introducimos nuestra contraseña, esperamos a que termine la ejecución y seguimos los pasos que se indican.

A continuación se mostrarán los 3 scripts que contienen los comentarios (#) suficientes para entender su funcionamiento:

Script1

```
echo "Script 1 iniciado"

echo "Creando el directorio live...."

#Accede al directorio base
cd ~
#Comprueba que el modulo squashfs esta instalado
modprobe squashfs

#Crea el directorio live que será el directorio principal
mkdir ~/live
#Copia la imagen del Cd en este directorio
mv ubuntu-6.06.1-desktop-i386.iso ~/live
#Accede al directorio live
cd ~/live

echo "Creado el directorio live"

#Crea un directorio para montar el Cd: mnt
mkdir mnt
#Monta la imagen en mnt
mount -o loop ubuntu-6.06.1-desktop-i386.iso mnt

echo "Montado el cd-rom"

echo "Extrayendo el contenido del cd-rom...."

#Crea un directorio para guardar el contenido del Cd y poder
manipularlo: extract-cd
mkdir extract-cd
#Copia en este directorio todo el contenido del Cd excepto el sistema
de ficheros
rsync --exclude=/casper/filesystem.squashfs -a mnt/ extract-cd

echo "Extraido el contenido del cd-rom"

#Crea un directorio para montar el sistema de ficheros: squashfs
mkdir squashfs
#Monta el sistema de ficheros en squashfs
mount -t squashfs -o loop mnt/casper/filesystem.squashfs squashfs

echo "Montado el squashfs"

echo "Copiando contenido al directorio edit...."

#Crea un directorio para copiar el contenido del sistema de ficheros y
poder manipularlo: edit
mkdir edit
#Copia el contenido de squashfs a edit
cp -a squashfs/* edit/

echo "Creado y llenado el directorio edit"

echo "Desmontando mnt y squashfs..."
```

```
#Desmonta la imagen del Cd
umount mnt
#Desmonta el sistema de ficheros
umount squashfs

#-----------------------------------------------------------------
#Se copian los ficheros de configuración de nuestro PC al sistema de
ficheros
#-----------------------------------------------------------------
echo "Copiando ficheros de configuracion..."

cp /etc/resolv.conf edit/etc/
cp /etc/hosts edit/etc/
cp /etc/apt/sources.list edit/etc/apt/

echo "Copiados los ficheros de configuracion a edit"

echo "Copiando la aplicacion...."

#-----------------------------------------------------------------
#Se copian los ficheros propios al sistema de ficheros
#-----------------------------------------------------------------

#Se copia la aplicación al directorio /home

cp -r /home/aplicacion edit/home/

#Se copia un lanzador al directorio /etc/skel. Este lanzador estará en
el directorio base del
#usuario que ejecute el Cd

cp /home/jose/aplicacion edit/etc/skel/

echo "Copiada la aplicacion a edit/home/app2"

#Se copia el script 2 al directorio raiz para luego poder llamarlo al
hacer el chroot
cp ~/script2 ~/live/edit/

echo "Entrando en la jaula: teclee './script2'"

#Vamos a hacer que todas las acciones a partir de ahora se ejecuten
sobre el directorio edit
chroot edit
```

Script2

```
echo "Estamos dentro de la jaula, instalando paquetes...."

#Se montan dispositivos necesarios para realizar acciones dentro de la
jaula
```

```
mount -t proc none /proc
mount -t sysfs none /sys

#Actualiza los paquetes disponibles
apt-get -y update

echo "Quitando...."

#--------------------------------------------------------------------
#Eliminación de paquetes innecesarios:
#
#     -Paquetes de idioma: se borran todos (195MB)
#     -Paquetes de juegos: se borran todos
#     -Reproductor de Video "Totem"
#     -Gestor de Scanner: Xsane
#     -Mensajería instantánea: Gaim
#--------------------------------------------------------------------
apt-get remove -y --purge \
language-pack-* \
language-pack-gnome-* \
language-support-* \
gnome-games \
totem \
xsane \
gaim

echo "Poniendo...."

#--------------------------------------------------------------------
#Instalación de de paquetes necesarios:
#
#     -Paquetes de idioma: Español
#     -Gtk2 para python
#     -Paquetes de programación: wxPython
#     -Navegador de internet: Mozilla Firefox
#     -Herramientas forenses: The Sleuth Kit
#     -Emulador: QEmu
#     -Extractor de imagenes: dcfldd
#     -Editores de texto: Gedit y Scite
#     -Reproductor de musica: Xmms
#     -Visor de documentos PDF: Evince
#     -Visor de documentos CHM: XChm
#     -Reproductor de video: VLC
#     -Compresor de archivos: File Roller
#--------------------------------------------------------------------

apt-get -y --force-yes install \
language-pack-es \
language-pack-es-base \
language-pack-gnome-es \
language-pack-gnome-es-base \
python-gtk2 \
python-wxgtk2.6 \
firefox \
sleuthkit \
qemu \
dcfldd \
gedit \
scite \
```

```
xmms \
evince \
xchm \
vlc \
file-roller


echo "Cambiando parametros de idioma al espaÃ±ol ..."

#Accede al directorio donde estan los scripts que configurarán el
sistema
#de ficheros

cd usr/share/initramfs-tools/scripts/casper-bottom/

#Cambia las variables locales para establecerlas al español:

#El teclado
sed 's/kbd=us/kbd=es/g' 19keyboard > 19keyboard2
#El idioma
sed 's/locale=en_US.UTF-8/locale=es_ES.UTF-8/g' 14locales > 14locales2

mv 19keyboard2 19keyboard
mv 14locales2 14locales

#Y les da permisos de ejecucion
chmod ugo+x 19keyboard 14locales

echo "Reconstruyendo el kernel..."

#Vuelve al directorio base
cd -
#Al cambiar los scripts se debe generar de nuevo el nucleo
mkinitramfs -o /initrd.gz 2.6.15-26-386

echo "Limpiando..."

#Limpiamos los paquetes descargados
apt-get clean
rm -rf /tmp/*

echo "Saliendo del chroot: teclee 'exit' y pulse intro"

echo "A continuacion teclee 'sudo ./script3' y pulse intro (duracion
de 10 a 15 minutos)"

#Desmonta los dispositivos montados al inicio del script
umount /proc
umount /sys
```

Script3

```
echo "Estamos fuera del chroot, moviendo el fichero initrd.gz...."

#Accede al directorio live
cd ~/live
#Copia el nuevo nucleo
mv edit/initrd.gz extract-cd/casper/

#Borra el script anterior
rm edit/script2

echo "Cambiando el idioma de la pantalla inicial al espaÃ±ol...."

#Instala los paquetes necesarios para poder reconstruir el menu que
aparece al iniciar el CD
apt-get -y --force-yes install dpkg-dev gfxboot
#Instala el codigo del menu
apt-get -y --force-yes source gfxboot-theme-ubuntu
#Accede al directorio
cd gfxboot-theme-ubuntu*/
#Lo construye pero en idioma español
make DEFAULT_LANG=es
#Copia el nuevo menu
cp -af boot/* ../extract-cd/isolinux/

echo "Fin de la configuracion del CD-Live"

echo "Creando manifest...."

#Accede al directorio live
cd ~/live

#Cambia los permisos del manifest para poder modificarlo
chmod +w extract-cd/casper/filesystem.manifest

#Crea un nuevo manifest a partir de las aplicaciones instaladas en el
sistema de ficheros
chroot edit dpkg-query -W --showformat='${Package} ${Version}\n' >
extract-cd/casper/filesystem.manifest

#Crea una copia del manifest
cp extract-cd/casper/filesystem.manifest extract-
cd/casper/filesystem.manifest-desktop

#Para que sea el manifest del directorio cambiando una palabra
sed -ie '/ubiquity/d' extract-cd/casper/filesystem.manifest-desktop

echo "Borrando anterior sistema de ficheros...."

#Borra el antiguo sistema de ficheros
rm extract-cd/casper/filesystem.squashfs

echo "Creando sistema de ficheros a partir del directorio edit...."

#Crea el nuevo sistema de ficheros a partir del directorio edit (puede
tardar de 5 a 10 minutos)
mksquashfs edit extract-cd/casper/filesystem.squashfs

echo "Fin de la creacion de sistema de ficheros"
```

```
echo "Creando md5....."

#Accede al directorio live
cd ~/live
#Borra el anterior valor MD5
rm extract-cd/md5sum.txt
#Crea el nuevo valor MD5
(cd extract-cd && find . -type f -print0 | xargs -0 md5sum >
md5sum.txt)

echo "Fin de la creacion del md5, ultimo paso, creando imagen...."

#Accede al contenido del CD
cd ~/live/extract-cd
#Crea la imagen a partir de este directorio
#--Nota: El nombre de la imagen resultante será "ubuntu-prueba.iso" y
puede modificarse en la linea de abajo

mkisofs -r -V "$IMAGE_NAME" -cache-inodes -J -l -b
isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-load-
size 4 -boot-info-table -o ../ubuntu-prueba.iso .

echo "Imagen creada, buen trabajo"

#Si desea emular la imagen creada con 'qemu' descomente la linea
siguiente:

#qemu -cdrom ../ubuntu-prueba.iso -boot d

#Si desea que se inicie el KQemu para emular la imagen mas
rapidamente:
# - Descargue de internet la ultima version de KQemu
# - Mueva el fichero kqemu.kmdr a su directorio base
# - Instale el ejecutor de scripts 'kommander', descomentando la
siguiente linea:
#
#apt-get -y --force-yes install kommander
#
# - Y por ultimo inicie la interfaz grafica y elija la imagen desde
ella (descomente las 3 lineas siguientes):

cd ~
echo "Iniciando KQemu..."
kmdr-executor ~/kqemu.kmdr >&error_kqemu.txt &
exit 0

#Si desea que se grabe directamente la imagen una vez creada
descomente esta linea:

#cdrecord dev=/dev/cdrom ../ubuntu-prueba.iso

echo "Fin del script3"
```

Pruebas con imágenes

Debido a la naturaleza de este proyecto y a la dificultad para encontrar imágenes de prueba, las pruebas realizadas no han sido las habituales para una aplicación normal.

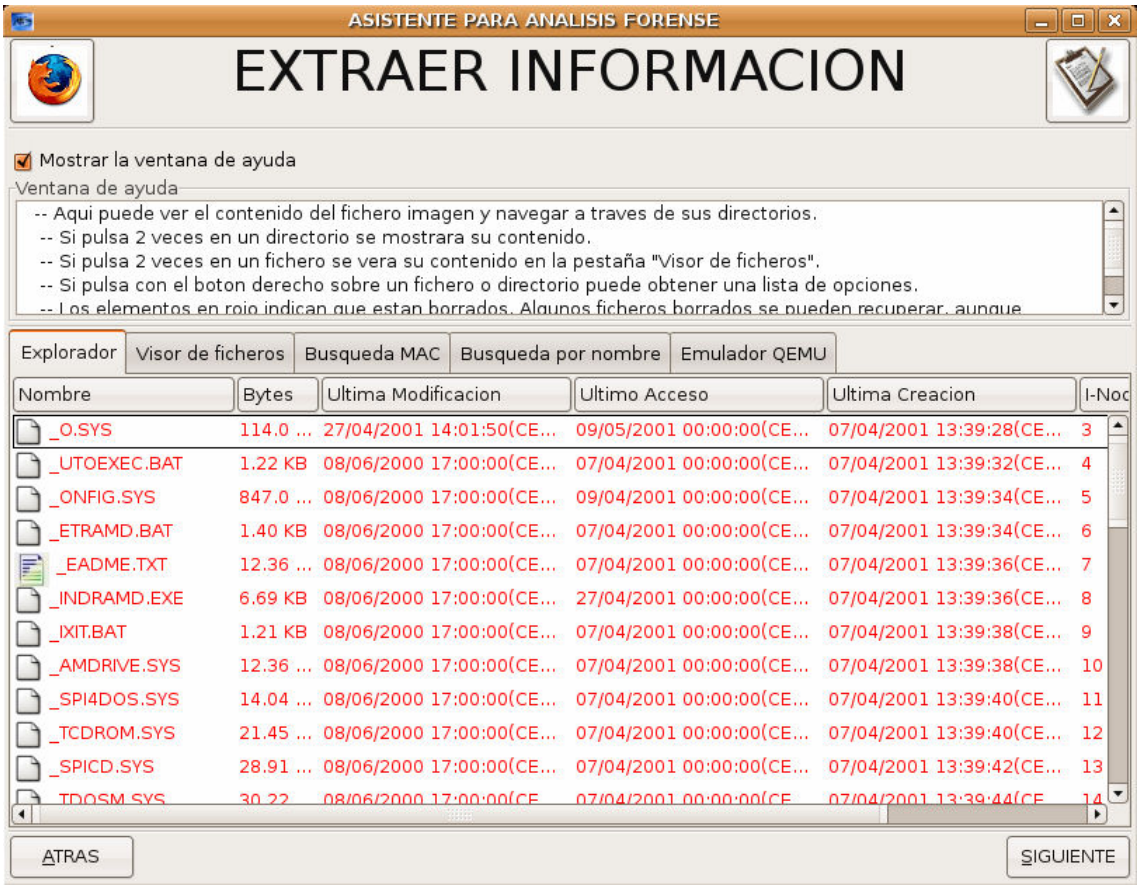
Se han tomado varias imágenes de distintos sistemas de ficheros y se han introducido en la aplicación realizando diversas pruebas sobre ellas según la naturaleza de la misma y su contexto. Los resultados han sido los siguientes:

Disquette de arranque MS-DOS:

Se ha tomado un disquette de arranque de MS-DOS y se ha hecho un formateo rápido desde Windows XP, de manera que al finalizar el formateo se instalasen los archivos de un sistema de ficheros de nuevo.

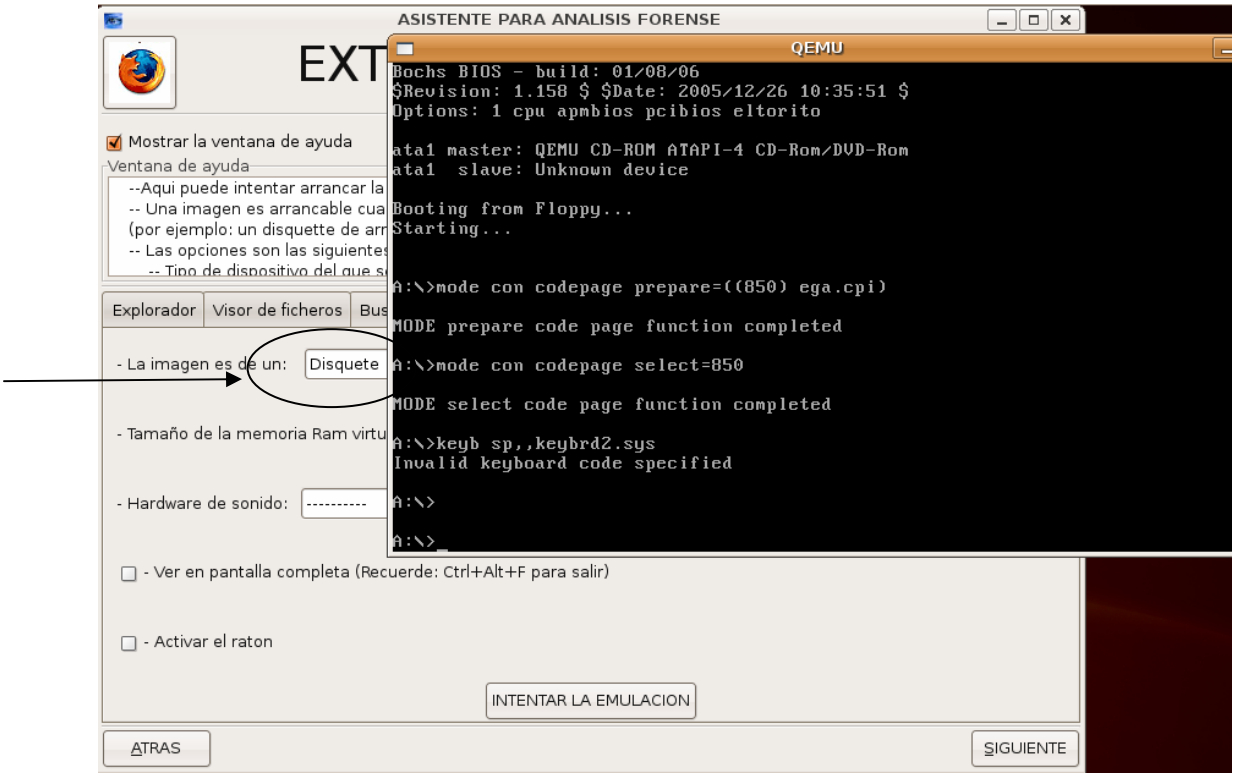
Al introducir la imagen el resultado fue este:

Es sistema de ficheros era reconocido, aunque el sistema operativo que se obtiene no es correcto. No obstante se puede analizar la imagen y los ficheros mostrados son los siguientes:



Se puede observar que los ficheros que había antes de formatear aparecen como ficheros borrados, ya que el formateo no ha sido a nivel de unidades de datos (formateo completo). Mas abajo se encuentran los mismos ficheros, pero sin borrar.

Al realizar una emulación de esta imagen el resultado es satisfactorio aunque hay que modificar algunos parámetros para que funcione, ya que la imagen es de un disquette.



Disquette perteneciente al reto nº 26 del poyecto HoneyNet

Se ha descargado [77] la imagen de un disquette con formato FAT 16 desde la pagina web de HoneyNet.org que contiene un reto para los analistas forenses. El contenido no es real, sino que esta manipulado para que su análisis sea más complejo.

El reto consiste consiste en que este contiene infinidad de pruebas que implican a un comprador y a su traficante de drogas, y debemos buscar pruebas en él que inculpen a ambos. Según la propia web, este es uno de los retos más difíciles, e incluye el uso de técnicas de esteganografía (descubrir datos ocultos en imágenes). Los enigmas que encierra y sus soluciones se muestran a continuación:

- **¿Quién es el que probablemente le suministra la droga a Jimmy Jungle?**

Probablemente sea John Smith. Para averiguar esto hay que encontrar una imagen JPEG en el sector 16896 y dentro de esta imagen, usando técnicas de esteganografía se puede ver el siguiente documento con formato Word:

```
Dear John Smith:

My biggest dealer (Joe Jacobs) got busted. The day of our scheduled
meeting, he never showed up. I called a couple of his friends and they
told me he was brought in by the police for questioning. I'm not sure
what to do. Please understand that I cannot accept another shipment
from you without his business. I was forced to turn away the delivery
boat that arrived at Danny's because I didn't have the money to pay
the driver. I will pay you back for the driver's time and gas. In the
future, we may have to find another delivery point because Danny is
starting to get nervous.

Without Joe, I can't pay any of my bills. I have 10 other dealers who
combined do not total Joe's sales volume.

I need some assistance. I would like to get away until things quiet
down up here. I need to talk to you about reorganizing. Do you still
have the condo in Aruba? Would you be willing to meet me down there?
If so, when? Also, please take a look at the map to see where I am
currently hiding out.

Thanks for your understanding and sorry for any inconvenience.

Sincerely,

Jimmy Jungle
```

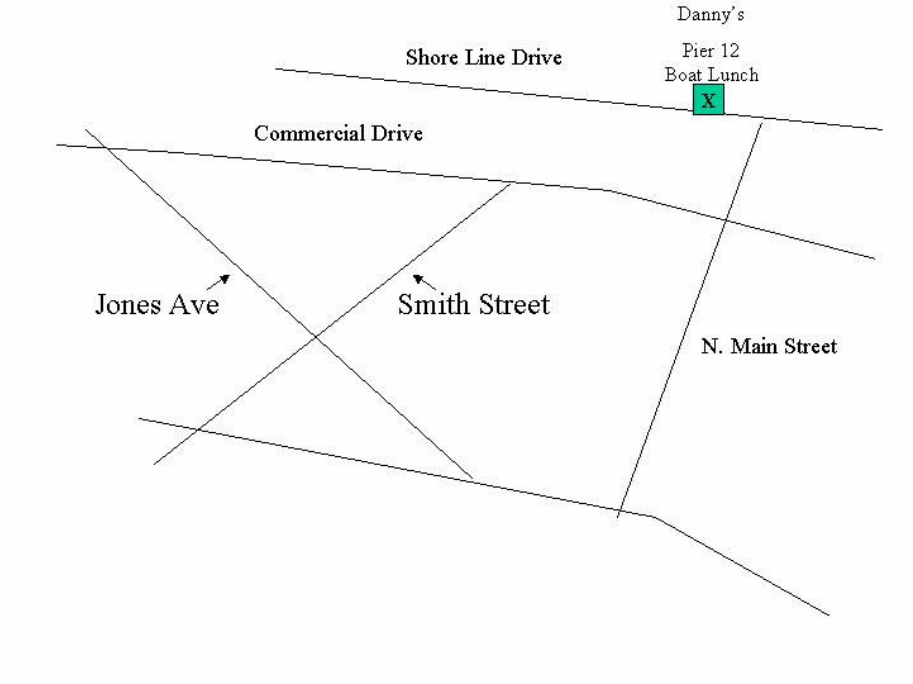
- **¿Cuál es la dirección postal del supuesto vendedor de Jimmy Jungle?**

Usando el comando “strings” sobre el fichero imagen se encuentra la siguiente dirección:

```
1212 Main Street
Jones, FL 00001
```

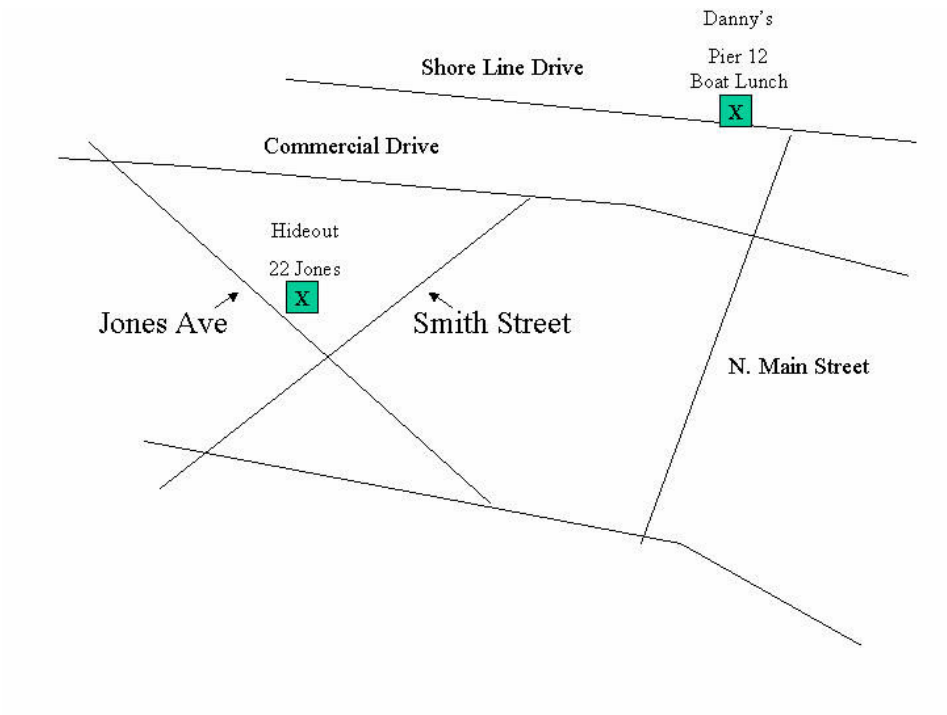
- **¿Cuál es la localización exacta donde Jimmy Jungle recibe las drogas?**

En el documento recuperado en la primera cuestión se puede ver que las drogas se dejaban en un sitio llamado Danny’s, y se enviaban en un barco. En la imagen recuperada se puede ver donde esta este emplazamiento:



- **¿Dónde esta escondido actualmente Jimmy Jungle?**

En el 22 de Jones Avenue. Esto podemos verlo en otra imagen de tipo BitMap recuperada en el sector 49664:



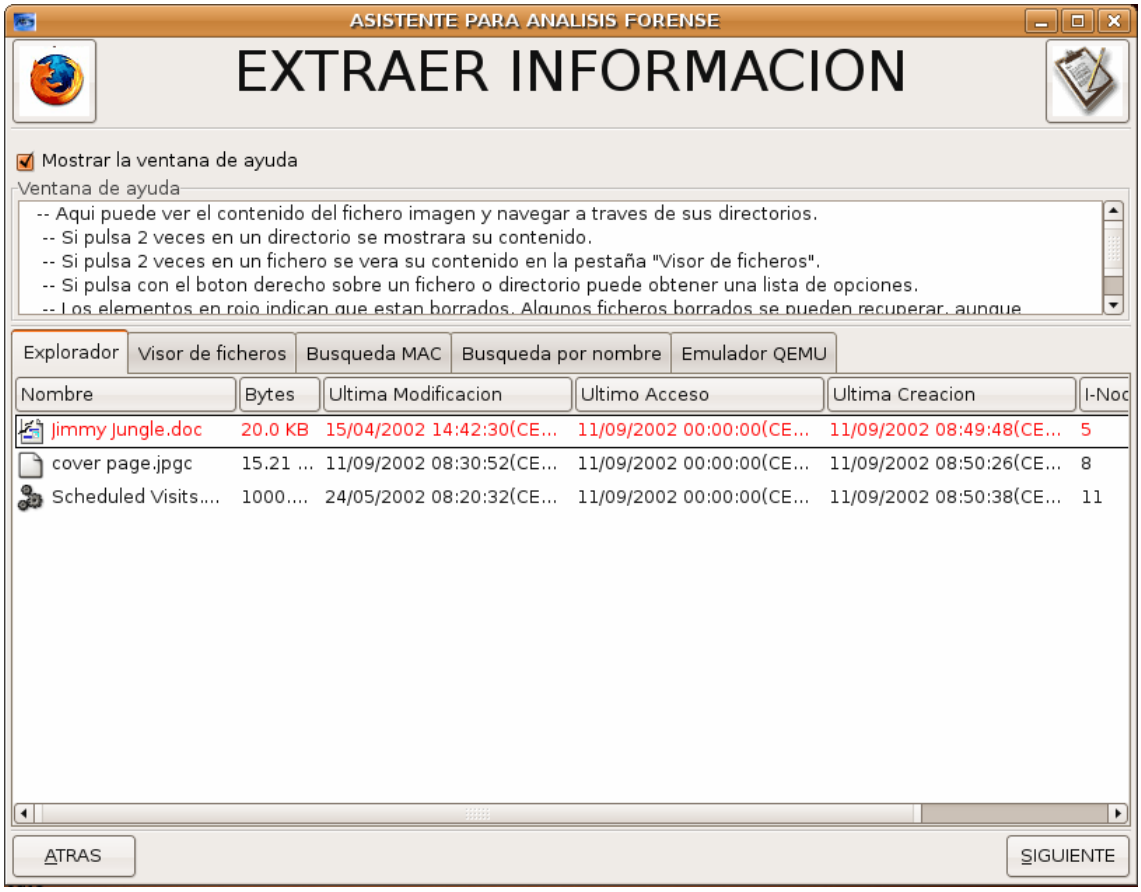
- **¿Qué clase de coche conduce Jimmy Jungle?**

Un Mustang azul de 1978. Usando técnicas de esteganografía se puede extraer de la imagen anterior un fichero de sonido (.Wav) en el que se le oye decirlo claramente.

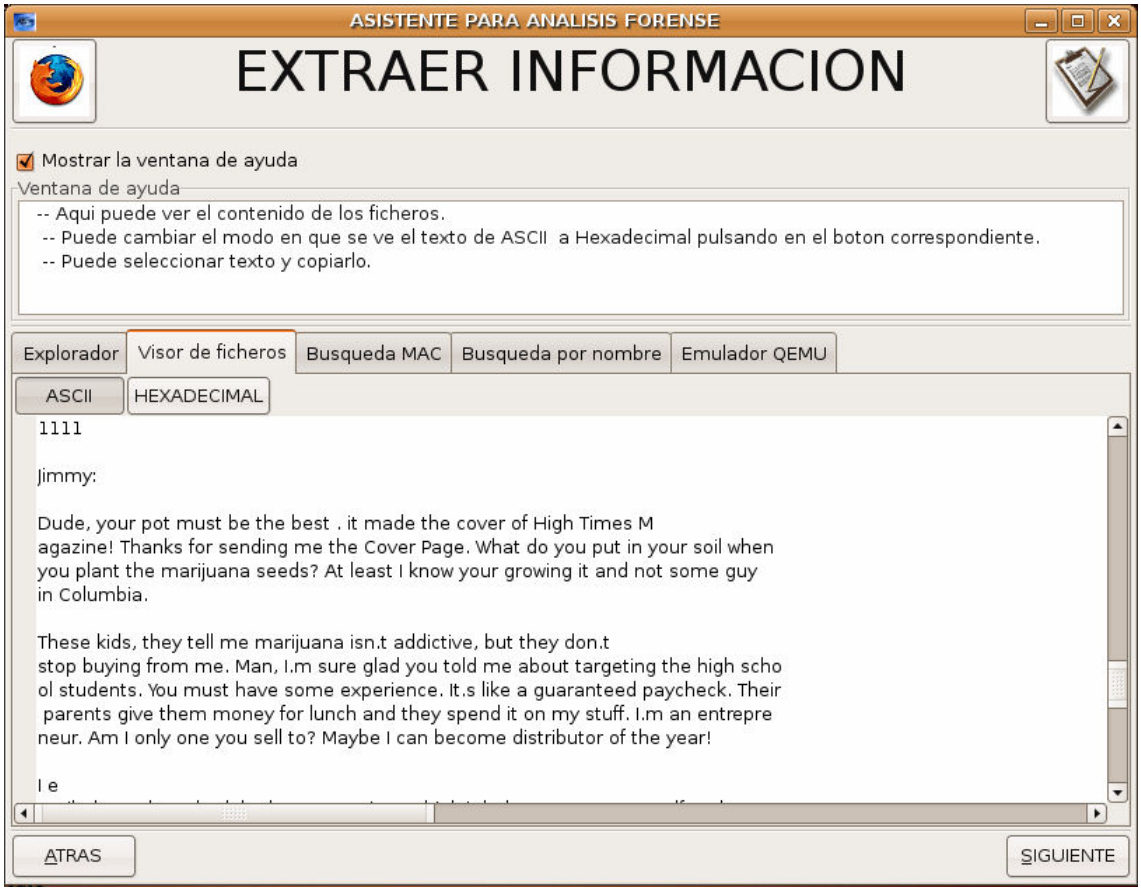
Análisis realizado con la aplicación:

Al introducir la imagen en la aplicación este fue el resultado:

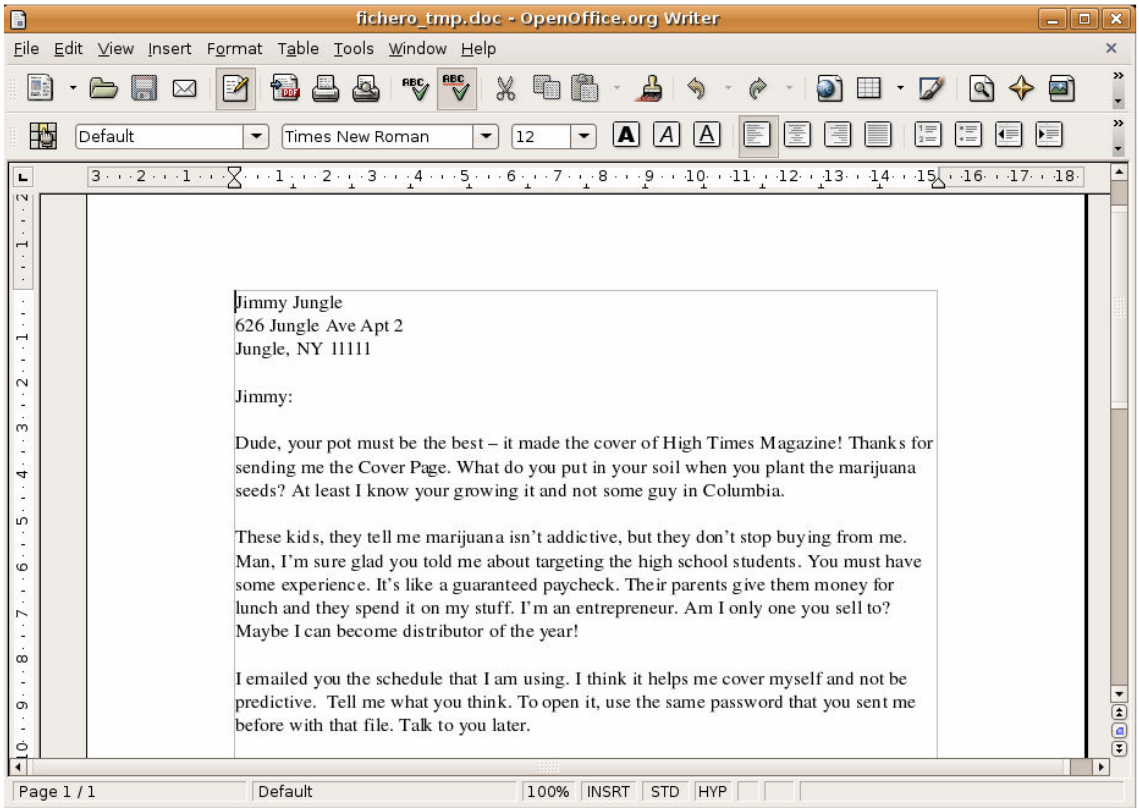
Y los ficheros que contenía:



De estos tres ficheros hay un fichero .doc borrado que contiene una carta escrita por el vendedor de droga, aunque su contenido no entra dentro de las preguntas que debemos responder:



Vamos a ver su contenido más claramente con Open Office:



Ademas hay otros 2 ficheros:

Cover Page.jpgc: Es un fichero que no contiene nada y que esta ahí para despistar a los analistas novatos que renombran el fichero a jpg.

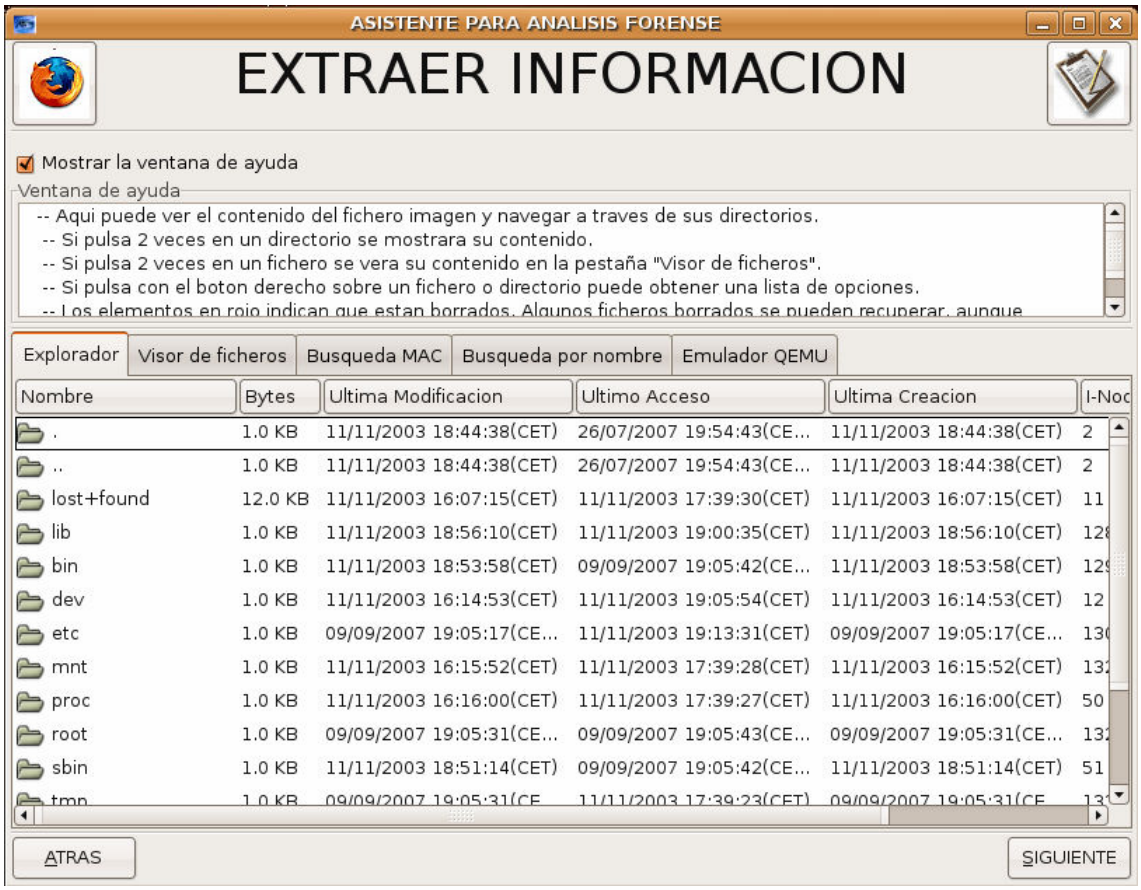
Sheduled Visists.exe: Es un fichero Zip protegido por una contraseña. Esta contraseña se debe buscar ejecutando el comando "strings" sobre la imagen. Esta búsqueda sería una locura en una imagen de mayor tamaño pero en esta se puede ver claramente una cadena de texto: "pw=goodtimes", por lo que ya tenemos la contraseña. Al intentar descomprimir el fichero nos damos cuenta de que es otro señuelo puesto para hacer perder el tiempo a los investigadores.

Con todo esto nos damos cuenta de que el análisis de esta imagen no es posible sin tener a mano herramientas de esteganografía (como por ej, Invisible Secrets 2002), y herramientas de búsquedas de ficheros a nivel de aplicación (por ej, FOREMOST).

Imagen bootable de Linux

Este es un fichero de 10MB con formato Ext2 de Linux, extraído de los ficheros de los ficheros de prueba de QEmu. Contiene la funcionalidad suficiente para ser autoarrancable. Al introducirlo en la aplicación el resultado es el siguiente:

El contenido de la imagen se muestra a continuación.



Y explorando este sistema de ficheros podemos comprobar los distintos ficheros y directorios que componen este núcleo de Linux.

Intentamos la emulación sin introducir opciones y el resultado de la emulación se muestra en la captura de pantalla siguiente:

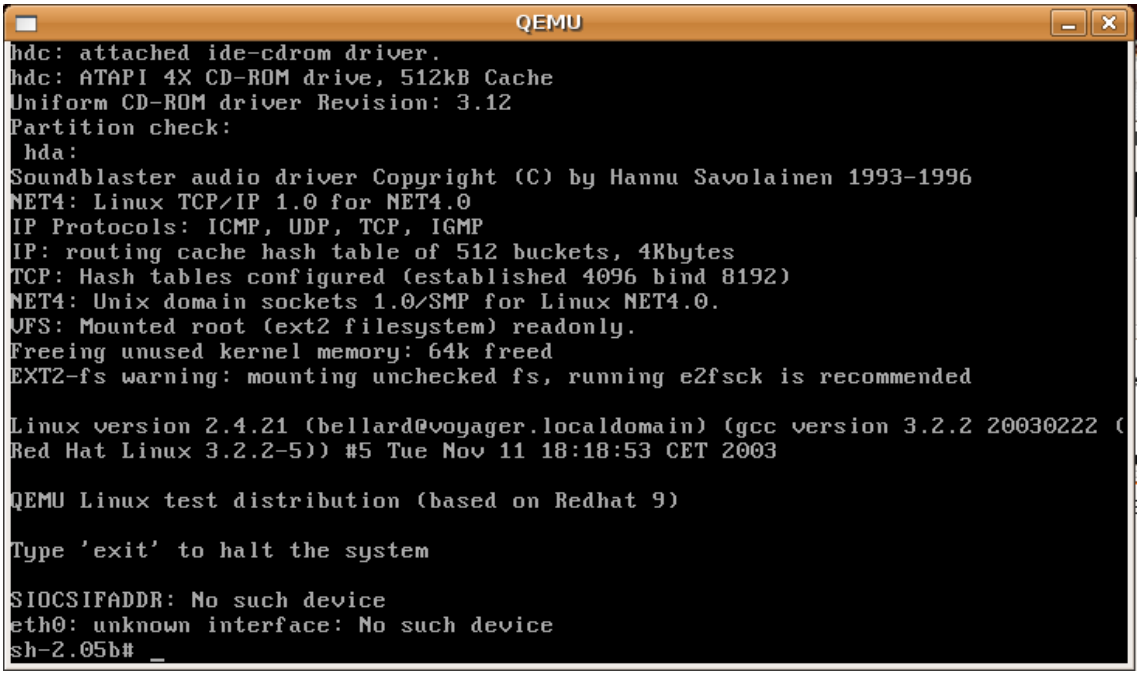


Imagen partida del "I Reto Rediris de Análisis Forense"

http://www.rediris.es/cert/ped/reto/ficheros.html

Descargamos un fichero comprimido llamado "192.168.10.tar.bz2" de la pagina web del I Reto Rediris de Análisis Forense [78]. Este fichero es una imagen de un sistema RedHat Linux, dividida en los siguientes ficheros:

```
b258f21b93e0eaa1f605dfd47d3f66f4  192.168.3.10-hda1.dd /boot
0b826f207f81bbc294bd059302a3558a  192.168.3.10-hda5.dd /usr
87669b6e6939619cdbc8ff8dfba574c4  192.168.3.10-hda6.dd /home
29ac32347b0bdda0659c061b46dce9e4  192.168.3.10-hda7.dd /var
4eed1213ed2aaa48f26e3edffd8a888d  192.168.3.10-hda8.dd /
c7da26612f6f7b318fb79064e2909500  192.168.3.10-hda9.dd swap
```

La prueba del I Reto de Análisis Forense consistía en analizar un sistema supuestamente comprometido por un Hacker, y conseguir responder a las siguientes preguntas cuya respuesta se relata tras cada una de ellas:

- ¿Quién ha realizado el ataque ?, (dirección IP de los equipos implicados en el ataque)

Sin duda esta es la pregunta más difícil de responder ya que una dirección ip se puede falsificar o se pueden usar ordenadores zombi para realizar ataques.

En cualquier caso, la dirección del atacante queda reflejada en el fichero “wtmp”:

```
ftp      ftpd7052     200.47.186.114   Fri Aug 23 00:22    gone - no logout
ftp      ftpd7049     200.47.186.114   Fri Aug 23 00:21    gone - no logout
```

- ¿Comó se realizo el ataque ? (Vulnerabilidad o fallo empleado para acceder al sistema)

La intrusión se realizó a las 00:12:15 del dia 23 de Agosto de 2002 y durante un par de horas los atacantes estuvieron realizando diversas acciones en el sistema para tomar el control del mismo, ademas de instalar herramientas que garanticen su futuro acceso. Para ello introducirse en el sistema aprovecharon una vulnerabilidad del cliente wu-FTP version 2.6.1. A continuación instalaron un rootkit llamado NeroD que inicia una puerta trasera en en la forma de un servidor ssh.

- ¿Qué hizo el atacante? (Que acciones realizo el atacante una vez que accedío al sistema, ¿por qué accedió al sistema?).

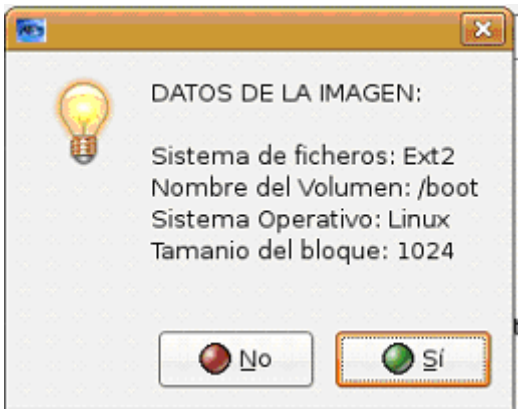
El atacante instaló un “bouncer” de IRC llamado psyBNC. Este programa sirve como Proxy para poder conectarse y chatear en canales IRC ocultando la dirección IP del que lo usa, ademas de otras funcionalidades. Junto con este se instaló también un bot de IRC llamado energyMech que amplia las posibilidades que ofrece un servicio IRC. Se instaló además un Sniffer para capturar contraseñas en la red de la computadora comprometida. Y por último se usaron diversas herramientas de un toolkit llamado AWU, que permiten descubrir nuevas vulnerabilidades FTP como la usada para entrar en el sistema.

Análisis con la aplicación

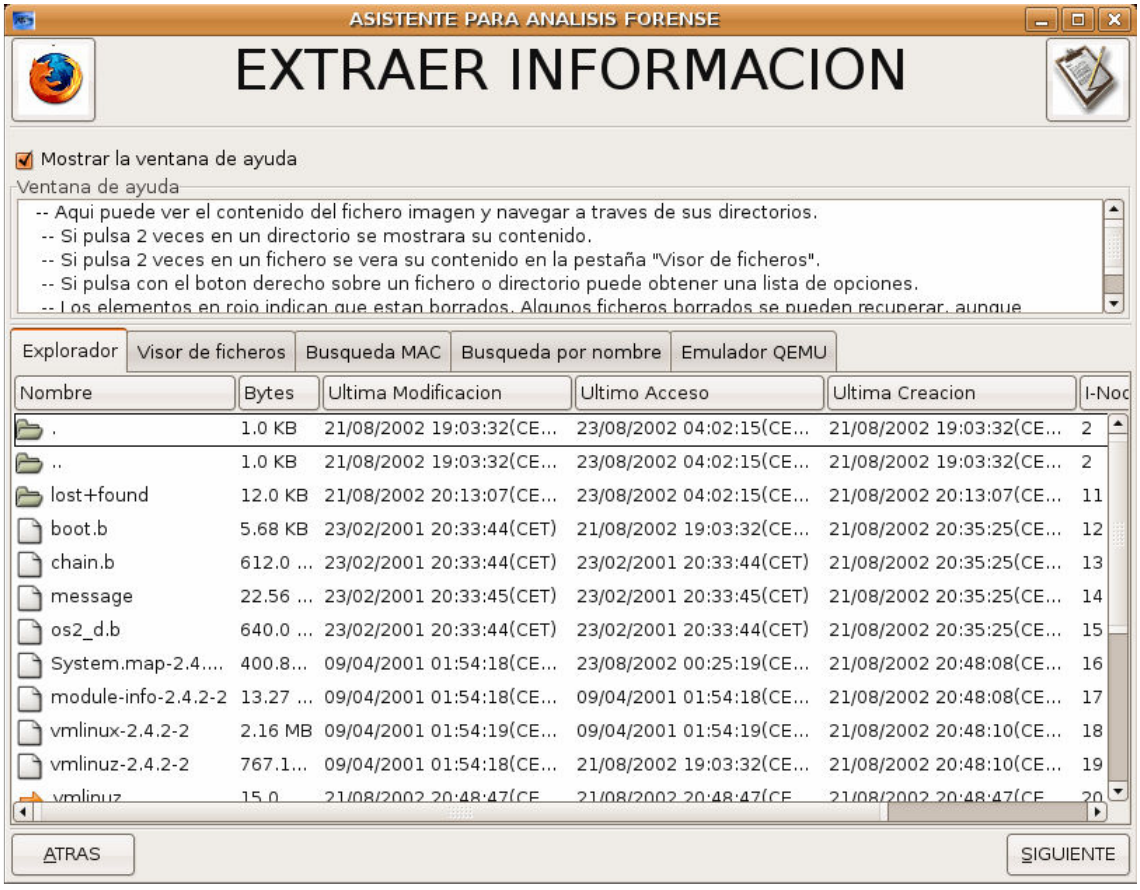
Para realizar este análisis se ha partido de la fecha de inicio del ataque (23 de Agosto de 2002) para buscar los ficheros implicados. El resultado de introducir cada una de las imágenes en la aplicación y realizares el siguiente:

192.168.3.10-hda1.dd (/boot)

Al seleccionar este fichero la aplicación muestra los siguiente:



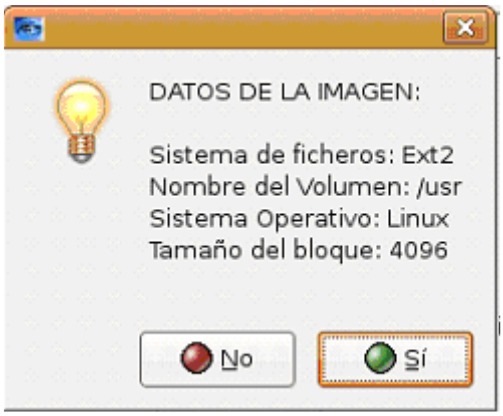
Veamos el contenido del sistema de ficheros:



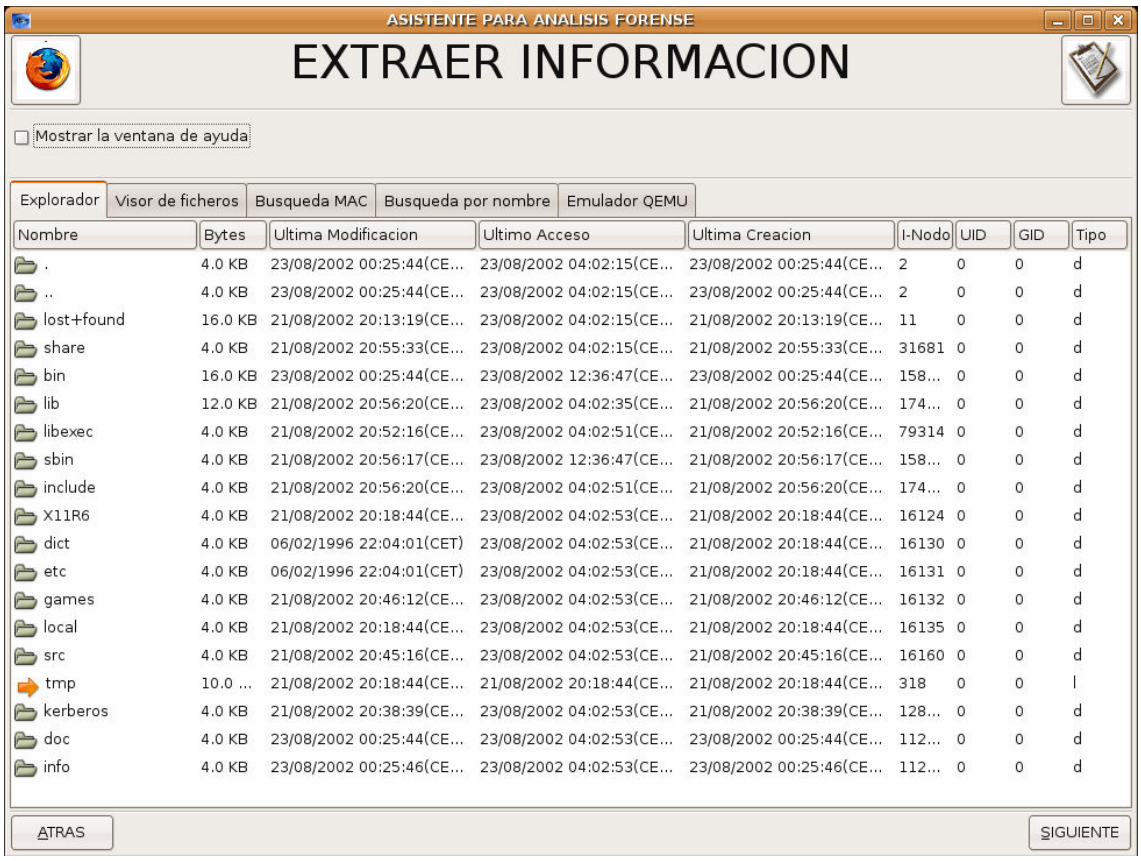
Esta imagen no contiene nada interesante salvo los ficheros normales de arranque de redhat.

192.168.3.10-hda5.dd (/usr)

Al introducir este fichero la aplicación nos dice lo siguiente:

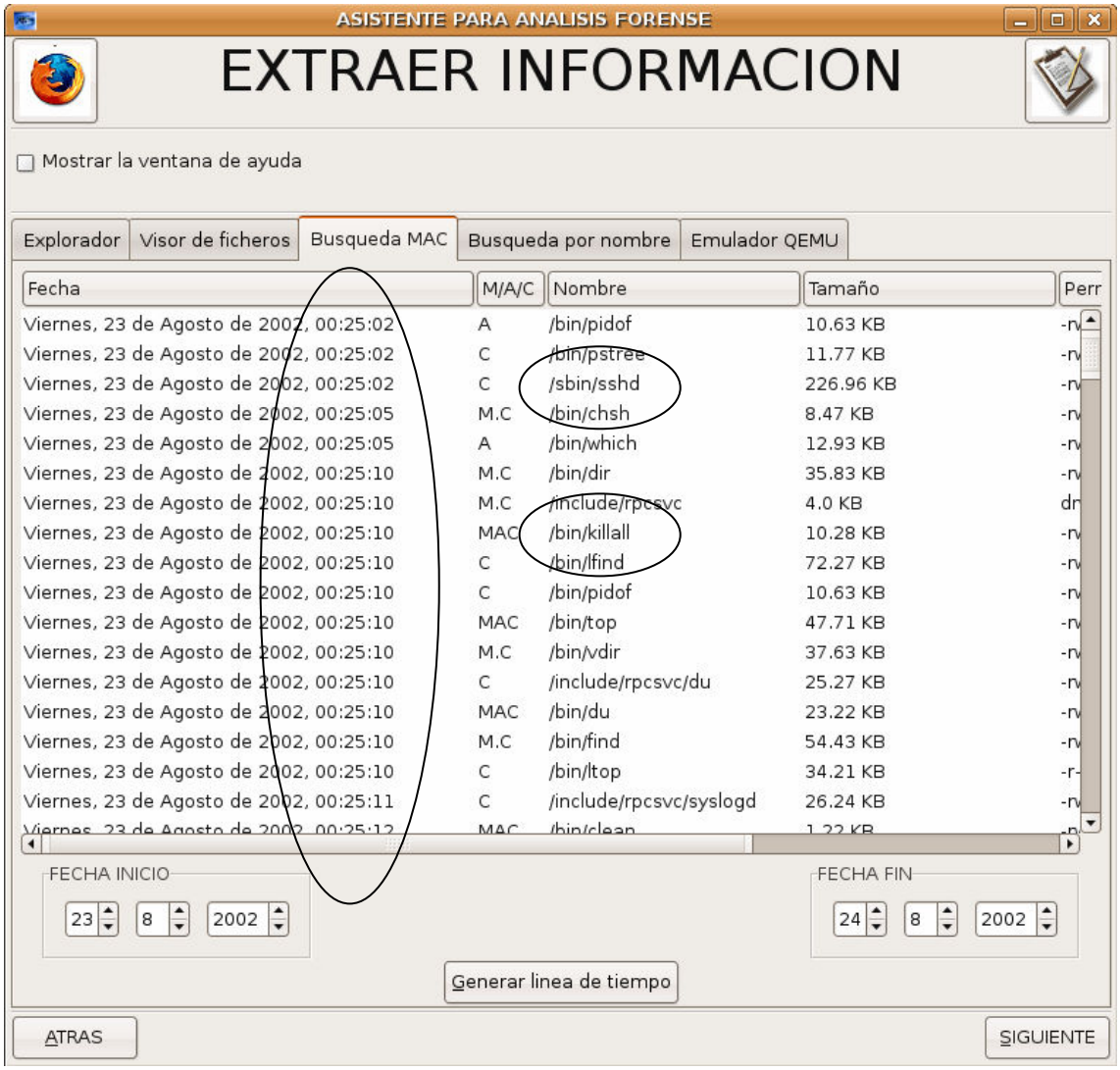


Veamos ahora el sistema de ficheros que contiene:



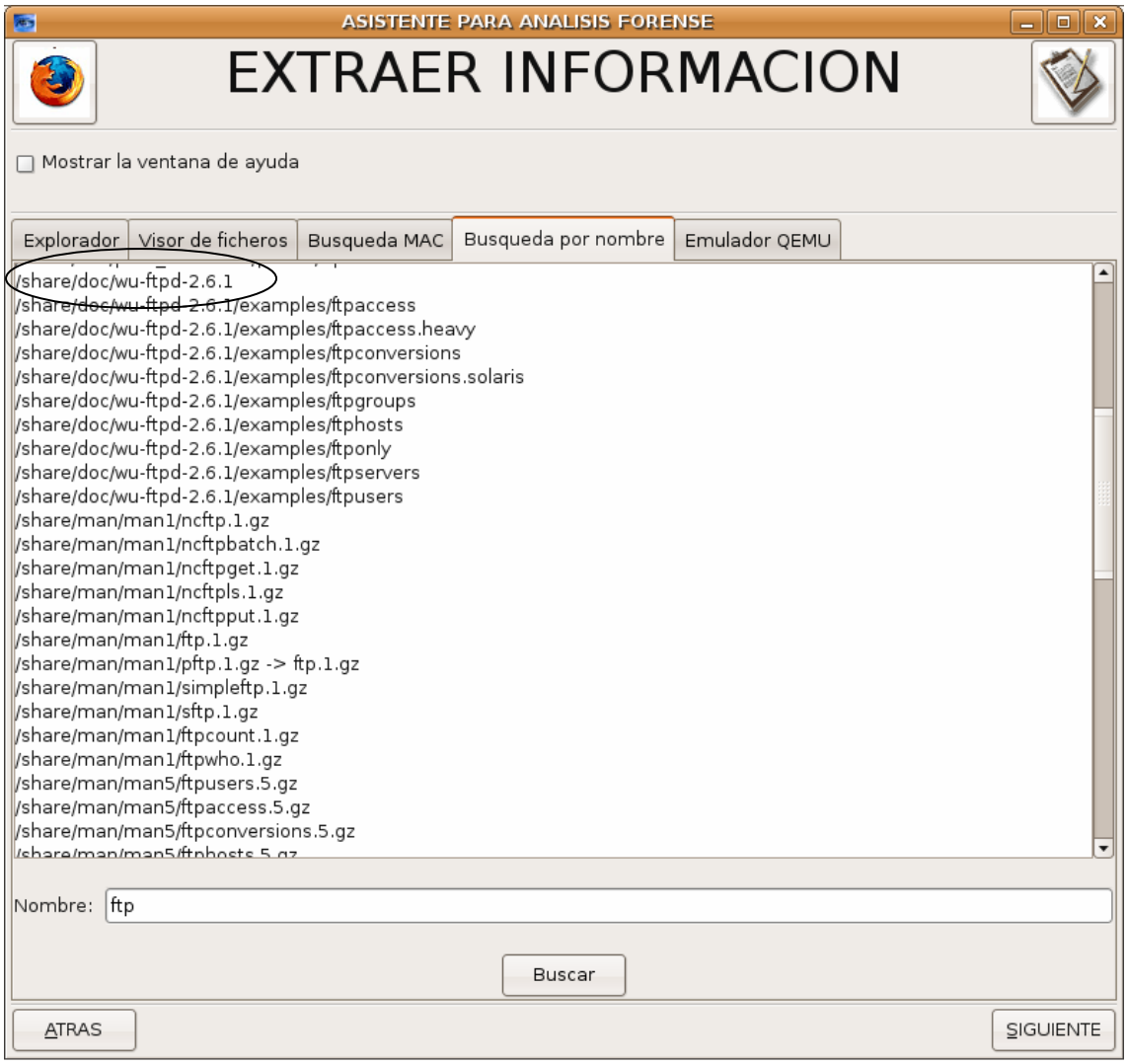
Nombre	Bytes	Ultima Modificacion	Ultimo Acceso	Ultima Creacion	I-Nodo	UID	GID	Tipo
.	4.0 KB	23/08/2002 00:25:44(CE...	23/08/2002 04:02:15(CE...	23/08/2002 00:25:44(CE...	2	0	0	d
..	4.0 KB	23/08/2002 00:25:44(CE...	23/08/2002 04:02:15(CE...	23/08/2002 00:25:44(CE...	2	0	0	d
lost+found	16.0 KB	21/08/2002 20:13:19(CE...	23/08/2002 04:02:15(CE...	21/08/2002 20:13:19(CE...	11	0	0	d
share	4.0 KB	21/08/2002 20:55:33(CE...	23/08/2002 04:02:15(CE...	21/08/2002 20:55:33(CE...	31681	0	0	d
bin	16.0 KB	23/08/2002 00:25:44(CE...	23/08/2002 12:36:47(CE...	23/08/2002 00:25:44(CE...	158...	0	0	d
lib	12.0 KB	21/08/2002 20:56:20(CE...	23/08/2002 04:02:35(CE...	21/08/2002 20:56:20(CE...	174...	0	0	d
libexec	4.0 KB	21/08/2002 20:52:16(CE...	23/08/2002 04:02:51(CE...	21/08/2002 20:52:16(CE...	79314	0	0	d
sbin	4.0 KB	21/08/2002 20:56:17(CE...	23/08/2002 12:36:47(CE...	21/08/2002 20:56:17(CE...	158...	0	0	d
include	4.0 KB	21/08/2002 20:56:20(CE...	23/08/2002 04:02:51(CE...	21/08/2002 20:56:20(CE...	174...	0	0	d
X11R6	4.0 KB	21/08/2002 20:18:44(CE...	23/08/2002 04:02:53(CE...	21/08/2002 20:18:44(CE...	16124	0	0	d
dict	4.0 KB	06/02/1996 22:04:01(CET)	23/08/2002 04:02:53(CE...	21/08/2002 20:18:44(CE...	16130	0	0	d
etc	4.0 KB	06/02/1996 22:04:01(CET)	23/08/2002 04:02:53(CE...	21/08/2002 20:18:44(CE...	16131	0	0	d
games	4.0 KB	21/08/2002 20:46:12(CE...	23/08/2002 04:02:53(CE...	21/08/2002 20:46:12(CE...	16132	0	0	d
local	4.0 KB	21/08/2002 20:18:44(CE...	23/08/2002 04:02:53(CE...	21/08/2002 20:18:44(CE...	16135	0	0	d
src	4.0 KB	21/08/2002 20:45:16(CE...	23/08/2002 04:02:53(CE...	21/08/2002 20:45:16(CE...	16160	0	0	d
tmp	10.0 ...	21/08/2002 20:18:44(CE...	21/08/2002 20:18:44(CE...	21/08/2002 20:18:44(CE...	318	0	0	l
kerberos	4.0 KB	21/08/2002 20:38:39(CE...	23/08/2002 04:02:53(CE...	21/08/2002 20:38:39(CE...	128...	0	0	d
doc	4.0 KB	23/08/2002 00:25:44(CE...	23/08/2002 04:02:53(CE...	23/08/2002 00:25:44(CE...	112...	0	0	d
info	4.0 KB	23/08/2002 00:25:46(CE...	23/08/2002 04:02:53(CE...	23/08/2002 00:25:46(CE...	112...	0	0	d

A simple vista no hay nada extraño, por lo que vamos a ver los ficheros cuyos sellos de tiempo se modificaron el dia 23 de agosto de 2002:



Como vemos a partir de la hora en la que se inició el ataque se producen muchas llamadas a binarios del sistema operativo, entre ellas a “sshd” y “killall”, y todas ellas se producen en cuestion de segundos.

Buscaremos el cliente FTP que tuvo la vulnerabilidad buscando el nombre “ftp” en el sistema de ficheros:

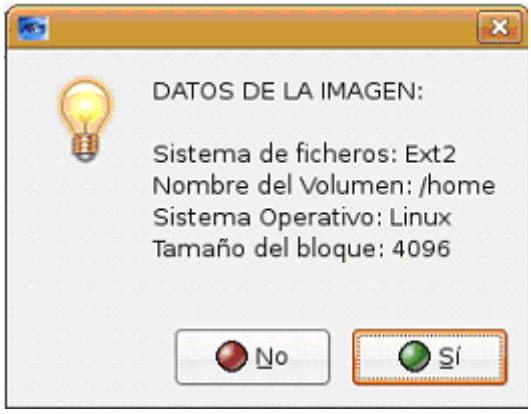


Encontramos en /share/doc/wu-ftpd-2.6.1 el cliente ftp que buscabamos.

En esta partición no existen más ficheros relevantes.

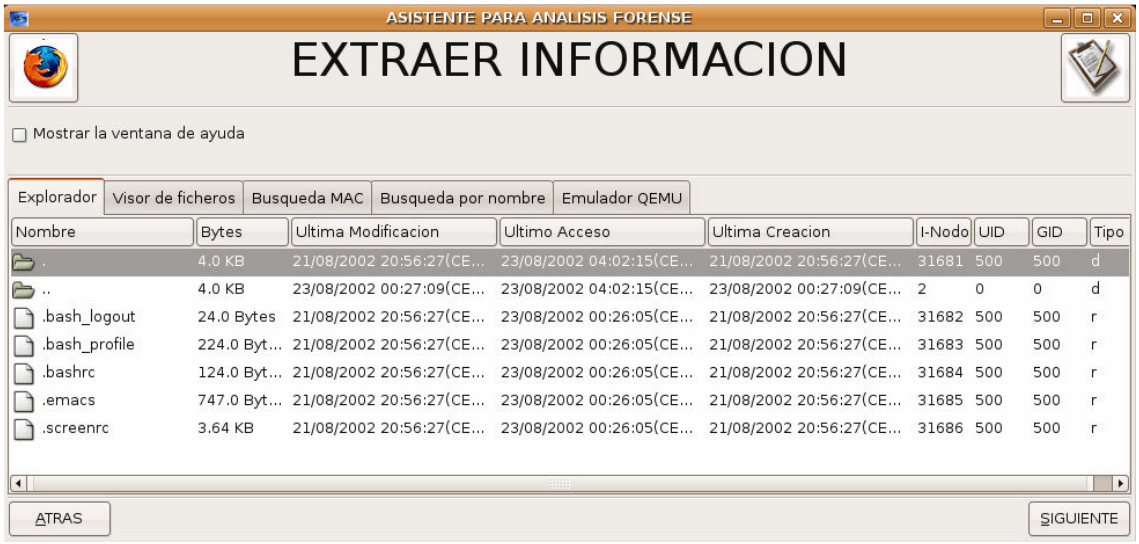
192.168.3.10-hda6.dd (/home)

En esta partición encontramos algunas cosas interesantes. Al introducir la imagen el resultado es:



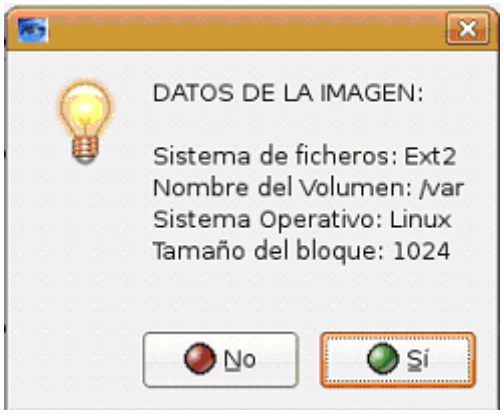
Veamos lo que contiene el sistema de ficheros:

Llaman la atención dos directorios: “pepelu” y “nerod”. Ambos contienen los mismos ficheros:

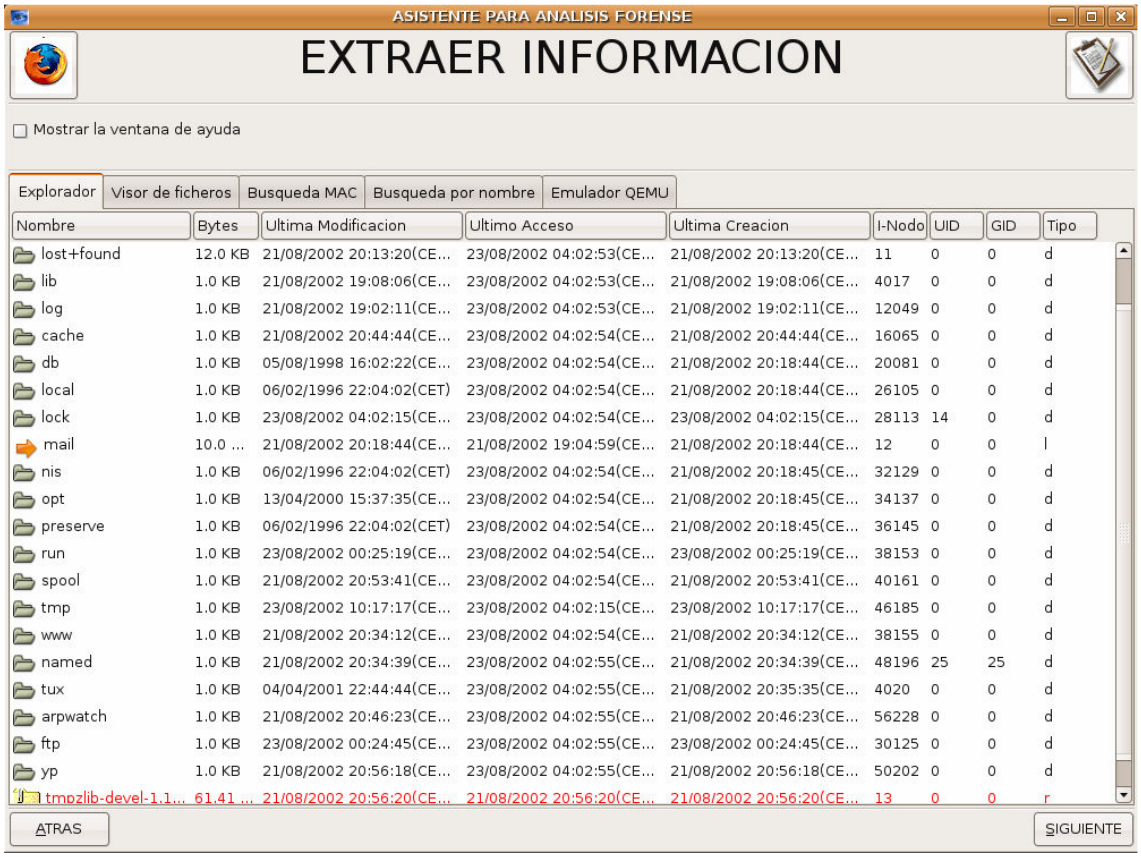


192.168.3.10-hda7.dd (/var)

En esta imagen encontramos mucha información valiosa. Al introducirla nos muestra lo siguiente:

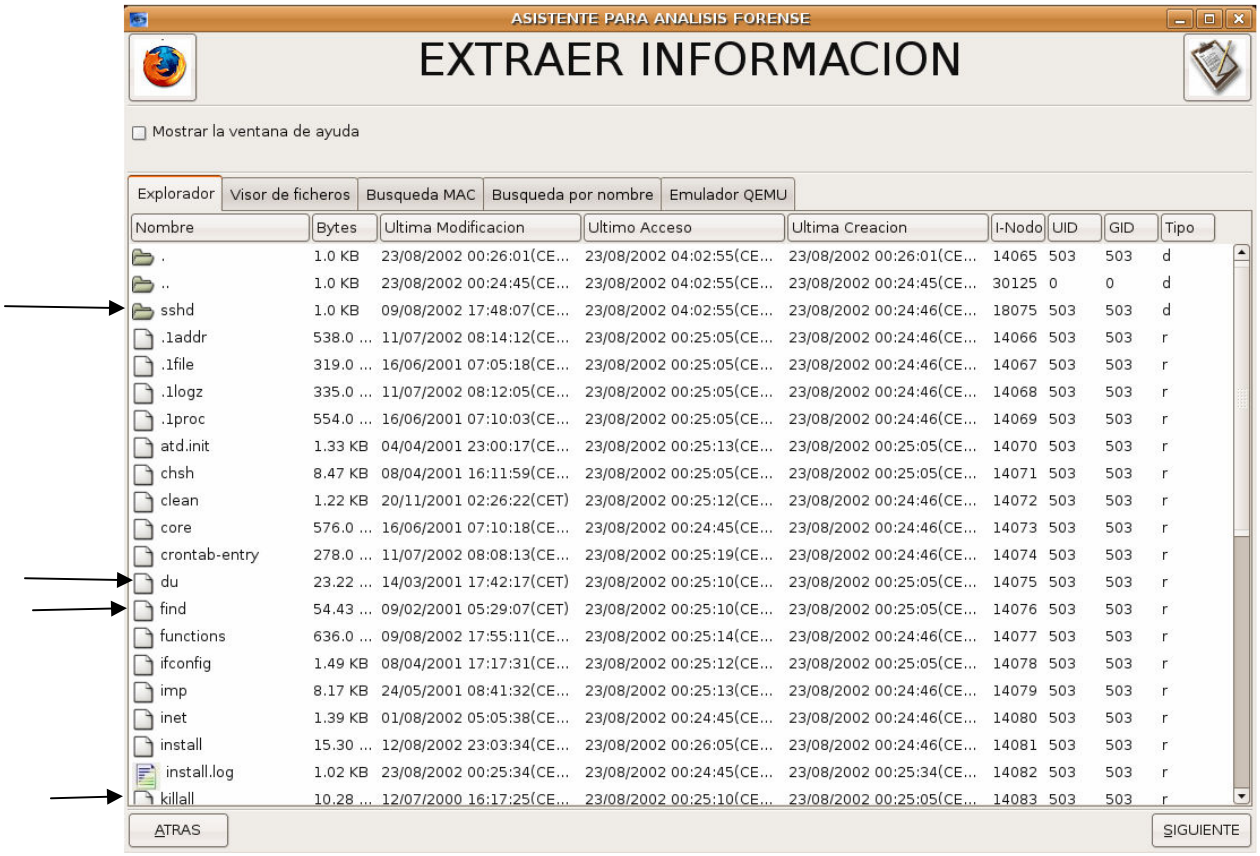


Veamos ahora el contenido del sistema de ficheros:



A simple vista no vemos nada extraño salvo un fichero borrado que no se puede recuperar (i-nodo=0) y una carpeta llamada “ftp” que nos puede interesar. Veamos su contenido:

Vemos que contiene el archivo de instalación de NeroD (nerod.tar.gz) y el directorio nerod que pasamos a mostrar a continuación:

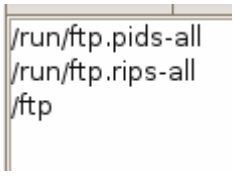


Nombre	Bytes	Ultima Modificacion	Ultimo Acceso	Ultima Creacion	I-Nodo	UID	GID	Tipo
.	1.0 KB	23/08/2002 00:26:01(CE...	23/08/2002 04:02:55(CE...	23/08/2002 00:26:01(CE...	14065	503	503	d
..	1.0 KB	23/08/2002 00:24:45(CE...	23/08/2002 04:02:55(CE...	23/08/2002 00:24:45(CE...	30125	0	0	d
sshd	1.0 KB	09/08/2002 17:48:07(CE...	23/08/2002 04:02:55(CE...	23/08/2002 00:24:46(CE...	18075	503	503	d
.1addr	538.0 ...	11/07/2002 08:14:12(CE...	23/08/2002 00:25:05(CE...	23/08/2002 00:24:46(CE...	14066	503	503	r
.1file	319.0 ...	16/06/2001 07:05:18(CE...	23/08/2002 00:25:05(CE...	23/08/2002 00:24:46(CE...	14067	503	503	r
.1logz	335.0 ...	11/07/2002 08:12:05(CE...	23/08/2002 00:25:05(CE...	23/08/2002 00:24:46(CE...	14068	503	503	r
.1proc	554.0 ...	16/06/2001 07:10:03(CE...	23/08/2002 00:25:05(CE...	23/08/2002 00:24:46(CE...	14069	503	503	r
atd.init	1.33 KB	04/04/2001 23:00:17(CE...	23/08/2002 00:25:13(CE...	23/08/2002 00:25:05(CE...	14070	503	503	r
chsh	8.47 KB	08/04/2001 16:11:59(CE...	23/08/2002 00:25:05(CE...	23/08/2002 00:25:05(CE...	14071	503	503	r
clean	1.22 KB	20/11/2001 02:26:22(CET)	23/08/2002 00:25:12(CE...	23/08/2002 00:24:46(CE...	14072	503	503	r
core	576.0 ...	16/06/2001 07:10:18(CE...	23/08/2002 00:24:45(CE...	23/08/2002 00:24:46(CE...	14073	503	503	r
crontab-entry	278.0 ...	11/07/2002 08:08:13(CE...	23/08/2002 00:25:19(CE...	23/08/2002 00:24:46(CE...	14074	503	503	r
du	23.22 ...	14/03/2001 17:42:17(CET)	23/08/2002 00:25:10(CE...	23/08/2002 00:25:05(CE...	14075	503	503	r
find	54.43 ...	09/02/2001 05:29:07(CET)	23/08/2002 00:25:10(CE...	23/08/2002 00:25:05(CE...	14076	503	503	r
functions	636.0 ...	09/08/2002 17:55:11(CE...	23/08/2002 00:25:14(CE...	23/08/2002 00:24:46(CE...	14077	503	503	r
ifconfig	1.49 KB	08/04/2001 17:17:31(CE...	23/08/2002 00:25:12(CE...	23/08/2002 00:25:05(CE...	14078	503	503	r
imp	8.17 KB	24/05/2001 08:41:32(CE...	23/08/2002 00:25:13(CE...	23/08/2002 00:24:46(CE...	14079	503	503	r
inet	1.39 KB	01/08/2002 05:05:38(CE...	23/08/2002 00:24:45(CE...	23/08/2002 00:24:46(CE...	14080	503	503	r
install	15.30 ...	12/08/2002 23:03:34(CE...	23/08/2002 00:26:05(CE...	23/08/2002 00:24:46(CE...	14081	503	503	r
install.log	1.02 KB	23/08/2002 00:25:34(CE...	23/08/2002 00:24:45(CE...	23/08/2002 00:25:34(CE...	14082	503	503	r
killall	10.28 ...	12/07/2000 16:17:25(CE...	23/08/2002 00:25:10(CE...	23/08/2002 00:25:05(CE...	14083	503	503	r

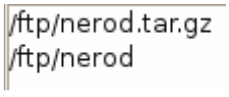
En esta imagen puede apreciar el contenido del rootkit cuyos binarios coinciden con los que se muestran en la línea de tiempos realizada para la imagen 192.168.3.10-hda5.dd.

Vamos realizar una búsqueda de varios nombres en esta imagen:

"ftp"



"nerod"



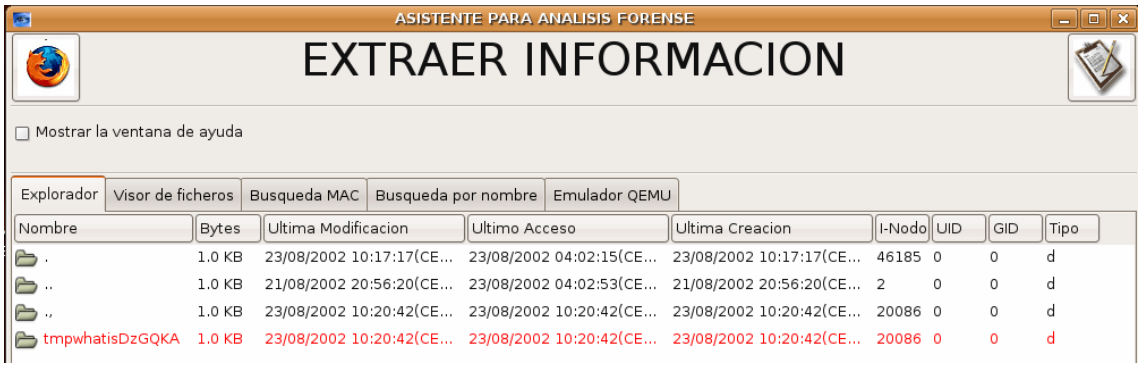
"psybnc"

```
/tmp/../psybnc
/tmp/../psybnc/log/psybnc.log
/tmp/../psybnc/src/psybnc.c
/tmp/../psybnc/src/psybnc.o
/tmp/../psybnc/psybncchk
/tmp/../psybnc/psybnc.conf
/tmp/../psybnc/psybnc (deleted-realloc)
/tmp/../psybnc/psybnc.pid
/tmp/../psybnc/psybnc.conf.old
```

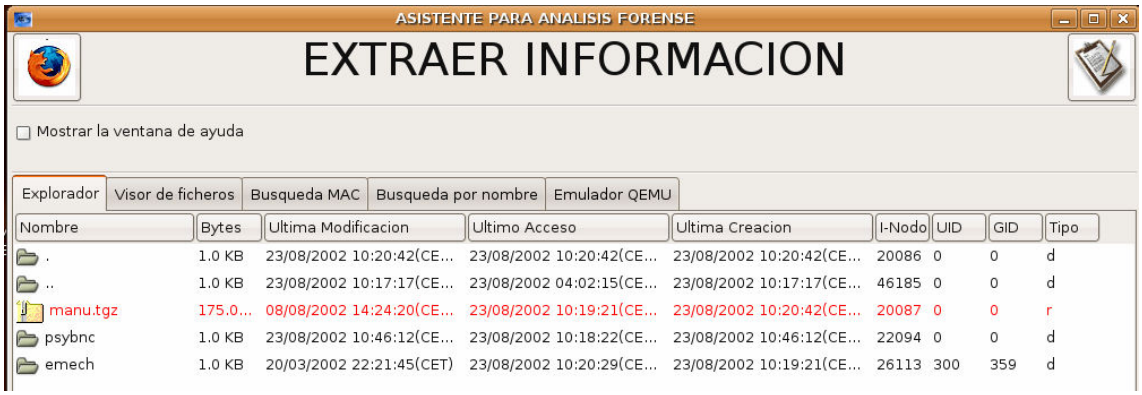
"mech"

```
/tmp/../emech
/tmp/../emech/mech.set
/tmp/../emech/mech.session
/tmp/../emech/emech.users
/tmp/../emech/mech.levels
/tmp/../emech/emech.users.save
/tmp/../emech/mech.pid
```

Por tanto vemos que en el directorio /tmp podemos encontrar mas evidencias de la intrusión:

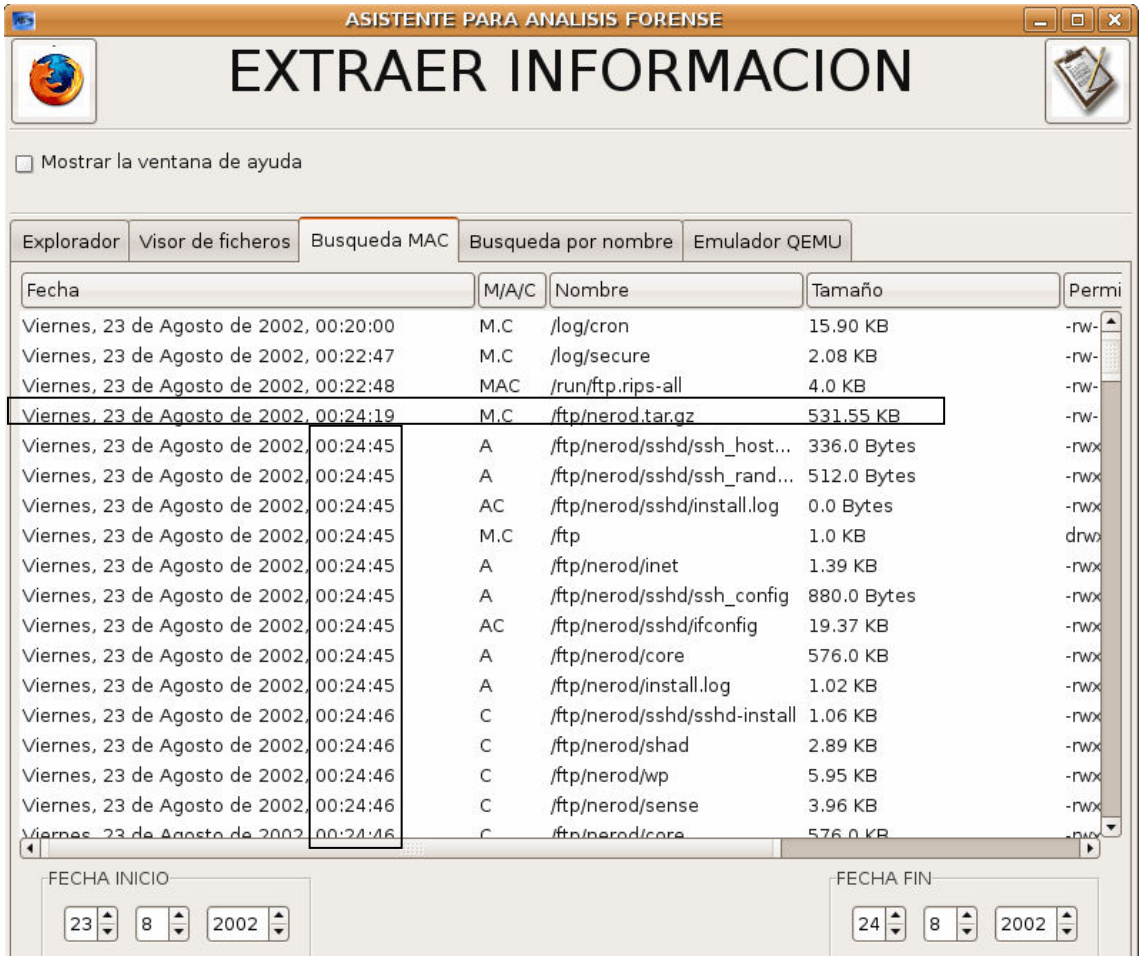


Existen en este directorio otros 2 directorios con el mismo i-nodo: uno llama "." (para confundirlo con ".." y un directorio borrado llamado "tmpwhatisDzGQKA" que no es relevante ya que ambos tienen el mismo i-nodo. Veamos el contenido del directorio ".":



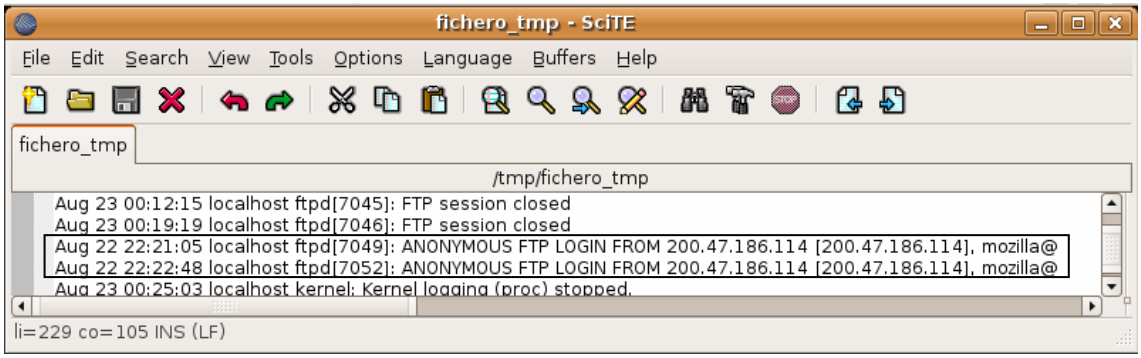
Encontramos aquí un fichero borrado llamado "manu.tgz" que contiene los instalables de psyBNC y EnergyMECH.

Vamos ahora a realizar una línea de tiempo en el dia 23 de Agosto de 2002 para ver si encontramos la actividad de alguno de estos ficheros.



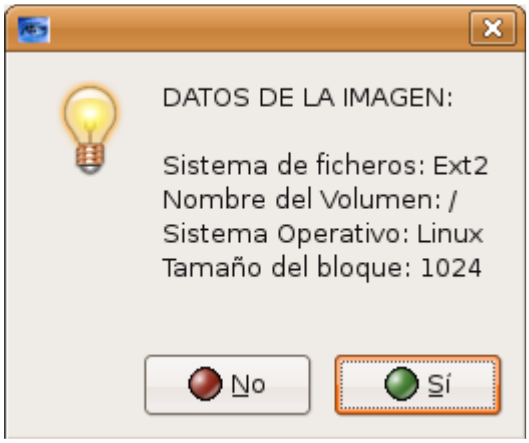
En esta imagen se puede apreciar como se copia el rootkit al sistema a las 00:24:19 y 26 segundos mas tarde comienza su ejecución.

Para ver la dirección IP del atacante recuperamos el fichero /log/messages y encontramos:



192.168.3.10-hda8.dd (/)

Al introducir la imagen el resultado es el siguiente:



Veamos el sistema de ficheros que contiene:

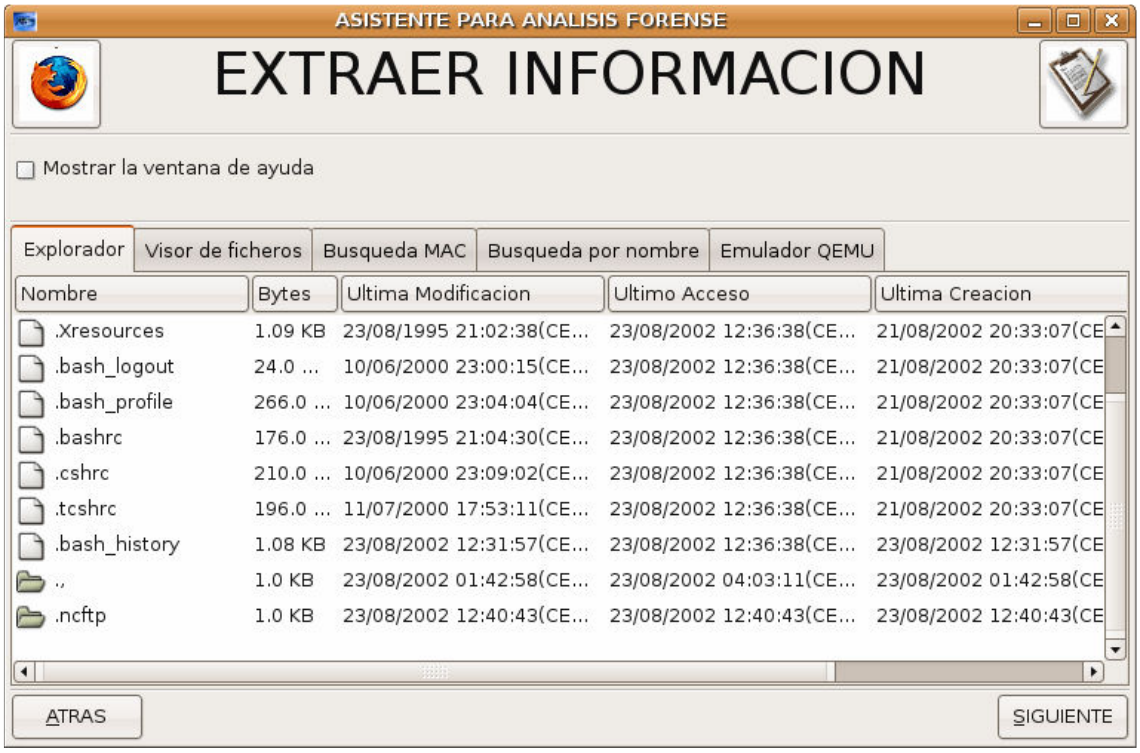
.	1.0 KB	21/08/2002 20:52:37(CE...	23/08/2002 12:36:50(CE...	21/08/2002 20:52:37(CE...	2	0	0	d
..	1.0 KB	21/08/2002 20:52:37(CE...	23/08/2002 12:36:50(CE...	21/08/2002 20:52:37(CE...	2	0	0	d
lost+found	12.0 KB	21/08/2002 20:13:05(CE...	23/08/2002 04:02:15(CE...	21/08/2002 20:13:05(CE...	11	0	0	d
boot	1.0 KB	21/08/2002 20:13:23(CE...	21/08/2002 20:13:23(CE...	21/08/2002 20:13:23(CE...	4017	0	0	d
home	1.0 KB	21/08/2002 20:13:23(CE...	21/08/2002 20:13:23(CE...	21/08/2002 20:13:23(CE...	8033	0	0	d
usr	1.0 KB	21/08/2002 20:13:23(CE...	21/08/2002 20:13:23(CE...	21/08/2002 20:13:23(CE...	12049	0	0	d
var	1.0 KB	21/08/2002 20:13:23(CE...	21/08/2002 20:13:23(CE...	21/08/2002 20:13:23(CE...	16065	0	0	d
proc	1.0 KB	21/08/2002 20:13:23(CE...	21/08/2002 20:13:23(CE...	21/08/2002 20:13:23(CE...	20081	0	0	d
tmp	1.0 KB	23/08/2002 04:02:01(CE...	23/08/2002 00:25:19(CE...	23/08/2002 04:03:12(CE...	22089	0	0	d
dev	80.0 KB	23/08/2002 10:20:29(CE...	23/08/2002 12:36:51(CE...	23/08/2002 10:20:29(CE...	24097	0	0	d
etc	3.0 KB	23/08/2002 00:56:39(CE...	23/08/2002 12:36:41(CE...	23/08/2002 00:56:39(CE...	26105	0	0	d
bin	2.0 KB	23/08/2002 00:25:34(CE...	23/08/2002 12:36:47(CE...	23/08/2002 00:25:34(CE...	36145	0	0	d
lib	3.0 KB	21/08/2002 20:49:16(CE...	23/08/2002 04:03:08(CE...	21/08/2002 20:49:16(CE...	44177	0	0	d
mnt	1.0 KB	21/08/2002 20:18:44(CE...	23/08/2002 04:03:11(CE...	21/08/2002 20:18:44(CE...	48193	0	0	d
opt	1.0 KB	23/08/1999 18:03:42(CE...	23/08/2002 04:03:11(CE...	21/08/2002 20:18:44(CE...	58233	0	0	d
root	1.0 KB	23/08/2002 12:40:43(CE...	23/08/2002 12:36:45(CE...	23/08/2002 12:40:43(CE...	60241	0	0	d
sbin	3.0 KB	23/08/2002 00:25:34(CE...	23/08/2002 12:36:47(CE...	23/08/2002 00:25:34(CE...	62249	0	0	d
misc	1.0 KB	03/03/2001 06:30:45(CET)	03/03/2001 06:30:45(CET)	21/08/2002 20:39:10(CE...	22155	0	0	d
.automount	1.0 KB	07/04/2001 22:53:32(CE...	23/08/2002 04:03:11(CE...	21/08/2002 20:52:37(CE...	42334	0	0	d

Realizaremos una línea de tiempo para ver qué ocurrió el dia 23 de Agosto de 2002 para tener una idea de donde buscar:

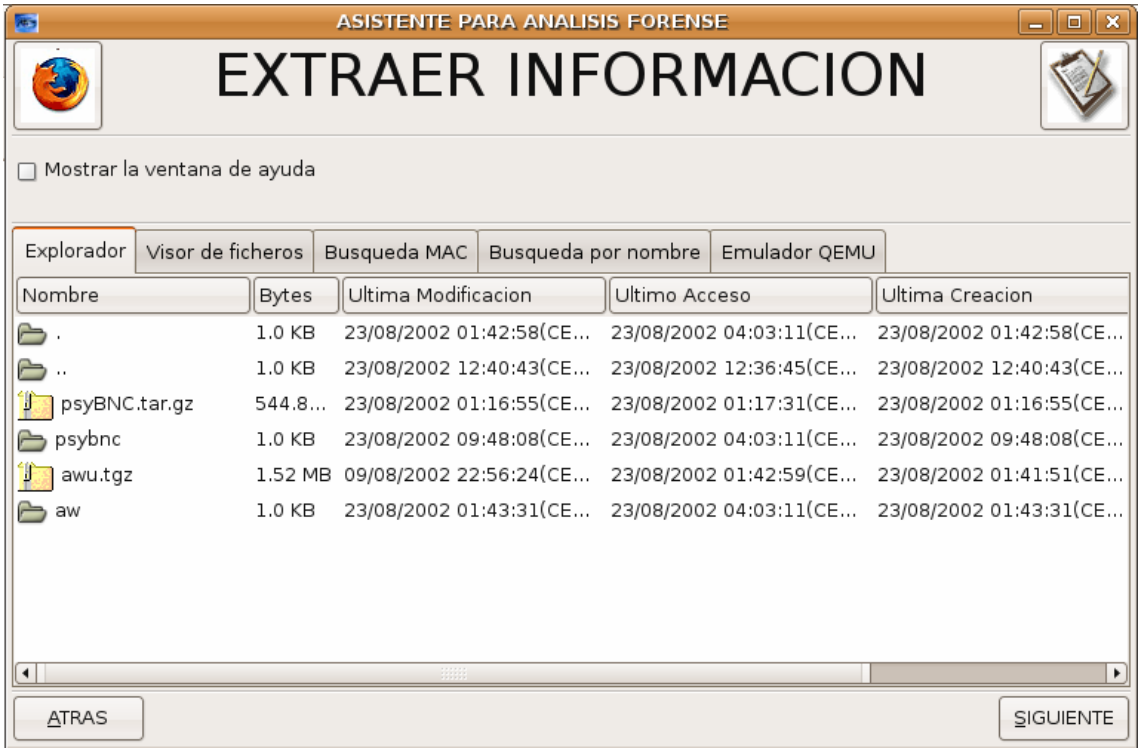
Viernes, 23 de Agosto de 2002, 00:56:39		MAC	/etc/shadow.lock	5.0 Bytes
Viernes, 23 de Agosto de 2002, 00:56:39		AC	/etc/shadow-	856.0 Bytes
Viernes, 23 de Agosto de 2002, 01:16:55		M.C	/root/../psyBNC.tar.gz	544.88 KB
Viernes, 23 de Agosto de 2002,	01:17:30	C	/etc/rc.d/rc2.d/K87portm...	2.0 KB
Viernes, 23 de Agosto de 2002,	01:17:30	AC	/root/../psybnc/help/SETU...	85.0 Bytes
Viernes, 23 de Agosto de 2002,	01:17:30	AC	/root/../psybnc/help/ADD...	434.0 Bytes
Viernes, 23 de Agosto de 2002,	01:17:30	AC	/root/../psybnc/help/ENCR...	673.0 Bytes
Viernes, 23 de Agosto de 2002,	01:17:30	AC	/root/../psybnc/help/SWIT...	268.0 Bytes
Viernes, 23 de Agosto de 2002,	01:17:30	AC	/root/../psybnc/help/PROX...	279.0 Bytes
Viernes, 23 de Agosto de 2002,	01:17:30	AC	/root/../psybnc/help/LISTE...	143.0 Bytes
Viernes, 23 de Agosto de 2002,	01:17:30	AC	/root/../psybnc/help/DELB...	183.0 Bytes
Viernes, 23 de Agosto de 2002,	01:17:30	AC	/root/../psybnc/help/PLAY...	222.0 Bytes
Viernes, 23 de Agosto de 2002,	01:17:30	AC	/root/../psybnc/help/BRE...	101.0 Bytes
Viernes, 23 de Agosto de 2002,	01:17:30	AC	/root/../psybnc/help/ADD...	339.0 Bytes
Viernes, 23 de Agosto de 2002,	01:17:30	AC	/root/../psybnc/scripts/INFO	136.0 Bytes
Viernes, 23 de Agosto de 2002,	01:17:30	AC	/root/../psybnc/help/ADD...	412.0 Bytes
Viernes, 23 de Agosto de 2002,	01:17:30	AC	/root/../psybnc/help/DELO...	268.0 Bytes
Viernes, 23 de Agosto de 2002,	01:17:30	AC	/root/../psybnc/help/DCC...	240.0 Bytes

Se puede apreciar como en el directorio /root existe otro fichero directorio llamado "..," que contiene más evidencias. A las 01:16:55 se realiza la copia del instalador psyBNC.tar.gz y 35 segundos mas tarde se produce su instalación.

Veamos pues el contenido del directorio /root:

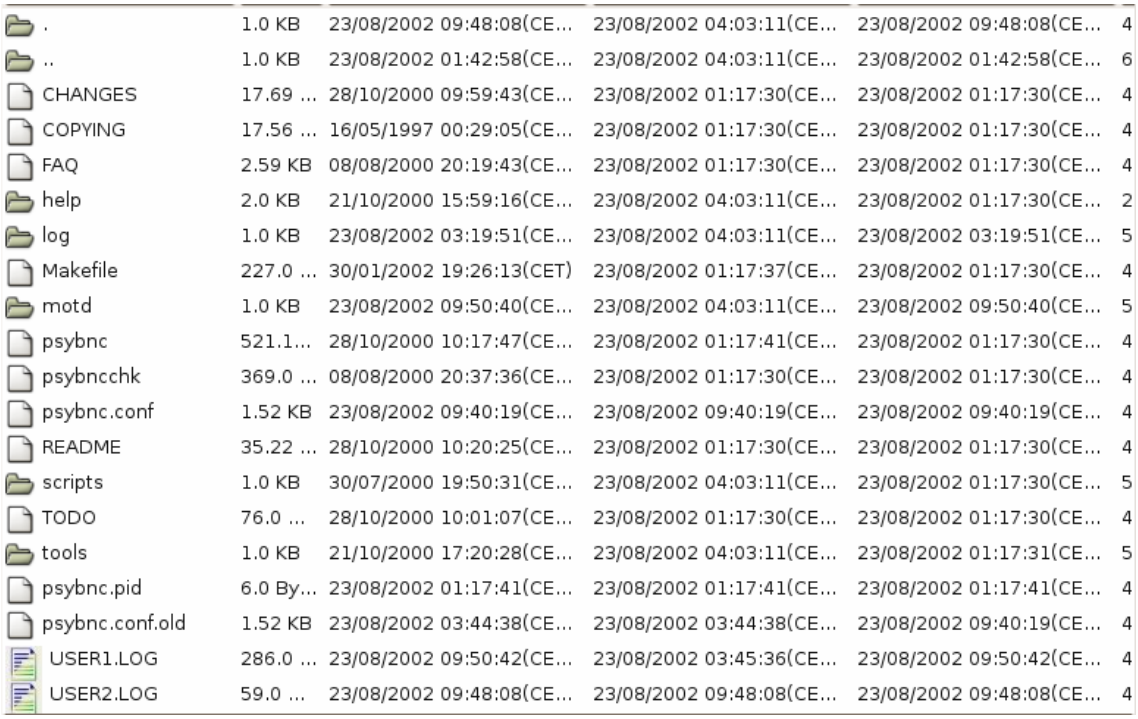


Vemos que contiene un directorio llamado “.,” cuyo contenido es el siguiente:



Aquí podemos apreciar los archivos “psyBNC.tar.gz” y “awu.tgz” asi como los directorios que contienen ambas herramientas.

El contenido de psybnc es el siguiente:



.	1.0 KB	23/08/2002 09:48:08(CE...	23/08/2002 04:03:11(CE...	23/08/2002 09:48:08(CE...	4
..	1.0 KB	23/08/2002 01:42:58(CE...	23/08/2002 04:03:11(CE...	23/08/2002 01:42:58(CE...	6
CHANGES	17.69 ...	28/10/2000 09:59:43(CE...	23/08/2002 01:17:30(CE...	23/08/2002 01:17:30(CE...	4
COPYING	17.56 ...	16/05/1997 00:29:05(CE...	23/08/2002 01:17:30(CE...	23/08/2002 01:17:30(CE...	4
FAQ	2.59 KB	08/08/2000 20:19:43(CE...	23/08/2002 01:17:30(CE...	23/08/2002 01:17:30(CE...	4
help	2.0 KB	21/10/2000 15:59:16(CE...	23/08/2002 04:03:11(CE...	23/08/2002 01:17:30(CE...	2
log	1.0 KB	23/08/2002 03:19:51(CE...	23/08/2002 04:03:11(CE...	23/08/2002 03:19:51(CE...	5
Makefile	227.0 ...	30/01/2002 19:26:13(CET)	23/08/2002 01:17:37(CE...	23/08/2002 01:17:30(CE...	4
motd	1.0 KB	23/08/2002 09:50:40(CE...	23/08/2002 04:03:11(CE...	23/08/2002 09:50:40(CE...	5
psybnc	521.1...	28/10/2000 10:17:47(CE...	23/08/2002 01:17:41(CE...	23/08/2002 01:17:30(CE...	4
psybncchk	369.0 ...	08/08/2000 20:37:36(CE...	23/08/2002 01:17:30(CE...	23/08/2002 01:17:30(CE...	4
psybnc.conf	1.52 KB	23/08/2002 09:40:19(CE...	23/08/2002 09:40:19(CE...	23/08/2002 09:40:19(CE...	4
README	35.22 ...	28/10/2000 10:20:25(CE...	23/08/2002 01:17:30(CE...	23/08/2002 01:17:30(CE...	4
scripts	1.0 KB	30/07/2000 19:50:31(CE...	23/08/2002 04:03:11(CE...	23/08/2002 01:17:30(CE...	5
TODO	76.0 ...	28/10/2000 10:01:07(CE...	23/08/2002 01:17:30(CE...	23/08/2002 01:17:30(CE...	4
tools	1.0 KB	21/10/2000 17:20:28(CE...	23/08/2002 04:03:11(CE...	23/08/2002 01:17:31(CE...	5
psybnc.pid	6.0 By...	23/08/2002 01:17:41(CE...	23/08/2002 01:17:41(CE...	23/08/2002 01:17:41(CE...	4
psybnc.conf.old	1.52 KB	23/08/2002 03:44:38(CE...	23/08/2002 03:44:38(CE...	23/08/2002 09:40:19(CE...	4
USER1.LOG	286.0 ...	23/08/2002 09:50:42(CE...	23/08/2002 03:45:36(CE...	23/08/2002 09:50:42(CE...	4
USER2.LOG	59.0 ...	23/08/2002 09:48:08(CE...	23/08/2002 09:48:08(CE...	23/08/2002 09:48:08(CE...	4

Y el contenido de aw:

.	1.0 KB	23/08/2002 01:43:31(CE...	23/08/2002 04:03:11(CE...	23/08/2002 01:43:31(CE...
..	1.0 KB	23/08/2002 01:42:58(CE...	23/08/2002 04:03:11(CE...	23/08/2002 01:42:58(CE...
auto	205.0 ...	23/01/2002 00:21:27(CET)	23/08/2002 01:42:58(CE...	23/08/2002 01:42:58(CE...
awu	1.26 KB	23/01/2002 00:20:27(CET)	23/08/2002 01:43:31(CE...	23/08/2002 01:42:58(CE...
awu.list	231.0 ...	23/01/2002 04:05:18(CET)	23/08/2002 01:42:58(CE...	23/08/2002 01:42:58(CE...
Makefile	597.0 ...	21/01/2002 23:38:26(CET)	23/08/2002 01:42:58(CE...	23/08/2002 01:42:58(CE...
nodupe.c	5.41 KB	20/01/2002 07:46:27(CET)	23/08/2002 01:42:58(CE...	23/08/2002 01:42:58(CE...
oops.c	1.03 KB	20/01/2002 08:23:16(CET)	23/08/2002 01:42:58(CE...	23/08/2002 01:42:58(CE...
pscan2.c	5.73 KB	22/01/2002 18:37:17(CET)	23/08/2002 01:42:58(CE...	23/08/2002 01:42:58(CE...
ss.c	6.07 KB	04/04/2002 22:29:08(CE...	23/08/2002 01:42:58(CE...	23/08/2002 01:42:58(CE...
ssvuln.c	3.27 KB	21/01/2002 23:32:54(CET)	23/08/2002 01:42:58(CE...	23/08/2002 01:42:58(CE...
targets	4.89 KB	20/01/2002 07:46:27(CET)	23/08/2002 01:42:58(CE...	23/08/2002 01:42:58(CE...
wu	373.1...	20/01/2002 08:21:24(CET)	23/08/2002 01:42:58(CE...	23/08/2002 01:42:58(CE...
x2	1.32 MB	28/01/2002 22:51:59(CET)	23/08/2002 01:42:58(CE...	23/08/2002 01:42:59(CE...
nodupe.o	3.75 KB	04/04/2002 22:30:44(CE...	23/08/2002 01:42:58(CE...	23/08/2002 01:42:59(CE...
oops.o	1.61 KB	04/04/2002 22:30:45(CE...	23/08/2002 01:42:58(CE...	23/08/2002 01:42:59(CE...
oops	12.77 ...	04/04/2002 22:30:45(CE...	23/08/2002 01:42:58(CE...	23/08/2002 01:42:59(CE...
outpu	3.62 KB	26/03/2002 02:46:04(CET)	23/08/2002 01:42:58(CE...	23/08/2002 01:42:59(CE...
doit4me	389.0 ...	21/01/1997 05:28:09(CET)	23/08/2002 01:42:58(CE...	23/08/2002 01:42:59(CE...
ss	16.56 ...	04/04/2002 22:30:46(CE...	23/08/2002 01:42:58(CE...	23/08/2002 01:42:59(CE...

192.168.3.10-hda9.dd (swap)

El resultado de introducir esta imagen es el siguiente:

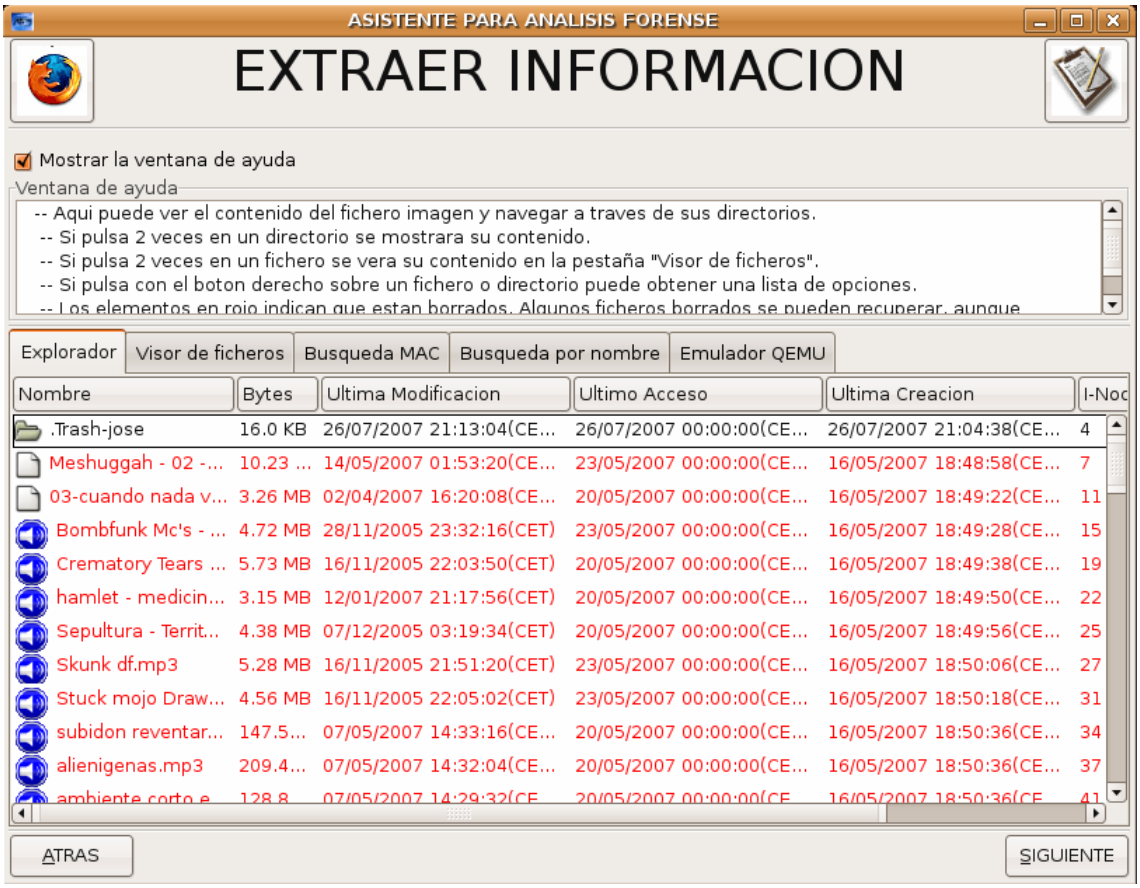
Esto es asi porque no se reconoce ningun dato relevante en la imagen: no tiene nombre, ni tamaño del sector, por lo que se devuelve el error correspondiente.

Mp3 de 256 MB

Se ha tomado una imagen de un dispositivo de música MP3 de 256MB de capacidad. En este MP3 contenía ficheros borrados hace tiempo (difícilmente recuperables), por lo que se ha borrado un fichero y un directorio y a continuación se ha tomado otra imagen para intentar recuperar estos ficheros borrados.

Veamos el proceso seguido. Al introducir la imagen en la aplicación el resultado fue:

Y los ficheros mostrados fueron los siguientes:



Intentamos recuperar los ficheros borrados y el resultado fue satisfactorio:

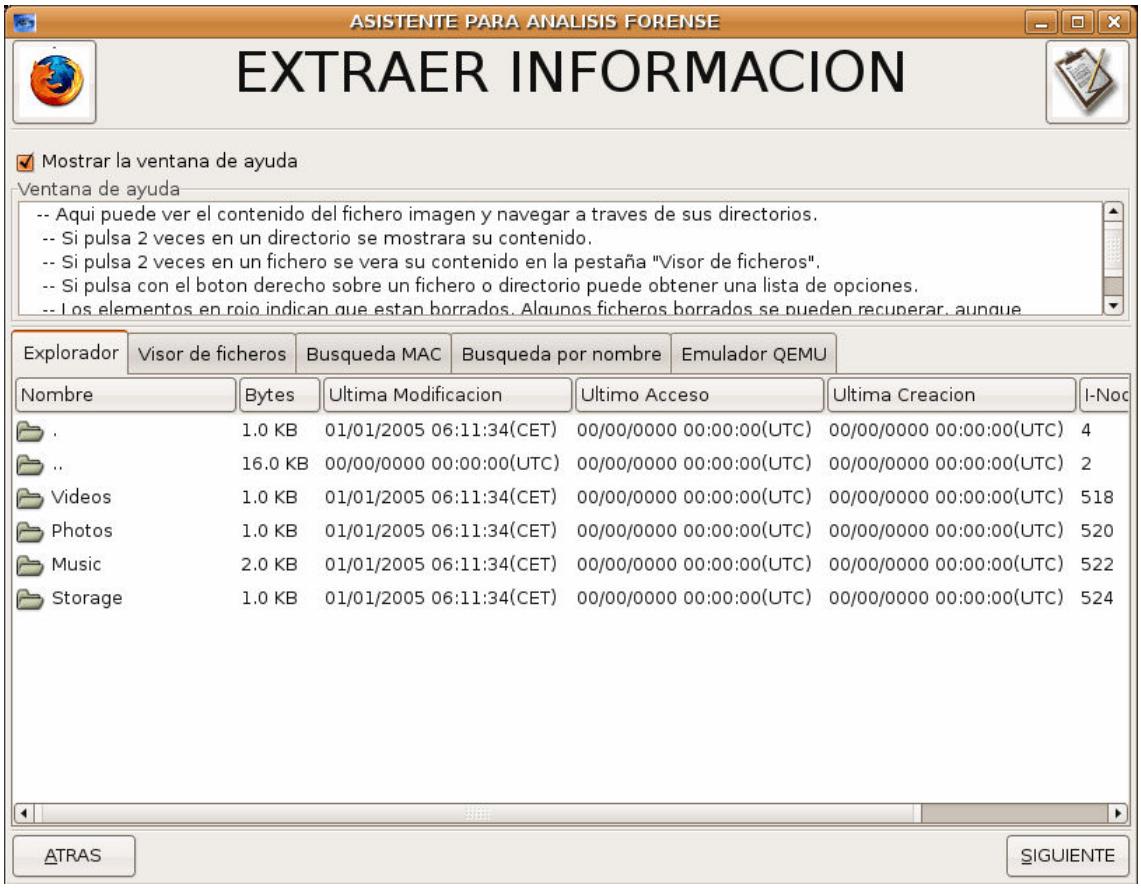
Al intentar recuperar el directorio borrado se consiguieron recuperar los ficheros que contenía, aunque el contenido de los mismos no era exacto y las canciones se mezclaban unas con otras.

Movil Motorola

Se extrajo una imagen de un movil Motorota con una memoria interna de 64MB. Esta prueba se hizo simplemente para ver si se podia extraer una imagen completa del movil y buscar datos privados como Codigo Pin, Mensajes borrados, etc.

Al insertarlo en la aplicación el resultado fue el siguiente:

Podemos ver que lo unico que se muestra son los datos multimedia almacenados en el teléfono móvil.


ASISTENTE PARA ANALISIS FORENSE
EXTRAER INFORMACION
Mostrar la ventana de ayuda
Ventana de ayuda
-- Aqui puede ver el contenido del fichero imagen y navegar a traves de sus directorios.
-- Si pulsa 2 veces en un directorio se mostrara su contenido.
-- Si pulsa 2 veces en un fichero se vera su contenido en la pestaña "Visor de ficheros".
-- Si pulsa con el boton derecho sobre un fichero o directorio puede obtener una lista de opciones.
-- Los elementos en rojo indican que estan borrados. Algunos ficheros borrados se pueden recuperar, aunque
Explorador
Visor de ficheros
Busqueda MAC
Busqueda por nombre
Emulador QEMU
Nombre
Bytes
Ultima Modificacion
Ultimo Acceso
Ultima Creacion
I-Noc
. 1.0 KB 01/01/2005 06:11:34(CET) 00/00/0000 00:00:00(UTC) 00/00/0000 00:00:00(UTC) 4
.. 16.0 KB 00/00/0000 00:00:00(UTC) 00/00/0000 00:00:00(UTC) 00/00/0000 00:00:00(UTC) 2
Videos 1.0 KB 01/01/2005 06:11:34(CET) 00/00/0000 00:00:00(UTC) 00/00/0000 00:00:00(UTC) 518
Photos 1.0 KB 01/01/2005 06:11:34(CET) 00/00/0000 00:00:00(UTC) 00/00/0000 00:00:00(UTC) 520
Music 2.0 KB 01/01/2005 06:11:34(CET) 00/00/0000 00:00:00(UTC) 00/00/0000 00:00:00(UTC) 522
Storage 1.0 KB 01/01/2005 06:11:34(CET) 00/00/0000 00:00:00(UTC) 00/00/0000 00:00:00(UTC) 524
ATRAS
SIGUIENTE

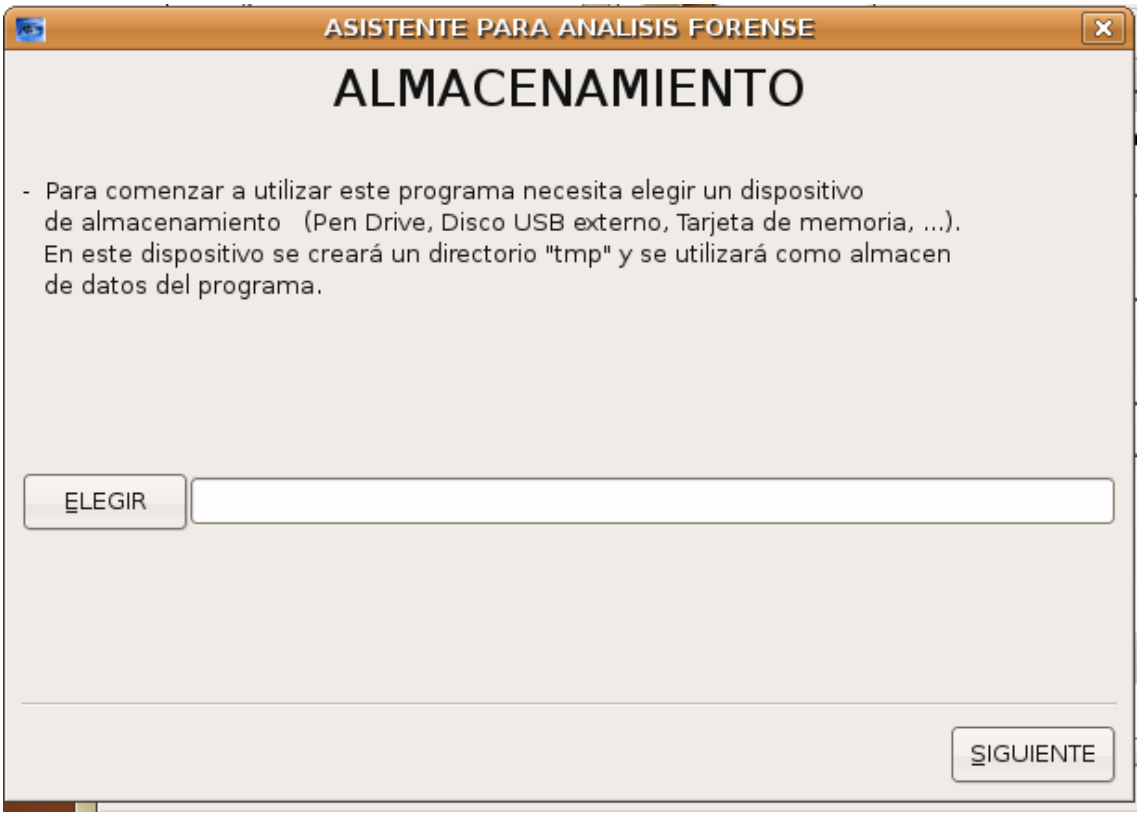
Manual de usuario

Para usar la aplicación desde el CD-Live, una vez arrancado simplemente hay que abrir una consola y teclear:

./aplicacion

Y se abrirá la ventana de presentación del programa:

A continuación debe pulsar en el botón Comenzar para pasar a la siguiente pantalla:



En ella debe introducir la ruta de un dispositivo de almacenamiento EXTERNO que tenga permisos de escritura. Esto quiere decir que los discos duros locales no servirán como dispositivo de almacenamiento. Un dispositivo válido puede ser un Pen Drive, una tarjeta de memoria, un Disco Duro USB esterno, etc. Por ejemplo /dev/sdc puede ser la ruta de un Pen Drive introducido en el PC.

En caso de no conocer la ruta del dispositivo, puede pulsar en el boton Elegir y se mostrarán una lista con los dispositivos con permisos de escritura que hay en el sistema:

Pinche en uno de ellos y su ruta se introducirá automáticamente en el cuadro correspondiente.

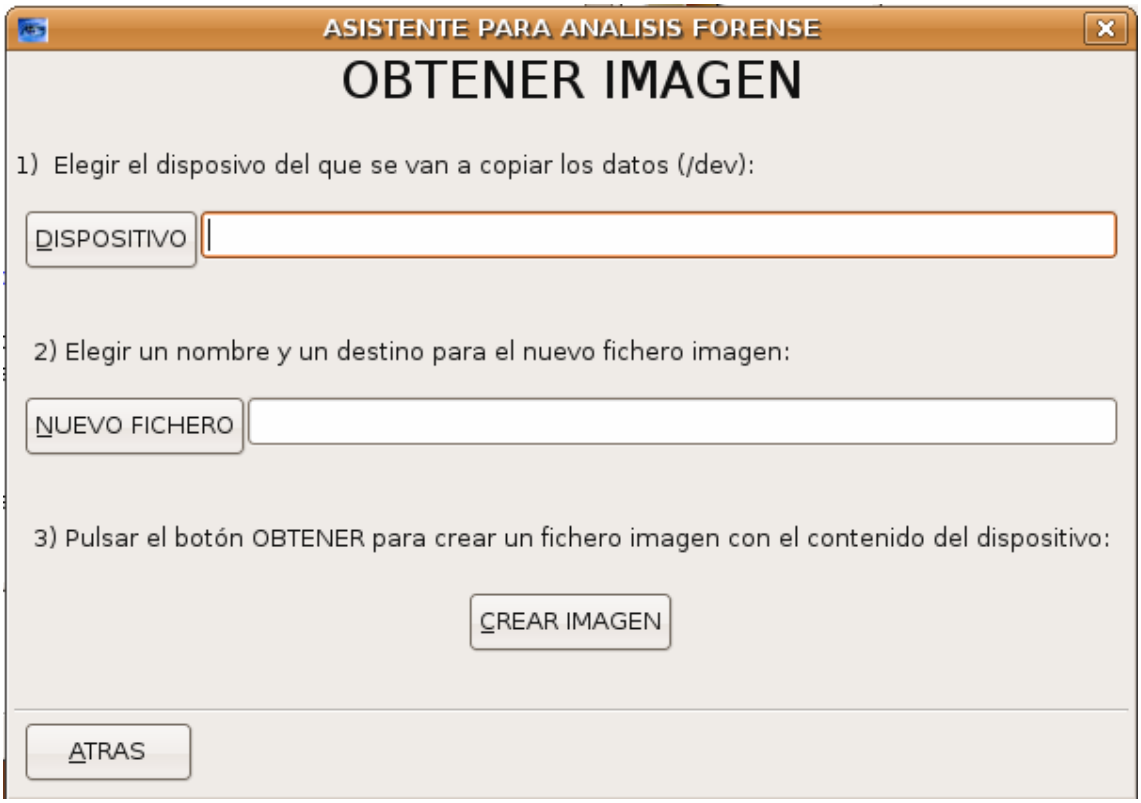
A continuación pulse el botón Siguiente para pasar a la siguiente pantalla. En este momento se habrá creado un directorio llamado /tmp en el dispositivo elegido y en este directorio se guardarán los datos temporales de la aplicación así como las imágenes adquiridas por el usuario y los ficheros recuperados. Ahora vemos lo siguiente:

En esta pantalla debemos elegir una imagen para poder analizarla. Para ello tenemos 3 opciones:

- Elegir un fichero imagen: Pulsando el boton Elegir se abrirá una ventana con la que podrá buscar un fichero imagen dentro del equipo.
- Elegir un fichero imagen de prueba: Existen 3 ficheros que el usuario puede probar
    - Diquette Arranque MS-DOS
    - Imagen Autoarrancable Linux
    - Disquette del reto forense nº26 del proyecto HoneyNet

    Eligiendo cualquiera de estas imágenes se introducirá su ruta automáticamente en el cuadro correspondiente:

- Obtener un fichero imagen a partir de un dispositivo del PC: Se abrirá una nueva ventana que le mostrará los pasos a seguir

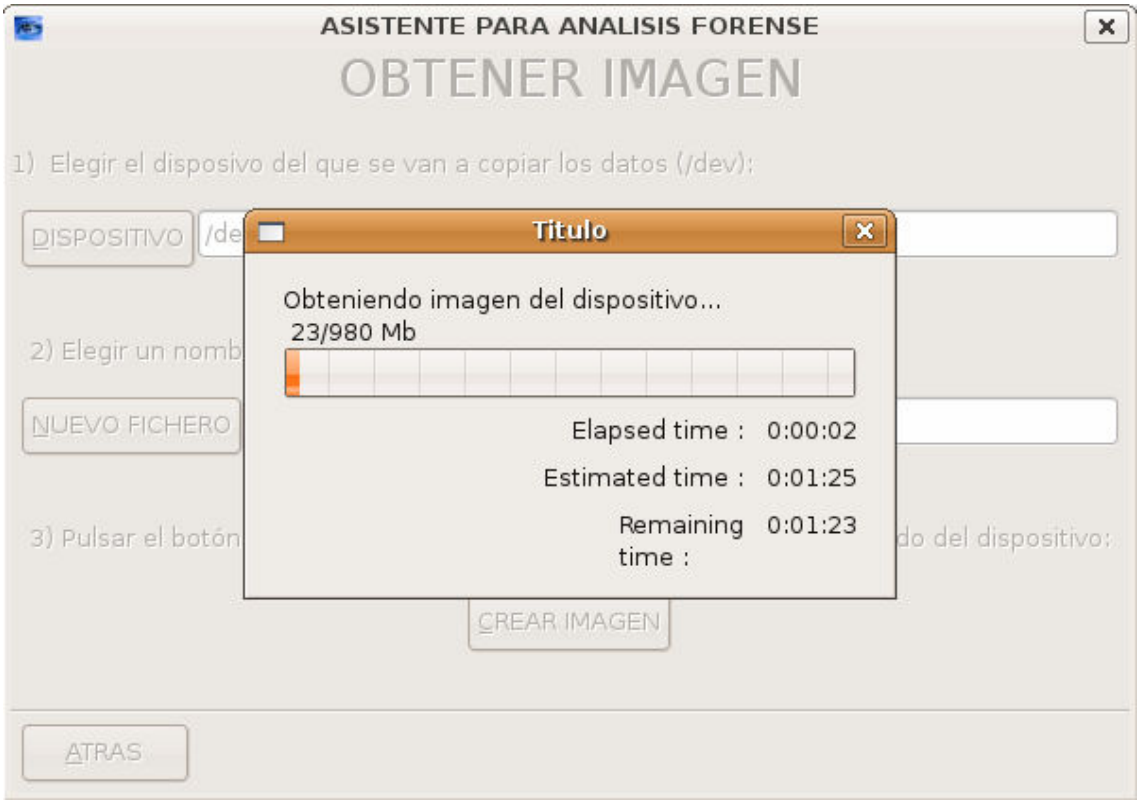


Primero debe elegir el dispositivo de origen, introduciendo su ruta directamente o eligiendolo de una lista de dispositivos que se muestra al pulsar el boton Origen:

En esta ocasión se mostrarán todos los dispositivos de los que se pueda extraer una imagen, es decir, los discos duros, cd rom, disqueteras, usb, etc.

Ahora debe elegir el fichero de destino. Si introduce un nombre directamente, se almacenará en el directorio /tmp creado en el dispositivo de almacenamiento. Si pulsa el botón Destino podrá elegir la ruta y el nombre del fichero mediante una ventana.

A continuación debe puslar el botón Obtener y se mostrará una barra de progreso indicando el porcentaje y el tiempo restante:



Al terminar el proceso, si la copia ha tenido éxito se mostrará un mensaje indicando el valor MD5 de la copia.

Una vez obtenida la imagen pulsamos en el botón Atras y la ruta de la imagen se introducirá en el cuadro correspondiente de la ventana anterior.

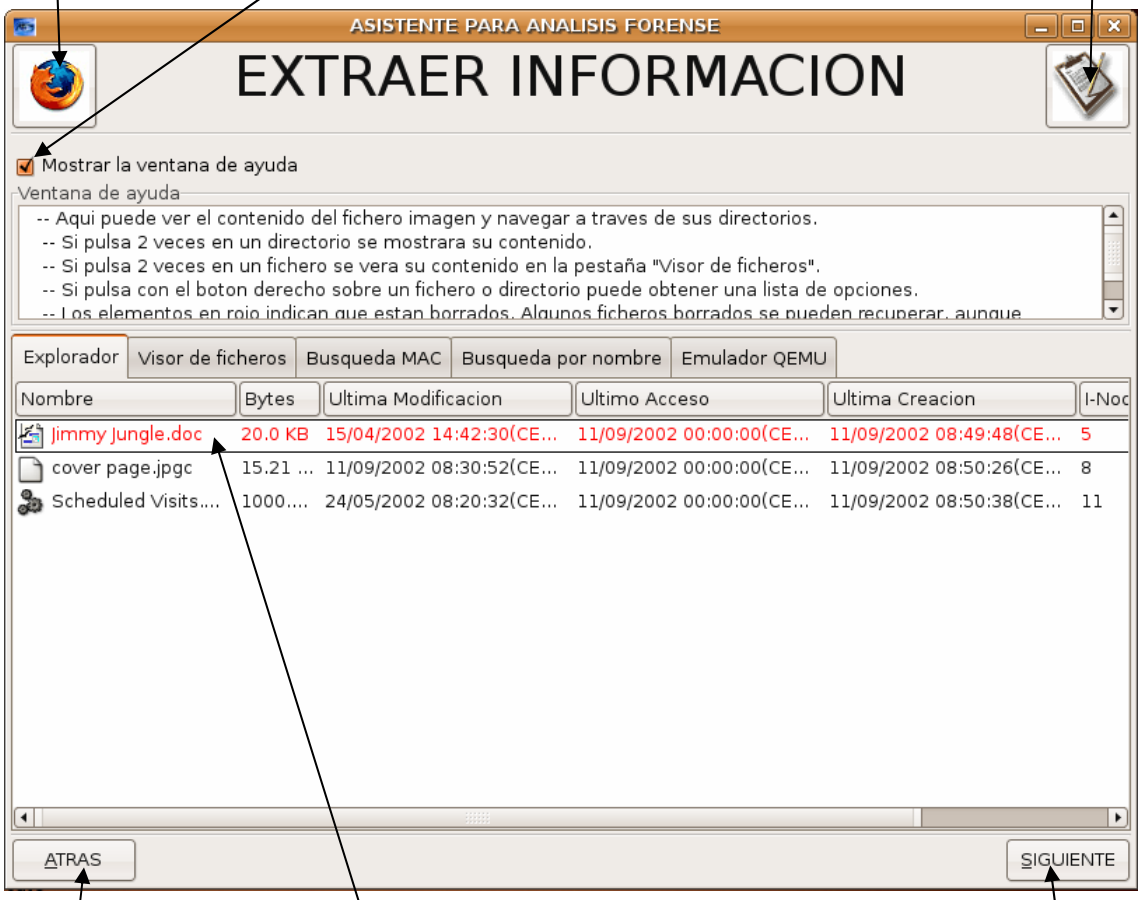
Una vez elegida una imagen de cualquiera de estas 3 maneras pulsamos el botón Siguiente para continuar con el asistente. Si la aplicación reconoce el formato de la imagen y ésta contiene un sistema de ficheros correcto, se mostrarán los datos del sistema de ficheros:

Una vez revisados estos datos podemos elegir si analizar esta imagen o no. Para aceptar pulsamos en Si y pasaremos a la siguiente pantalla en la que se podrá analizar la imagen seleccionada.

Pulsando este botón se abrirá el explorador de Internet con el buscador Google

Pulsando este botón se abrirá un editor de textos para introducir una nueva anotación en el fichero notas.txt

Pulsando aquí se puede ocultar/mostrar la Ventana de ayuda



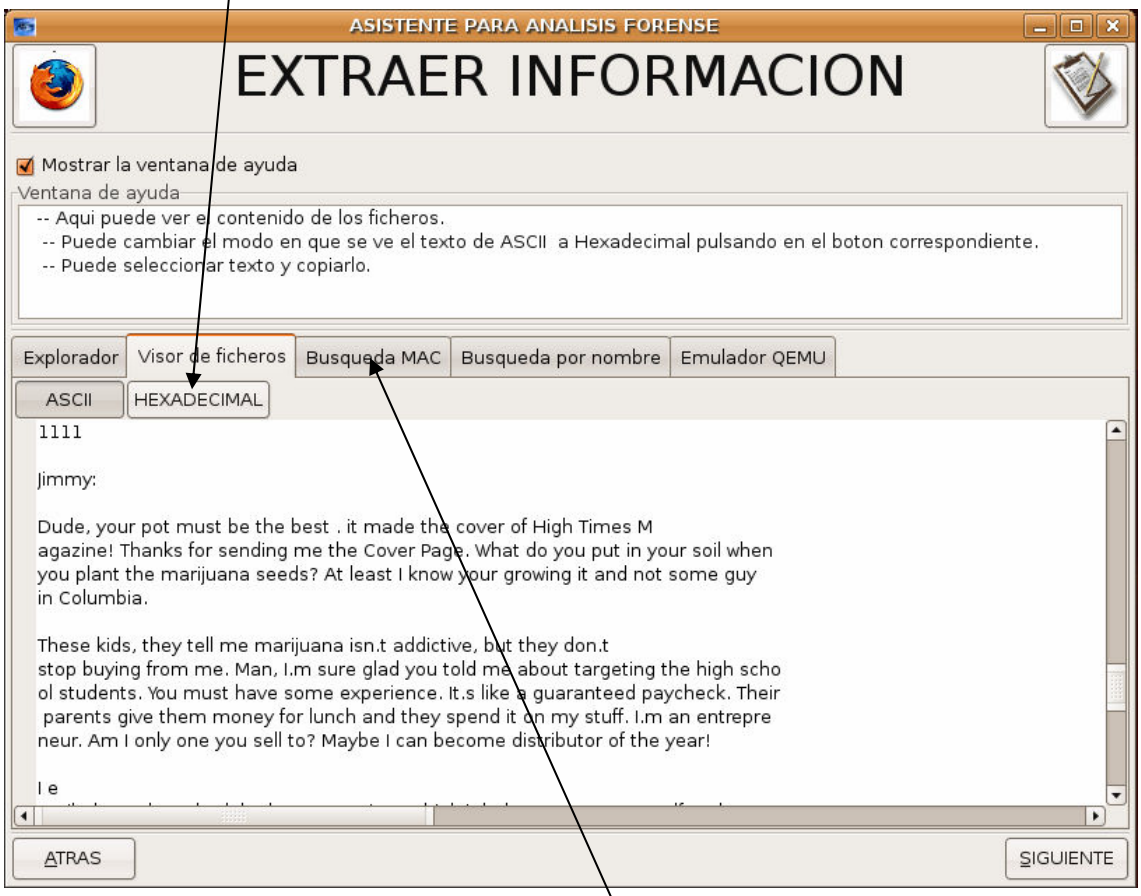
Aquí se muestran los ficheros que contiene la imagen. En caso de estar en rojo significa que estan borrados.

Pulsando aquí se vuelve a la pantalla para elegir el fichero imagen

Si pulsa aquí se accede a la siguiente ventana: Conclusiones

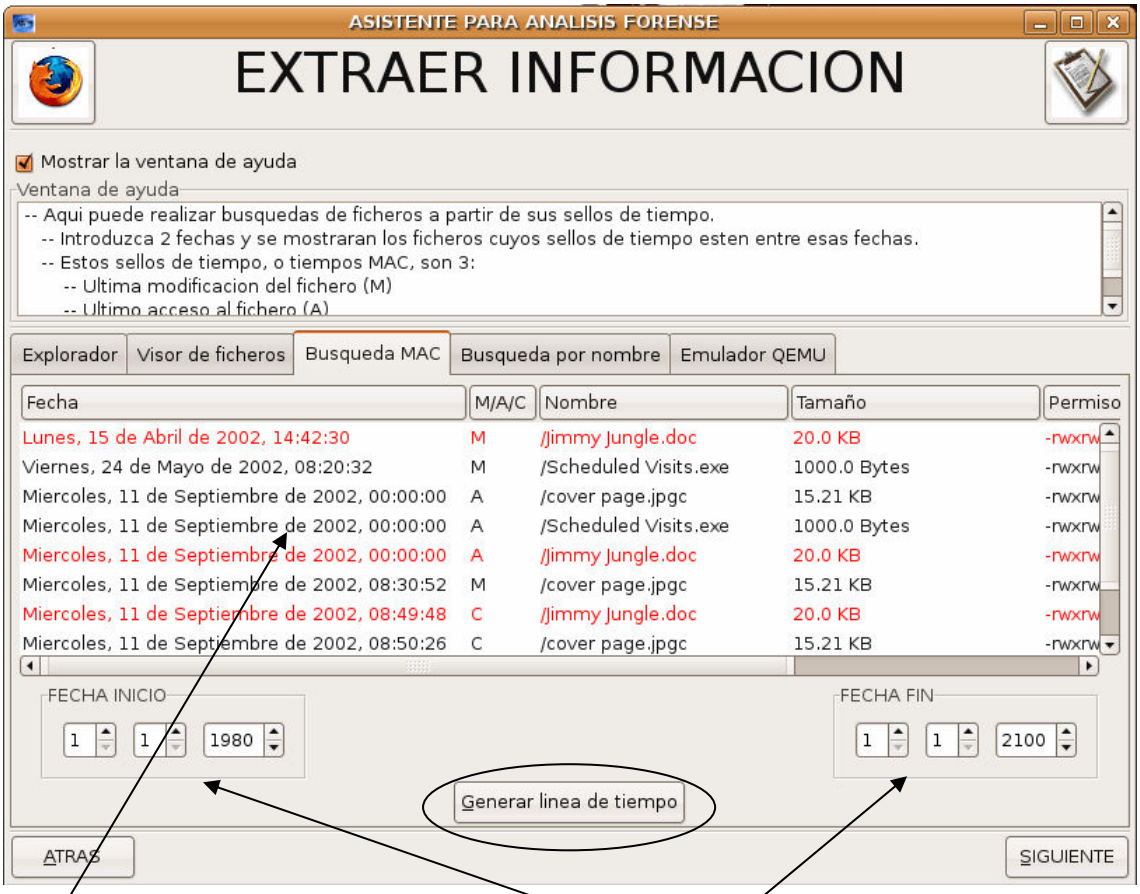
Si pulsamos con el botón derecho sobre un item(un fichero o directorio) se mostrará un menú contextual con una lista de opciones:

Si pulsa dos veces con el ratón sobre un fichero se verá su contenido en la siguiente pestaña: visor de ficheros. Puede ver su contenido en formato ASCII o en HEXADECIMAL.



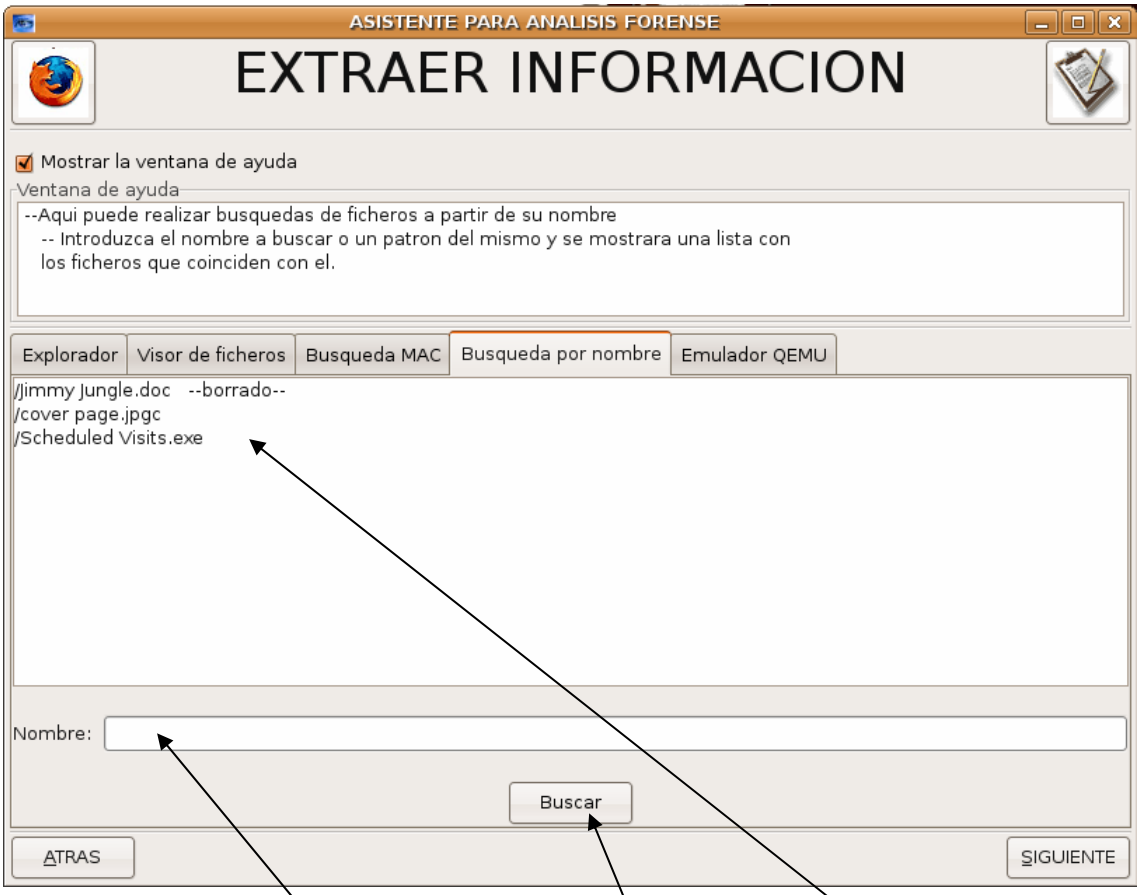
Si pulsa dos veces sobre un directorio se mostrarán los ficheros y directorios que contiene en la misma pestaña.

La siguiente pestaña se llama Busqueda MAC y muestra los ficheros cuya actividad (Creado, Accedido o Modificado) este entre dos fechas:



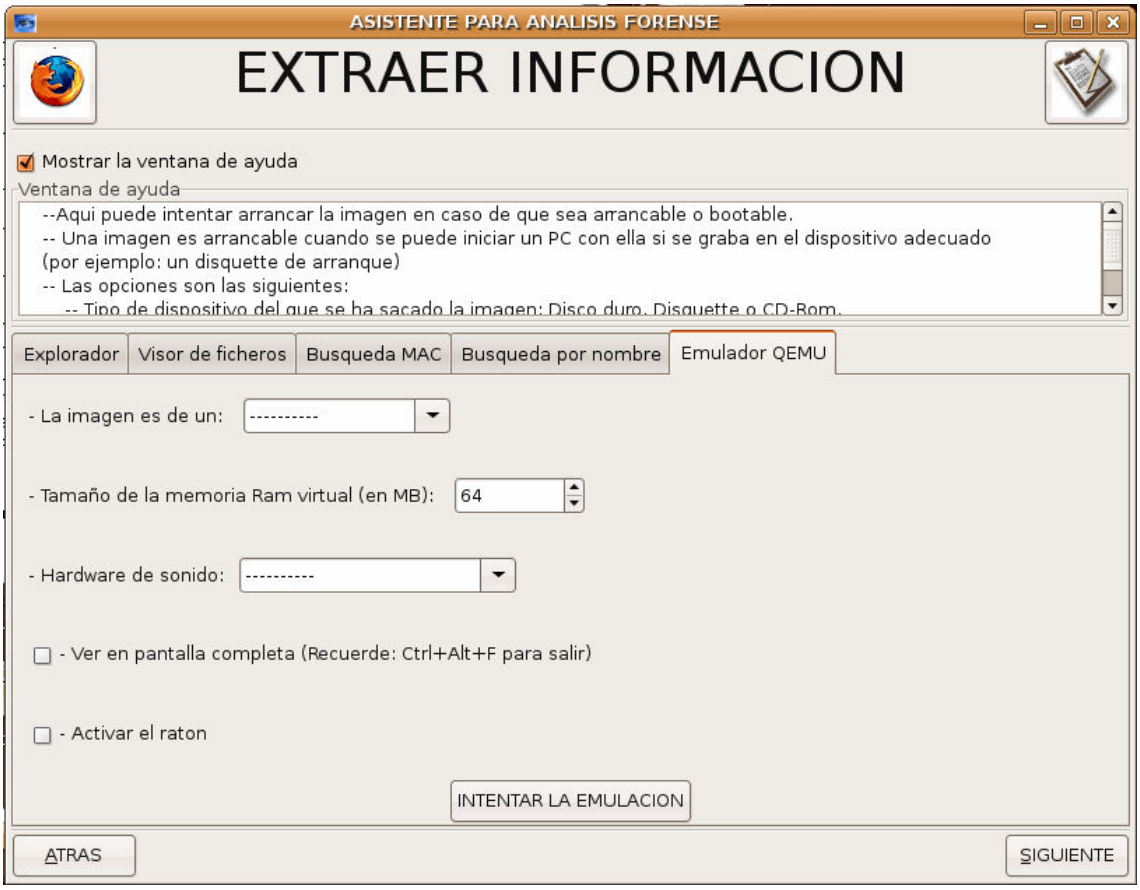
Este listado se genera una vez introducidas las fechas y pulsado el botón "Generar línea de tiempo".

La pestaña "Busqueda por nombre" realiza una busqueda por todo el sistema de ficheros hasta encontrar el/los nombre/s de ficheros/directorios que coincidan con un patron introducido:

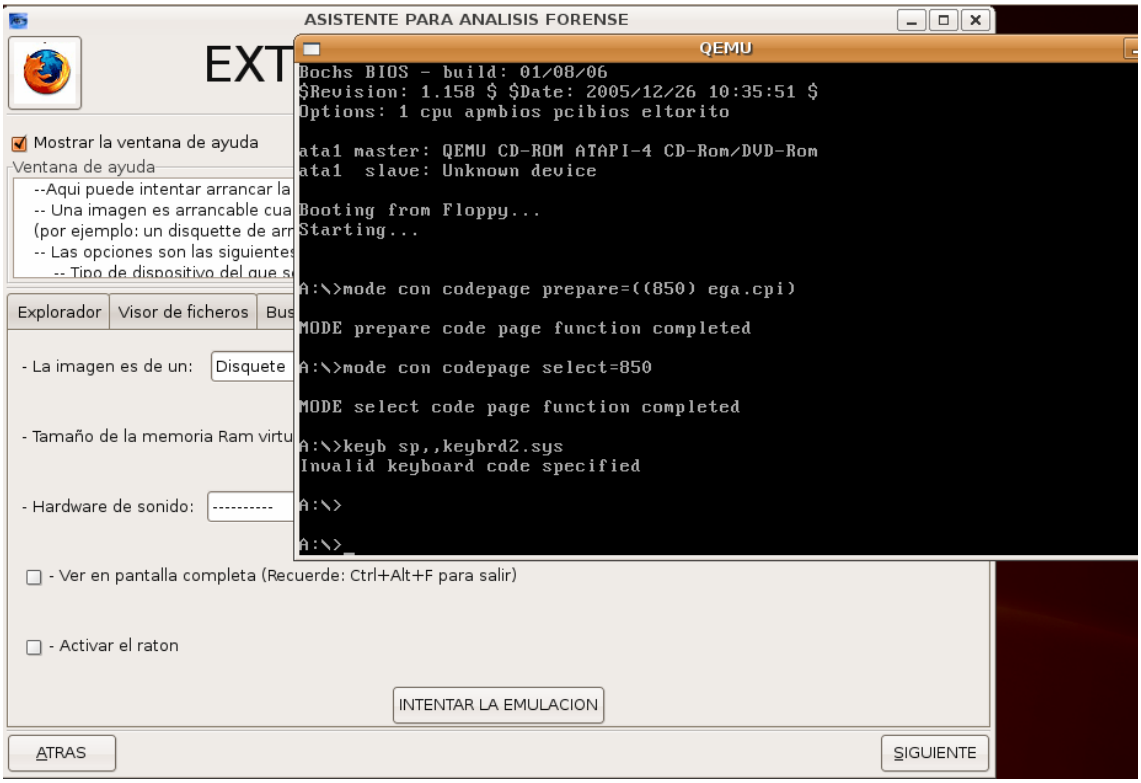


Si no se introduce ningún nombre para buscar, se mostrarán todos los ítems que contenga el fichero imagen una vez pulsado el botón Buscar.

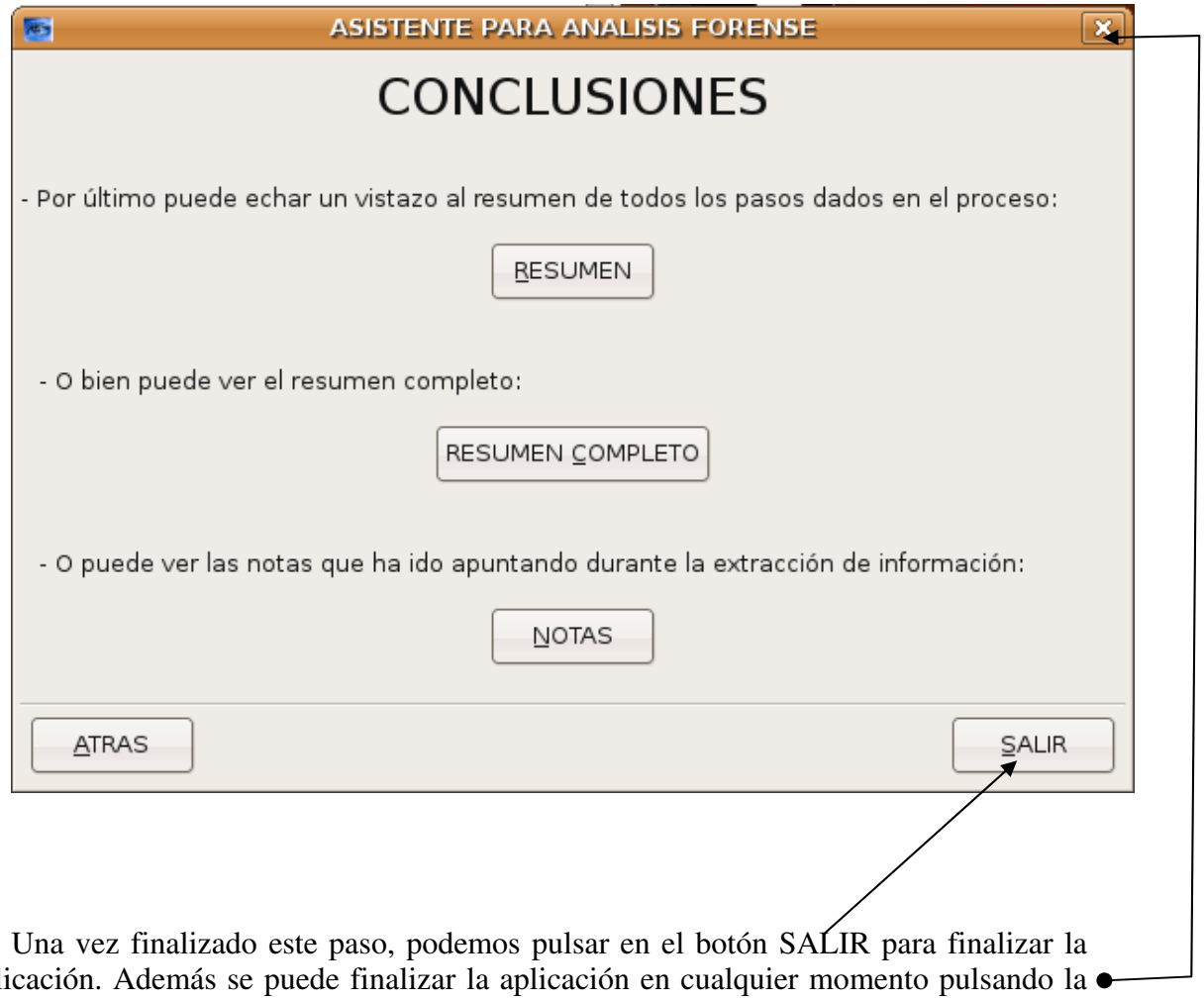
La última pestaña es un virtualizador de sistemas o emulador, basado en QEmu:



Introduciremos las opciones que más se adecúen a nuestro fichero imagen y pulsaremos en INTENTAR LA EMULACION. Si todo sale bien debe salir una ventana auxiliar mostrando el progreso de la emulación:



Y por último, la última ventana nos permite reflexionar sobre los pasos dados en la investigación. En el transcurso de la misma se deben haber tomado notas y además el programa ha ido anotando todos nuestros pasos para después poder verlos en un resumen simple o en uno extendido.



Una vez finalizado este paso, podemos pulsar en el botón SALIR para finalizar la aplicación. Además se puede finalizar la aplicación en cualquier momento pulsando la X de la equina superior derecha de todas las ventanas.

CONCLUSIONES

Aspectos a mejorar

Funcionalidad de la aplicación:

- Otras Busquedas:
    - Buscar ficheros por un numero de i-nodo
    - Buscar ficheros por tamaño
- Busquedas y recuperaciones de ficheros a nivel de aplicación: Para esto puede usarse una aplicación como FOREMOST que busca ficheros en un fichero imagen mediante un fichero de "numero mágicos" que lleva asociado cada tipo de fichero.
- Comunicación con otras herramientas con interfaz gráfica:
    - Autopsy: Interfaz web para The Sleuth Kit.
    - Pyflag: Programa escrito en Python destinado al análisis forense y especializado en el análisis de logs.
    - KQemu: es un acelerador de Qemu escrito en un script para Kommander. Para su ejecución es necesario instalar el intérprete Kommander en el sistema (70MB).
    - Antivirus: como por ejemplo, Clam-AV o F-Prot
    - Antirootkits: como por ejemplo, RKHunter.
- Mejorar la recuperación de ficheros: para recuperar ficheros se utiliza el comando icat que devuelve un fichero dado su inodo. En caso de estar borrado el inodo puede ser erroneo o puede que se haya escrito encima de los datos del fichero, por lo que sería conveniente aumentar la potencia de la recuperación de ficheros con otras herramientas existentes en caso de que icat falle.
- Uso de varios algoritmos para calcular el HASH: Md5 es rápido, pero tambien se puede usar SHA-1, RSA, etc.
- Realizar búsquedas de cadenas sobre el fichero imagen: el comando "strings" devuelve las cadenas de texto que contiene un fichero imagen. Usando este comando se podría ampliar la funcionalidad de la aplicación.
- Ordenación de ficheros: por tipo (fichero/directorio), nombre, metadatos (inodo, sellos de tiempo, dirección lógica de las unidades de datos).

- Comprobar si un fichero es de un tipo conocido por su valor MD5 buscando en una base de datos local o de Internet. Existen bases de datos con los valores hash de los ficheros que componen el sistema operativo de multitud de sistemas Unix, Windows y MAC.

- Incluir herramientas avanzadas: criptografía, esteganografía, etc.

Cd-Live:

- Hacer que la aplicación se ejecute al iniciar el CD.

- Cambiar el fondo de pantalla.

- Cambiar el icono que se muestra por defecto al insertar el Cd.

- Eliminar la parte correspondiente a Windows.

Resumen

Primeramente se ha llevado a cabo un repaso sobre la historia y los diferentes modelos presentados para el análisis forense de computadoras. Siguiendo el modelo de Casey se han visto las diferentes tareas que ocupa y se ha profundizado en la más importante y compleja: el análisis de los datos o extracción de información. Dentro de este hemos visto las distintas partes a revisar dentro de un dispositivo y pasando por las distintas capas de un sistema de ficheros se han revisado las técnicas usadas para recuperar datos.

A continuación se han recopilado todas las herramientas que se usan o se han usado en los análisis forenses de computadoras, desde la adquisición de datos al examen de los mismos, pasando por herramientas específicas para una parte de un sistema de ficheros (por ej, metadatos), hasta llegar a las herramientas de análisis de discos.

En base a lo visto anteriormente se ha planteado una aplicación asistente que aparte de servir como herramienta proporcionando acceso a las herramientas, sirva de forma didáctica a un usuario que carece de conocimientos sobre este área. Usando herramientas de libre distribución como Python, The Sleuth Kit, o QEMU, se ha desarrollado una aplicación útil e intuitiva que guía al usuario sobre el proceso de análisis forense y sobre el manejo de la propia aplicación.

Esta aplicación se ha incluido en un Cd-Live autoarrancable de manera que pueda ser probada al arrancar con este Cd. Para la creación de este Cd se han realizado varios scripts que automatizan en cierto modo el proceso.

Posteriormente se ha probado la aplicación sobre varios sistemas de ficheros, LINUX y FAT, realizando pruebas en relación a la naturaleza de la imagen. Aquí se comprueba que la aplicación a pesar de ser bastante completa, esta lejos de ser más que una aplicación con una mera función didáctica y en cualquier caso, de recuperación de datos, ya que un análisis forense implica el uso de todas las técnicas y herramientas conocidas por el investigador, así como la realización de largos exámenes, exhaustivos y precisos.

Bibliografía

Metodología forense

[1] **Computer Forensics: Computer Crime Scene Investigation**, John R. Vacca, Charles River Media © 2002 (731 paginas) ISBN:1584500182

[2] **An Extended Model of Cybercrime Investigations**, de Séamus Ó Ciardhuáin, International Journal of Digital Evidence Summer 2004, Volume 3, Issue 1

[3] **Electronic Crime Scene Investigation: A guide for first responders**, U.S. Department of Justice

[4] **Report From the First Digital Forensic Research Workshop** (**DFRWS**), Agosto de 2001, Utica, New Cork

[5] **An Examination of Digital Forensic Models**, de Mark Reith, Clint Carr, Gregg Gunsch, International Journal of Digital Evidence Fall 2002, Volume 1, Issue 3

[6] **Getting Physical with the Digital Investigation Process**, de Brian Carrier y Eugene H. Spafford, International Journal of Digital Evidence Fall 2003, Volume 2, Issue 2

[7] **Digital Evidence and Computer Crime: Forensic Science**, Computers, and the Internet, Segunda edición, por Eoghan Casey, Academic Press 2004, ISBN:0121631044

Legislación

[8] www.policia.es/bit

Herramientas Software

[9] LINReS: http://www.niiconsulting.com/innovation/linres.html
[10] SMART: http://www.asrdata.com
[11] The BlackBag Macintosh Forensic Software: http://www.blackbagtech.com/software_mfs.html
[12] MacForensicsLab : http://www.macforensicslab.com/
[13] BringBack: http://www.toolsthatwork.com/bringback.htm
[14] EnCase: http://www.guidancesoftware.com/
[15] FBI: http://www.nuix.com.au
[16] Forensics Toolkit (FTK): http://www.accessdata.com/products/ftk/
[17] ILOOK Investigator: http://www.ilook-forensics.org/
[18] SafeBack: http://www.forensics-intl.com/safeback.html
[19] X-Ways Forensics: http://www.x-ways.net/forensics/index-m.html

[20] Pro Discover Forensics: http://www.techpathways.com/ProDiscoverWindows.htm
[21] Autopsy: http://www.sleuthkit.org/autopsy/desc.php
[22] Foremost: http://foremost.sf.net/
[23] FTimes: http://ftimes.sourceforge.net/FTimes/index.shtml
[24] GFZip: http://www.nongnu.org/gfzip/
[25] Gpart: http://www.stud.uni-hannover.de/user/76201/gpart/
[26] Magic Rescue: http://jbj.rapanden.dk/magicrescue/
[27] PyFlag: http://pyflag.sourceforge.net/
[28] Scalpel: http://www.digitalforensicssolutions.com/Scalpel/
[29] Scrounge-Ntfs: http://memberwebs.com/nielsen/software/scrounge/
[30] The Sleuth Kit: http://www.sleuthkit.org/
[31] The Coroner's Toolkit: http://www.porcupine.org/forensics/tct.html
[32] ZeitLine: http://projects.cerias.purdue.edu/forensics/timeline.php
[33] ZeitLIne2: http://sourceforge.net/projects/zeitline/
[34] AntiWord: http://www.winfield.demon.nl/
[35] Catdoc y XLS2CSV: http://www.45.free.net/~vitus/software/catdoc/
[36] JHead: http://www.sentex.net/~mwandel/jhead/
[37] VINETTO: http://vinetto.sourceforge.net/
[38] Word2x: http://word2x.sourceforge.net/
[39] WvWare: http://wvware.sourceforge.net/
[40] XPdf: http://www.foolabs.com/xpdf/
[41] Metadata Assistant: http://www.payneconsulting.com/products/metadataent/
[42] Galleta: http://www.foundstone.com/resources/proddesc/galleta.htm
[43] Pasco: http://www.foundstone.com/resources/proddesc/pasco.htm
[44] Rifiuti: http://www.foundstone.com/resources/proddesc/rifiuti.htm
[45] Yim2Text: http://www.1vs0.com/tools.html
[46] NetIntercept: http://www.sandstorm.net/products/netintercept
[47] Sguil: http://sguil.sourceforge.net/
[48] Snort: http://www.snort.org/
[49] Tcpdump: http://www.tcpdump.org
[50] Tcpextract: http://tcpxtract.sourceforge.net/
[51] Tcpflow: http://www.circlemud.org/~jelson/software/tcpflow/
[52] TrueWitness: http://www.nature-soft.com/forensic.html
[53] Etherpeek: http://www.wildpackets.com/products/etherpeek/overview
[54] BringBack: http://www.toolsthatwork.com/
[55] ByteBack DRIS: http://www.toolsthatwork.com
[56] RAID Reconstructor: http://www.runtime.org/raid.htm
[57] Salvation Data: http://www.salvationdata.com
[58] Partition Table Doctor: http://www.ptdd.com/index.htm
[59] TestDisk: http://www.cgsecurity.org/wiki/TestDisk
[60] Ewfacquire: https://www.uitwisselplatform.nl/projects/libewf/
[61] Adepto (Grab): http://www.e-fense.com/helix/
[62] GNU ddrescue: http://www.gnu.org/software/ddrescue/ddrescue.html
[63] dd_rescue: http://www.garloff.de/kurt/linux/ddrescue/
[64] rdd: http://sourceforge.net/projects/rdd
[65] Qemu: http://www.qemu.org

[66] VMWare: http://www.vmware.com/
[67] Biew: http://biew.sourceforge.net/en/biew.html
[68] Hex WorkShop: http://www.bpsoft.com
[69] Khexedit: http://docs.kde.org/stable/en/kdeutils/khexedit/index.html
[70] WinHex: www.winhex.com
[71] DBAN: http://dban.sourceforge.net/
[72] Lenovo SDD: http://www-307.ibm.com/pc/support/site.wss/document.do?sitestyle=lenovo&lndocid=MIGR-56394
[73] Timestomp: http://www.metasploit.com/projects/antiforensics/timestomp.exe
[74] Slacker: http://www.metasploit.com/projects/antiforensics/slacker.exe

Creación del CD-Live de Ubuntu

[75] Cd-Live Ubuntu, HowTo: https://help.ubuntu.com/community/LiveCD

[76] Ubuntu Live Cd (español): http://formacion.cnice.mec.es/materiales/43/cd/cap1/maqueta1_frame.htm

Imágenes forenses

[77] Scan26: http://www.honeynet.org/scans/scan26/

[78] I Reto Forense de Rediris: http://www.rediris.es/cert/ped/reto/ficheros.html

ANEXOS

A) HARDWARE PARA ANÁLISIS FORENSE DE COMPUTADORAS

En el mundo de la informática forense no solo se cuenta con herramientas software para las investigaciones, tambien existe un enorme abanico de soluciones hardware para el apoyo en estas tareas. Ya que esta investigación se sale del ámbito de este proyecto, tan solo se muestran las imágenes de los mismos a modo de ilustración para el lector.

Copiadoras duplicadoras

Su función es la de tomar un dispositivo y realizar una copia de manera rápida. En la imagen de la derecha vemos un dispositivo amarillo que contiene un disco duro en su interior y que realiza una copia de otro disco duro conectado a él.

PCs Previewers: Equipos de investigación en caliente no intrusivos

Son equipos muy caros y sofisticados usados para conectar a un PC que este en funcionamiento (en caliente) y extraer todos los datos útiles para un investigador de manera automática.

Bloqueadores de escritura (Write Blockers)

Se trata de unos dispositivos que impiden que se escriba en un disco duro (generalmente). Su utilidad radica en que al conectar un disco duro y arrancar con el normalmente los sistemas operativos realizan alguna escritura en el mismo, como por ejemplo, actualizar la fecha de acceso.

Estaciones de trabajo forenses

Son equipos muy caros e incluyen toda la funcionalidad hardware y software necesaria para realizar un buen análisis forense. Existen muchos tipos de estaciones de trabajo forenses, aunque se pueden distinguir las que son fijas de las que son transportables. En cualquier caso el tamaño de ambas suele ser bastante grande.

Análisis de Red

Son dispositivos que se conectan entre el sistema a analizar y la conexión de red que va al mismo. Realizan logs con todas los datos que se transmitan y reciban, además de remarcar y sacar conclusiones con datos relevantes o peligrosos.