RESUMEN - TESIS DE GRADO

AUTORES	**LEONARD DAVID LOBO PARRA**
FACULTAD	**DE INGENIERÍAS**
PLAN DE ESTUDIOS	**INGENIERÍADE SISTEMAS**
DIRECTOR	**DEWAR WILMER RICO BAUTISTA**
TÍTULO DE LA TESIS	**DESARROLLO E IMPLEMENTACIÓN DEL ANÁLISIS DIGITAL FORENSE UTILIZANDO UNA METODOLOGÍA POST-MORTEM.**

RESUMEN
(70 palabras aproximadamente)

La presente investigación tiene por objeto estudiar el análisis digital forense; específicamente la técnica post-mortem, tomando como referencia el análisis en caliente; tambien se estudian diversas metodologías y estándares internacionales, identificando el modelo que más se adecue, para aplicarlo a laboratorios donde tras poner a prueba diversas herramientas sobre el S.O. Backtrack, se escoge la suit AUTOPSY y a partir de este proceso se construye la metodología practica para el análisis forense post-mortem “SIGLAS”

CARACTERÍSTICAS			
PÁGINAS: 119	PLANOS:	ILUSTRACIONES: 30	CD-ROM: 1

DESARROLLO E IMPLEMENTACIÓN DEL ANÁLISIS DIGITAL FORENSE UTILIZANDO UNA METODOLOGÍA POST-MORTEM.

LEONARD DAVID LOBO PARRA

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA
FACULTAD DE INGENIERÍAS
INGENIERÍADE SISTEMAS
OCAÑA
2014

DESARROLLO E IMPLEMENTACIÓN DEL ANÁLISIS DIGITAL FORENSE UTILIZANDO UNA METODOLOGÍA POST-MORTEM.

LEONARD DAVID LOBO PARRA

Trabajo de grado presentado para optar el título de Ingeniero de Sistemas

Director
DEWAR WILMER RICO BAUTISTA
Magister en Ciencias Computacionales

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA
FACULTAD DEINGENIERÍAS
INGENIERÍADE SISTEMAS
OCAÑA
2014

CONTENIDO

LISTA DE FIGURAS

Pág.

LISTA DE ANEXOS

Pág.

RESUMEN

La presente investigación tiene por objeto estudiar el análisis digital forense; específicamente la técnica post-mortem, tomando como referencia el análisis en caliente; también se estudian diversas metodologías y estándares internacionales, identificando el modelo que más se adecue, para aplicarlo a laboratorios donde tras poner a prueba diversas herramientas sobre el S.O. Backtrack, se escoge la suit AUTOPSY y a partir de este proceso se construye la metodología practica para el análisis forense post-mortem "SIGLAS"

INTRODUCCIÓN

La presente investigación pretende ilustrar el trabajo a realizar con respecto a la investigación "Desarrollo e implementación de un análisis digital forense usando la metodología post-mortem". El entorno de estudio es el laboratorio del semillero de investigación SIGLAS (Gnu Linux And Security) de la universidad Francisco de Paula Santander Ocaña; y fundamentalmente se busca establecer la efectividad de la metodología post-mortem dentro de la realización de un análisis digital forense, puesto que existen dos maneras para realizar dicha investigación. En primera instancia encontramos la metodología en caliente o en vivo, donde el equipo o terminal a analizar se encuentra encendida, y por consiguiente los programas en ejecución son visibles y la totalidad de la funcionalidad de la máquina, no obstante, también están latentes las trampas del atacante que pudieran dejar fuera de servicio el equipo u ocultar el ataque en sí mismo.

Cuando el análisis se realiza sobre el disco duro es importante tener en cuenta que la forma de apagado es sumamente importante, así como también la forma en la que se adquiere, preserva y analizar la información (cadena de custodia), puesto que de ello depende la credibilidad y fiabilidad que esta pueda llegar a tener en un estrado judicial como prueba científica.

Para realizar todo el proceso de peritaje informático es indispensable contar con herramientas que faciliten y viabilicen el proceso, en el presente artículo se describirá el uso de la distribución BACKTRACK 5 r1.

1. DESARROLLO E IMPLEMENTACIÓN DEL ANÁLISIS DIGITAL FORENSE UTILIZANDO UNA METODOLOGÍA POST-MORTEM.

1.1 PLANTEAMIENTO DEL PROBLEMA

En la actualidad hay una marcada tendencia al aumento de ataques informáticos provenientes de variadas fuentes que ponen al descubierto las falencias de diversos sistemas operativos y de seguridad. Dichos ataques pueden ser recibidos por usuarios comunes de la red o por usuarios tan importantes como entidades gubernamentales, muchas veces poniendo en riesgo informaciones tan secretas como las financieras o secretos de estado; por estas razones, se ha visto la necesidad imperiosa de desarrollar métodos o aplicaciones con el fin de contrarrestarlas.

1.2 FORMULACION DEL PROBLEMA

¿Constituye la metodología post-mortem una técnica eficiente para realizar un análisis forense?

1.3 JUSTIFICACION

En los últimos años se ha visto un incremento en los ataques informáticos a objetivos tan diversos como personas particulares, entidades financieras o agencias gubernamentales de tal forma que se ha aumentado la vulnerabilidad de muchos sistemas; por esta razón, se ha visto la necesidad de desarrollar e implementar diversas técnicas que permitan hacer un correcto análisis forense digital. Cuando suceden los ataques y la seguridad informática falla, es necesario evaluar los ataques, ya sea para propósitos judiciales o para analizar los motivos por los cuales se vulneró el sistema con el fin de mejorar la seguridad. Este análisis se puede realizar en las modalidades on line y post – mortem, presentando cada una ventajas y desventajas; en este proyecto, se utilizará el análisis informático forense post – mortem, con el fin de evaluar el daño o las implicaciones causados por cierto ataque informático mediante el uso de varias metodologías preestablecidas.

1.4 OBJETIVOS DE LA INVESTIGACIÓN

1.4.1 General. Desarrollar e implementar el análisis forense digital utilizando una metodología post-mortem.

1.4.2 Específicos. Analizar las diferentes metodologías existentes con el fin de seleccionar una de ellas para la evaluación de un ataque informático.

Conocer y evaluar las distintas herramientas y técnicas de adquisición y análisis de datos de un dispositivo de almacenamiento físico proponiendo una opción optima, funcional y eficiente, para implementar el análisis forense.

Evaluar el potencial desempeño del análisis forense post-mortem en ataques informáticos estableciendo el alcance y las implicaciones de las actividades ilícitas realizadas por un intruso en un sistema.

Proponer un procedimiento estructurado y específico para la aplicación del análisis digital forense empleando la metodología post-mortem en casos prácticos.

Es fundamental a la hora de realizar un medición y estructuración de la consecución de los anteriores objetivos, el trazar actividades e indicadores y permitan establecer la efectividad en dicho proceso. Para tal efecto se propone el siguiente esquema:

OBJETIVO GENERAL	OBJETIVO ESPECIFICO	ACTIVIDADES	INDICADOR
Desarrollar e implementar análisis forense digital utilizando una metodología post-mortem.	•Analizar las diferentes metodologías existentes con el fin de seleccionar una de ellas para la evaluación de un ataque informático.	* Revisión bibliográfica de metodologías existentes	* Metodología escogida
		*Comparación con estándares internacionales	
		* Comparación experimental	
	•Conocer y evaluar las distintas herramientas y técnicas de adquisición y análisis de datos de un dispositivo de almacenamiento físico proponiendo una opción optima, funcional y eficiente, para implementar el análisis forense.	* consulta bibliográfica y comparación de las herramientas, distribuciones de software y siuits de análisis forense	* Suit forense *Herramientas de apoyo * Técnica para la adquisición de datos
		* comparar experimentalmente las diferentes técnicas de adquisición de datos.	
	* Evaluar el potencial desempeño del análisis forense post-mortem en ataques informáticos estableciendo el alcance y las implicaciones de las actividades ilícitas realizadas por un intruso en un sistema.	* Diseño e implementación de los laboratorios propuestos	Informe de desempeño y efectividad del análisis.
		* Diseño e implementación de un ataque informático	
		* Realización de análisis digital forense para cada caso propuesto.	
		* Comparación de los resultados obtenidos.	

	• Proponer un procedimiento estructurado y específico para la aplicación del análisis digital forense empleando la metodología post-mortem en casos prácticos.	* Análisis de procedimientos recurrentes en los distintos laboratorios.	Metodología practica propuesta.
		* Estructuración y formalización de procedimientos recurrentes mediante la comparación con estándares internacionales y metodologías existentes.	
		* Formulación de la propuesta metodológica para el análisis digital forense con metodología post-mortem.	

1.5 DELIMITACIONES

1.5.1 Geográfica. Este proyecto se llevara a cabo en las instalaciones de la Universidad Francisco de Paula Santander de Ocaña.

1.5.2 Conceptuales. Durante el desarrollo de la investigación se encontraron una serie de conceptos, de teorías, pensamientos que dan lugar a una información más precisa sobre el tema. Estos conceptos de análisis forense, de ataques informáticos, del software Backtrack y el marco legal vigente para este tipo de casos.

1.5.3 Temporales. El proyecto tendrá un tiempo de realización de 48 semanas, de acuerdo a las actividades a realizar.

1.5.4 Operativas. Este proyecto se va a realizar en el área de análisis forense informático la cual es un área de investigación incipiente en la Universidad Francisco de Paula Santander Ocaña. Uno de los posibles obstáculos que puede presentarse en el ejercicio investigativo del ataque informático es que no se cuenta con el suficiente conocimiento en dicha área, por tal motivo se hace indispensable la asesoría de expertos en el área.

2. MARCO REFERENCIAL

2.1 ANTECEDENTES HISTÓRICOS

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"[1].

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 Crimen 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 informática 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

[1]Análisis forense Informático. Historia del Análisis forense digital [En Línea] Disponible en Internet en: <http://wwwdi.ujaen.es/~mlucena/bin/proy_forense.pdf>

almacenada o transmitida en formato binario"[2]. Más tarde "binario" cambió a "digital". La evidencia digital incluye hardware, audio, video, teléfonos móviles, impresoras, etc.

Ese mismo año se celebra el primer Simposios 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.

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.

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 están limitadas solamente a los agentes de la ley y se han desarrollado también en el mundo empresarial. Aun 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 líneas 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

[2]Análisis forense. Cómo investigar un incidente de seguridad [En Línea] Disponible en Internet en: <http://www.k-nabora.com/index.php/blog/AnA-lisis-forense.-CA-mo-investigar-un-incidente-de-seguridad-248.html>

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[3], 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)
¿Cómo se realizó el ataque? (Vulnerabilidad o fallo empleado para acceder al sistema)
¿Qué hizo el atacante? (Qué acciones realizó el atacante una vez que accedió 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, El sevier comenzó la publicación de una nueva revista llamada "Digital Investigation: The International Journal of Digital Forensics and Incident Response"

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.

2.2 MARCO TEORICO

Las *RFC*[4] por sus siglas en inglés (*Request For Comments*) son un conjunto de documentos que sirven de referencia para la comunidad de Internet, que describen, especifican y asisten en la implementación, estandarización y discusión de la mayoría de las normas, los estándares, las tecnologías y los protocolos relacionados con Internet y las redes en general. La metodología que se utiliza con las *RFC* es asignarle a cada una un número único que la identifique y que es el consecutivo de la última *RFC* publicada. Una *RFC* ya publicada

[3]Seguridad en la red y análisis forense, Hades. Universidad de Murcia – Facultad de Informática. Murcia, España: Administración y seguridad en redes.
[4]RequestForComments. [on line] Disponible en internet en: http://es.kioskea.net/contents/internet/rfc.php3. [citado el 29 de junio de 2011]

jamás puede modificarse, no existen varias versiones de una RFC. Lo que se hace, en cambio, es escribir una nueva *RFC* que deje obsoleta o complemente una *RFC* anterior.

El RFC 3227[5] Directrices para la colección de archivos y pruebas, proporciona un interesante paso a paso de las buenas prácticas para los administradores de sistemas con directrices sobre a recopilación y archivo de las pruebas pertinentes a incidentes de seguridad.

La ggestión de incidentes de seguridad de la información basada en la norma ISO 27001[6] en su inciso A.13.1 garantizar que los eventos y las debilidades de la seguridad de la información asociados con los sistemas de información están comunicados de una forma que permita tomar acciones correctivas precisas.

Por otro lado la realización de informes de eventos de seguridad de información es normada por la ISO 27001 en su sección A.13.1.2 la cual busca garantizar que un enfoque consistente y efectivo es aplicado en la administración de los incidentes de seguridad de información, así mismo la norma ofrece soporte respecto a responsabilidades y procedimientos en la parte A.13.2.2 y también se habla del aprendizaje desde los incidentes de seguridad de la información en el inciso A.13.2.3 así como de Recolección de la evidencia.

El estándar ISO 20000 proporciona unos aspectos de carácter técnico para el manejo de incidentes, no obstante para comprender el estándar de dicha norma resulta casi obligatorio hacer una referencia a otro estándar del mundo de las TI: ITIL (Information Technology Infraestructure Library). ITIL es un entorno de trabajo que engloba la "Gestión de Servicios de Tecnologías de la Información" (TI). Reúne un conjunto de las mejores prácticas recogidas por la "Oficina Gubernamental de Comercio Británica" donde se describen los procesos necesarios para administrar el área TI eficazmente, a fin de optimizar beneficios y garantizar la integración de los servicios en la cadena de valor de las unidades de negocio. Constituye pues una biblioteca de "Buenas Prácticas de la Gestión de Servicios de TI"[7]

ISO 20000 constituye el estándar reconocido internacionalmente para la gestión de servicios de TI. Mencionar cómo la serie ISO 20000 proviene de la adecuación de la BS 15000. El estándar ISO 20000 se divide en dos partes, al finalizar la primera nos trompamos con la sección dedicada a procesos de resolución dentro de la cual se orienta al manejo de incidentes de seguridad en los incisos 8.1 Antecedentes, 8.2 Gestión del incidente y 8.3 Gestión del problema.

---

[5] Request for Comments: 3227 Guidelines for Evidence Collection and Archiving. [on line] Disponible en internet: http://www.ietf.org/rfc/rfc3227.txt

[6] Norma ISO 27001 [on line] Disponible en internet en: http://es.scribd.com/doc/25034834/NORMA-ISO27001

[7] soluciones en las empresas de ti mediante la aplicación de un sistema de gestión ISO 20000 parte 1 integrado a un sistema ISO 27001 e ISO 9001, José Mª Zubieta Guillén, escuela técnica superior de ingenieros industriales y de telecomunicación, 2010.[En Línea] Disponible en Internet en: http://academica-e.unavarra.es/bitstream/handle/2454/2166/577243.pdf?sequence=1

El proceso DS8 de la norma administrar la mesa de servicios e incidentes de la COBIT[8] dice "El Soporte debe responde de manera oportuna y efectiva a las consultas y problemas de los usuario TI ", en otras palabras debe hacerse cargo de la necesidad inmediata del usuario, dejando muchas veces la solución del problema técnico de fondo para otra oportunidad y otro grupo de especialistas. Es para la búsqueda de la solución del problema técnico de fondo donde entrar a tallar la Solución de Problemas y en el marco de la metodología COBIT es el proceso DS10 Gestión de Problemas.

2.3 MARCO CONCEPTUAL

2.3.1 Análisis Forense Digital. Conjunto de principios y técnicas que comprende el proceso de adquisición, conservación, documentación, análisis y presentación de evidencias digitales y que llegado el caso puedan ser acepta-das legalmente en un proceso judicial.

2.3.2 Análisis forense en frio (Post-mortem). El análisis en frío recoge pruebas del sistema afectado cuando éste ya ha sido apagado. El ataque ya ha terminado y, al igual que en los casos policiales, es necesario recoger pruebas para conocer los hechos ocurridos.

2.3.3 Análisis forense en caliente (On-line). El análisis en caliente recoge pruebas en el sistema afectado estando éste todavía encendido. El ataque continúa en marcha, por lo que es un momento propicio para recoger evidencias (ya que tras el apagado de la máquina, algunas de éstas se perderán).

2.3.4 Encriptación informática. La encriptación informática es simplemente la codificación de la información que se va a enviar a través de la red (Internet). Para poder descodificarla es necesario un software o una clave que sólo conocen el emisor y el receptor de esta información.

La encriptación de la informática se hace cada vez más necesaria debido al aumento de los robos de claves, número de cuentas corrientes, y en general toda la información que viaja por la red[9].

2.3.5 Evidencia digital. Conjunto de datos en formato binario, esto es, comprende los ficheros, su contenido o referencias a éstos (meta-datos) que se encuentren en los soportes físicos o lógicos del sistema atacado.

2.3.6 Seguridad Informática[10]. Consiste en la protección conferida a un sistema de información automatizado con el fin de alcanzar los objetivos aplicables de preservar la integridad, disponibilidad y confidencialidad de los recursos del sistema de información.

---

[8] administrar la mesa de servicios e incidentes de la COBIT, proceso D8, [En Línea] Disponible en Internet en: http://www.slideshare.net/Bluedelacour/ds8-administrar-la-mesa-de-servicio-y-los-incidentes

[9]Análisis forense digital. Encriptación de disco, burla de tecnología. Dragonjar [En Línea] Disponible en Internet en: <http://www.dragonjar.org/burlada-tecnologia-de-encriptacion-de-discos.xhtml>

[10]WILLIAM STALLINGS. NETWORK SECURITY ESSENTIAL, Applications and Standars. Fourthedition. Año 2011.

Esta definición presenta tres objetivos principales que se encuentran en el corazón de la seguridad informática.

Confidencialidad. Este término se refiere a dos conceptos relacionados:

Confidencialidad de los datos: Asegura que la información privada o confidencial no esté disponible o revelada apersonas no autorizadas.

Privacidad: Asegura que el control de las personas o influir en la información que relacionados con ellos puede ser recogida y almacenada y que, por quién y para quién información puede ser revelada.

Integridad. Este término se refiere a dos conceptos relacionados:

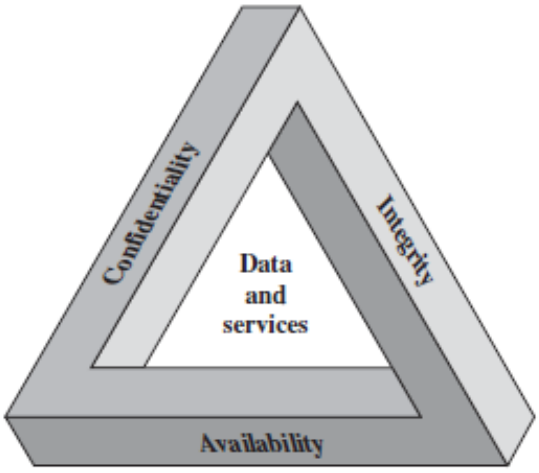
Integridad de los datos: Asegura que la información y los programas sólo se cambian de una manera específica y autorizada.

La integridad del sistema: Asegura que un sistema que pretende realizar una función deseada de una forma intacta, sin autorización deliberada o involuntaria no manipule el sistema.

Disponibilidad .Asegura que el sistemas trabaje inmediatamente y el servicio no se le niegue a los usuarios autorizados.

Estos tres conceptos forman lo que se refiere a menudo como la tríada de la CIA (**Figura 1**).Los tres conceptos abarcan los objetivos fundamentales de seguridad para ambos, datos y servicios de computación e información.

Figura 1. Requerimientos de seguridad informática.



Fuente. William Stallings, NETWORK SECURITY ESSENTIALS applications and standards. Fourth edition,

2.3.7 Incidente de Seguridad Informática. Es considerado como una violación o intento de violación de la política de seguridad, de la política de uso adecuado o de las buenas prácticas de utilización de los sistemas informáticos.

Se categorizan dichos incidentes en:

Incidentes de Denegación de Servicios (DoS): Son un tipo de incidentes cuya finalidad es obstaculizar, dañar o impedir el acceso a redes, sistemas o aplicaciones mediante el agotamiento de sus recursos
Incidentes de código malicioso: Cualquier tipo de código ya sea, virus, gusano, "caballo de Troya", que pueda ejecutarse en un sistema e infectarlo.
Incidentes de acceso no autorizado: Se produce cuando un usuario o aplicación accede, por medio de hardware o software, sin los permisos adecuados a un sistema, a una red, a una aplicación o los datos.
Incidentes por uso inapropiado: Se dan cuando los usuarios se "saltan" la política de uso apropiado de los sistemas (por ejemplo ejecutando aplicaciones P2P en la red interna de la organización para la descarga de música)[11].

2.4 MARCO LEGAL

2.4.1 Leyes para la regulación en las telecomunicaciones en Colombia. Ley 1273 de 2009"De la protección de la información y de los datos"[12]

Congreso de la república. Por medio de la cual se modifica el Código Penal, se crea un nuevo bien jurídico tutelado -denominado "de la protección de la información y de los datos"-y se preservan integralmente los sistemas que utilicen las tecnologías de la información y las comunicaciones, entre otras disposiciones.

EL CONGRESO DE COLOMBIA

Decreta:

Artículo1o.Adiciónese el Código Penal con un Título VII BIS denominado "De la Protección de la información y de los datos".

De los atentados contra la confidencialidad, la integridad y la disponibilidad de los datos y de los sistemas informáticos. Artículo 269ª. *Acceso abusivo a un sistema informático*. El que, sin autorización o por fuera de lo acordado, acceda en todo o en parte a un sistema informático protegido o no con una medida de seguridad, o se mantenga dentro

[11]Experiencias de análisis forense en México. Departamento de Seguridad en Cómputo / UNAM-CERT en Jornadas de Análisis Forense. Madrid, Septiembre 2005.
[12]34SECRETARIASENADO, Ley 1273 de 2009 [en línea].<http://www.secretariasenado.gov.co/senado/basedoc/ley/2009/ley_1273_2009.html> [citado el 11 de abril de 2010]

del mismo en contra de la voluntad de quien tenga el legítimo derecho a excluirlo, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1.000 salarios mínimos legales mensuales vigentes.

Artículo 269B. *Obstaculización ilegítima de sistema informático o red de telecomunicación.* El que, sin estar facultado para ello, impida u obstaculice el funcionamiento o el acceso normal a un sistema informático, a los datos informáticos allí contenidos, o a una red de telecomunicaciones, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1000 salarios mínimos legales mensuales vigentes, siempre que la conducta no constituya delito sancionado con una pena mayor.

Artículo 269C. *Interceptación de datos informáticos*. El que, sin orden judicial previa intercepte datos informáticos en su origen, destino o en el interior de un sistema informático, o las emisiones electromagnéticas provenientes de un sistema informático que los transporte incurrirá en pena de prisión de treinta y seis (36) a setenta y dos (72) meses.

Artículo 269D. *Daño Informático*. El que, sin estar facultado para ello, destruya, dañe, borre, deteriore, altere o suprima datos informáticos, o un sistema de tratamiento de información o sus partes o componentes lógicos, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1.000 salarios mínimos legales mensuales vigentes.

Artículo 269E. *Uso de software malicioso*. El que, sin estar facultado para ello, produzca, trafique, adquiera, distribuya, venda, envíe, introduzca o extraiga del territorio nacional software malicioso u otros programas de computación de efectos dañinos, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1.000 salarios mínimos legales mensuales vigentes.

Artículo 269F. *Violación de datos personales*. El que, sin estar facultado para ello, con provecho propio o de un tercero, obtenga, compile, sustraiga, ofrezca, venda, intercambie, envíe, compre, intercepte, divulgue, modifique o emplee códigos personales, datos personales contenidos en ficheros, archivos, bases de datos o medios semejantes, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1000 salarios mínimos legales mensuales vigentes.

Artículo 269G. *Suplantación de sitios web para capturar datos personales*. El que con objeto ilícito y sin estar facultado para ello, diseñe, desarrolle, trafique, venda, ejecute, programe o envíe páginas electrónicas, enlaces o ventanas emergentes, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1.000 salarios mínimos legales mensuales vigentes, siempre que la conducta no constituya delito sancionado con pena más grave.

En la misma sanción incurrirá el que modifique el sistema de resolución de nombres de dominio, de tal manera que haga entrar al usuario a una IP diferente en la creencia de que

acceda a su banco o a otro sitio o de personal confianza, siempre que la conducta no constituya delito sancionado con pena más grave.

La pena señalada en los dos incisos anteriores se agravará de una tercera parte a la mitad, si para consumarlo el agente ha reclutado víctimas en la cadena del delito.

Artículo 269H. *Circunstancias de agravación punitiva*: Las penas imponibles de acuerdo con los artículos descritos en este título, se aumentarán de la mitad a las tres cuartas partes si la conducta se cometiere:

Sobre redes o sistemas informáticos o de comunicaciones estatales u oficiales o del sector financiero, nacionales o extranjeros.
Por servidor público en ejercicio de sus funciones.
Aprovechando la confianza depositada por el poseedor de la información o por quien tuviere un vínculo contractual con este.
Revelando o dando a conocer el contenido de la información en perjuicio de otro.
Obteniendo provecho para sí o para un tercero.
Con fines terroristas o generando riesgo para la seguridad o defensa nacional.
Utilizando como instrumento a un tercero de buena fe.
Si quien incurre en estas conductas es el responsable de la administración, manejo o control de dicha información, además se le impondrá hasta por tres años, la pena de inhabilitación para el ejercicio de profesión relacionada con sistemas de información procesada con equipos computacionales.

De los atentados informáticos y otras infracciones.

Artículo 269I. *Hurto por medios informáticos y semejantes*. El que, superando medidas de seguridad informáticas, realice la conducta señalada en el artículo 239manipulando un sistema informático, una red de sistema electrónico, telemático u otro medio semejante, o suplantando a un usuario ante los sistemas de autenticación y de autorización establecidos, incurrirá en las penas señaladas en el artículo 240de este Código.

Artículo 269J. *Transferencia no consentida de activos*. El que, con ánimo de lucro y valiéndose de alguna manipulación informática o artificio semejante, consiga la transferencia no consentida de cualquier activo en perjuicio de un tercero, siempre que la conducta no constituya delito sancionado con pena más grave, incurrirá en pena de prisión de cuarenta y ocho (48) a ciento veinte (120) meses y en multa de 200 a 1.500 salarios mínimos legales mensuales vigentes. La misma sanción se le impondrá a quien fabrique, introduzca, posea o facilite programa de computador destinado a la comisión del delito descrito en el inciso anterior, o de una estafa. Si la conducta descrita en los dos incisos anteriores tuviere una cuantía superior a 200 salarios mínimos legales mensuales, la sanción allí señalada se incrementará en la mitad.

Art.236. recuperación de información dejada al navegar por internet u otros medios tecnológicos que produzcan efectos equivalentes.

Art 275. elementos materiales probatorios y evidencia física.

Mensaje de datos, como el intercambio electrónico de datos, Internet, correo electrónico, telegrama, télex, telefax o similar, regulados por la Ley 527 de1999 o las normas que la sustituyan, adicionen o reformen.

2.4.2 Licencias para el uso del Software Libre

Licencias GPL[13]: Una de las más utilizadas es la Licencia Pública General de GNU (GNU GPL). El autor conserva los derechos de autor (*copyright*), y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL. Esto hace que sea imposible crear un producto con partes no licenciadas GPL: el conjunto tiene que ser GPL.

Es decir, la licencia GNU GPL posibilita la modificación y redistribución del software, pero únicamente bajo esa misma licencia. Y añade que si se reutiliza en un mismo programa código "A" licenciado bajo licencia GNU GPL y código "B" licenciado bajo otro tipo de licencia libre, el código final "C", independientemente de la cantidad y calidad de cada uno de los códigos "A" y "B", debe estar bajo la licencia GNU GPL.

Copyleft[14]. *Copyleft* o copia permitida comprende a un grupo de derechos de autor caracterizados por eliminar las restricciones de distribución o modificación impuestas por el copyright, con la condición de que el trabajo derivado se mantenga con el mismo régimen de derechos de autor que el original. Bajo tales licencias pueden protegerse una gran diversidad de obras, tales como programas informáticos, arte, cultura y ciencia, es decir prácticamente casi cualquier tipo de producción creativa. *Copyleft* dice que cualquiera que redistribuye el software, con o sin cambios, debe dar la libertad de copiarlo y modificarlo más. *Copyleft* garantiza que cada usuario tiene libertad.

BSD. Llamadas así porque se utilizan en gran cantidad de software distribuido junto a los sistemas operativos BSD. El autor, bajo tales licencias, mantiene la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario.

Creative Commons. Las licencias Creative Commons o CC están inspiradas en la licencia GPL de la *Free Software Foundation*. No son, sin embargo, un tipo de licenciamiento de software. La idea principal es posibilitar un modelo legal ayudado por herramientas informáticas, para así facilitar la distribución y el uso de contenidos.

---

[13]Free Software Foundation, Inc. GNU OperatingSystem: Licencias. [en linea] http://www.gnu.org/licenses/licenses.es.html> [citado el 11 de Abril de 2010]

[14]FundaciónCopyleft. Copyleft. [en línea] < http://fundacioncopyleft.org/es/9/que-es-copyleft> [citado el 11de Abril de 2010]

2.4.3 Ley 842 de 2003

TITULO IV
CODIGO DE ETICA PARA EL EJERCICIO DE LA INGENIERIA EN GENERAL Y SUS PROFESIONES AFINES Y AUXILIARES
CAPITULO I

Disposiciones generales

Artículo 29. *Postulados éticos del ejercicio profesional.* El ejercicio profesional de la Ingeniería en todas sus ramas, de sus profesiones afines y sus respectivas profesiones auxiliares, debe ser guiado por criterios, conceptos y elevados fines, que propendan a enaltecerlo; por lo tanto deberá estar ajustado a las disposiciones de las siguientes normas que constituyen su Código de Ética Profesional.

Parágrafo. El Código de Ética Profesional adoptado mediante la presente ley será el marco del comportamiento profesional del ingeniero en general, de sus profesionales afines y de sus profesionales auxiliares y su violación será sancionada mediante el procedimiento establecido en el presente título.

Artículo 30. Los ingenieros, sus profesionales afines y sus profesionales auxiliares, para todos los efectos del Código de Ética Profesional y su Régimen Disciplinario contemplados en esta ley, se denominarán "Los profesionales".

CAPITULO II

Artículo 33.*Deberes especiales de los profesionales para con la sociedad* Son deberes especiales de los profesionales para con la sociedad:

Interesarse por el bien público, con el objeto de contribuir con sus conocimientos, capacidad y experiencia para servir a la humanidad.

Coopera para el progreso de la sociedad, aportando su colaboración intelectual y material en obras culturales, ilustración técnica, ciencia aplicada e investigación científica.

Aplicar el máximo de su esfuerzo en el sentido de lograr una clara expresión hacia la comunidad de los aspectos técnicos y de los asuntos relacionados con sus respectivas, profesiones y su ejercicio;

Artículo 34.*Prohibiciones especiales a los profesionales respecto de la sociedad.* Son prohibiciones especiales a los profesionales respecto de la sociedad:

Ofrecer o aceptar trabajos en contra de las disposiciones legales vigentes, o aceptar tareas que excedan la incumbencia que le otorga sutítulo y su propia preparación.

Imponer su firma, a título gratuito u oneroso, en planos, especificaciones, dictámenes, memorias, informes, solicitudes de licencias urbanísticas, solicitudes de licencias, informes, solicitudes de licencias urbanísticas, solicitudes de licencias de construcción y toda otra documentación relacionada con el ejercicio profesional, que no hayan sido estudiados, controlados o ejecutados personalmente.

Artículo 37. *Deberes de los profesionales para con sus colegas y demás profesionales.* Son deberes de los profesionales para con sus colegas y demás profesionales de la ingeniería:

Respetar y reconocer la propiedad intelectual de los demás profesionales sobre sus diseños y proyectos.

Artículo 38.*Prohibiciones a los profesionales respecto de sus colegas y demás profesionales.* Son prohibiciones a los profesionales, respecto de sus colegas y demás profesionales de la ingeniería:

Utilizar sin autorización de sus legítimos autores y para su aplicación en trabajos profesionales propios, los estudios, cálculos, planos, diseños y software y demás documentación perteneciente a aquellos, salvo que la tarea profesional lo requiera, caso en el cual se deberá dar aviso al autor de tal utilización.

2.4.4 Ley de Derecho de Autor. Hace referencia sobre la protección de la información que con intención y sin derecho reproduzca, con infracción del encabezamiento del artículo 41 de esta Ley, en forma original o elaborada, íntegra o parcialmente, obras del ingenioso quien introduzca en el país, almacene, distribuya, venda o ponga de cualquier otra manera en circulación reproducciones ilícitas de las obras del ingenio o productos protegidos por esta Ley.

2.4.5 La legislación de derechos de autor en Colombia. Mediante decisión 351 de la comisión del acuerdo de Cartagena de diciembre de 1993, que está respaldada por la ley 44 de 1993 y por la ley 23 de 1982. Estas normas otorgan amplia e importante protección a los programas de software, convirtiendo ilícita la copia de programas sin consentimiento de los titulares de los derechos de autor, con excepción de la copia de seguridad.

2.4.6 Norma Técnica Colombiana NTC 4490,1160 y 130837[15]. Estas normas establecen la presentación uniforme de referencias bibliográficas para publicaciones seriadas, libros, folletos y para fuentes de información electrónicas, con el fin de facilitar la identificación de los mismos o de una de sus partes.

2.4.7 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

[15]INSTITUTO COLOMBIANO DE NORMAS TÉCNICASY CERTIFICACIÓN. Tesis y otros trabajos de grado. Bogotá : ICONTEC, 2002

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 hipertónicos 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 responde 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)[16].

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.

[16] Comercio Electrónico. David Hazael Torres Castañeda. [en línea]. Disponible en Internet En: http://www.eumed.net/ce/2012/tcgz.html

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.

3. DISEÑO METODOLÓGICO

3.1 TIPO DE INVESTIGACIÓN

El proceso estará fundamentado en una investigación descriptiva[17], la cual consiste en llegar a conocer situaciones y actitudes predominantes de un objeto de estudio a través de la descripción exacta de las actividades, objetos, procesos a llevar a cabo la investigación. El objetivo no se limita a la recolección de datos, sino a la predicción e identificación de las relaciones que existen entre dos o más variables. Los investigadores no son tabuladores, sino que recogen los datos sobre la base de una hipótesis o teoría, exponen y resumen la información de manera cuidadosa y luego analizan minuciosamente los resultados, a fin de extraer generalizaciones significativas que contribuyan al conocimiento.

Esta investigación se define como descriptiva, ya que por medio de los laboratorios que se realicen, se analizará el comportamiento de las diferentes funcionalidades y potencial que tiene la metodología de análisis fornece post-mortem.

3.2 POBLACIÓN

La población está conformada por administrador del laboratorio del Grupo de Investigación en Teleinformática y Desarrollo de Software (GITYD).

3.3 MUESTRA

Considerando que la población es solo el administrador del laboratorio del Grupo de Investigación en Teleinformática y Desarrollo de Software (GITYD), se toma como muestra toda la población involucrada en el proceso.

3.4 TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE LA INFORMACIÓN

Para la presente investigación es necesario aplicar la técnica de observación estructurada[18] la cual se lleva a cabo cuando se pretende probar una hipótesis o cuando se quiere hacer una descripción sistemática de algún fenómeno. El instrumento mediante el cual se va a obtener la información para aplicar esta técnica es una ficha de observación para los laboratorios que se van a realizar durante el desarrollo del presente estudio. Después de un correcto y completo estudio del análisis forense informático post – mortem se realizará el análisis del ataque. La recolección de datos se hará mediante evaluación directa del ataque informático después de realizado éste. Para este propósito se utilizará la metodología de Casey y la distribución de Linux Backtrack especializada en seguridad informática

[17]GrajalesTevni. TIPOS DE INVESTIGACION [en línea]. Disponible en Internet En: http://tgrajales.net/investipos.pdf

[18]Puente Wilson. TÉCNICAS DE INVESTIGACIÓN [en línea]. Disponible En: Internet En: http://www.rrppnet.com.ar/tecnicasdeinvestigacion.htm

3.4.1 Selección de la metodología. Dentro de la selección de la metodología que se aplicara en el estudio aquí realizado se tuvieron en cuenta 3 aspectos, el primero fue la revisión bibliográfica de metodologías existentes, el segundo fue la comparación ente las distintas metodologías y el tercero lo constituye la comparación práctica. Para tal efecto se realiza un matriz de comparación la cual permite establecer cuál de la metodología se adapta de mejor manera al estudio, permitiendo de este modo evacuar los dos primeros aspectos de evaluación.

A continuación se describen los ítems de evaluación aplicados en la matriz, cada uno de ellos posee un peso que oscila entre 1 y 5

Adaptabilidad a estándares internacionales. Fundamentalmente se compara con el modelo propuesto por el EDRF y RFC 3227.
Documentación disponible.
Adaptabilidad y practicidad en la implementación experimental.
Compatibilidad con la metodología post-mortem

Matriz de comparación de metodologías

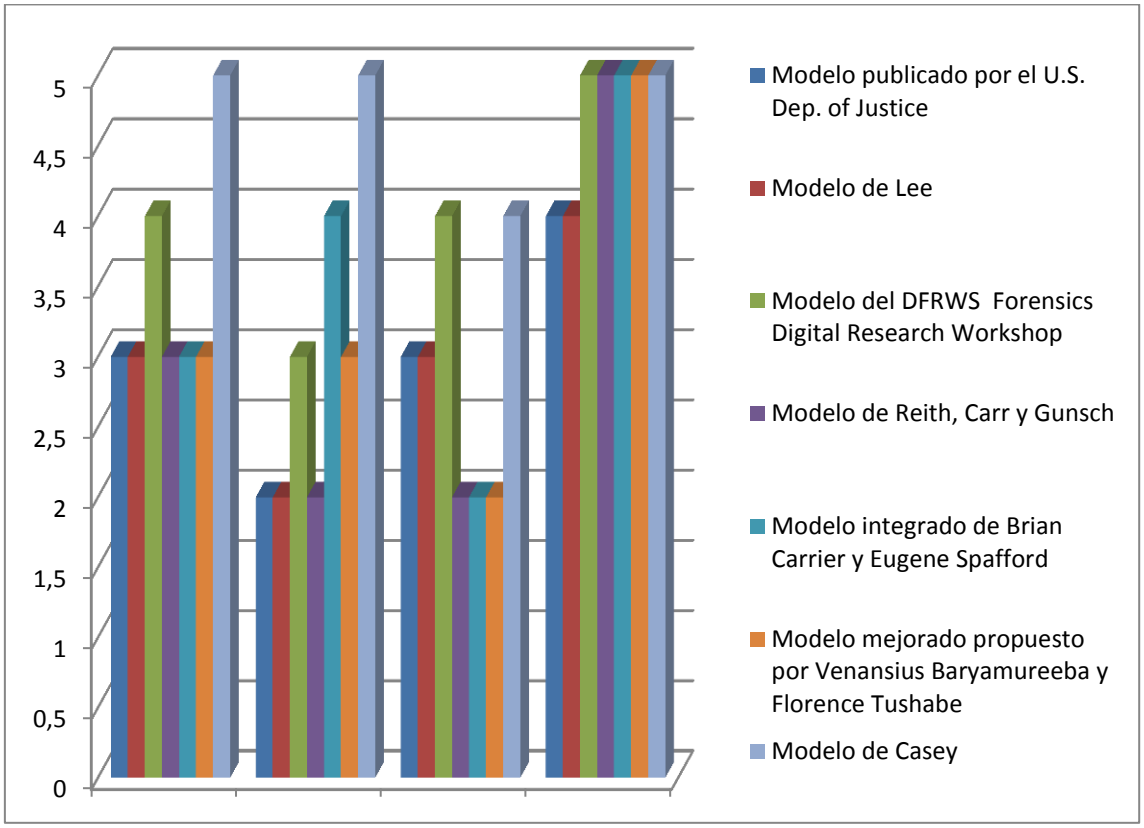
Id	Metodología	Fases	Ítems de evaluación				
			A	B	C	D	T
01	**Modelo publicado por el U.S. Dep. of Justice**	1. Identificación 2. Conservación 3. Análisis 4. Presentación	3	2	3	4	12
02	**Modelo de Lee**	1.Reconocimiento 2. Identificación 3. Individualización 4. Reconstrucción	3	2	3	4	12
03	**Modelo del DFRWS Forensics Digital Research Workshop**	1. La identificación 2. La preservación 3. La colección 4. El examen 5. El análisis 6. La presentación 7. La decisión	4	3	4	5	16
04	**Modelo de Reith, Carr y Gunsch**	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	3	2	2	5	12

05	**Modelo integrado de Brian Carrier y Eugene Spafford**	**Fases de Preparación:** 1. preparación de operaciones. 2. preparación de infraestructuras. **Fases de Despliegue:** 1. Detección y Notificación. 2. Confirmación y Autorización. **Fases de Investigación Física de la escena del crimen:** 1. conservación. 2. Inspección. 3. Documentación. 4. búsqueda y recolección. 5. reconstrucción 6. Presentación. **Fases de Investigación de la Escena Digital del Delito:** 1. conservación. 2. Inspección. 3. Documentación. 4. búsqueda y recolección. 5. reconstrucción 6. Presentación. **Fase de revisión**	3	4	2	5	14
06	**Modelo mejorado propuesto por Venansius Baryamureeba y Florence Tushabe**	**Fases de despliegue:** 1.Detección y Notificación 2. Investigación Física de la escena del delito. 3. Investigación Digital de la escena del delito. 4. Confirmación. 5. Informe **Fases de Hipótesis:** 1. Investigación digital de la escena del delito 2. Autorización **Fases Dinamita:** 1. Investigación Física de la escena del delito. 2. Investigación Digital de la escena del delito. 3. Reconstrucción. 4. Comunicación. 5. Revisión	3	3	2	5	13

07	**Modelo de Casey**	1. Autorización y preparación 2. Identificación 3. Documentación, Adquisición y Conservación 4. Extracción de Información y Análisis 5. Reconstrucción 6. Publicaciones	5	5	4	5	19

Fuente: Laboratorios del investigador.

Figura 2: Representación gráfica de la comparación de metodologías



Fuente: Laboratorios del investigador.

Como puede observarse en el gráfico y en la matriz de la cual se abstrae el mismo, la metodología CASEY es la que obtiene los resultados más altos y tras realizar la concerniente evaluación práctica se demuestra que si es la más óptima para la realización de este estudio.

3.4.2 Modelo de Casey (2004). 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[19] referencia que recoge los siguientes pasos:

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

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.

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

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.

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.

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.

Examen y Análisis. Con todos los datos obtenidos en las etapas anteriores podemos tener una idea de dónde 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.

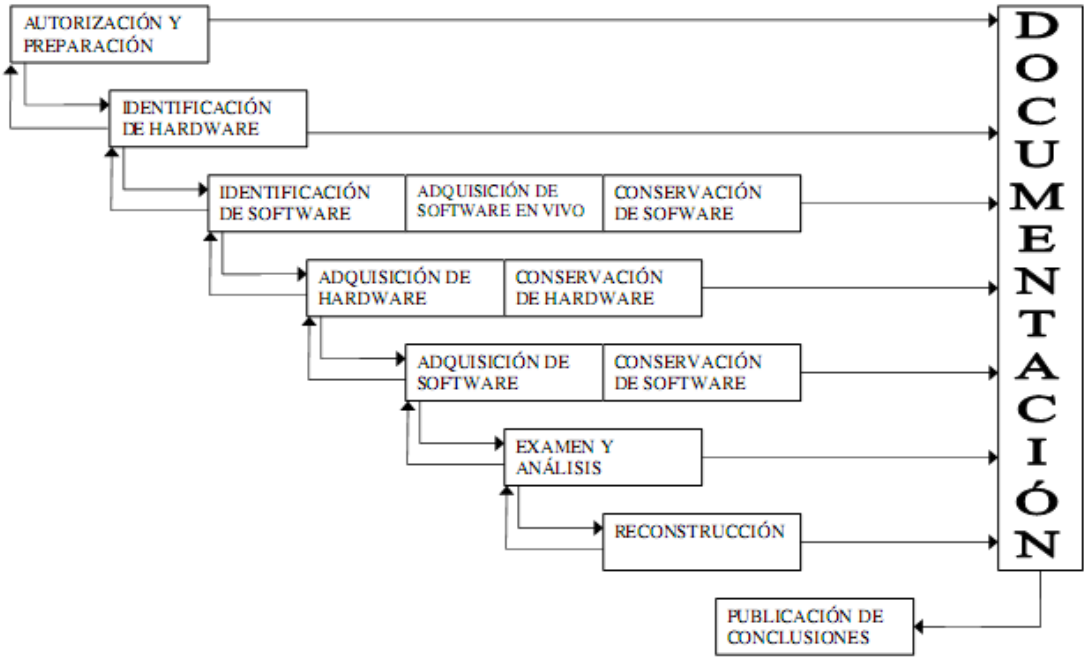
Reconstrucción. Una vez que tenemos datos suficientes debemos ser capaces de responder a las preguntas ¿Qué pasó? ¿Quién lo hizo? ¿Cuándo? ¿Dónde? y en última instancia ¿por qué?

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.

[19]Easey, Eoghan. Digital Evidence and Computer Crime, Third Edition: Forensic Science, Computers, and the Internet. Academic Press. p. 840. ISBN 978-0123742681.

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.

Figura 3. Esquema general modelo Casey



Fuente: Herramienta de apoyo para el análisis forense de computadoras, José Arquillo Cruz, Escuela Politécnica Superior de Jaén, Septiembre, 2007

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 a cabo 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 sobrescribir 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 cómo 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 está 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.

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 iníciales 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.

Figura 4. Formato de cadena de custodia

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					

Fuente: Herramienta de apoyo para el análisis forense de computadoras, José Arquillo Cruz, Escuela Politécnica Superior de Jaén, Septiembre, 2007

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.

Identificación de Hardware. Hay muchos productos computarizados que pueden tener evidencias recogidos en, como teléfonos, 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.

Identificación del software. Generalmente se considera que todo el contenido del hardware identificado contiene potencialmente 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:

Figura 5. Lista de volatilidad

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

Fuente: Herramienta de apoyo para el análisis forense de computadoras, José Arquillo Cruz, Escuela Politécnica Superior de Jaén, Septiembre, 2007

Un ejemplo de orden de volatilidad podría ser:

Registros y cache
Tablas de enrutamiento
Cache Arp
Tabla de procesos en ejecución
Estadísticas y módulos Kernel

Memoria principal
Ficheros temporales del sistema
Memoria secundaria
Configuración del Router
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 está 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 está 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 SISTEMAS OPERATIVOS 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.

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: ¿qué

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.

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. ¿Qué hacer si una computadora está 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.

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 cómo 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 auto-arrancable, 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 dé 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.

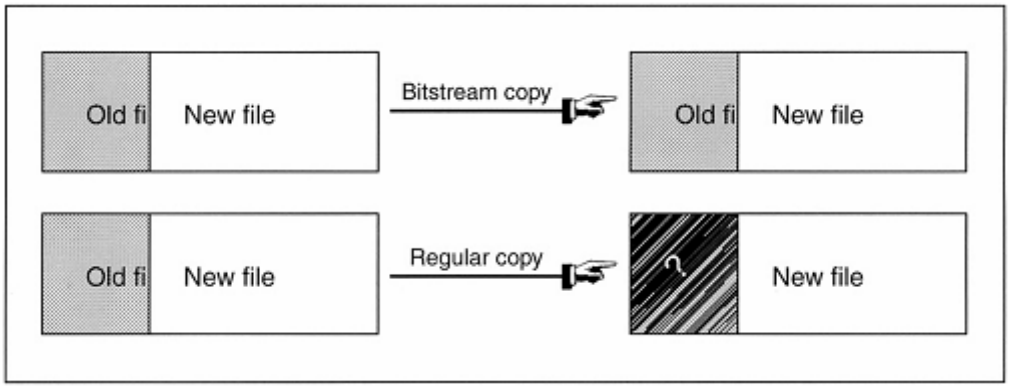
Hay además un riesgo de que el sistema haya sido modificado con el fin de ocultar o destruir las evidencias (por ejemplo, usando un rootkit). Por ejemplo, si un investigador busca ficheros log en una computadora, puede haber ficheros logs borrados en el espacio libre que pueden serle útiles. Cuando se 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 6. Comparación entre una copia normal y una bit a bit.



Fuente: Easey, Eoghan. Digital Evidence and Computer Crime, Third Edition: Forensic Science, Computers, and the Internet. Academic Press. p. 840. ISBN 978-0123742681.

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.

Examen y análisis

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 SISTEMAS OPERATIVOS 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.

Búsqueda y recopilación de información. En esta etapa es fundamental tener claro cuáles serían las categorías que llegarían a ser causales de delito o simple suspicacia para de este modo facilitar la búsqueda.

Reconstrucción. Una reconstrucción investigativa nos ayuda a obtener una imagen más completa del delito: que ha pasado, quien causó los eventos, cuando, donde, cómo 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, qué y donde): los componentes de un delito, su posición e interacción.
Análisis Funcional (como): qué fue posible e imposible

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.

Análisis relacional. En un esfuerzo para identificar relaciones entre sospechosos, la víctima 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 está conectado. Los investigadores deben decidir cuánto 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.

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 cómo 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 cómo 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.

Publicación de conclusiones. La última fase de un análisis de evidencias digitales es integrar todo el conocimiento y conclusiones en un informe final que dé 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 cómo y dónde 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 por qué otras explicaciones son improbables o imposibles demuestra que el método científico fue 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, severo). 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 des-encriptaron, 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 los 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 cuándo nos refereimos a una evidencia digital en un contexto dado y cualificar las conclusiones apropiadamente.

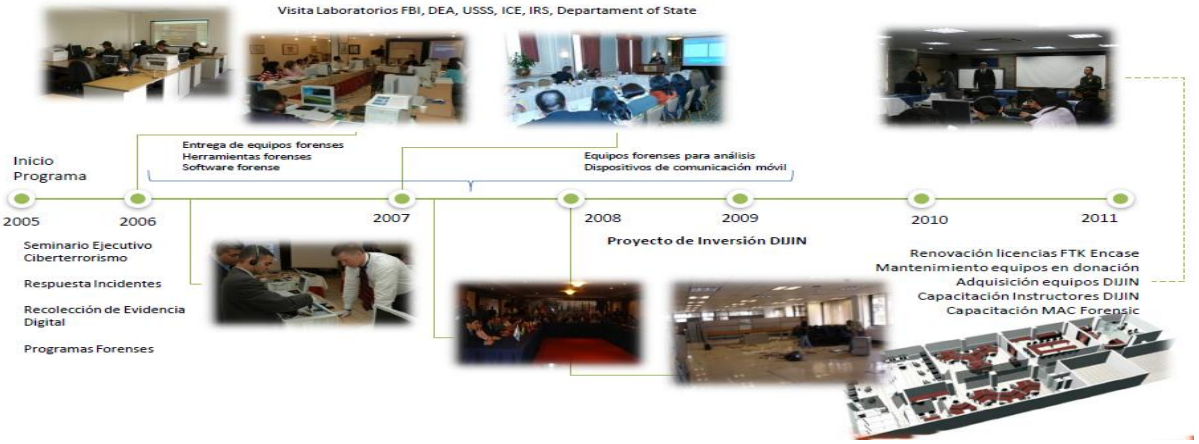
3.5 PROCESAMIENTO Y ANÁLISIS DE LA INFORMACION

El respectivo análisis de la información se realizara cuando la fecha de observación estructurada contenga los resultados de los laboratorios, los cuales se generaran durante la ejecución de la investigación como se muestra en el cronograma de actividades.

4. DIAGNOSTICO SITUACIONAL

La informática forense es una disciplina relativamente nueva y poco aplicada en Colombia. Por ello, creemos que no existe una metodología que sea referencia nacional, la cual permita garantizar que el proceso forense cumpla con su cometido de aplicar técnicas científicas y analíticas e infraestructura tecnológica para identificar, preservar, analizar y presentar evidencia, de manera que sea admisible en un proceso legal. Se Considera que en muchos casos, la manipulación de la evidencia se hace de una forma equivocada, si no se respetan los protocolos internacionales del manejo de evidencia digital, sin embargo, lo que se toma como mundialmente válido son las mejores prácticas, como las del Servicio Secreto de EUA y recomendaciones de organismos especializados como el NIST[20]; así como también el RFC 3227[21] y esfuerzos internacionales como CP4DF[22] o el proyecto CTOSE[23]; por lo que es necesario sentar las bases de un procedimiento que permita recuperar y preservar la información de un dispositivo de almacenamiento, como un disco duro. Por otra parte en Colombia se pueden ver los esfuerzos por avanzar en el tema ya que con el nacimiento de GITEC-DIJIN[24] en el año de 2008, y su progresivo crecimiento tanto técnico como en infraestructura, en el marco de su estrategia de implementación tecnológica, se crea un punto departida en la estandarización de la investigación forense en el país.

Figura 7. Desarrollo GITEC-DIJIN



Fuente: seminario internacional seguridad de la información: nuevos retos "recolección de evidencias". Policía Nacional. Subintendente, Yair Vanegas Rodríguez. Octubre 2011

[20] National Institute of Standards and Technology. [on line] Disponible en internet en: http://www.nist.gov/public_affairs/general_information.cfm

[21] Request For Comments. [on line] Disponible en internet en: http://www.ietf.org/rfc/rfc3227.txt

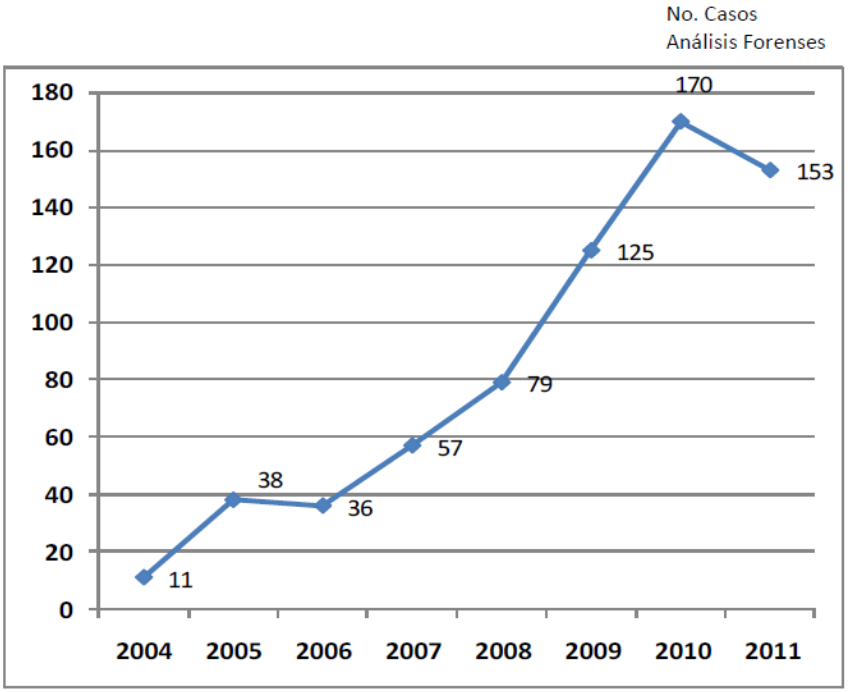
[22] Code of practices for *Digital Forensics.* [on line] Disponible en internet en: http://cp4df.sourceforge.net/porque.html

[23] Cyber Tools On-Line Search for Evidence[on line] Disponible en internet en: http://www.shellsec.net/articulo/CTOSE-un-proyecto-para-garantizar-la-seguridad-de-las-operaciones-electronicas/

[24]

Actualmente el grupo de investigación GITEC-DIJIN ha realizado enormes avances desde lo institucional con miras al fortalecimiento y evolución de la informática forense en Colombia; el trabajo de la DIJIN desde sus 45 seccionales y 145 expertos en ciber-terrorismo en aspectos como la extracción de archivos borrados, fragmentos de archivos, Desciframiento de contraseñas de archivos, Reconstrucción de actividades WEB – historial, Archivos ocultos – códigos malicioso y la cobertura en varias de las más importantes ciudades del país (Cali, Medellín, Bucaramanga, Barranquilla); han dado como resultado el crecimiento exponencial de los casos forneces entre 2004 y 2011(ver figura 8)

Figura 8. Casos forense entre 2004 y 2011



AÑO	GB
2004	804
2005	1795
2006	2767
2007	4900
2008	11.039
2009	20.480
2010	32.844
2011	40.670

No. Gigabytes
Analizados

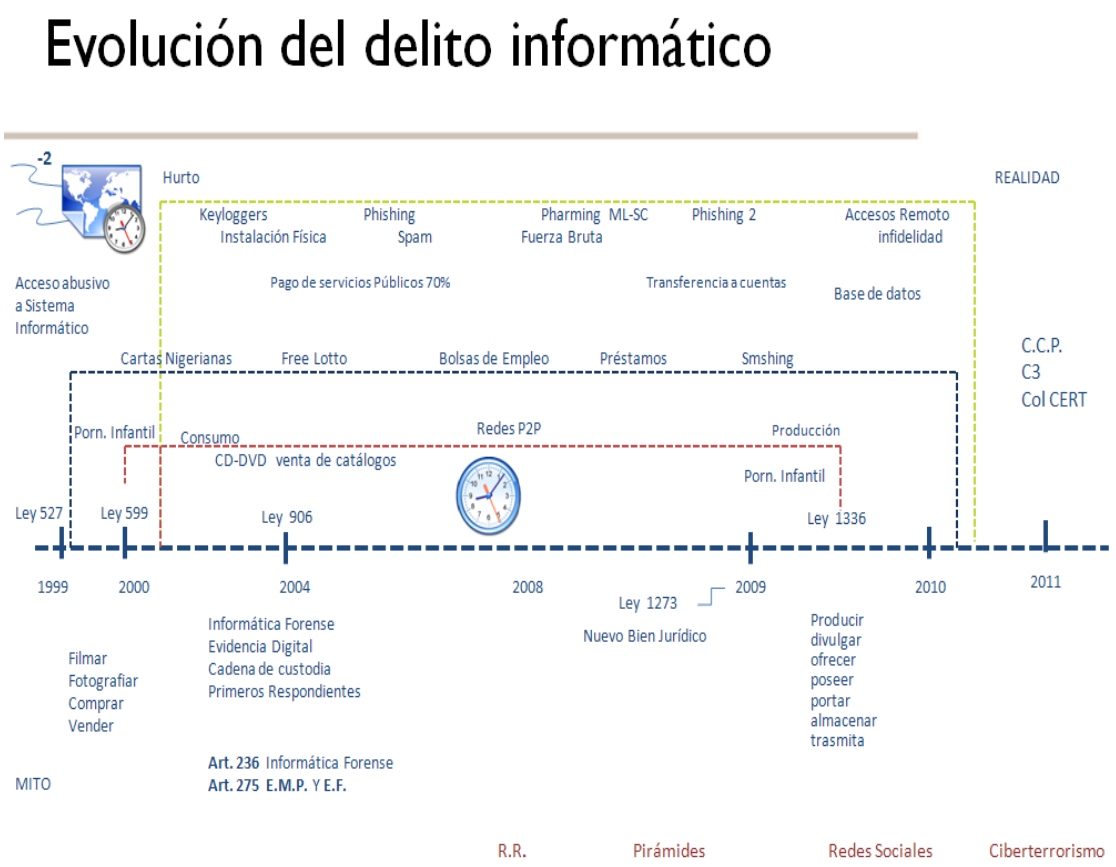
Fuente: seminario internacional seguridad de la información: nuevos retos "recolección de evidencias". Policía Nacional. Subintendente, Yair Vanegas Rodríguez. Octubre 2011

El CONPES No. 3701 de 2011[25] que busca Contrarrestar el incremento de las amenazas informáticas que afectan la infraestructura critica del país contempla la creación de Grupo de respuesta a incidentes cibernéticos de Colombia colCERT, como respuesta al creciente número de incidentes en el país, así como la diversificación de los delitos informáticos. Dicho ente junto con el COMANDO CONJUNTO CIBERNETICO que es el Equipo encargado de la defensa del país en el ciberespacio y el CENTRO CIBERNÉTICO POLICIAL que es el equipo encargado de la seguridad ciudadana en el ciberespacio; colaboran activamente en la resolución de incidentes en lo que se refiere a Asistencia técnica, Coordinación en la gestión de incidentes, Asistencia ante emergencias, Desarrollo

[25] Compes No. 3701 de 2011. [en línea]. Disponible En: Internet En: http://www.mintic.gov.co/index.php/docs-normatividad?pid=698&sid=741:3701

de capacidades operativas, Proveer información estratégica de inteligencia, Asesoramiento y apoyo en ciber-defensa así como en la Coordinación de respuesta ante incidentes.

Figura 9. Evolución de los delitos informáticos



Fuente: Lineamientos de Política para Ciber-seguridad y Ciber-defensa, Centro Cibernético Policial, Policía Nacional de Colombia, marzo 2012

Algunos autores (como Brian Carrier y Eugene Spafford) prefieren omitir el término forense, ya que entonces el análisis variará en función de las leyes del país o países en los que se pretende presentar las evidencias como prueba judicial, y sin embargo, existen pautas comunes a la hora de realizar el análisis, al margen de las leyes que imperen en el país determinado. También es posible realizar un análisis de sistemas de información en un entorno corporativo, donde el término forense pierde su sentido.

Para Carrier el análisis forense se reduce a una investigación digital y esta es un proceso en el que se desarrollan y ponen a prueba las hipótesis que respondan a preguntas sobre los eventos digitales. Esto se hace utilizando el método científico en el que se desarrolla una hipótesis con evidencia que nos encontramos y luego probar la hipótesis mediante la

búsqueda de más evidencia que muestra que la hipótesis es imposible. La evidencia digital es un objeto digital que contiene información confiable que apoya o refuta una hipótesis[26].

4.1 SELECCIÓN DE LAS HERRAMIENTAS Y SUIT FORENSE

El análisis forense requiere de una gran cantidad de elementos para poder su ejecución, además de la experticia del investigador, es fundamental contar con las herramientas lógicas que proporcionen un apoyo confiable, pues es en este aspecto que la investigación descarga la mayor parte de los elementos sensibles.

El proceso de selección de la suit forense empleada en esta investigación se fundamentó en aspectos tanto prácticos como funcionales y teóricos. Teniendo en cuenta que la investigación se realiza en el contexto de la universidad Francisco de Paula Santander Ocaña, y más específicamente del semillero SIGLAS (Gnu/ Linux And Security), era prioritario que la herramienta/s seleccionada fuese basada en software libre; partiendo de este hecho, a continuación se desglosan los ítems evaluados en la selección:

Orientación al análisis forense con la metodología post mortem: en este numeral se pretende establecer si la herramienta soporta el uso de imágenes bit a bit para realizar el estudio, y así mismo si sus opciones de indagación no requieren del sistema en ejecución. Pues es posible realizar un análisis post mortem sin requerir extraer una imagen bit a bit.
Software libre: dentro de los aspectos planteados al inicio de la investigación el software constituye uno de los más importantes, y es por tal que se aplica un mayor puntaje a aquellas herramientas que poseen esta característica.

Soporte: si la herramienta escogida para el estudio posee un soporte actualizado, constituye prenda de garantía en aras de validar las evidencias o hallazgos dentro del estudio.

Impacto sobre el sistema: una buena herramienta de análisis digital debe hacer un mínimo o ningún impacto sobre el sistema pues esto corrompe la evidencia.

Plataformas soportadas: Es muy común encontrarnos con análisis de ataques o incidentes en sistemas opetarivos Linux, Windows, ios entre otros, por lo cual la herramienta seleccionada debe cubrir la mayor cantidad de sistemas operativos.

Sistema operativo Linux: Es una de las premisa que la herramienta escogida funcione sobre el sistema operativo Backtrack.

Accesibilidad/disponibilidad: el poder adquirir la herramienta con facilidad (descargar, costos, etc) es un aspecto prioritario.

[26] Carrier, Brian. File System Forensic Analysis. Addison-Wesley, 2005

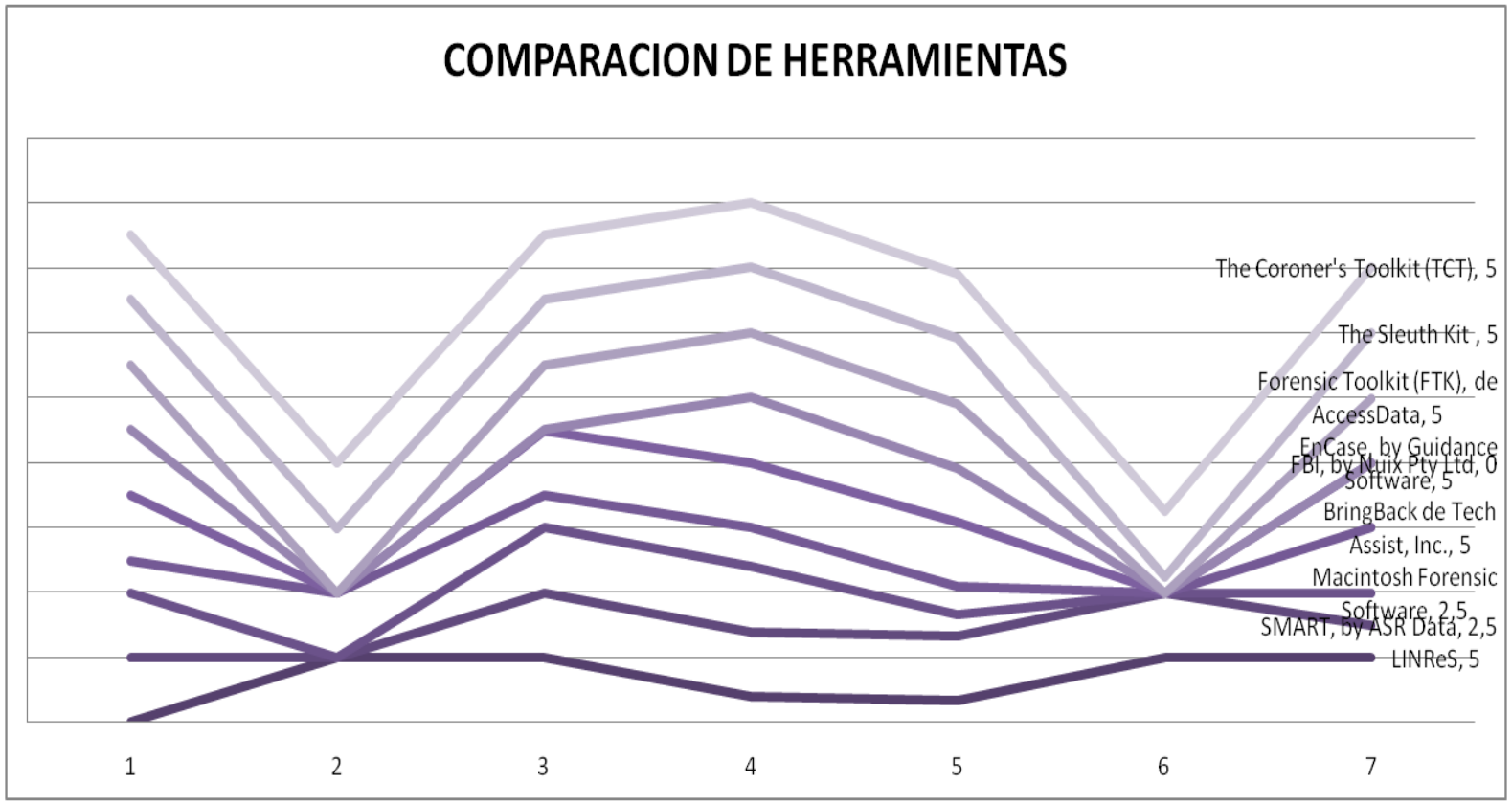
COMPARACION DE SUIT FORENSE

Figura 10: Comparación de herramientas forense

					PLATAFORMAS SOPORTADAS						
HERRAMIENTA	ORIENTACION AL POST-MORTEM	SOFTWARE LIBRE	SOPORTE	IMPACTO SOBRE EL SISTEMA	LINUX	WINDOWS	MAC	PUNTOS	S.O. (LINUX)	ACCESIBILIDAD/DISPONIBILIDAD	PUNTAJE
LINReS	0	5	5	2	0	1.66	0	1.66	5	5	22
SMART, by ASR Data	5	0	5	5	1.66	1.66	1.66	5	5	2.5	22.5
Macintosh Forensic Software	5	0	5	5	0	0	1.66	1.6	0	2.5	17.5
BringBack de Tech Assist, Inc.	2.5	5	2.5	3	0.9	1.25	0	2.25	0	5	18
EnCase, by Guidance Software	5	0	5	5	1.66	1.66	1.66	5	0	5	20
FBI, by Nuix Pty Ltd	5	0	0	5	1.66	1.66	0.8	4	0	0	10
Forensic Toolkit (FTK), de AccessData	5	0	5	5	1.66	1.66	1.66	5	0	5	20
ILook Investigator	5	0	5	5	1.66	1.66	1.66	5	0	0	15
Safeback de NTI & Armor Forensics	5	0	2.5	5	0.8	1.6	0	2.4	0	2	14.5
X-Ways Forensics, de X-Ways AG	5	0	5	5	1.66	1.4	0	2.2	0	2.5	17.5
Prodiscover, de Techpathway	5	0	5	5	0	1.66	0	1.6	0	2.5	17.5
AFFLIB	5	5	5	5	1.66	1.66	1.66	5	5	5	30
Autopsy	5	5	5	5	1.25	1.25	1.25	1.25	5	5	30
FOREMOST	5	5	5	5	1.66	1.66	1.66	5	5	5	30
The Sleuth Kit	5	5	5	5	1.25	1.25	1.25	1.25	5	5	30
The Coroner's Toolkit (TCT)	5	5	5	5	1.66	1.66	1.66	5	5	5	30
Zeitline	5	5	5	5	1.66	1.66	1.66	5	5	5	30
Valoraciones altas											
Herramientas de apoyo											
Valoraciones bajas											

Fuente: Laboratorios del investigador y análisis bibliográfico.

Figura 11: Representación gráfica de la comparación de herramientas forense



Fuente: Laboratorios del investigador y análisis bibliográfico.

4.1.1 Autopsy Y The Sleuth Kit. El navegador Forense Autopsy es una interfaz gráfica para las herramientas de análisis de investigación digital en línea de comando contenidas en Sleuth Kit. Ambos unidos pueden analizar discos UNIX y Windows, además de sistemas de archivos (NTFS, FAT, UFS1/2 y Ext2/3). Autopsy y Sleuth Kit son Open Source (Código Abierto) y pueden ser ejecutados en plataformas UNIX. Como Autopsy se basa en HTML, se puede conectar al servidor Autopsy desde cualquier plataforma utilizando un navegador HTML. Autopsy proporciona una interfaz tipo "Manejador de Archivos", y muestra detalles sobre datos borrados y estructuras del sistema de archivos.[27]

Modos de Análisis[28]

Análisis "En reposo": ocurre cuando un sistema dedicado para análisis es utilizado para examinar los datos de un sistema sospechoso. En este caso, Autopsy y Sleuth Kit son ejecutados en un entorno confiable, típicamente en un laboratorio. Autopsy y TSK soportan formatos de archivos AFF (Advanced Forensic Format), Expert Witness, y raw (en bruto).

Técnicas de búsqueda de evidencia

Listado de Archivos: Analiza los archivos y directorios, incluyendo los nombres de archivos eliminados y nombres de archivos basados en Unicode

Contenido de Archivos: Los contenidos de archivos pueden ser visualizados en bruto (raw), hexadecimal o en cadenas ASCII extraídas. Cuando los datos son interpretados, Autopsy los esteriliza para prevenir daño al sistema de análisis local. Autopsy no utiliza ningún lenguaje script en el lado del cliente.

Base de Datos de HASH: Para identificar rápidamente archivos desconocidos como confiables o dañidos se realizan operaciones de búsqueda en una base de datos de hashs. Autopsy utiliza NIST (National Software Reference Library "NSRL") y bases de datos de archivos conocidos como confiables o dañinos creadas por los usuarios.

Ordenando por tipo de archivo: Para identificar archivos de un tipo conocido, se ordenan los archivos basándose en sus firmas internas. Autopsy puede extraer también solamente imágenes gráficas (incluyendo miniaturas). La extensión de un archivo puede ser comparada también con un tipo de archivo para identificar archivos que pueden tener su extensión modificada para ocultarlos

Línea de tiempo de la actividad de archivos: En algunos casos, el tener una línea de tiempo de la actividad de los archivos puede ayudar a identificar áreas de un sistema de archivo que podría contener evidencia. Autopsy puede crear líneas de tiempo que contienen

[27] Título: Autopsy en español, Alonso Eduardo Caballero Quezada, Noviembre 7 del año 2009, [en línea]. Disponible En: http://www.reydes.com/archivos/autopsy_reydes.pdf
[28] Soporte en línea de TSK [en línea]. Disponible En: http://wiki.sleuthkit.org

entradas de los tiempos de Modificación, Acceso, y Cambio (MAC) de archivos asignados y sin asignar.

Búsqueda de palabras clave: Las búsquedas de palabras clave en una imagen de un sistema de archivos puede ser realizada utilizando cadenas ASCII y expresiones regulares. Las búsquedas pueden ser realizadas en la imagen completa del sistema de archivos o solamente en el espacio sín asignar. Se puede crear un archivo índice para una búsqueda más rápida. Las cadenas buscadas frecuentemente pueden ser fácilmente configuradas dentro de Autopsy de búsquedas automatizadas.

Análisis de Meta Datos: Las estructuras de Meta Datos contienen detalles sobre archivos y directorios. Autopsy permite visualizar detalles de cualquier estructura de Meta Datos en el sistema de archivos. Esto es útil para la recuperación de contenido eliminado. Autopsy buscará los directorios para identificar la ruta completa de un archivo que tiene asignada la estructura.

Análisis de Unidades de Datos: Las unidades de datos son el lugar donde se almacena el contenido del archivo. Autopsy permite visualizar el contenido de cualquier unidad de datos en una variedad de formatos incluyendo ASCII, volcado hexadecimal, y cadenas. Se proporciona también el tipo de archivo y Autopsy buscará las estructuras de Meta Datos para identificar cuales unidades de datos tiene asignada.

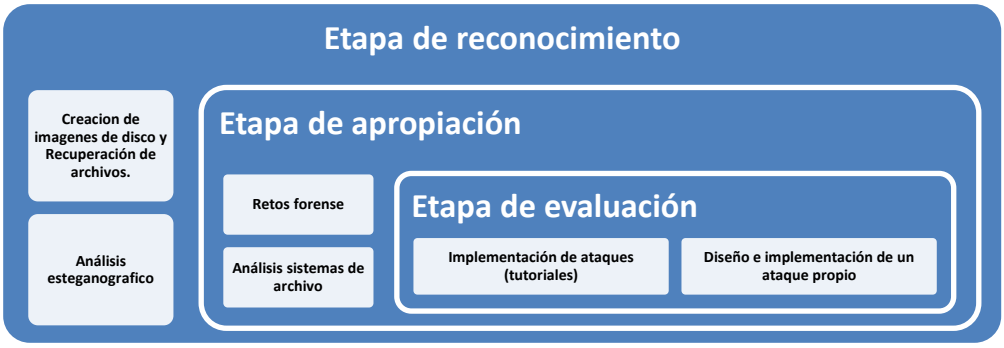
Detalles de la imagen: Los detalles del sistema de archivos puede ser visualizados, incluyendo la disposición sobre el disco y tiempos de actividad. Este modo proporciona información que es de utilidad durante la recuperación de datos.

4.2 DISEÑO DE LABORATORIOS

Las pruebas realizadas en la presente investigación, se basan en escenarios experimentales con ambientes controlados, dentro de, los cuales se recrearon incidentes de seguridad que comprometían estaciones de trabajo a diferentes niveles; así mismo se recurrió a retos forenses para la adquirían de imágenes.

4.2.1 Estructura de laboratorios. Los laboratorios realizados a lo largo de esta investigación comprenden distintos niveles de experticia dependiendo de la fase en la cual se encontrase el proyecto, así las cosas podemos definir tres etapas dentro de la elaboración de los laboratorios:

Figura 12: Etapas del diseño e implementación de laboratorios



Fuente: Laboratorios del investigador.

4.2.2 Etapa de reconocimiento: Es la primera fase de los laboratorios y tiene lugar al inicio de la investigación. El objeto es este punto es conocer la herramienta y los procedimientos que optimizan el uso de la misma, es por tal que se parte de los aspecto fundamentales como la creación de copias Bit a bit de dispositivos de almacenamiento, recuperación de archivos y análisis esteganofraficos.

Laboratorios:

Análisis del sistema de archivos FAT (ANEXO 1)
Recuperación de archivos borrados memoria USB (ANEXO 2)
Análisis esteganografico de archivos de imagen (.jpg, .bmp) (ANEXO 3)

4.2.3 Etapa de apropiación: una vez conocida la herramienta era fundamental conocer los diversos sistemas de archivo mediante el análisis de los mismos atreves de la herramienta seleccionada, por otro lado los retos forenses[29] proporcionaron la experticia en el tema puesto que a pesar de ser actividades diseñadas, cumplen un objetivo didáctico sumamente importante.

Laboratorios:

Reto forense flisol 2010(ANEXO 4)
Reto forense 2 del hacker.net (anexo 5)
Reto forense 4 del proyecto honeynet (anexo 6)

4.2.4 Etapa de evaluación: Para este punto el dominio de la herramienta y lo conocimientos teóricos permite a la investigador desarrollar análisis a incidentes muchos más profundos y elaborados, siguiendo casos de taques documentados desde la implementación miasma y diseñando ataques propios, a los cuales se les realiza el análisis respectivo y de este modo se establece la efectividad del análisis forense.

[29] Honeynet Project Challenges, en línea]. Disponible En: http://www.honeynet.org/challenges

Escalada de privilegios con meta EXPLOIT FRAMEWORK (anexo 9)
Ataque diseñado por el investigador (ANEXO 10)

4.3 METODOLOGÍA PROPUESTA SIGLAS

Con la llegada de la sistematización de procesos, las diversas corporaciones y empresas pasaron a tener toda su información digital, con lo cual se agilizaron e hicieron mucho más eficientes los procesos; no obstante, esto trajo un nuevo problema, y este fue la seguridad puesto que con la instauración de la información como principal activo de las empresas, también se hizo el más apetecido por los atacantes.

Si hay algo claro en el mundo de la información es que no existe un sistema que sea totalmente seguro, y evádeteme mente se presentan incidentes de seguridad, ante los cuales las preguntas son muchas ¿Cuál fue el origen del ataque? ¿Qué implicaciones tuvo el incidente? ¿Qué medidas son necesarias para evitar situaciones similares en el futuro? Es entonces cuando el análisis digital forense hace su aparición para despejar todas las dudas, puesto que durante, y después del incidente fuere cual fuere su origen, el atacante deja rastros, los cuales son habidos de ser estudiados tanto en entornos de red como en el espacio local de la terminal atacada, así mismo la memoria RAM y demás dispositivos que puedan albergar información como lo son smatphone , impresoras, memorias USB, discos flexibles, DVD entre otros, son medios habidos de ser analizados.

No obstante dentro de la gran cantidad de enfoques y conceptos que encierra el análisis forense, es claro que la implementación de una metodóloga practica que indique de modo directo la forma en la cual realizar la investigación, constituye poco menos que una quimera pues la complejidad misma del estudio impediría tal iniciativa, pero a pesar de ello se pueden ahondar esfuerzos para proporcionar a una guía que indique los pasos tanto metodológicos como prácticos en un sentido amplio que permita al investigador partir de un punto claro; en tal sentido se pretende a continuación fundamentándose en la metodología Casey 2000 y el modelo del Electronic Discovery Reference Model (EDRM), realizar la construcción de dicha guía.

Figura 13: Construcción de la metodología propuesta

<table>
<tr><th>METODOLOGÍA CASEY</th><th>MODELO EDRM</th><th colspan="5">CONSTRUCCIÓN DE LA METODOLOGÍA PROPUESTA (Análisis teórico)</th></tr>
<tr><td>Autorización y preparación</td><td>Administración de la información</td><td>Planteamiento del escenario</td><td>Perspectiva del análisis y tipos de análisis</td><td></td><td></td><td></td></tr>
<tr><td>Identificación</td><td>Identificación</td><td colspan="5">Identificar el hardware y software</td></tr>
<tr><td rowspan="2">Documentación, adquisición y conservación</td><td rowspan="2">Preservación y colección</td><td rowspan="2">Adquisición de datos del disco duro</td><td>Adquisición física</td><td>Apagado del sistema</td><td rowspan="2">Escribir los datos de salida y destino ubicación</td><td rowspan="6">Aplicación de la Suit Autopsy + The Sleuth Kit, herramienta DD, editor hexadecimal, Foremost, Sacalpel, Gpart…….</td></tr>
<tr><td>Bloqueadores de escritura de hardware</td></tr>
<tr><td rowspan="2">Extracción de información y análisis</td><td rowspan="2">Proceso revisión y análisis</td><td rowspan="2">Sistema de ficheros</td><td>Adquisición lógica</td><td>Algoritmos criptográficos formato de archivo</td></tr>
<tr><td colspan="3">Categoría meta-datos
Categoría contenido
Categoría meta-datos
Categoría nombre de fichero
Categoría aplicación</td></tr>
<tr><td>Reconstrucción</td><td>Producción</td><td>Análisis de líneas de tiempo y modificación de ficheros.</td><td>Formulación de posibles eventos</td><td>Evaluación de hipótesis</td><td>Escenario reconstruido</td></tr>
<tr><td>Publicación de conclusiones</td><td>Publicación</td><td>Informes forenses</td><td>Artículos</td><td></td><td></td><td></td></tr>
</table>

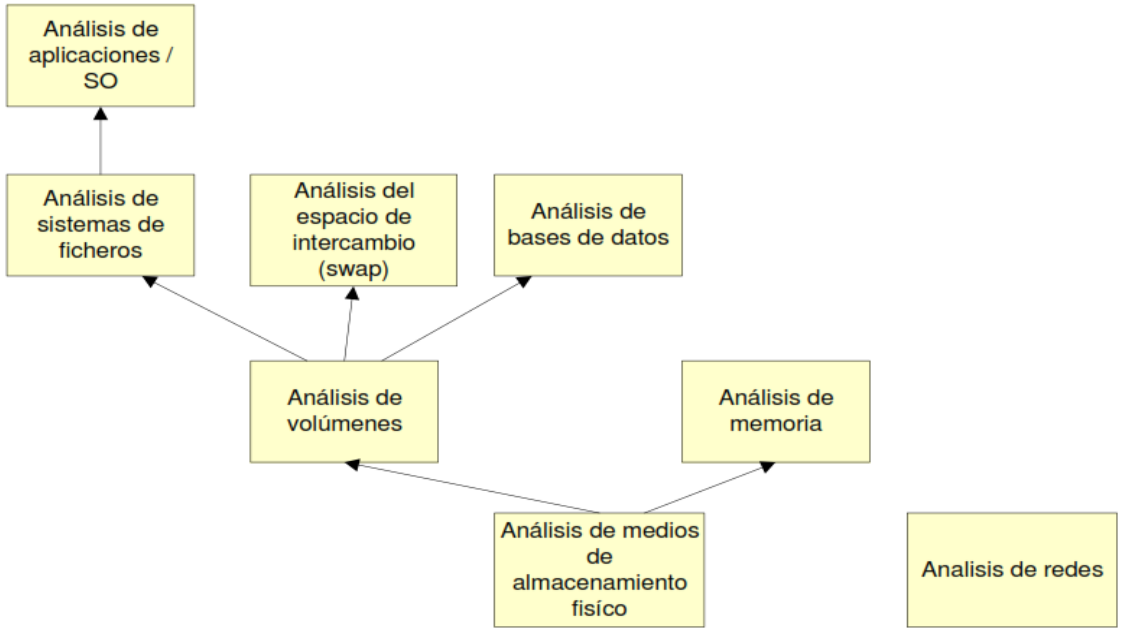
Fuente: Laboratorios del investigador y análisis bibliográfico.

4.3.1 Planteamiento de análisis. En esta etapa el investigador se encuentra con el escenario tanto físico como contextual de incidente, se realizan averiguaciones de ser necesario y se inicia con el diseño del análisis dependiendo de las necesidades de la empresa y de la situación misma pues podría necesitarse diferentes tipos de análisis para diferentes tipos de incidente, bien sea una análisis de redes, en caliente, a sistemas de información entre otros.

4.3.2 Perspectiva del análisis. Algunos autores (como Brian Carrier y Eugene Spafford) prefieren omitir el término forense, ya que entonces el análisis variará en función de las leyes del país o países en los que se pretende presentar las evidencias como prueba judicial, y sin embargo, existen pautas comunes a la hora de realizar el análisis, al margen de las leyes que imperen en el país determinado.

También es posible realizar un análisis de sistemas de información en un entorno corporativo, donde el término forense pierde su sentido. En este trabajo seguiré su filosofía, eludiendo los detalles legales del análisis de sistemas de información y omitiendo, por tanto, el término forense. Existen dos áreas independientes de análisis de sistemas de información: el análisis de medios físicos (por ejemplo, discos duros) y el análisis basado en dispositivos de comunicación (análisis de redes). En este trabajo nos centraremos en la primera área.

Figura 14. Niveles de análisis digital según la estructura de los datos a analizar



Fuente: Carrier, Brian. File System Forensic Analysis. Addison-Wesley, 2005

Los dispositivos de almacenamiento no-volátil (como un disco duro o una memoria flash) están normalmente organizados en volúmenes. Hay dispositivos, como los disquetes, que

únicamente poseen un volumen. Las particiones son utilizadas para dividir un volumen en otros más pequeños. A su vez, se pueden organizar volúmenes para formar otros más grandes (como en los sistemas RAID). Los volúmenes de los dispositivos pueden contener sistemas de ficheros (lo más común), espacios de intercambio (denominados swap) o bases de datos. Es necesario conocer la información referente a los volúmenes porque nos permitirá encontrar datos ocultos y descubrir donde se encuentran los sistemas de archivos. Los sistemas de ficheros son estructuras de datos que permiten a las aplicaciones crear, leer y escribir ficheros en el disco.

El análisis de un sistema de ficheros nos permite encontrar ficheros, recuperar ficheros borrados y encontrar ficheros ocultos. Para poder analizar los ficheros encontrados necesitamos analizarlos a nivel de aplicación, ya que su estructura depende de las aplicaciones que los crearon (y del sistema operativo sobre el que estas funcionaban). El análisis de aplicación se divide a su vez en varias categorías, como pueden ser:

Análisis de sistemas operativos, en el que se examinan los ficheros de configuración y salida del SO para obtener información sobre los sucesos ocurridos.
Análisis de programas ejecutables, en el que se examinan los sucesos que podrían desencadenan los mismos.
Análisis de imágenes, en el que se trata de buscar información en las fotografías, como por ejemplo quién está en la foto o quién la tomó y cuándo. También se buscan indicios de esteganografía
Análisis de videos, como el utilizado con webcams o cámaras de vigilancia, en el que se busca, al igual que en el análisis de imágenes, información de quién aparece, dónde y cuándo fue grabado.

4.3.3 Tipos de análisis. Dentro de la presente investigación se optó por abordar los aspectos referentes al análisis de medios de almacenamiento físicos no volátiles, y en un sentido un poco más exacto siguiendo la rama descrita en la Figura 10. Se abordaran el análisis de volúmenes, de ficheros y de sistemas operativos y aplicaciones. Los dispositivos de almacenamiento que se utilizan para el almacenamiento no volátil están organizados típicamente en volúmenes. El volumen es una recopilación de los lugares de almacenamiento que un usuario o aplicación puede escribir y leer. Hay dos conceptos importantes en esta capa. Uno es el particionamiento, donde nos dividimos un solo volumen en múltiples pequeños volúmenes, y el otro es el montaje, donde se conjugan varios volúmenes en un mayor volumen, que más tarde puede ser dividida. Ejemplos de esta categoría incluyen partición DOS tablas, particiones de Apple y matrices RAID. Algunos medios, como disquetes, no tienen datos en esta capa, y todo el disco es un volumen. Tendremos que analizar los datos en el nivel de volumen para determinar dónde se encuentra el sistema de archivos u otros datos y para determinar dónde pueden encontrar los datos ocultos.

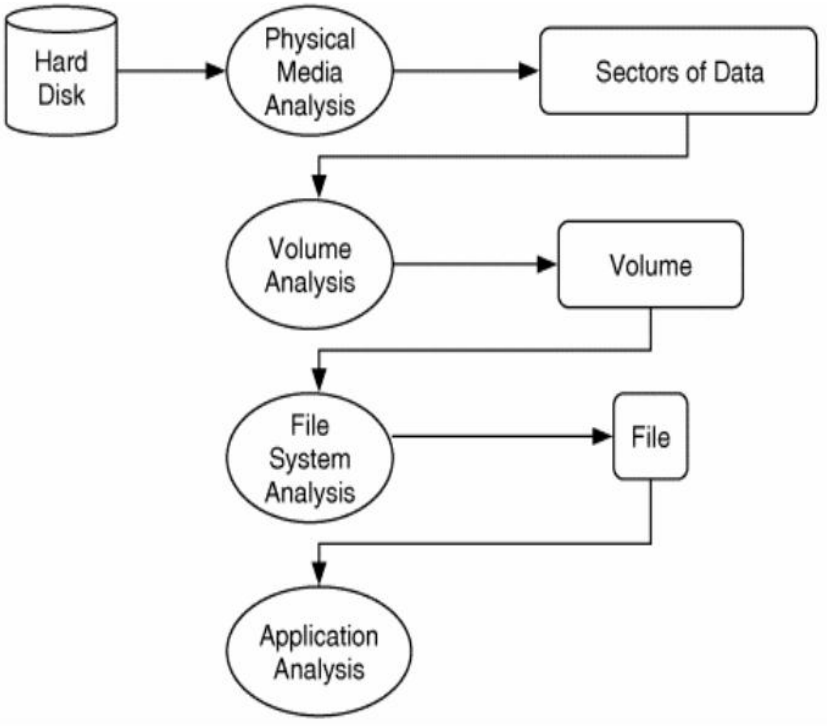
Dentro de cada volumen pueden haber cualquier tipo de datos, pero el contenido más comunes son sistemas de archivos. Otros volúmenes pueden contener una base de datos o ser utilizado como un espacio de intercambio temporal (similar a la Archivo de paginación

de Windows). Los sistemas de archivos, son una colección de estructuras de datos que permitan una aplicación crear, leer y escribir archivos.

Analizamos un sistema de archivos para encontrar archivos, recuperar archivos borrados, y para encontrar los datos ocultos. El resultado del sistema de archivos análisis podría ser el contenido del archivo, fragmentos de datos y metadatos asociados a archivos. Para entender lo que está dentro de un archivo, tenemos que saltar a la capa de aplicación. La estructura de cada archivo se basa en la aplicación o el sistema operativo que creó el archivo. Por ejemplo, desde el archivo perspectiva de sistema, un archivo de registro de Windows no es diferente de una página HTML, ya que son los dos archivos. Internamente, tienen estructuras muy diferentes y se necesitan diferentes herramientas para analizar cada uno.

Análisis de aplicaciones es muy importante, y es aquí donde podremos analizar la configuración de archivos para determinar qué programas se ejecutan, o para determinar qué JPEG es un formato de fotografía. Podemos ver el proceso de análisis en la Figura 11. Este muestra un disco que se analiza para producir una secuencia de bytes, que se analizan en la capa de volumen para producir volúmenes. Los volúmenes son analizados en la capa del sistema de archivos para generar un archivo. El archivo se analiza a continuación, en la aplicación capa.

Figura 15. Proceso de análisis de datos desde el nivel físico hasta al nivel de aplicación.



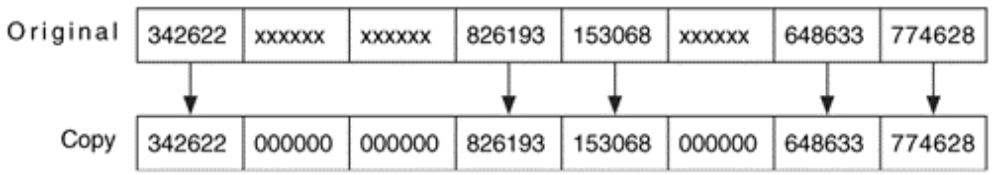
Fuente: Carrier, Brian. File System Forensic Analysis. Addison-Wesley, 2005

4.3.4 Adquisición de datos del disco duro. A la hora de adquirir los datos debemos tener en cuenta que los mismos están interpretados en diferentes niveles (discos, volúmenes, ficheros, etc.) y que puede que sea necesario realizar una copia de todo el disco o solo de unos cuantos ficheros, dependiendo del nivel al que creamos que se encuentran los datos que nos proporcionaran evidencias. La adquisición de datos desde un dispositivo (supongamos que queremos obtener los datos a nivel de disco) tiene dos fases[30]: la lectura de la información y la posterior escritura. Como se ha visto con anterioridad, una de las premisas del análisis digital debe ser modificar los datos originales lo menos posible. Para ello se pueden utilizar durante esta fase, bloqueadores de escritura (hardware o software) que se interponen entre el controlador y el disco duro y evitan que se realicen las operaciones de escritura.

La adquisición es una parte crucial del proceso de investigación, y el Instituto Nacional de Estándares y Tecnología (NIST)[31] ha llevado a cabo pruebas en las herramientas de adquisición comunes. The Computer Forensic Tool Testing (CFTT)[32] en el NIST desarrolla los requisitos y pruebas estuches para herramientas de imágenes de disco. Los resultados y las especificaciones se pueden encontrar en su sitio Web Aseveraciones.
Control de errores

Cuando una herramienta de adquisición está leyendo datos de un disco, que tiene que ser capaz de manejar errores. Los errores pueden ser causados por un problema físico en el que toda la unidad ya no obras, o los errores podrían estar en un número limitado de sectores. Si sólo un número limitado de sectores están dañados, una adquisición normal puede ocurrir, a condición de que la herramienta de adquisición maneje adecuadamente los errores. El comportamiento de aceptación general para hacer frente a un sector defectuoso es registrar su dirección y escribir 0s para los datos que no se puede leer. Escribir 0s mantiene los otros datos en su ubicación correcta. Si el sector se ignoró en lugar de escribir 0s, la copia resultante sería demasiado pequeña, y la mayoría herramientas de análisis no funcionarían. La figura 12 muestra una serie de valores que están siendo adquiridos. Tres de los valores tiene errores y no se puede leer, por lo 0s se escriben en la copia.

Figura 16. El original tiene tres errores en los mismos que han sido substituido.



Fuente: Carrier, Brian. File System Forensic Analysis. Addison-Wesley, 2005

[30] Análisis forense de sistemas de información, Carmen Rodríguez Vázquez , [En Línea] disponible en: http://es.scribd.com/doc/96651301/Analisis-Forense-de-SI

[31] NIST: National Institute of Standards and Tecnology, [en línea]. Disponible en: http://www.nist.gov/index.html

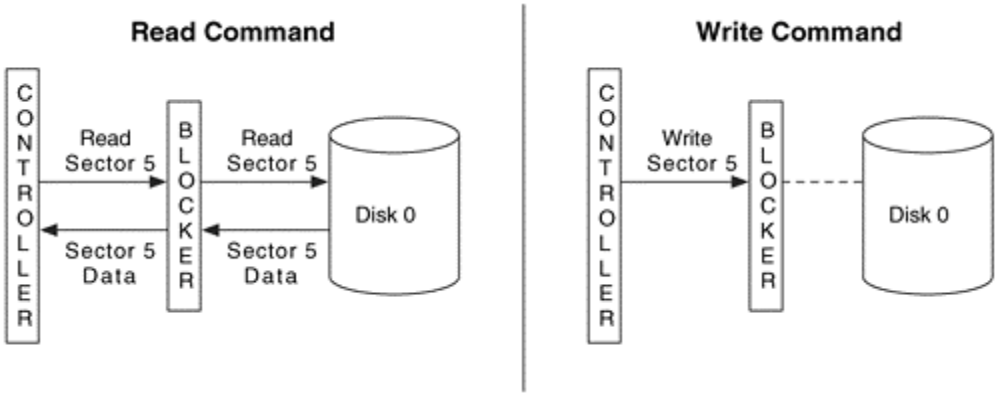
[32] CFTT: The Computer Forensic Tool Testing, [en línea]. Disponible en: http://www.cftt.nist.gov/disk_imaging.htm

Adquisición Física. La adquisición física comprende todos los procesos que involucren el hardware en sí mismo y el procedimiento que conlleva a dicho acto, es decir tanto la manipulación previa a la información, pues en esta etapa se garantiza buena parte de la integridad de la posible evidencia que pueda llegar ase colectada.

Apagado del sistema. Una de las decisiones principales a las que se deben tomar es enfrentar es si apagar o no el sistema una vez que sospechamos que ha podido ser comprometido. Y si decidimos apagarlo, debemos decidir cómo, o de un modo ordenado, o tirando del cable de alimentación. El análisis forense de un sistema vivo tiene a veces sus ventajas. Podemos hacer cosas como examinar la tabla de procesos para ver qué se está ejecutando, listar las conexiones de red o copiar lo que hay en memoria para examinarlo más tarde. Pero existen varios inconvenientes a la hora de examinar un sistema vivo, como por ejemplo que no estemos viendo lo que hay en realidad. Los rootkits modernos pueden ocultar datos y procesos fácilmente, implementando hooks a nivel de kernel por ejemplo. Un sistema muerto es más fácil de examinar, pudiéndose garantizar que después de su apagado no se han modificado o eliminado pruebas acerca del estado en que se encontraba el sistema. Pero, ¿cómo apagamos el sistema? Un apagado ordenado podría iniciar programas destinados a la limpieza de evidencias o, si el atacante tuviera aún peores intenciones, sobrescribir el firmware del disco duro o del sistema. Sea como fuere, al tirar del cable dejaríamos el sistema en un estado inconsistente o impediríamos que se escribiesen datos a los dispositivos.

Bloqueadores de escritura de hardware. Una de las directrices de investigación que hemos discutido en el capítulo 1 fue modificar el original datos como poco como sea posible. Hay muchas técnicas de adquisición que no modifican ninguno de los datos originales, pero los errores pueden ocurrir. Además, también hay algunas técnicas de adquisición que puede modificar los datos originales, y es posible que queramos evitarlo. Un protector de escritura de hardware es un dispositivo que se encuentra en la conexión entre un ordenador y un dispositivo de almacenamiento. Efectuará un seguimiento de los comandos que se están emitiendo y evita que el equipo de escritura de datos en el dispositivo de almacenamiento. Escribe bloqueadores de apoyo muchas interfaces de almacenamiento, tales como ATA, SCSI, Firewire (IEEE 1394), USB o Serial ATA. Estos dispositivos son especialmente importante cuando se utiliza un sistema operativo que puede montar el disco original, como Microsoft Windows.

Figura 17. La solicitud de lectura para el sector 5 se hace pasar a través del bloqueador de escritura, pero el comando de escritura para el mismo sector se bloquea antes de que llegue el disco.



Fuente: Carrier, Brian. File System Forensic Analysis. Addison-Wesley, 2005

Como todas las herramientas de investigación, las pruebas de los bloqueadores de escritura de hardware es importante, y el CFTT grupo en el NIST ha publicado un pliego de condiciones de bloqueadores de escritura de hardware[33] .La especificación clasifica los comandos ATA como no modificar, modificar, y la configuración. La especificación de estados comandos de modificación que deben ser bloqueados y devuelven opcionalmente éxito o el fracaso**.**

Escribir los datos de salida. Después de leer los datos del disco de origen, tenemos que escribir en alguna parte. Para tal efecto existen gran cantidad de métodos, los cuales permiten realizar la inspección de la información, por tal motivo se mostraran varios de ellos con hasta llegar al empleado en este estudio.

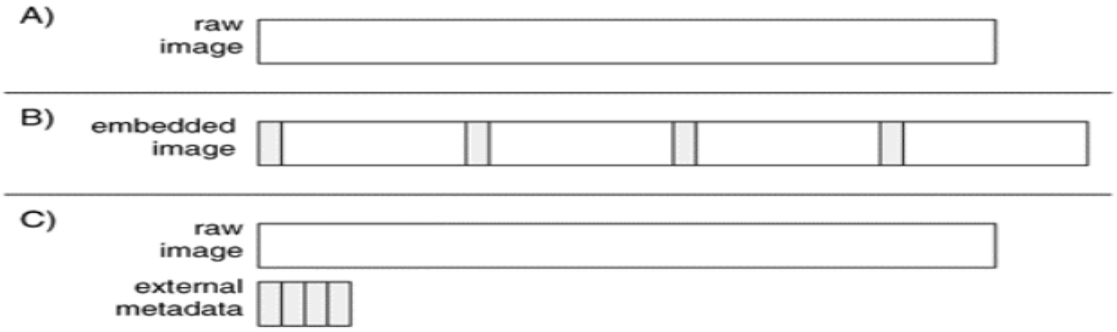
Destino ubicación. Cuando guardamos los datos, podemos escribir directamente en un disco o en un archivo. Antes de que existiera un software de análisis especializado, un investigador o bien arranca el sistema sospechoso o monta los discos en su sistema de análisis. Esta adquirió la unidad copiando los datos directamente a otro disco. En otras palabras, el sector 0 del disco de origen era idéntica a el sector 0 del disco de destino. El disco resultante se denomina con frecuencia un duplicado o una copia clonado. Este método puede causar problemas cuando el disco de destino es más grande que el disco de origen, ya que puede ser difícil saber exactamente donde termina la copia. Al adquirir directamente en el disco, se recomienda que el disco puede limpiar con ceros antes de la adquisición por lo que los datos no relacionados, posiblemente de una investigación previa, no se confunden con los datos desde el sistema sospechoso. Un segundo problema con la adquisición en el disco es que algunos sistemas operativos, como Microsoft Windows, intentarán montar cualquier disco y la copia pueden ser montada por el sistema de adquisición y tener sus datos cambiados. También puede ejecutar en dificultades si los discos originales y de

[33] Pliego de condiciones bloqueadores de escritura de hardware,NIST, [en línea]. Disponible en: http://www.cftt.nist.gov/hardware_write_block.htm

destino tienen diferentes geometrías, porque algunos de los estructuras de datos se basan en la geometría para describir ubicaciones. En la actualidad, la ubicación de salida más común es la de guardar los datos en un archivo en un disco duro o CD-ROM. Con un archivo, es fácil saber los límites de los datos y que los sistemas operativos no intenten montar de forma automática. El archivo se llama con frecuencia una imagen o una imagen duplicada. Muchas herramientas le permiten romper un archivo de imagen en pedazos más pequeños para que quepan en los CDso DVDs. Algunos investigadores se limpian los discos que almacenan archivos de imagen para que puedan testificar más fácilmente que no podría haber cualquier tipo de contaminación de casos anteriores.

Formato de archivo. Si guardamos los datos en un archivo, podemos elegir en qué formato va a ser la imagen. Una imagen raw contiene sólo los datos desde el dispositivo fuente, y es fácil de comparar la imagen con los datos de origen. Una imagen incrustada contiene información desde el dispositivo de origen y adicionales datos descriptivos acerca de la adquisición, como los valores de hash, fechas y horas. Algunas herramientas se crear una imagen sin procesar y guardar los datos descriptivos en un archivo separado. Recordemos que los valores hash, tales como CRC, MD5 y SHA-1, se utilizan para mostrar la integridad de los datos. Ejemplos de formatos de imagen se pueden ver en la Figura 14.

Figura 18. Los ejemplos de (A) una imagen RAW, (B) una imagen incrustada en los metadatos intercalan en los datos en bruto, y (C) una imagen con los datos almacenados en un formato en bruto y los metadatos almacenados en un segundo archivo.



Fuente: Carrier, Brian. File System Forensic Analysis. Addison-Wesley, 2005

En las implementaciones actuales de herramientas de adquisición, muchos de los formatos de imagen incrustados son patentados, tales como los de EnCase Guidance Software[34] y SafeBack del NTI[35], y algunos están documentados, como el formato utilizado por la tecnología Pathway's ProDiscover[36]. La mayoría de las herramientas de análisis importan imagenes sin procesar, por lo tanto, los formatos son más flexibles. La herramienta inteligente de ASR datos y los herramientas dcfldd / dccidd adquieren datos en un formato raw y tienen un archivo externo con datos adicionales.

[34] EnCase Forensic [en línea]. Disponible en: http://www.guidancesoftware.com/encase-forensic.htm
[35] SafeBack del NTI [en línea]. Disponible en: http://www.forensics-intl.com/safeback.html
[36] Pathway's ProDiscover [en línea]. Disponible en:http://www.techpathways.com/DesktopDefault.aspx

Algoritmos criptográficos. Un algoritmo consiste en un conjunto determinado y finito de pasos o acciones, con el objetivo de solucionar un problema. Los algoritmos criptográficos son funciones matemáticas, que son utilizadas en los procesos de encriptación o des encriptación de datos, que serán la entrada o parámetro de estas funciones. Si se hace referencia a procesos de encriptación o des encriptación, también existen funciones criptográficas como los hashing, o para la generación números aleatorios.

Modos de algoritmos criptográficos. En cuanto a los modos criptográficos, aplican a los algoritmos de cifrado simétrico. Un modo comúnmente combina el algoritmo con alguna forma de retroalimentación, o feedback, y una serie de operaciones. Estas operaciones serán simples, ya que la seguridad dependerá del algoritmo y no del modo que es utilizado. A continuación, se describen los diferentes modos que se encuentran como opción en las funcionalidades provistas por los entornos de programación.

Funciones hash seguras. La función hash unidireccional o función hash segura no sólo es importante para la autenticación de mensajes, sino también para las firmas digitales. En esta sección, se comenzará un análisis de los requisitos para una función hash segura. A continuación se tratará las funciones de control más importantes, las SHA.

Requisitos de la función hash. El propósito de una función hash es la de obtener una "huella" de un archivo, mensaje u otro bloque de datos. Para que resulte útil a la autenticación de mensajes, una función hash H debe poseer las siguientes propiedades:

H puede aplicarse a un bloque de datos de cualquier tamaño.
H produce una salida de tamaño fijo.
H (x) es relativamente fácil de computar para cualquier x dada, haciendo que tanto las implementaciones de hardware como software sean prácticas.
Para cualquier código h dado, es imposible desde el punto de vista computacional encontrar x tal que H(x) = h. Una función hash con esta propiedad se conoce como un solo sentido o unidireccional.
Para cualquier bloque dado x, es imposible desde el punto de vista computacional encontrar $y \neq x$ con H(y) = H(x). Una función hash con esta propiedad se conoce como resistencia débil a la colisión.
Es imposible desde el punto de vista computacional encontrar un par (x, y) tal que H(x) = H (y). Una función hash con esta propiedad se conoce como resistencia fuerte a la colisión.

Las tres primeras propiedades son requisitos para la aplicación práctica de una función hash a la autentificación de mensajes. La cuarta propiedad “unidireccional” dado un mensaje, es fácil generar un código, pero dado un código, es prácticamente imposible generar un mensaje. Esta propiedad es importante si la técnica de autenticación implica el uso de un valor secreto. El valor secreto no se envía; sin embargo, si la función hash no es unidireccional, un atacante puede descubrir fácilmente el valor secreto: si el atacante puede observar o interceptar una transmisión, obtiene el mensaje M y el código hash C =

H(SAB||M). Entonces el atacante invierte la función hash para obtener SAB||M = H-1(C).

Debido a que el atacante tiene ahora M y SAB||M, no tiene importancia recuperar SAB.
La segunda propiedad garantiza que es imposible encontrar un mensaje alternativo con el mismo valor hash cifrado. Si no se diera esta propiedad, un atacante podría ser capaz de la siguiente secuencia: en primer lugar, observar o interceptar un mensaje más su código hash cifrado, en segundo lugar, generar un código hash sin encriptar del mensaje, en tercer lugar, generar un mensaje alternativo con el mismo código hash. Una función hash que satisfaga las cinco primeras propiedades en la lista anterior se conoce como función hash débil. Si también posee la sexta propiedad, entonces se le conoce como una función hash robusta. La sexta propiedad protege contra un tipo sofisticado de ataque conocido como ataque basado en la paradoja del cumpleaños.

Además de proporcionar autentificación, un resumen de un mensaje también proporciona integridad de los datos. Realiza la misma función que una secuencia de comprobación de trama: si cualquier bit del mensaje se alteran accidentalmente durante la transmisión, el resumen del mensaje dará error.

Seguridad de las funciones hash: Al igual que con el cifrado simétrico, hay dos formas de atacar a una función hash segura: criptoanálisis y fuerza bruta, con los algoritmos de cifrado simétrico, el criptoanálisis de una función hash consiste en explotar las debilidades en el algoritmo lógico.

La fuerza de una función hash contra ataques de fuerza bruta depende únicamente de la longitud del código hash producido por el algoritmo. Para un código hash de longitud n, el nivel de esfuerzo requerido es proporcional a lo siguiente:

Figura 19. Tiempo requerido para romper un hash de orden n.

Preimage resistant	2^n
Second preimage resistant	2^n
Collision resistant	$2^{n/2}$

Fuente: William Stallings, NETWORK SECURITY ESSENTIALS applications and standards. Fourth edition.

Funciones Hash Simples: Todas las funciones hash funcionan con los siguientes principios generales. La entrada (mensajes, archivos, etc) es visto como una secuencia de bloques de n bits. La entrada se procesa bloque a bloque de forma iterativa para producir una función hash de n bits.

Escribir los datos de salida. Después de leer los datos del disco de origen, tenemos que escribir en alguna parte. Para tal efecto existen gran cantidad de métodos, los cuales permiten realizar la inspección de la información, por tal motivo se mostraran varios de ellos con hasta llegar al empleado en este estudio.

Destino ubicación. Cuando guardamos los datos, podemos escribir directamente en un disco o en un archivo. Antes de que existiera un software de análisis especializado, un investigador o bien arranca el sistema sospechoso o monta los discos en su sistema de análisis. Esta adquirió la unidad copiando los datos directamente a otro disco. En otras palabras, el sector 0 del disco de origen era idéntica a el sector 0 del disco de destino. El disco resultante se denomina con frecuencia un duplicado o una copia clonado. Este método puede causar problemas cuando el disco de destino es más grande que el disco de origen, ya que puede ser difícil saber exactamente donde termina la copia. Al adquirir directamente en el disco, se recomienda que el disco puede limpiar con ceros antes de la adquisición por lo que los datos no relacionados, posiblemente de una investigación previa, no se confunden con los datos desde el sistema sospechoso. Un segundo problema con la adquisición en el disco es que algunos sistemas operativos, como Microsoft Windows, intentarán montar cualquier disco y la copia pueden ser montada por el sistema de adquisición y tener sus datos cambiados. También puede ejecutar en dificultades si los discos originales y de destino tienen diferentes geometrías, porque algunos de los estructuras de datos se basan en la geometría para describir ubicaciones. En la actualidad, la ubicación de salida más común es la de guardar los datos en un archivo en un disco duro o CD-ROM. Con un archivo, es fácil saber los límites de los datos y que los sistemas operativos no intenten montar de forma automática. El archivo se llama con frecuencia una imagen o una imagen duplicada. Muchas herramientas le permiten romper un archivo de imagen en pedazos más pequeños para que quepan en los CDso DVDs. Algunos investigadores se limpian los discos que almacenan archivos de imagen para que puedan testificar más fácilmente que no podría haber cualquier tipo de contaminación de casos anteriores.

4.3.5 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 qué 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.

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 más 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 sistemas operativos 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 SISTEMAS OPERATIVOS 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 direccionales que un sistema operativo 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 sistema operativo puede usar diferentes estrategias para asignar unidades de datos. Normalmente un sistema operativo 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 sistema operativo 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 SISTEMAS OPERATIVOS 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 está 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.

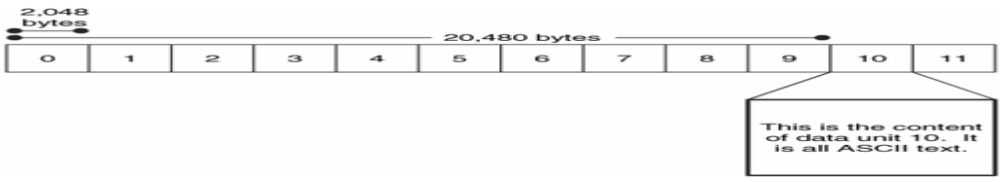
Es fácil esconder datos usando esta funcionalidad del sistema de ficheros, si existe (solo está en discos duros antiguos). Muchas herramientas que prueban la consistencia de un sistema de ficheros no verifican que una unidad de datos que está marcada como dañada esté actualmente dañada. Por lo tanto,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

Figura 20. Contenido de la unidad de datos



Fuente: File System Analysis, Brian Carrier, 2003

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 dónde 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 SISTEMAS OPERATIVOS; 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 SISTEMAS OPERATIVOS para ayudarnos a determinarlo. Esto es muy difícil, ya que requiere determinar la estrategia que usa el SISTEMAS OPERATIVOS 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 está 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 cómo analizar datos en esta categoría, vamos a pasar a ver como un usuario puede hacernos la vida más 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 SISTEMAS OPERATIVOS 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 SISTEMAS OPERATIVOS no escribía inmediatamente ceros en el disco. Más tarde el SISTEMAS OPERATIVOS 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, los SISTEMAS OPERATIVOS usarán las mismas unidades de datos. Un SISTEMA OPERATIVO 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.

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 SISTEMAS OPERATIVOS 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.

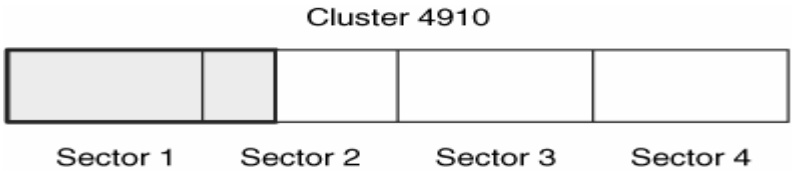
Slack Space. El "Slack space" es una de las palabras de moda en el análisis forense que mucha gente ha oído alguna vez. El Slack space ocurre cuando el tamaño de un fichero no es múltiplo del tamaño de la unidad de datos. Un fichero debe tener asignada una unidad de datos completa aunque muchas veces 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 SISTEMAS OPERATIVOS 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 SISTEMA OPERATIVO determina con que rellenar el contenido. El método obvio es rellenar el sector con ceros y esto es lo que la mayoría de SISTEMAS OPERATIVOS hacen. Esto es como rellenar la caja anterior con papel de periódico. Algunos SISTEMAS OPERATIVOS 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 SISTEMAS OPERATIVOS 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 SISTEMA OPERATIVO. El tercer y cuarto sectores pueden ser limpiados con ceros por el SISTEMAS OPERATIVOS, 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.

Figura 21. *Slack space de un fichero de 612 bytes en un cluster de 2048 bytes donde cada sector tiene 512 bytes.*



Fuente: Herramienta de apoyo para el análisis forense de computadoras, José Arquillo Cruz, 2007

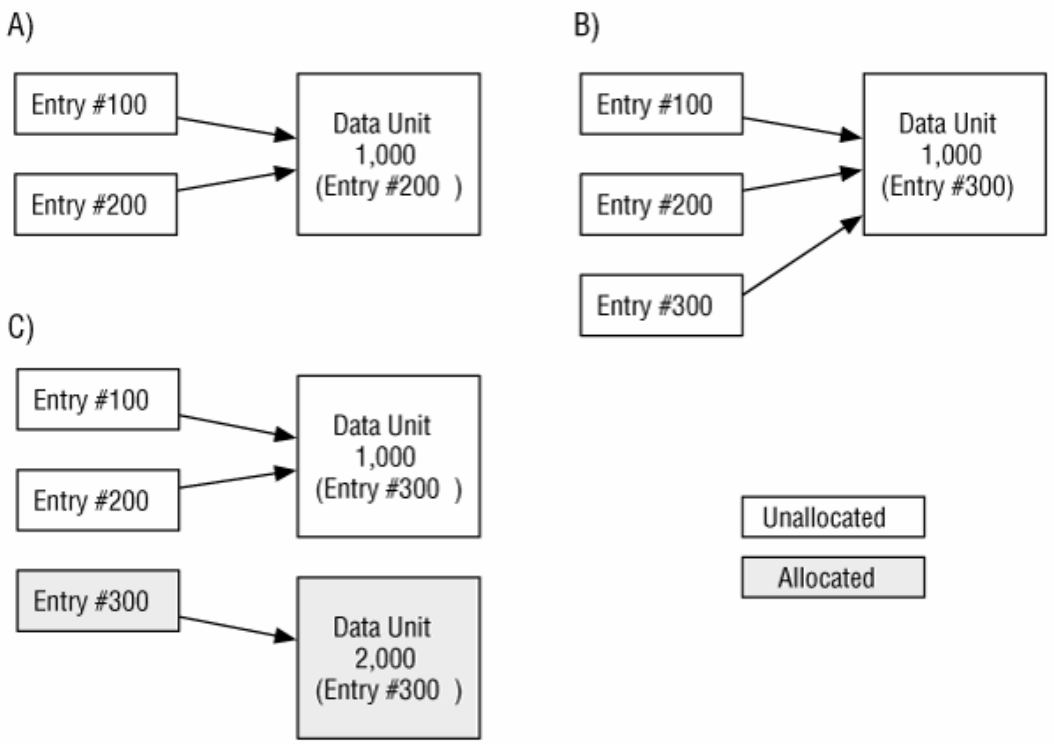
Una analogía común para el slack space es el video VHS. Supongamos una cinta VHS de 60 minutos. Una noche alguien graba 60 minutos de un episodio de una serie de TV. En otra ocasión decide ver de nuevo el episodio y al final rebobina la cinta. Otra noche graba 30 minutos de otro programa de TV. En ese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

Recuperación de ficheros basada en las 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 aún existen. Si la 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

Figura 22. La secuencia de estados donde los ficheros son asignados y borrados y en C no está claro de dónde vienen los datos de la unidad de datos 1000.



Fuente Herramienta de apoyo para el análisis forense de computadoras, José Arquillo Cruz, UNIVERSIDAD DE JAÉN, Escuela Politécnica Superior de Jaén 2007

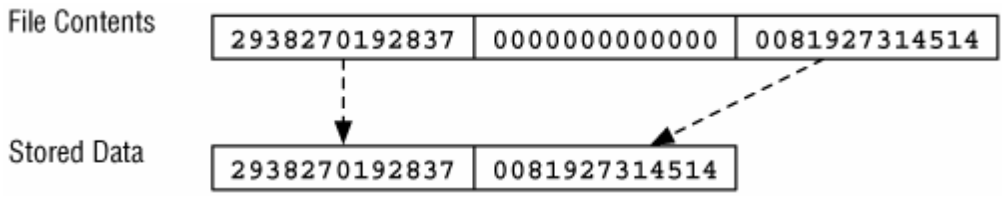
Necesitamos determinar cuál de las entradas se asignó al fichero más recientemente. Un método para hacer esto es usar la información temporal de cada entrada (u otros datos externos), pero puede que no seamos capaces de asegurarlo. Otro método es usar el tipo de

fichero, si la 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 última 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 los 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 está 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 está lleno de ceros.

Figura 23. Un fichero almacenado en formato "sparse" donde las unidades de datos con ceros no se escriben.



Fuente: File System Analysis, Brian Carrier, 2003

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 SISTEMAS OPERATIVOS cuando crea el fichero. Antes de que un fichero se escriba en disco, el SISTEMAS OPERATIVOS 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 SISTEMAS OPERATIVOS 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 metadatos para ver el contenido de los ficheros, buscar valores y localizar ficheros borrados.

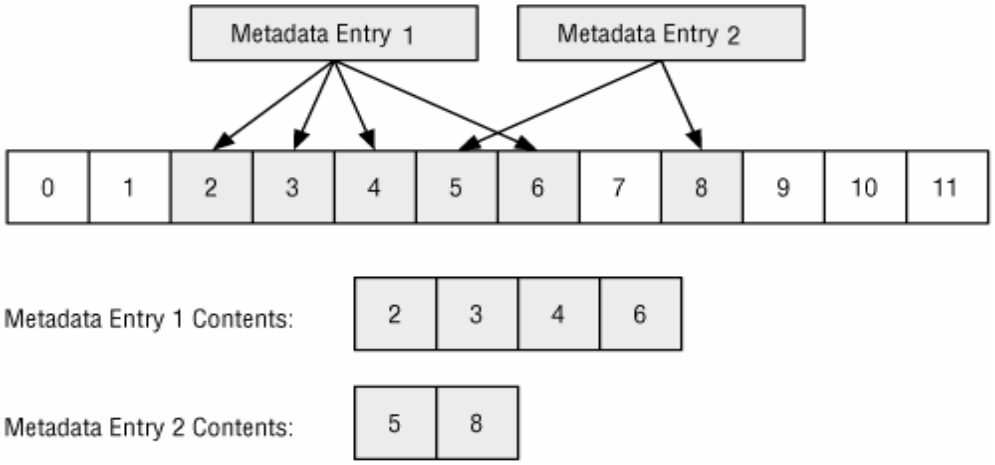
Búsqueda de metadatos. En muchos casos, analizamos los metadatos porque encontramos el nombre de un fichero que apunta a una estructura de metadatos específica y queremos aprender más sobre el fichero. Por lo tanto, necesitamos ubicar los metadatos y procesar su estructura de datos. Por ejemplo, si buscamos 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.

Figura 24. *Combinamos la información de las entradas de metadatos y las unidades de datos para ver el contenido de un fichero.*



Fuente. Análisis forense de sistemas de información, Carmen Rodríguez Vázquez

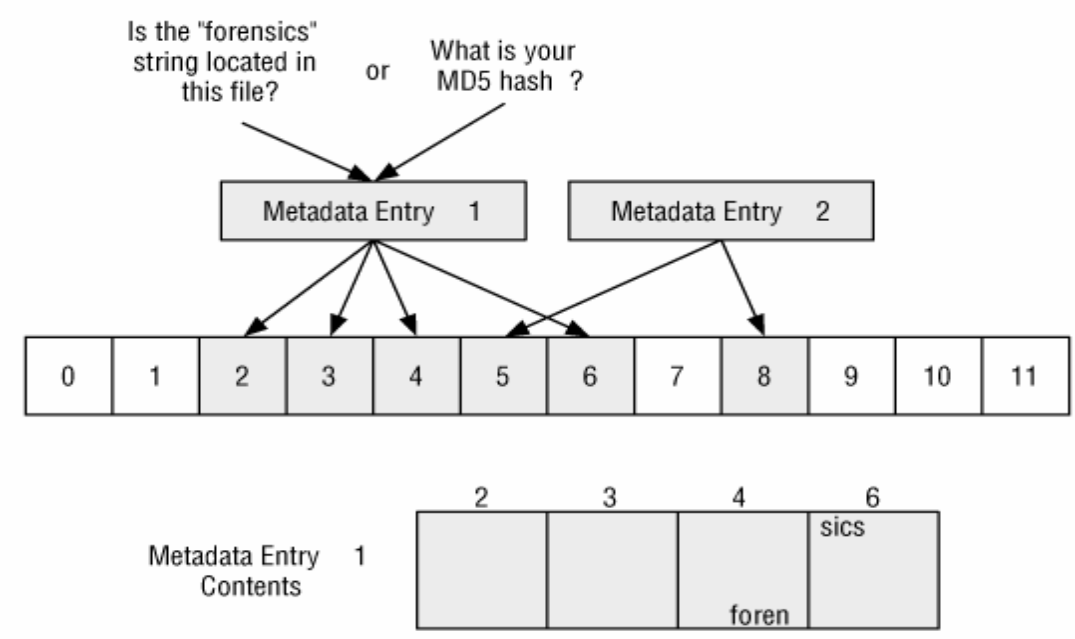
Durante este proceso, necesitamos mantener en mente el slack space porque el fichero puede no estar usando completamente el final de la unidad de datos.

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. 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.

Figura 25. Una búsqueda lógica en las unidades de datos asignadas a una entrada de metadatos.



Fuente: File System Analysis, Brian Carrier, 2003

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 re-use 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 cómo 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 diretorio C:\Windows podría ser:

Figura 26. Output del comando Mactime de The Sleuth Kit

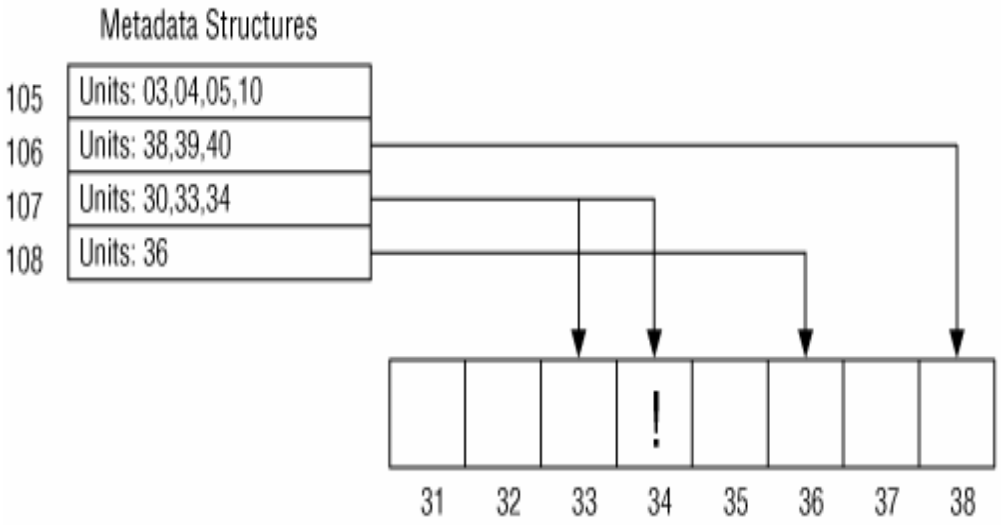
```
Wed Aug 11 2004 19:31:58     34528 .a. /system32/ntio804.sys

                             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
```

Fuente: File System Analysis, Brian Carrier, 2003

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 SISTEMAS OPERATIVOS 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 SISTEMAS OPERATIVOS 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 SISTEMAS OPERATIVOS no limpia los valores dirección cuando un fichero es borrado, este proceso puede identificar estructuras de meta-datos no asignadas.

Figura 27. Puede ser útil buscar entre las estructuras de meta-datos para encontrar una que tenga unidades de datos asignadas.



Fuente: Herramienta de apoyo para el análisis forense de computadoras, José Arquillo Cruz, 2007

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 SISTEMAS OPERATIVOS para determinarlo. Esto es muy dependiente de cada SISTEMAS OPERATIVOS 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.

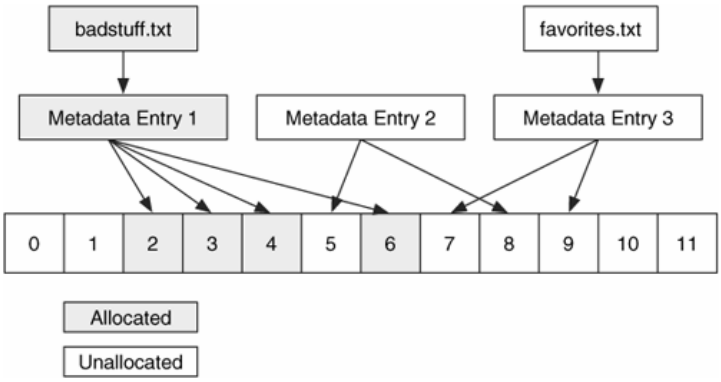
Categoría nombre de fichero. Esta categoría incluye los nombres de ficheros, que permiten al usuario referirse a un fichero por su nombre en vez de por su dirección de meta-datos. En esencia, esta categoría de datos incluye sólo el nombre de un archivo y su dirección de metadatos. Algunos sistemas de 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 16 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.

Figura 28. Podemos recuperar ficheros basándonos en su nombre, pero aun así se usarán técnicas de meta-datos para la recuperación.



Fuente: Herramienta de apoyo para el análisis forense de computadoras, José Arquillo Cruz, 2007

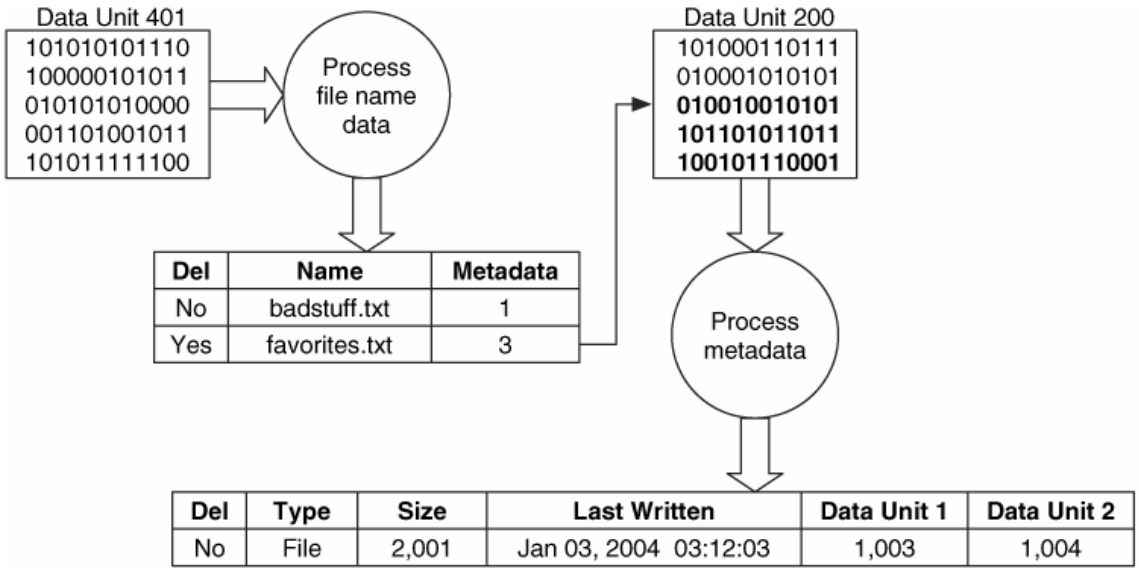
Para finalizar, si examinamos ficheros borrados desde la perspectiva de nombre de fichero, debemos tener en cuenta que las meta-datos y las unidades de datos podrían haber sido reasignadas a otro fichero. Además recordar que se necesita examinar las estructuras de meta-datos no asignadas para encontrar 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. Figura 17 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.

Figura 29. *Relación entre nombres de fichero y metadatos.*



Fuente: File System Analysis, Brian Carrier, 2003

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. 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 SISTEMAS OPERATIVOS, esto es difícil, ya que el SISTEMAS OPERATIVOS 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.

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 SISTEMAS OPERATIVOS o aplicación crea, podría ser designado como una característica en un sistema de ficheros. Por ejemplo, Acme Software podría decidir que sus SISTEMAS OPERATIVOS 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 SISTEMAS OPERATIVOS 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 SISTEMAS OPERATIVOS 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 el 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.

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.

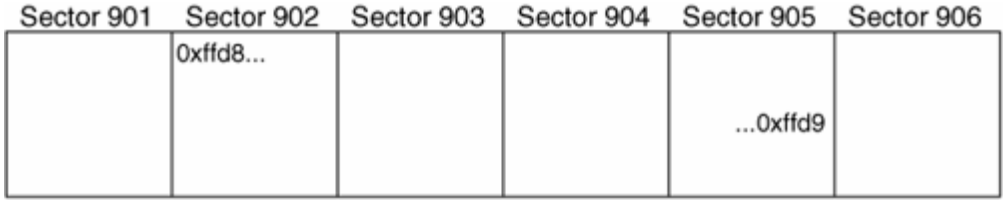
Una herramienta de ejemplo que realiza esto es FOREMOST[37]que ha sido desarrollada por los agentes especiales Kris Kendall y Jesse Kornblum de la ficina de Investigaciones especiales de las fuerzas aéreas de los Estados Unidos. Foremost analiza un sistema de

[37] Foremost [en línea]. Disponible en: http://foremost.sourceforge.net/

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 17 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.

Figura 30. *Bloques de datos en bruto en los que encontramos una imagen JPEG mediante su cabecera y su cola*



Fuente: File System Analysis, Brian Carrier, 2003

Una herramienta similar es LAZARUS (disponible en The Coroner's Toolkit[38]) 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 está 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 está claro si la ordenación está basada en la extensión o en la firma del fichero.

[38] The Coroner's Toolkit [en línea]. Disponible en: http://www.porcupine.org/forensics/tct.html

5. CONCLUSIONES

Una vez realizada la revisión bibliográfica de las metodologías aplicables al análisis forense con la metodología post mortem, e identificado que la más adecuada es la Casey, y tras evaluar las diversa herramientas forenses tomando la suit de The Sleuth Kit con su interfaz gráfica Autopsy como las más adecuadas para procesar imágenes bit a bit adquiridas mediante el comando DD de Linux, podemos afirmar que en los diferentes casos de estudio, la metodología post mortem ofrece una gran efectividad, aunque es de resaltar que dicha efectividad depende en su totalidad de 2 aspectos; el primero es el caso de estudio en sí mismo pues hay ocasiones en que no es posible apagar el equipo para realizar el estudio, o en otros caso el estudio se enfoca al sistema de información o a la red misma, es por ello que debe hacerse un análisis concienzudo del escenario. Por otro lado la experticia del investigador es la que en últimas define el éxito o fracaso del trabajo. En resumen y tras tener en cuenta los aspectos tanto metodológicos como prácticos y tras diseñar el modelo practico SIGLAS, la presente investigación verifico el desempeño del análisis post mortem en diversos casos de investigación forense ante lo cual se da por concluido este trabajo investigativo.

6. RECOMENDACIONES

A lo largo de la presente investigación se han realizado diversos laboratorios y reconstruido escenas de retos forenses para establecer la efectividad y alcances de el análisis digital forense mediante la metodología pos-mortem, y lo cierto es que esta muestra notorias ventajas con referencia a la metodología en caliente, ya que libra la posibilidad de activar bombas lógicas o rootkits que el atacante pudo implantar en el sistema, así mismo establece un minucioso proceso para la recolección identificación y estudio de las evidencias el cual se fundamenta tanto en estándares internacionales como el RFC3227, el estándar de la EDRM, metodologías como la Casey y obviamente el marco legal colombiano.

Por otro lado la fiabilidad de la evidencia no volátil se ve ligeramente opacada por la vigencia inmediata que los datos como los que residen en la memoria, puesto que pueden podrían proporcionar pistas para que de otro modo sean prácticamente imposibles de adquirir.

Es indudable que ambas técnicas son sumamente provechosas, y son pertinentes dependiendo de la situación específica a la cual se desee aplicar.

Si viene es cierto que existen múltiples herramientas, también es claro que teniendo que dentro del contexto mundial el incremento en las actividades delictiva informática es enorme, el costo de las aplicaciones propietarias es a su vez sumamente alto, razón por la cual el hecho de la implementación de un estudio con herramientas completamente libres da cierto peso a la presente investigación.

Un aspecto sumamente curiosos que se descubrió en este análisis, lo constituye el hecho de que de cierto modo los procedimientos metodológicos son muy claro y existen muchos, no obstante, una guía de que hacer no es desglosado de una manera significativa, razón por la cual pudiéremos decir que esta fue una del as premisas fundamentales a la hora de establecer este estudio, pues el investigador forense debe, en la mayoría de los casos adquirir sus conocimiento a partir de la experiencia y esto definitivamente es algo sumamente productivo, pero la intención era establecer un marco de inicio a quien desease completar un investigación forense.

BIBLIOGRAFÍA

Apoyo para el análisis forense de computadoras, José Arquillo Cruz, Escuela Politécnica Superior de Jaén, Septiembre, 2007

Easey, Eoghan. Digital Evidence and Computer Crime, Third Edition: Forensic Science, Computers, and the Internet. Academic Press. p. 840. ISBN 978-0123742681.

Experiencias de análisis forense en México. Departamento de Seguridad en Cómputo / UNAM-CERT en Jornadas de Análisis Forense. Madrid, Septiembre 2005.

GUTIÉRREZ, Roberto. PÁRRIZAS, Ángel Alonso. Curso de Análisis Forense - TISSAT-24, 12 Enero 2005.

Seguridad en la red y análisis forense, Hades. Universidad de Murcia – Facultad de Informática. Murcia, España: Administración y seguridad en redes.

REFERENCIAS DOCUMENTALES ELECTRÓNICAS

Análisis forense digital. Encriptación de disco, burla de tecnología. Dragonjar [En Línea] Disponible en Internet en: <http://www.dragonjar.org/burlada-tecnologia-de-encriptacion-de-discos.xhtml >

Análisis forense Informático. Herramientas para análisis en Back Track Linux. Dragonjar [En Línea] Disponible en Internet en: <http://www.dragonjar.org/tag/herramientas>

Análisis forense Informático. Historia del Análisis forense digital [En Línea] Disponible en Internet en: <http://wwwdi.ujaen.es/~mlucena/bin/proy_forense.pdf>

Análisis Forense Y Crimen Electrónico. Técnicas y Herramientas para la Recolección de Datos [En Línea] Disponible en Internet en: <http://www.eficienciagerencial.com/tienda/temario/47.pdf>

Análisis forense. Cómo investigar un incidente de seguridad [En Línea] Disponible en Internet en: <http://www.k-nabora.com/index.php/blog/AnA-lisis-forense.-CA-mo-investigar-un-incidente-de-seguridad-248.html>

Autopsy en español, Alonso Eduardo Caballero Quezada, Noviembre 7 del año 2009, [en línea]. Disponible En: http://www.reydes.com/archivos/autopsy_reydes.pdf

Back Track Linux. BT – Penetration Testing Distribution [En Línea] Disponible en Internet en: <http://www.backtrack-linux.org/ >

Carrier, Brian. File System Forensic Analysis. Addison-Wesley, 2005
Code of practices for Digital Forensics. [on line] Disponible en internet en: http://cp4df.sourceforge.net/porque.html

Constitución política de Colombia. De la protección de la información y de los datos, Ley 1273 de 2009 [En Línea] Disponible en internet en: http://www.dmsjuridica.com/CODIGOS/LEGISLACION/LEYES/2009/LEY_1273_DE_2009.htm

Cyber Tools On-Line Search for Evidence[on line] Disponible en internet en: http://www.shellsec.net/articulo/CTOSE-un-proyecto-para-garantizar-la-seguridad-de-las-operaciones-electronicas/

Metodología Básica de Análisis Forense. Metodología de Análisis forense para responder a un incidente [En Línea] Disponible en Internet en: <http://www.dragonjar.org/metodologia-basica-de-analisis-forense-parte-1-de-4.xhtml>

National Institute of Standards and Technology. [on line] Disponible en internet en: http://www.nist.gov/public_affairs/general_information.cfm

Norma ISO 27001 [on line] Disponible en internet en: http://es.scribd.com/doc/25034834/NORMA-ISO27001

Request For Comments. [on line] Disponible en internet en: http://www.ietf.org/rfc/rfc3227.txt

Request for Comments: 3227 Guidelines for Evidence Collection and Archiving. [on line] Disponible en internet: http://www.ietf.org/rfc/rfc3227.txt

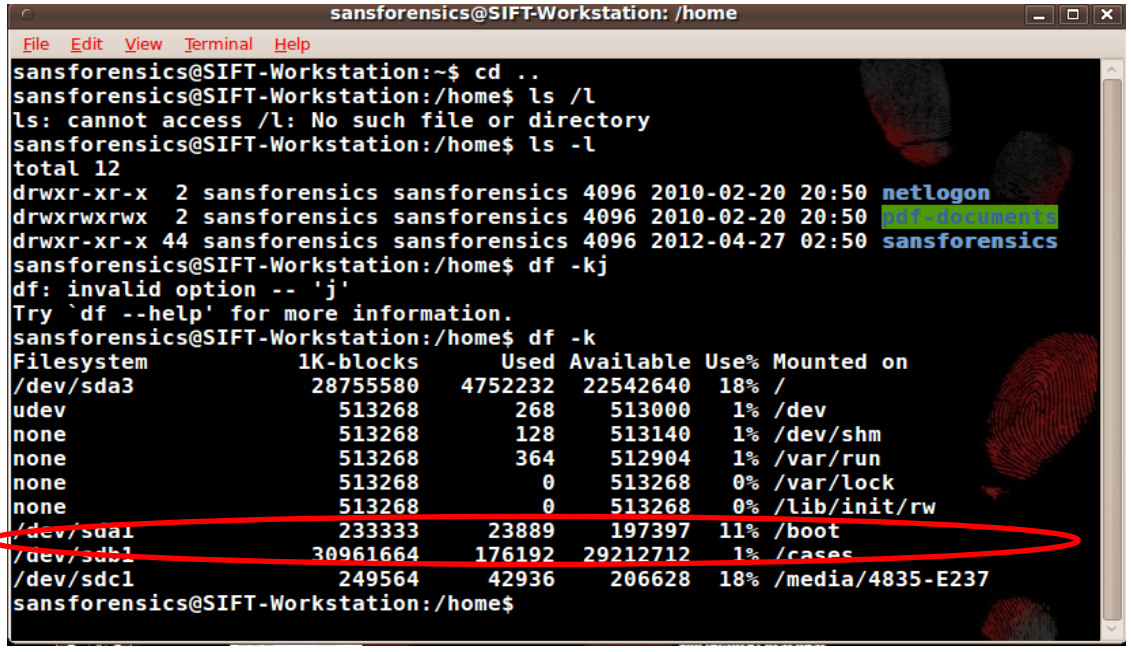
RequestForComments. [on line] Disponible en internet en: http://es.kioskea.net/contents/internet/rfc.php3. [citado el 29 de junio de 2011]

Soporte en línea de TSK [en línea]. Disponible En: http://wiki.sleuthkit.org

ANEXOS

Anexo 1. Análisis de sistema de archivos FAT

En el desarrollo del laboratorio utilizamos una memoria de tamaño 256MB que no fue montana en el sistema operativo de Windows para no dañar la evidencia. En el sistema Unix fue cargada en la unidad sdc1 tal como se muestra en la siguiente imagen.



Es necesario realizar la imagen de la memoria para lo cual creamos una carpeta para el desarrollo en del laboratorio (TALLER3).

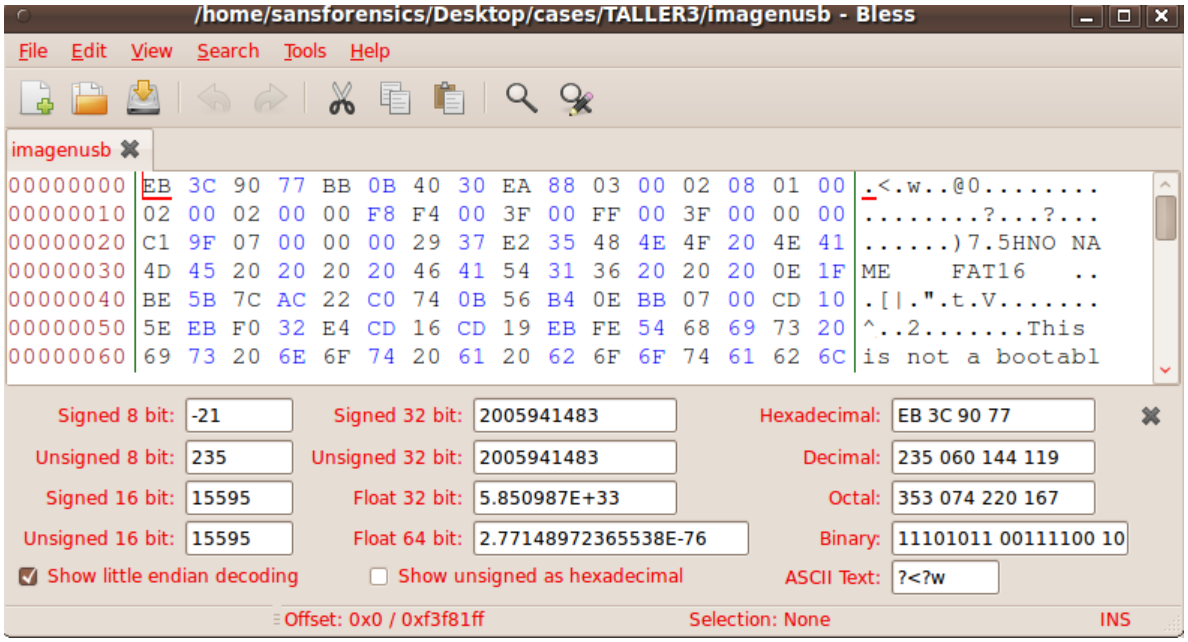
Nos ubicamos en la siguiente ruta: /home/forensics/Desktop/cases/TALLER3

```
sansforensics@SIFT-Workstation:~/Desktop/cases/TALLER 3$ cd ..
sansforensics@SIFT-Workstation:/cases$ ls -l
total 16
drwxrwxrwx 2 sansforensics sansforensics 4096 2011-07-22 17:38 EXAMPLE-DIR-YYYYM
MDD-###
drwx------ 2 root          root          4096 2012-04-16 14:41 lost+found
drwxr-xr-x 2 sansforensics sansforensics 4096 2012-04-28 13:29 TALLER3
drwxrwxrwx 2 sansforensics sansforensics 4096 2011-11-28 21:57 timeline-output-f
older
sansforensics@SIFT-Workstation:/cases$ cd TALLER3/
sansforensics@SIFT-Workstation:/cases/TALLER3$
```

Creamos la imagen con el nombre **imagenusb** con la siguiente la instrucción:

```
sansforensics@SIFT-Workstation:/cases/TALLER3$ sudo dd if=/dev/sdc1 of=imagenusb
499649+0 records in
499649+0 records out
255820288 bytes (256 MB) copied, 201.93 s, 1.3 MB/s
sansforensics@SIFT-Workstation:/cases/TALLER3$
```

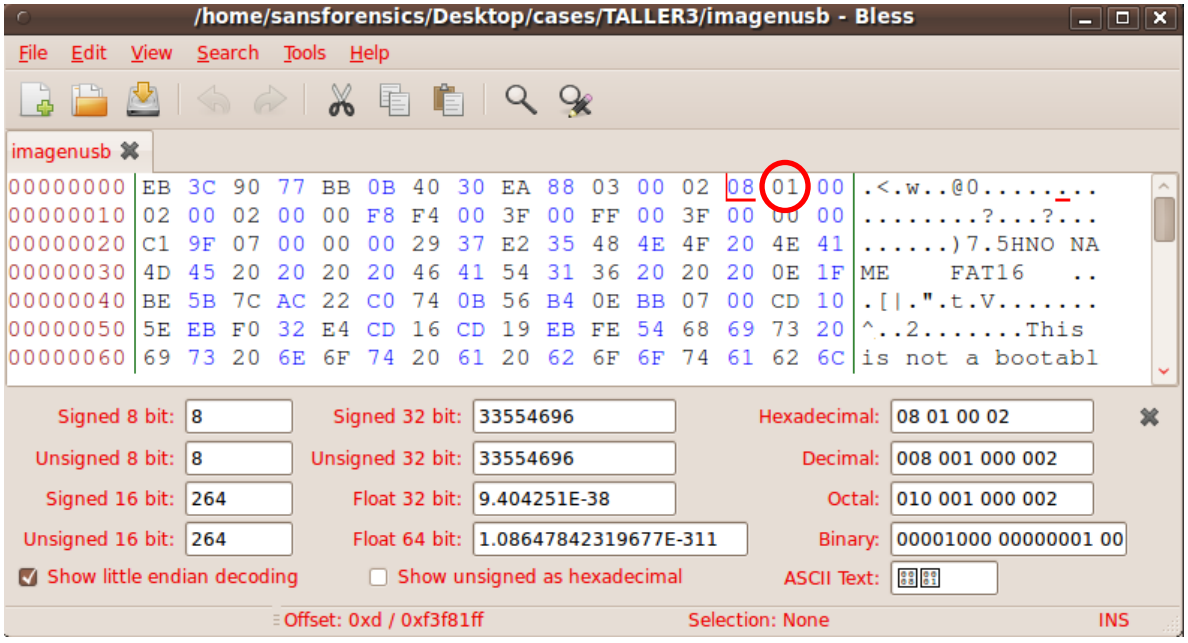
Abrimos la imagen creada con el editor Hexadecimal



Con lo anterior creamos y visualizamos la imagen creada de la memoria de 256 GB.

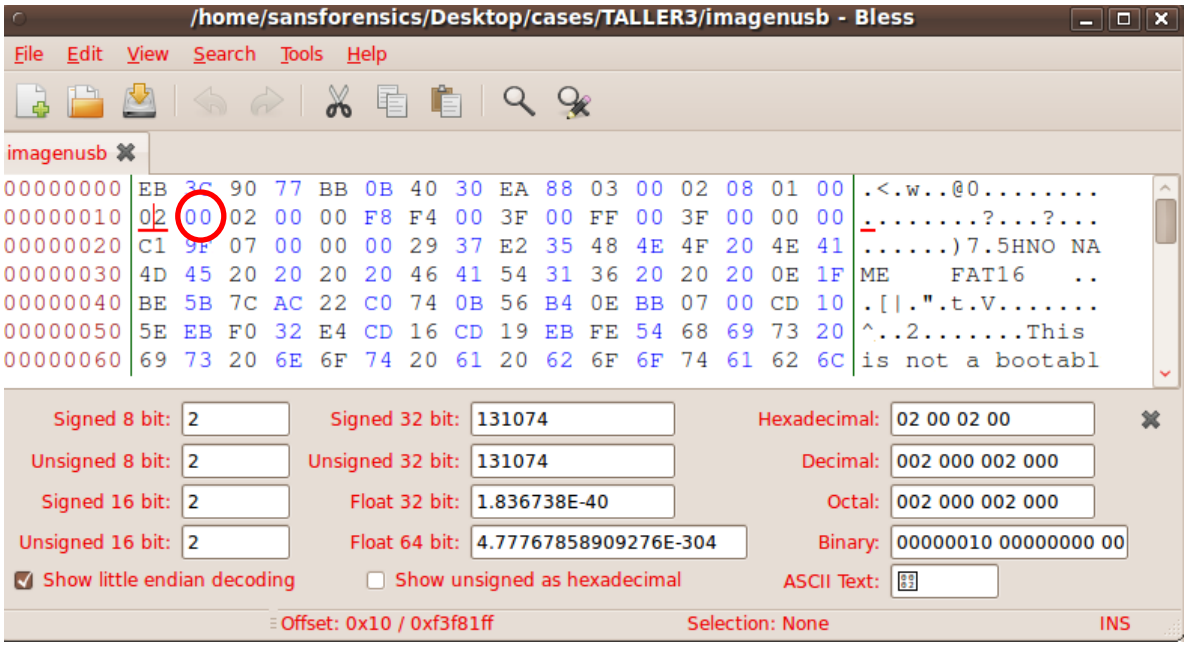
Identifiquemos el numero de sectores por cluster en el editor hexadecimal

El número de sectores por cluster son: 8

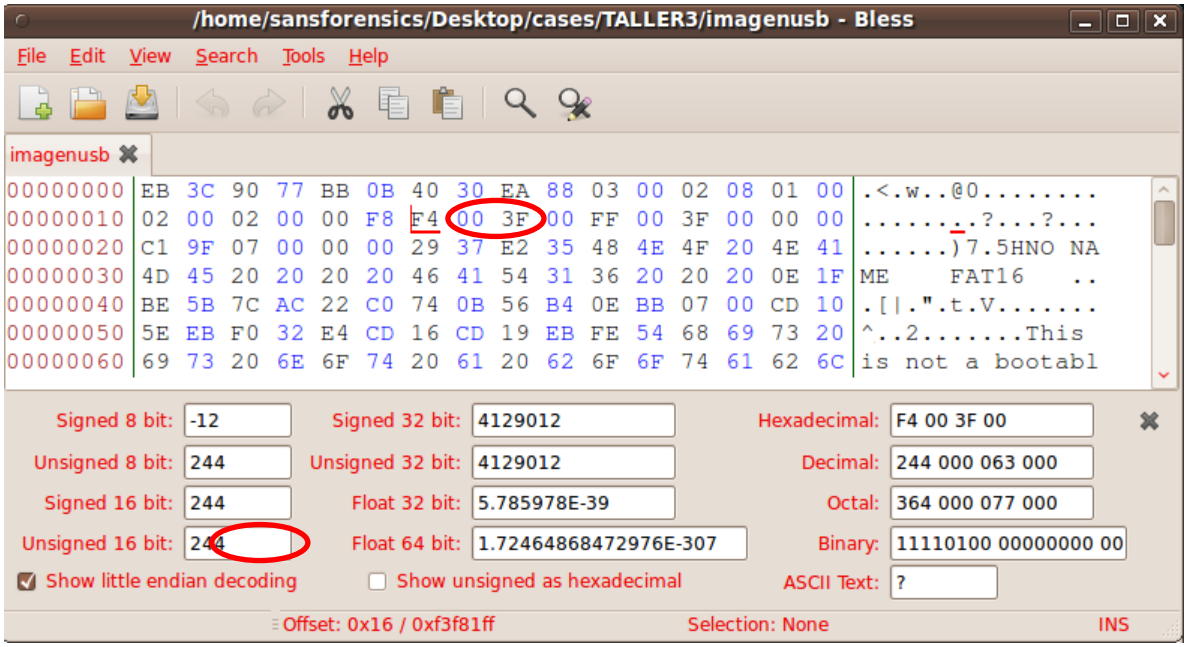


Identifiquemos la cadena de clústeres de 2 archivos

Inicialmente se identifican las tablas FAT que para nuestro caso fueron 2.

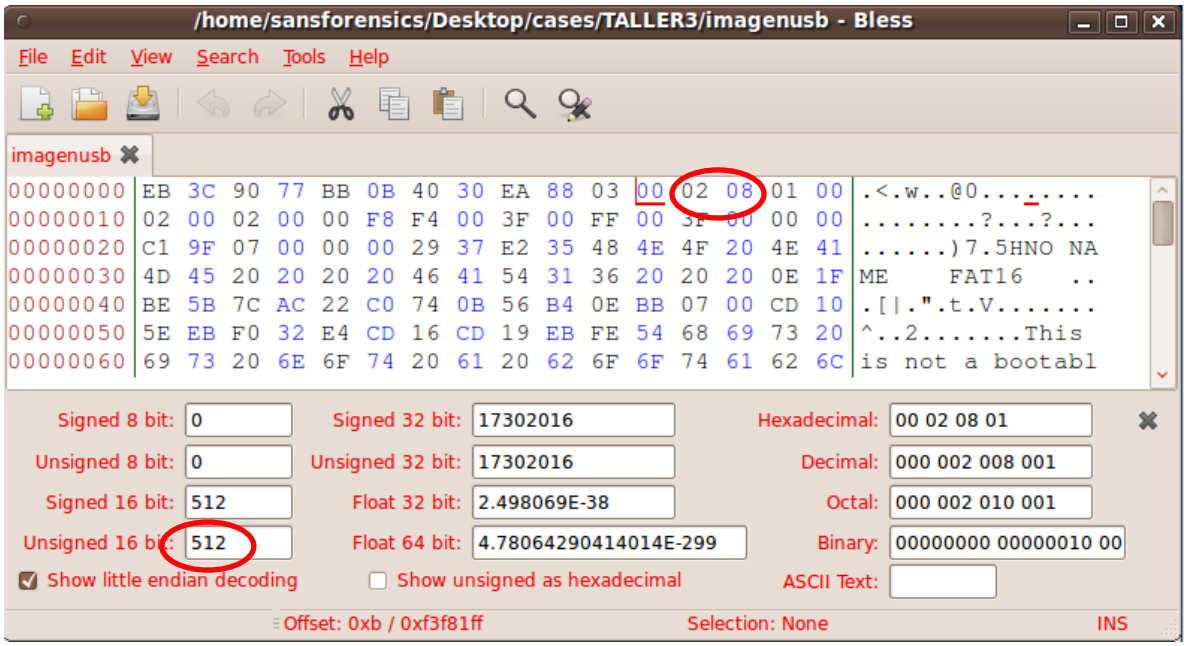


A continuación identificamos el tamaño del área FAT

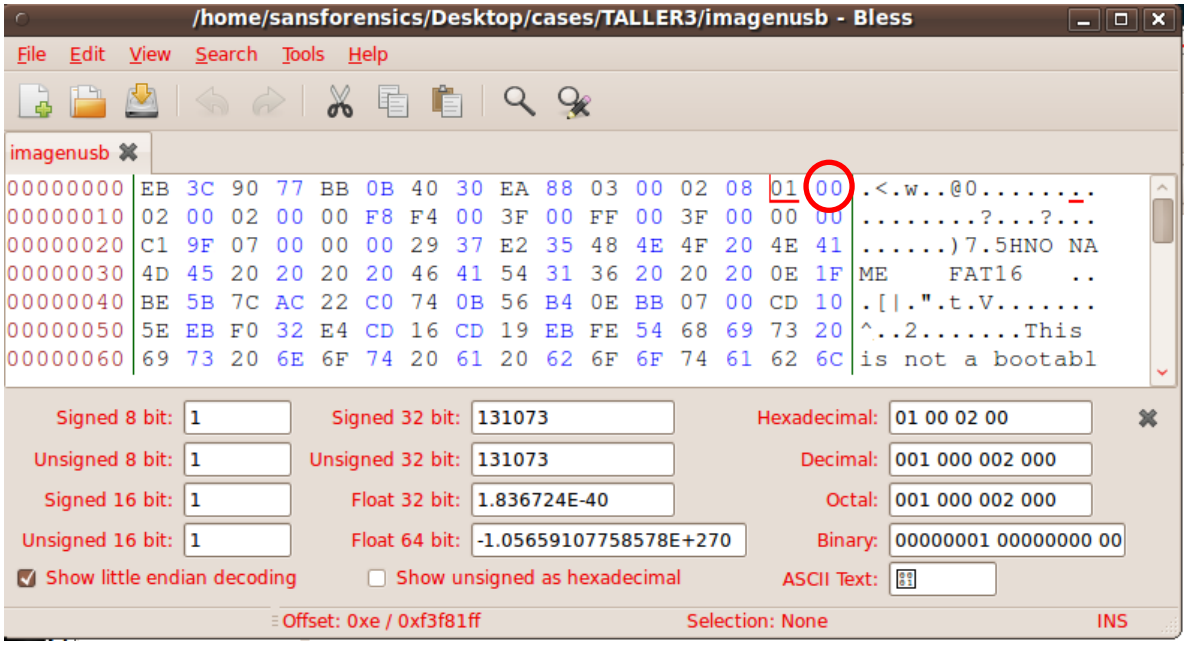


En este caso la FAT contiene 244 sectores.
Es decir un tamaño en bytes de 244*512=124928
Como se identificaron 2 tablas FAT se realiza el cálculo que es:
124928*2=249856 bytes

Identificamos el número de bytes por sector



Identificamos el número de sectores reservados que para este caso es 1

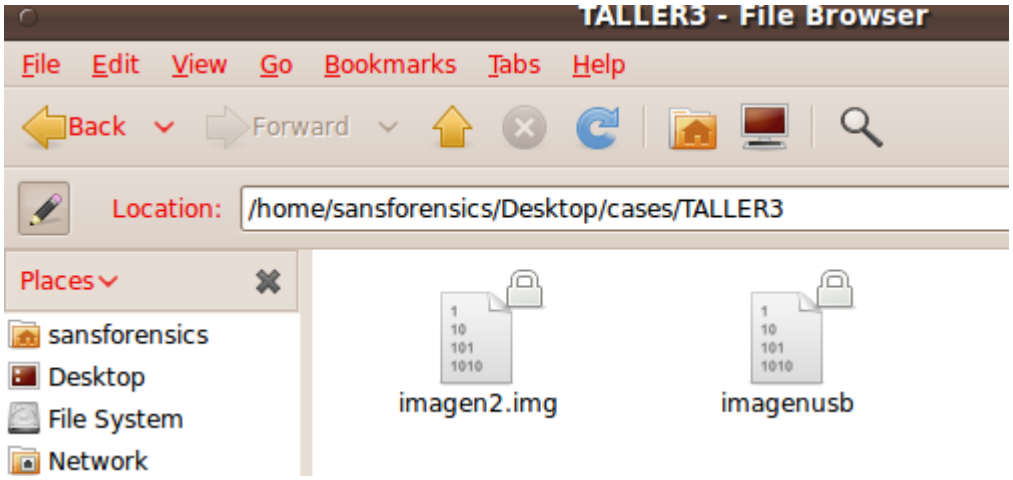


Con este valor calculamos el área reservada que para nuestro caso es de 512 con 1 sector reservado.

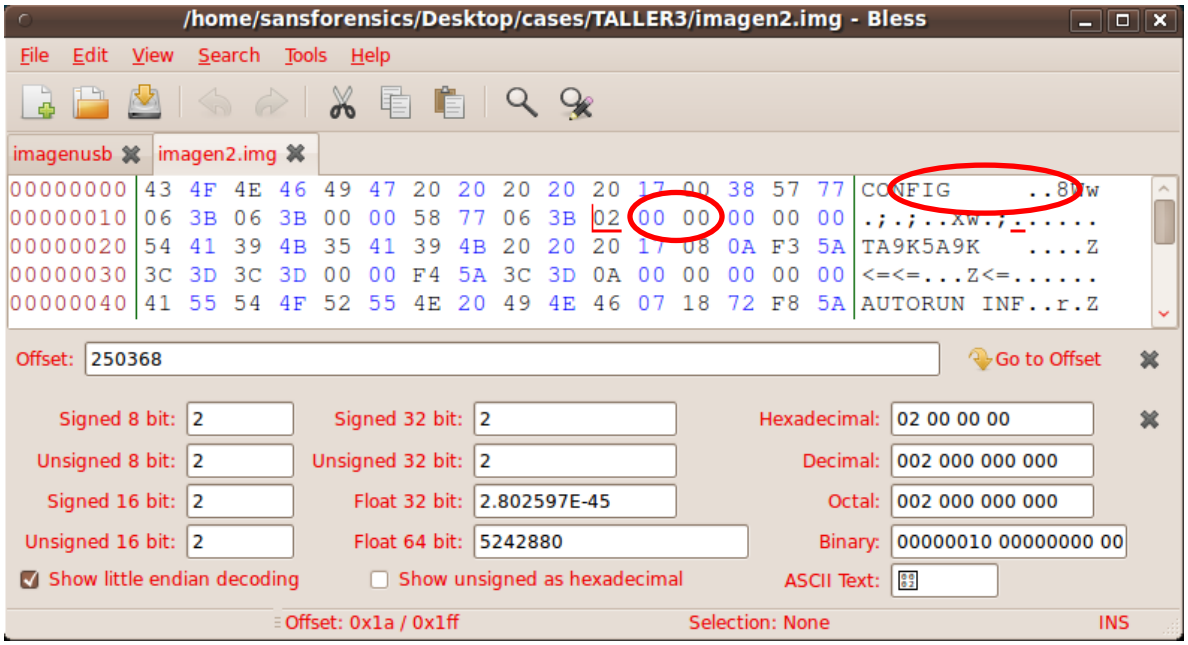
Área reservada= 512 bytes. Es necesario calcular el espacio ocupado en bytes del total de área reservada y de las tablas FAT

512 bytes + 249856 bytes= **250368 bytes**
En sectores serian=250880/512=**489 sectores**
Sacamos una imagen del espacio donde se encuentra el rootdirectory que para nuestro caso empieza en el sector 489

```
sansforensics@SIFT-Workstation:/cases/TALLER3$ sudo dd if=/dev/sdc1 of=imagen2.i
mg bs=512 count=1 skip=489
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0.00107763 s, 475 kB/s
sansforensics@SIFT-Workstation:/cases/TALLER3$
```



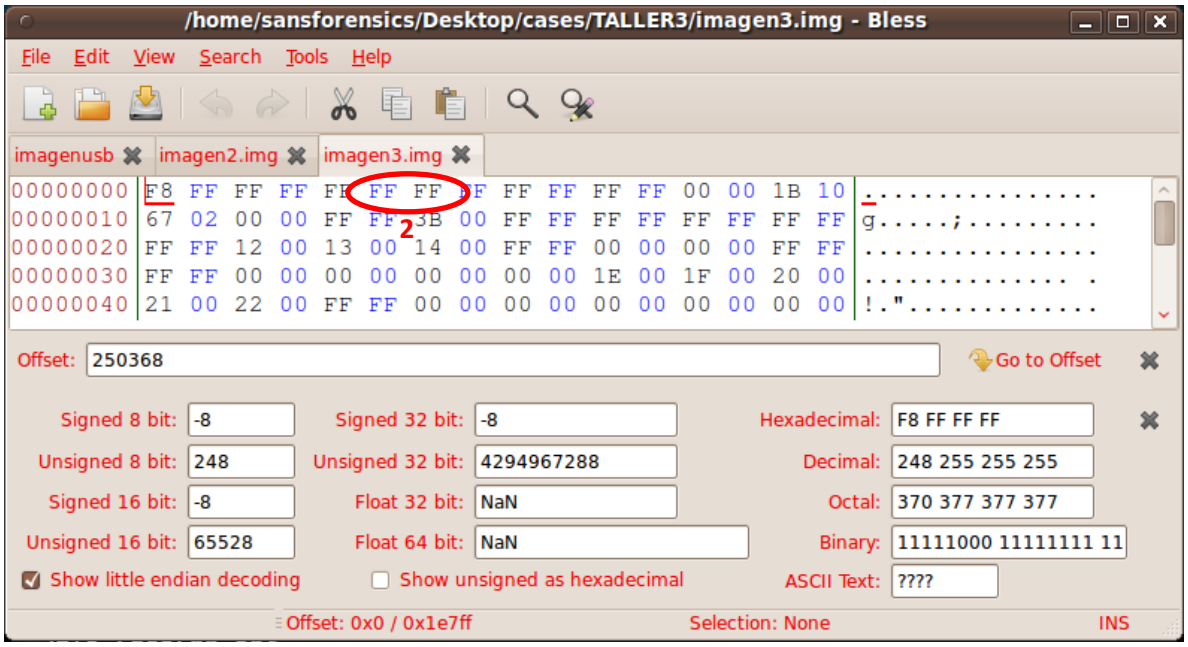
Al visualizar la imagen con el editor hexadecimal podemos ver las entradas de los archivos en el rootdirectory



Ahora para poder determinar la cadena de clustersquecomponenelarchivo hay queubicarlaentrada02delatablaFAT.ParaadquirirlatablaFATdebemosconocerelinicioyelfinal delatablaFAT.Elinicioesdespuésdelárea reservadaysehabíadefinidoqueestaocupaba1sector.Y porotroladoeltamañodelatabla FATes244sectores.Porlotantoseadquiereconelsiguientecomando.

```
sansforensics@SIFT-Workstation:/cases/TALLER3$ sudo dd if=/dev/sdc1 of=imagen3.i
mg bs=512 count=244 skip=1
244+0 records in
244+0 records out
124928 bytes (125 kB) copied, 0.0821802 s, 1.5 MB/s
sansforensics@SIFT-Workstation:/cases/TALLER3$
```

Buscarla cadenadeclustersdelarchivo CONFIG, lacualempiezaenelcluster2. Para nuestro caso únicamente utilizo un cluster.


/home/sansforensics/Desktop/cases/TALLER3/imagen3.img - Bless
File Edit View Search Tools Help
imagenusb
imagen2.img
imagen3.img
00000000 F8 FF FF FF FF FF FF FF FF FF FF FF 00 00 1B 10 ....
00000010 67 02 00 00 FF FF 3B 00 FF FF FF FF FF FF FF FF g.....;.........
00000020 FF FF 12 00 13 00 14 00 FF FF 00 00 00 00 FF FF ....
00000030 FF FF 00 00 00 00 00 00 00 00 1E 00 1F 00 20 00 ....
00000040 21 00 22 00 FF FF 00 00 00 00 00 00 00 00 00 00 !."..............
2
Offset: 250368
Go to Offset
Signed 8 bit: -8
Signed 32 bit: -8
Hexadecimal: F8 FF FF FF
Unsigned 8 bit: 248
Unsigned 32 bit: 4294967288
Decimal: 248 255 255 255
Signed 16 bit: -8
Float 32 bit: NaN
Octal: 370 377 377 377
Unsigned 16 bit: 65528
Float 64 bit: NaN
Binary: 11111000 11111111 11
Show little endian decoding
Show unsigned as hexadecimal
ASCII Text: ????
Offset: 0x0 / 0x1e7ff
Selection: None
INS

Anexo 2. Recuperación de archivos borrados memoria USB

Descomprimir la imagen enviada en formato .zip y se guarda en la carpeta cases
Ver las propiedades del sistema de archivos con el comando fsstat

```
CONTENT INFORMATION
------------------------------------------
Sector Size: 512
Cluster Size: 512
Total Cluster Range: 0 - 67582
Total Sector Range: 0 - 67582

$AttrDef Attribute Values:
$STANDARD_INFORMATION (16)   Size: 48-72   Flags: Resident
$ATTRIBUTE_LIST (32)   Size: No Limit   Flags: Non-resident
$FILE_NAME (48)   Size: 68-578   Flags: Resident,Index
$OBJECT_ID (64)   Size: 0-256   Flags: Resident
$SECURITY_DESCRIPTOR (80)   Size: No Limit   Flags: Non-resident
$VOLUME_NAME (96)   Size: 2-256   Flags: Resident
$VOLUME_INFORMATION (112)   Size: 12-12   Flags: Resident
$DATA (128)   Size: No Limit   Flags:
$INDEX_ROOT (144)   Size: No Limit   Flags: Resident
$INDEX_ALLOCATION (160)   Size: No Limit   Flags: Non-resident
$BITMAP (176)   Size: No Limit   Flags: Non-resident
$REPARSE_POINT (192)   Size: 0-16384   Flags: Non-resident
$EA_INFORMATION (208)   Size: 8-8   Flags: Resident
$EA (224)   Size: 0-65536   Flags:
$LOGGED_UTILITY_STREAM (256)   Size: 0-65536   Flags: Non-resident
```

Explorar la imagen con el comando ils, ver los atributos con fls y istat

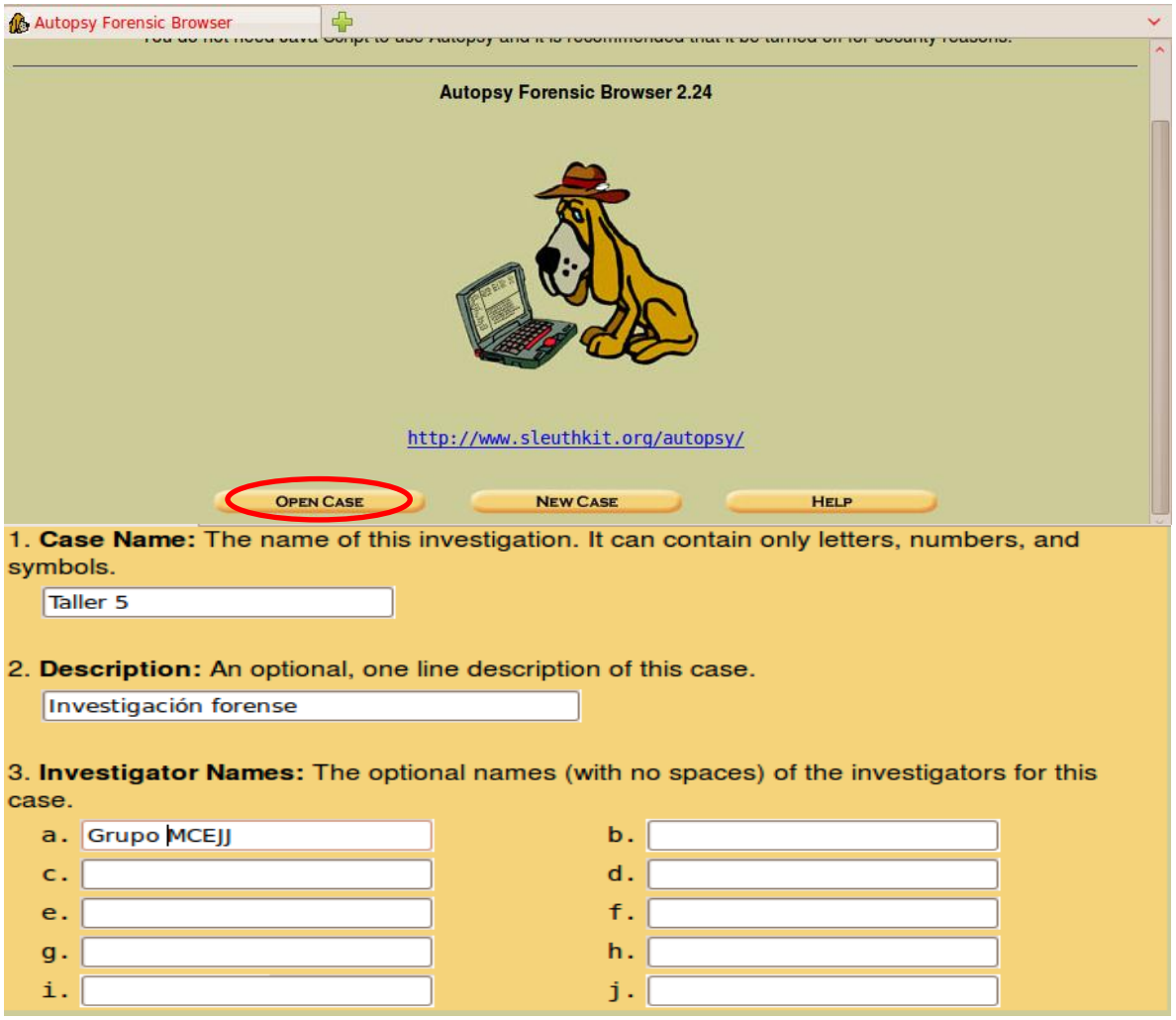
```
sansforensics@SIFT-Workstation:~/Desktop/cases/TALLER5$ ils ImgNTFS.img
class|host|device|start_time
ils|SIFT-Workstation||1336097871
st_ino|st_alloc|st_uid|st_gid|st_mtime|st_atime|st_ctime|st_crtime|st_mode|st_nl
ink|st_size
16|f|4294967295|0|1315582927|1315582927|1315582927|1315582927|555|0|0
17|f|4294967295|0|1315582927|1315582927|1315582927|1315582927|555|0|0
18|f|4294967295|0|1315582927|1315582927|1315582927|1315582927|555|0|0
19|f|4294967295|0|1315582927|1315582927|1315582927|1315582927|555|0|0
20|f|4294967295|0|1315582927|1315582927|1315582927|1315582927|555|0|0
21|f|4294967295|0|1315582927|1315582927|1315582927|1315582927|555|0|0
22|f|4294967295|0|1315582927|1315582927|1315582927|1315582927|555|0|0
23|f|4294967295|0|1315582927|1315582927|1315582927|1315582927|555|0|0
32|f|0|0|1314879703|1315583981|1315103979|1315583981|777|2|153664
33|f|0|0|1308100934|1315583981|1315105087|1315583981|777|2|577435
34|f|0|0|1314230388|1315583981|1315104059|1315583981|777|2|973647
35|f|0|0|1314592262|1315583981|1315103990|1315583981|777|2|3626858
36|f|0|0|1288713303|1315583981|1315105365|1315583981|777|2|136161
37|f|0|0|1314590537|1315583981|1315103999|1315583981|777|2|1656935
38|f|0|0|1315584113|1315584113|1315584113|1315584014|777|2|48
49|f|0|0|1265209122|1315584072|1301366803|1315584072|777|2|2890864
53|f|0|0|1257410307|1315584078|1301366805|1315584078|777|2|2884226
83|f|0|0|1315584146|1315584146|1315584146|1315584146|777|1|0
sansforensics@SIFT-Workstation:~/Desktop/cases/TALLER5$
```

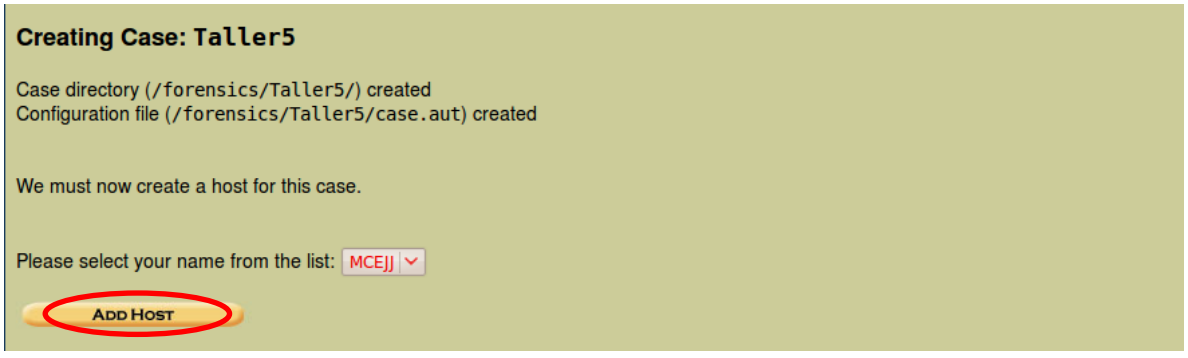
```
sansforensics@SIFT-Workstation:~/Desktop/cases/TALLER5$ istat ImgNTFS.img 16
MFT Entry Header Values:
Entry: 16        Sequence: 16
$LogFile Sequence Number: 0
Not Allocated File
Links: 0

$STANDARD_INFORMATION Attribute Values:
Flags: Hidden, System
Owner ID: 0
Security ID: 0  ()
Created:        Fri Sep  9 15:42:07 2011
File Modified:  Fri Sep  9 15:42:07 2011
MFT Modified:   Fri Sep  9 15:42:07 2011
Accessed:       Fri Sep  9 15:42:07 2011

Attributes:
Type: $STANDARD_INFORMATION (16-0)   Name: N/A   Resident   size: 48
```

```
sansforensics@SIFT-Workstation:~/Desktop/cases/TALLER5$ fls ImgNTFS.img
r/r 4-128-1:    $AttrDef
r/r 8-128-2:    $BadClus
r/r 8-128-1:    $BadClus:$Bad
r/r 6-128-1:    $Bitmap
r/r 7-128-1:    $Boot
d/d 11-144-2:   $Extend
r/r 2-128-1:    $LogFile
r/r 0-128-1:    $MFT
r/r 1-128-1:    $MFTMirr
r/r 9-128-2:    $Secure:$SDS
r/r 9-144-3:    $Secure:$SDH
r/r 9-144-4:    $Secure:$SII
r/r 10-128-1:   $UpCase
r/r 3-128-3:    $Volume
d/d 65-144-2:   .fseventsd
d/d 64-144-2:   .Trashes
r/r 67-128-2:   ._.Trashes
d/d 52-144-6:   sleuthkit-windows
d/d 27-144-6:   windows reference material
-/r * 32-128-4: Buenas_practicas_en_la_administracion_de_la_evidencia_digital_Do
c_GECTI 7.pdf
-/r * 33-128-4: HandbookOfCIS.pdf
-/r * 34-128-4: ProcedimientoGEneralCriminalistica.pdf
-/r * 35-128-4: Strengthening forensic science.pdf
-/r * 36-128-4: WritingReport.pdf
-/r * 37-128-4: 2011_CYBER_CRIME_survey.pdf
-/d * 38-144-7: WOanWareFOrensics
```

Montar la imagen “ImgNTFS.img” en el directorio compartido con Windows “cases” para su análisis correspondiente

Adicionar la imagen ImgNTFS.img en el software forense autopsy


Autopsy Forensic Browser
Autopsy Forensic Browser 2.24
http://www.sleuthkit.org/autopsy/
OPEN CASE
NEW CASE
HELP
1. Case Name: The name of this investigation. It can contain only letters, numbers, and symbols.
Taller 5
2. Description: An optional, one line description of this case.
Investigación forense
3. Investigator Names: The optional names (with no spaces) of the investigators for this case.
a. Grupo MCEJJ
b.
c.
d.
e.
f.
g.
h.
i.
j.


Creating Case: Taller5
Case directory (/forensics/Taller5/) created
Configuration file (/forensics/Taller5/case.aut) created
We must now create a host for this case.
Please select your name from the list: MCEJJ
ADD HOST

1. **Host Name:** The name of the computer being investigated. It can contain only letters, numbers, and symbols.

host1

2. **Description:** An optional one-line description or note about this computer.

MCEJJ

3. **Time zone:** An optional timezone value (i.e. EST5EDT). If not given, it defaults to the local setting. A list of time zones can be found in the help files.

America/bogota

4. **Timeskew Adjustment:** An optional value to describe how many seconds this computer's clock was out of sync. For example, if the computer was 10 seconds fast, then enter -10 to compensate.

-5

5. **Path of Alert Hash Database:** An optional hash database of known bad files.

Adding host: `host1` to case `Taller5`

Host Directory (`/forensics/Taller5/host1/`) created

Configuration file (`/forensics/Taller5/host1/host.aut`) created

We must now import an image file for this host

ADD IMAGE

Case: Taller5
Host: host1

No images have been added to this host yet

Select the Add Image File button below to add one

ADD IMAGE FILE CLOSE HOST

HELP

FILE ACTIVITY TIME LINES IMAGE INTEGRITY HASH DATABASES

VIEW NOTES EVENT SEQUENCER

1. **Location**
Enter the full path (starting with /) to the image file.
If the image is split (either raw or EnCase), then enter '*' for the extension.

ansforensics/Desktop/cases/TALLER5/ImgNTFS.img

2. **Type**
Please select if this image file is for a disk or a single partition.

○ Disk ◉ Partition

3. **Import Method**
To analyze the image file, it must be located in the evidence locker. It can be imported from its current location using a symbolic link, by copying it, or by moving it. Note that if a system failure occurs during the move, then the image could become corrupt.

○ Symlink ◉ Copy ○ Move

Local Name: images/ImgNTFS.img
Data Integrity: An MD5 hash can be used to verify the integrity of the image. (With split images, this hash is for the full image file)

○ Ignore the hash value for this image.
◉ Calculate the hash value for this image.
○ Add the following MD5 hash value for this image:

☐ Verify hash after importing?

File System Details

Analysis of the image file shows the following partitions:

Partition 1 (Type: ntfs)
Mount Point: C: File System Type: ntfs

Calculating MD5 (this could take a while)
Current MD5: DCD224E7AE90E2CD9C0BA3DEDAB1B46D
Testing partitions
Copying image(s) into evidence locker (this could take a little while)
Image file added with ID img1

Volume image (0 to 0 - ntfs - C:) added with ID vol1

OK ADD IMAGE

Case: Taller5
Host: host1

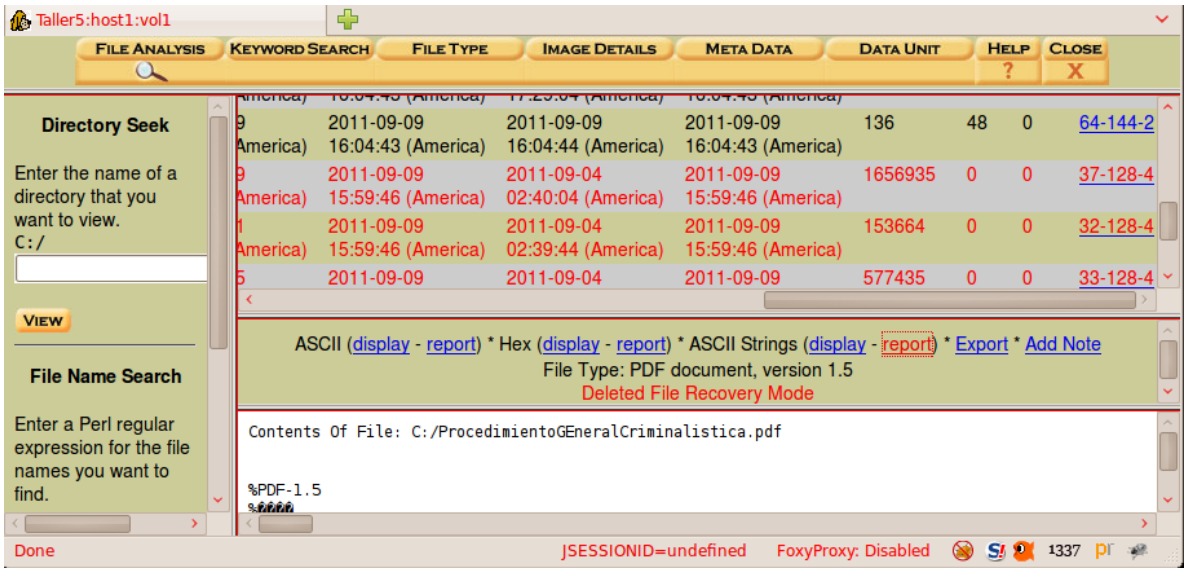
Select a volume to analyze or add a new image file.

CASE GALLERY HOST GALLERY HOST MANAGER

	mount	name	fs type	
◉	C:/	ImgNTFS.img-0-0	ntfs	details

ANALYZE ADD IMAGE FILE CLOSE HOST
HELP

FILE ACTIVITY TIME LINES IMAGE INTEGRITY HASH DATABASES
VIEW NOTES EVENT SEQUENCER



Recuperar un archivo borrado y ver sus atributos con istat

```
sansforensics@SIFT-Workstation:~/Desktop/cases/TALLER5$ ls
ImgNTFS.img  vol1-C..WritingReport.pdf
sansforensics@SIFT-Workstation:~/Desktop/cases/TALLER5$
```

Se visualizan los atributos del archivo con istat

Anexo 3. Reto forense flisol 2010

Su misión es analizar un disco flexible recuperado y responder las preguntas formuladas. Se necesita leer el reporte antes de continuar el reto. Como una investigación del mundo real se necesita tener alguna información adicional y alaguna evidencia, pero es la persona y sus conocimientos lo que responderán las preguntas.

Nombre del Archivo: image.zip

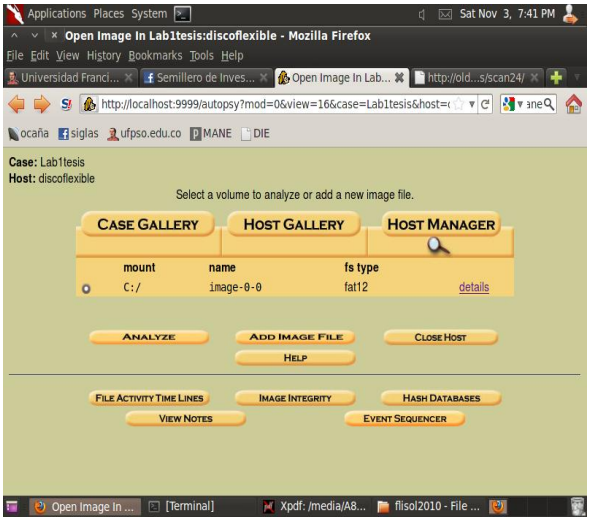
Hash MD5 del Archivo: b676147f63923e1f428131d59b1d6a72

Preguntas:

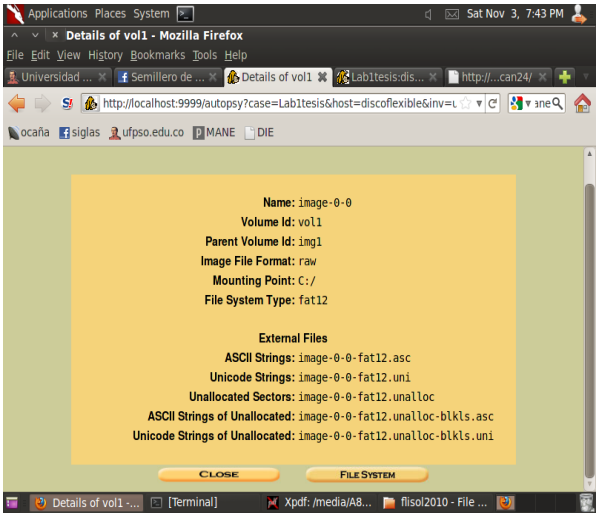
¿Quién es el proveedor de marihuana de Joe Jacobs y cual es la dirección del proveedor?
¿Qué dato crucial está disponible dentro de coverpage.jpg y porque el dato es crucial?
¿Qué (si hay) otras escuelas vecinas a Snith Hill Joe Jacobs frecuenta?
Para cada archivo, que procesos hizo el sospechoso para enmascarar de otros.
¿Qué procesos (usted como analista) realizó para examinar el contenido completo de cada archivo?

Verificación hash MD5:

Creación del caso.



Creación de indices de busqueda.



Tras analizar los archivos no se reconoce el “coverpage” como .JPG

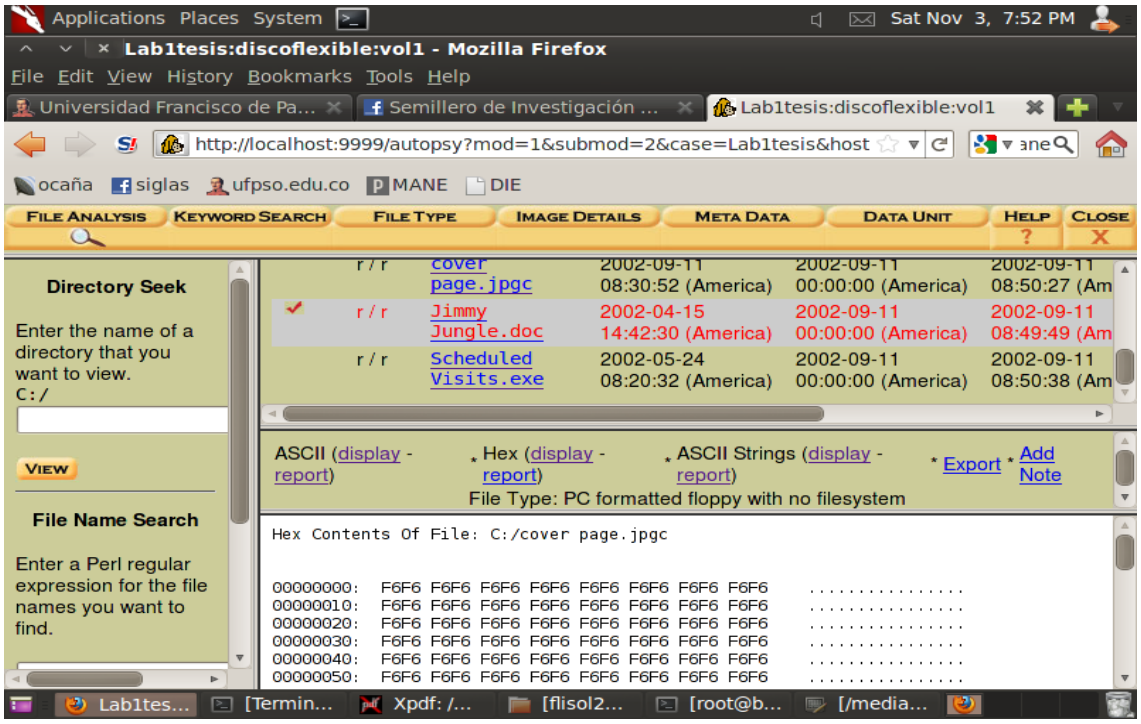
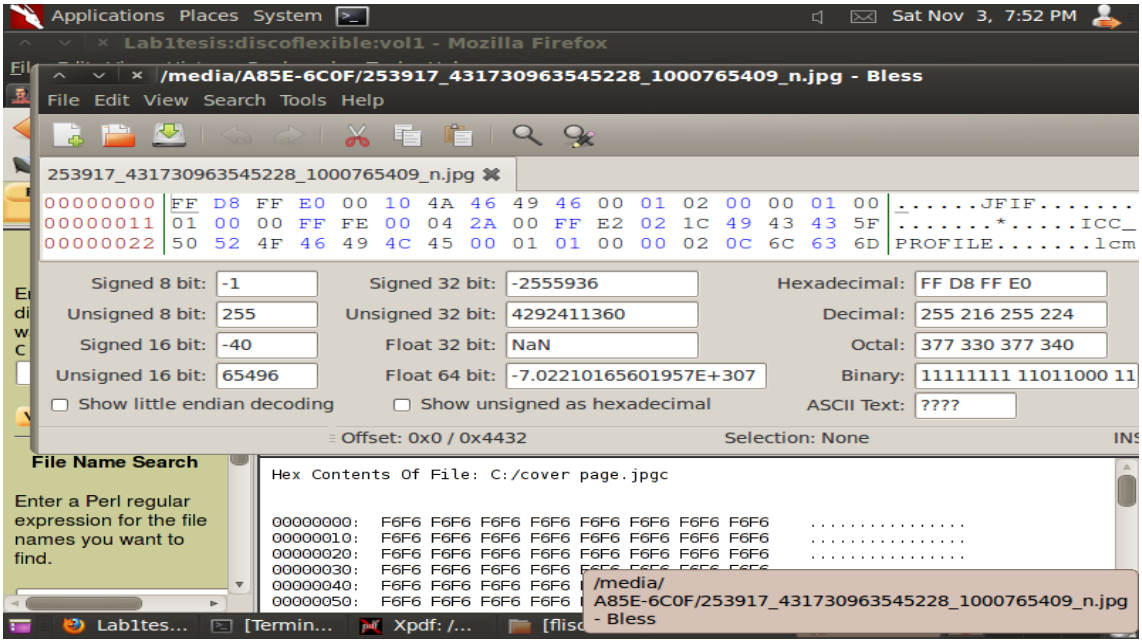
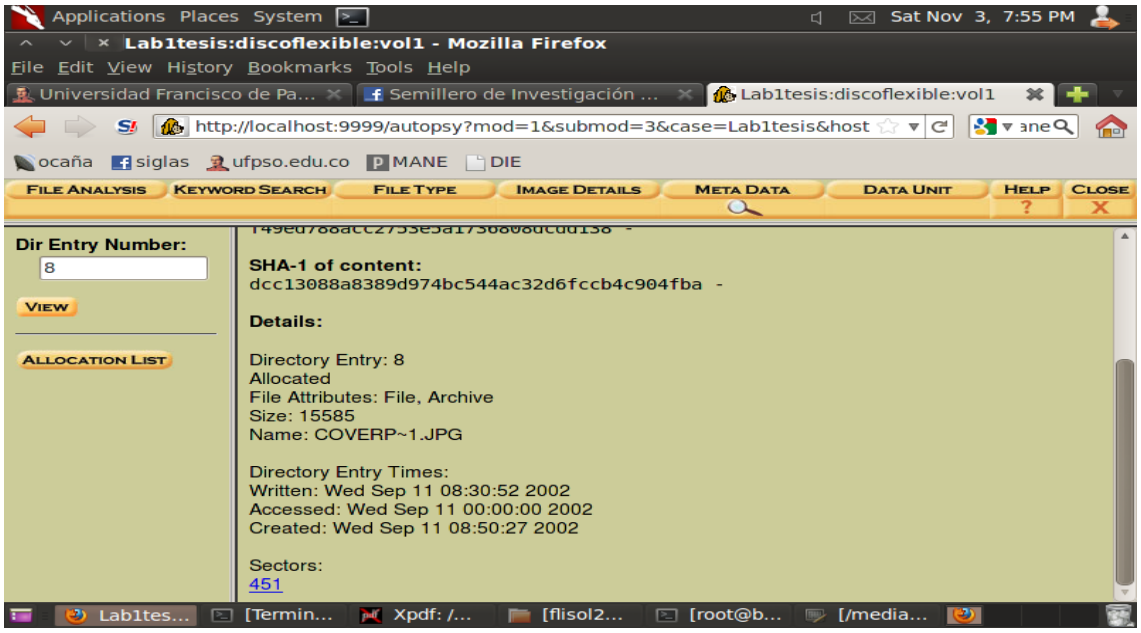


Figura: Comparación de la cabecera hexadecimal de un verdadero archivo JPG y la del "coverpage".



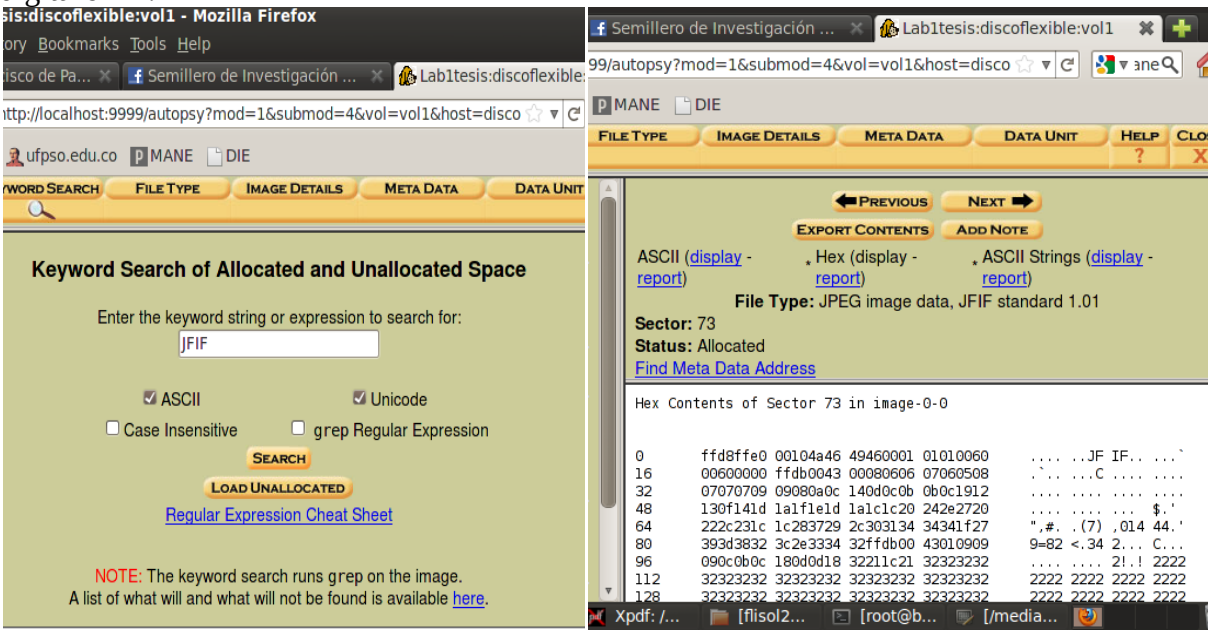
El tamaño no coincide con los sectores asignados.



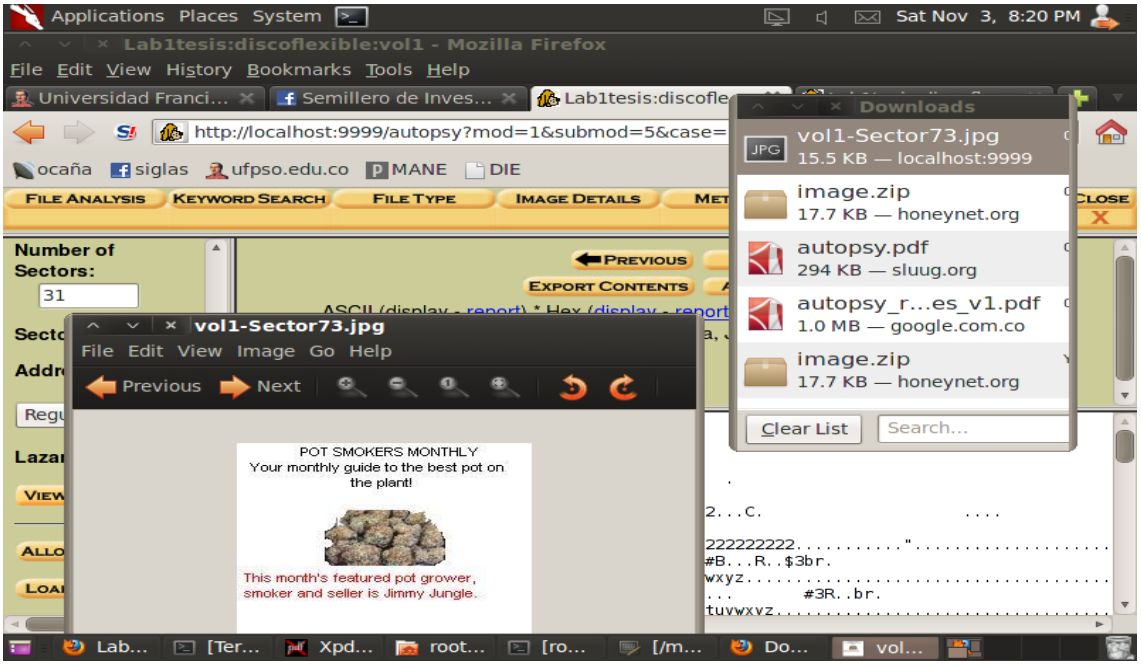
Como se observa en la imagen anterior el archivo “coverpage” tiene un tamaño de 15585 bytes pero solo se le asigna un sector (451) de 512 bytes de tamaño, lo cual no es consistente pues requería de 31 sectores (15585 / 512 = 31).

Búsqueda de palabra clave por la firma digital JFIF.

Figura: Coincidencia en el sector 73.



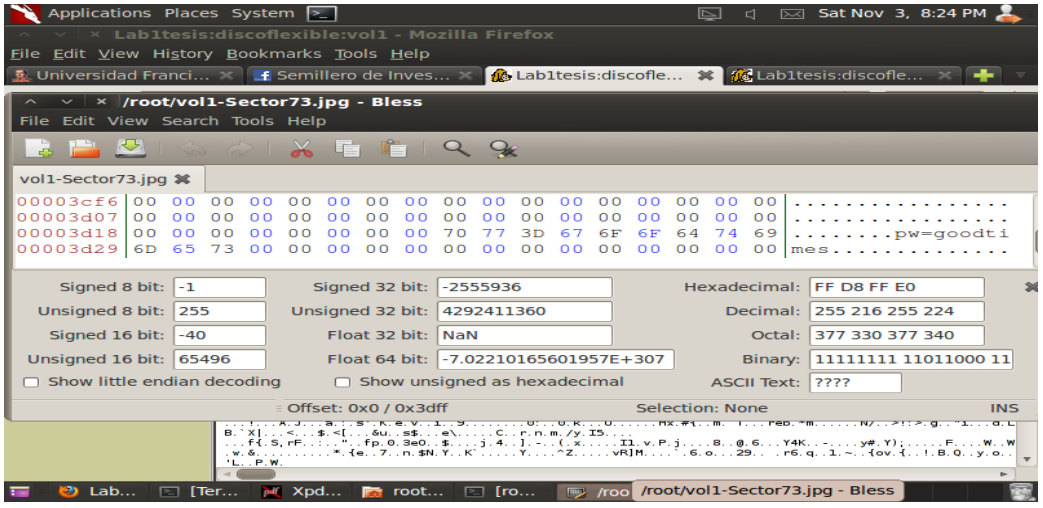
Se exporta el contenido contando desde el sector 73 hasta el 103 (31 sectores)



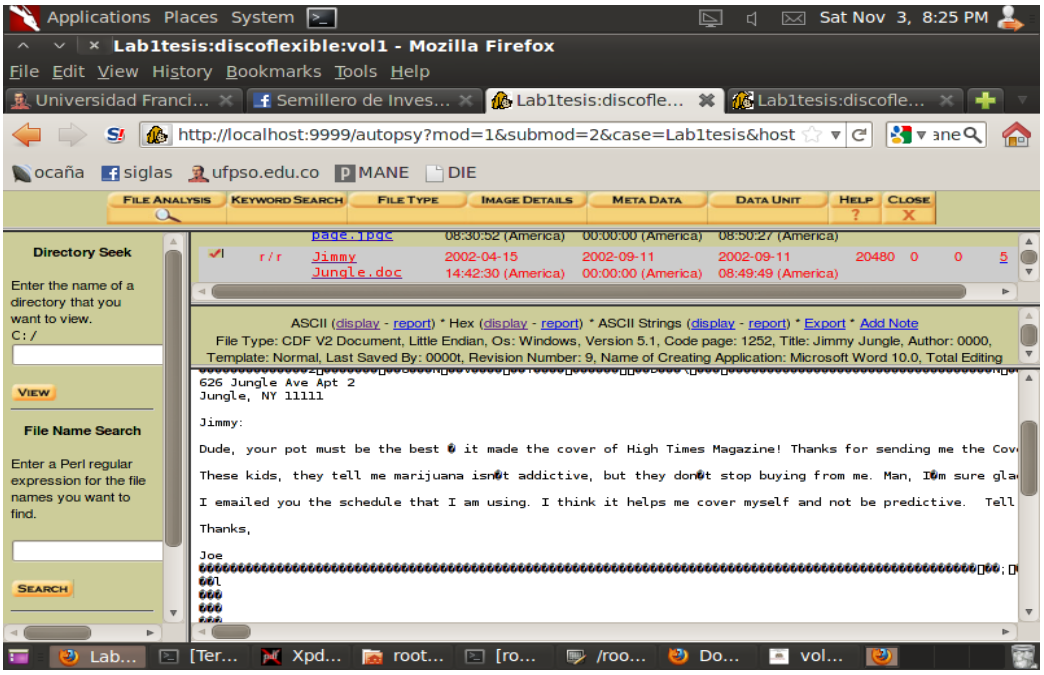
Se necesitan 31 sectores para almacenar 15585 bytes. Pero están asignados (36 ectores) del 73 hasta el 108. Pero solo 31 están asociados con el archivo; como se verifica más adelante; dado que la 104 y 105 están asignados a otro archivo. Por tal motivo también se extraerá los 36 sectores mediante la sentencia: **dd skip=73 bs=512 count=36 if=/media/hda3/image.dd of=/media/hda3/coverpage.jpg**

Se visualizarán en un editor hexadecimal, encontrando el texto ‘pw=goodtimes’.

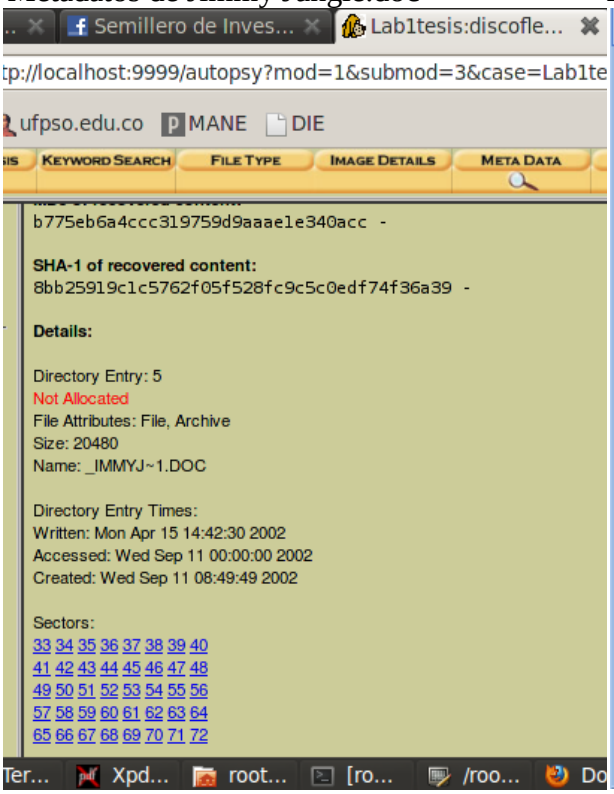
Visualización en editor hexadecimal del archivo extraído



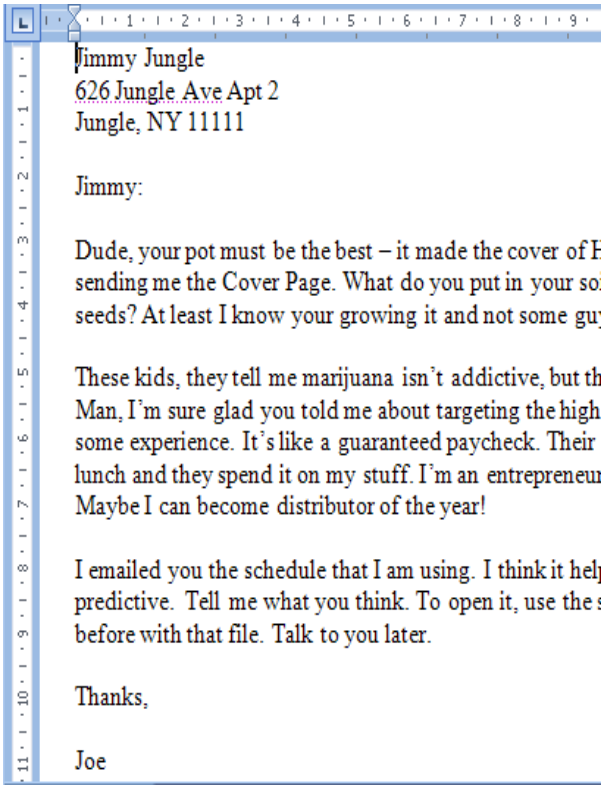
Se reconoce a Jimmy Jungle como un documento .DOC



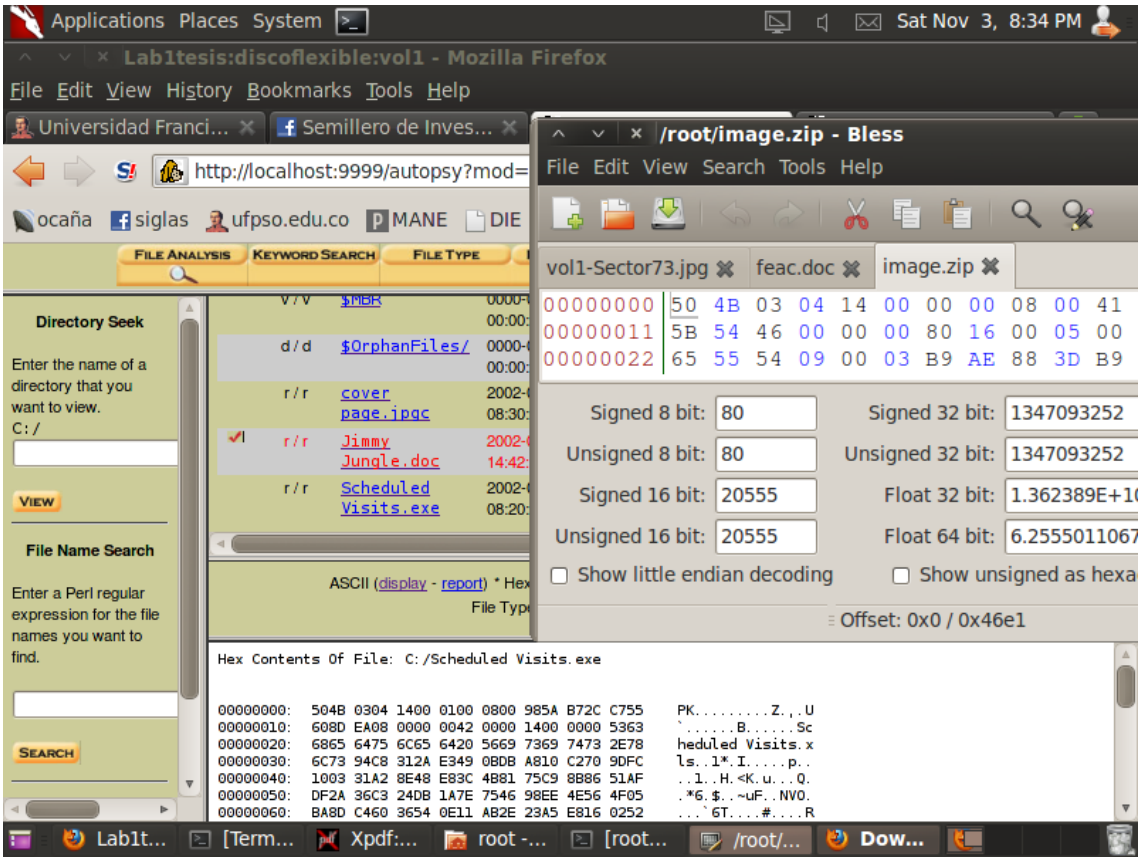
Metadatos de Jimmy Jungle.doc



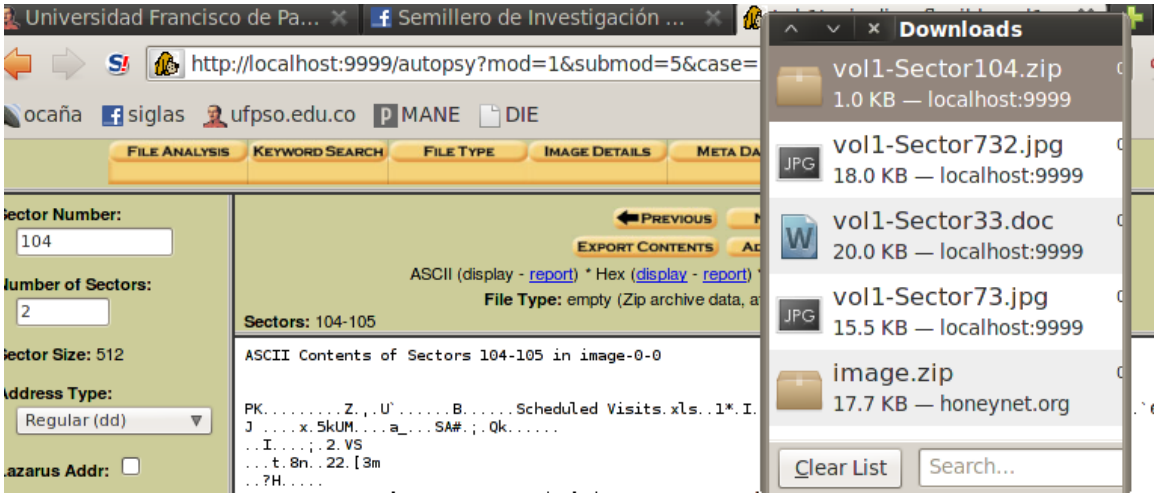
Extracción del archivo



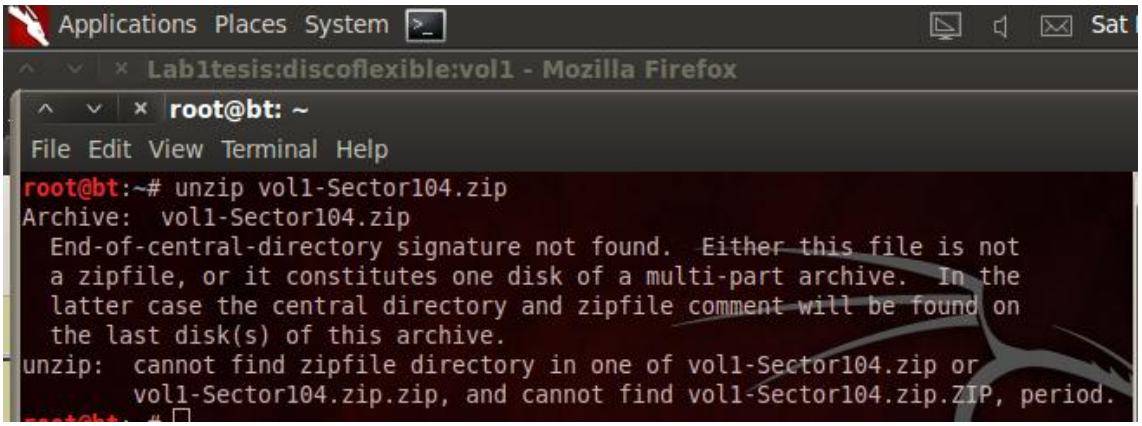
Visualizando el archivo Scheduled Visits.exe y se reconoce como .ZIP



Se extraen 104 y 105 asignados al Scheduled Visits.exe

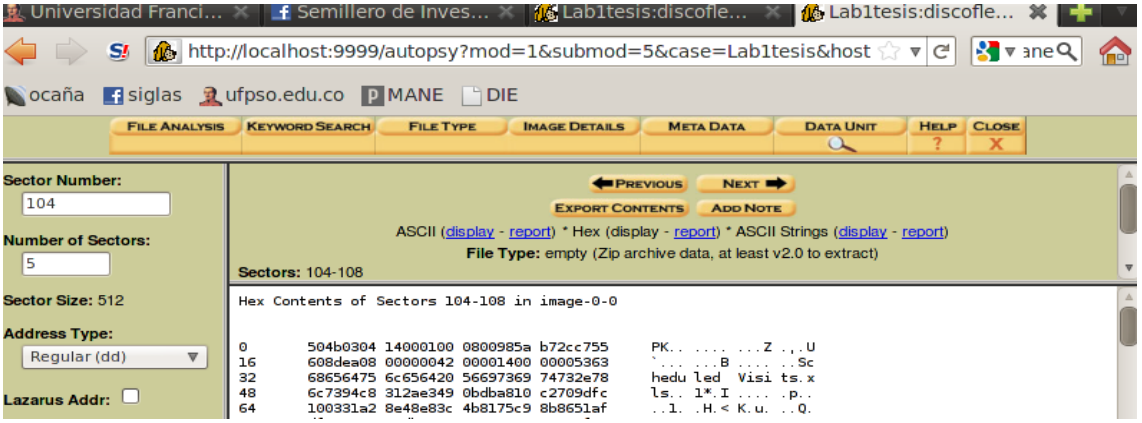


Al intentar descomprimir se observa un error. El archivo está incompleto.

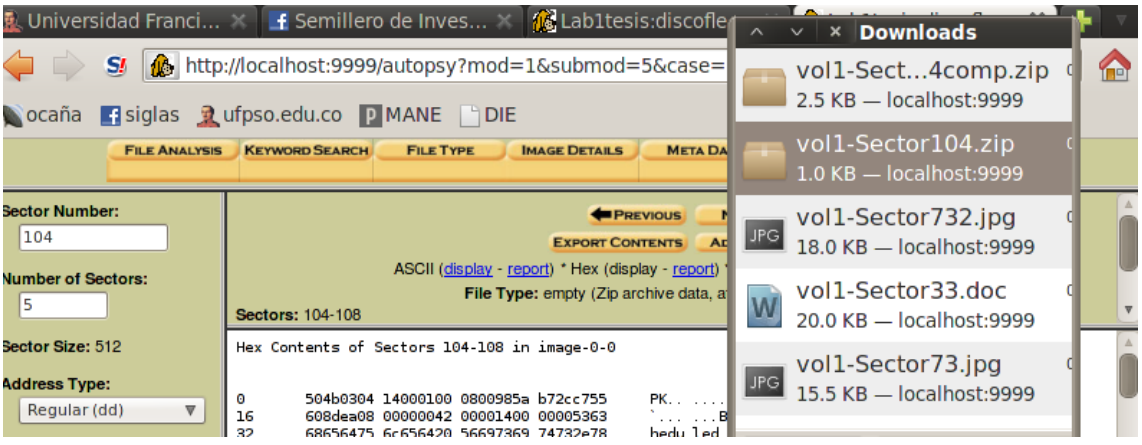


Laboratorios investigador

Se analizan los sectores del 104 al 108 y todos están asignados.



Se extrae el contenido de los sectores.



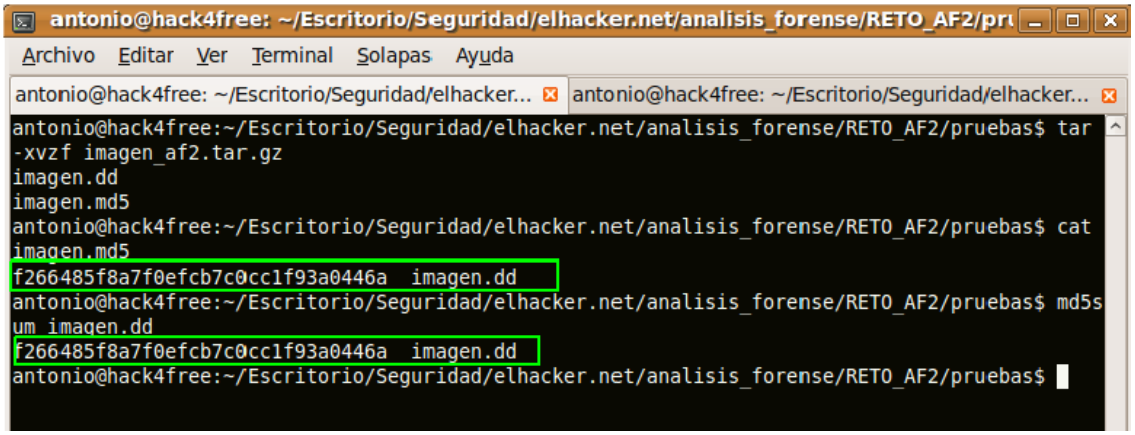
Contenido del archivo .ZIP.

	A	B	C
1	**Month**	**DAY**	**HIGH SCHOOLS**
2	2002		
3	April	Monday (1)	Smith Hill High School (A)
4		Tuesday (2)	Key High School (B)
5		Wednesday (3)	Leetch High School (C)
6		Thursday (4)	Birard High School (D)
7		Friday (5)	Richter High School (E)
8		Monday (1)	Hull High School (F)
9		Tuesday (2)	Smith Hill High School (A)
10		Wednesday (3)	Key High School (B)
11		Thursday (4)	Leetch High School (C)
12		Friday (5)	Birard High School (D)
13		Monday (1)	Richter High School (E)
14		Tuesday (2)	Hull High School (F)
15		Wednesday (3)	Smith Hill High School (A)
16		Thursday (4)	Key High School (B)
17		Friday (5)	Leetch High School (C)
18		Monday (1)	Birard High School (D)
19		Tuesday (2)	Richter High School (E)
20		Wednesday (3)	Hull High School (F)
21		Thursday (4)	Smith Hill High School (A)
22		Friday (5)	Key High School (B)
23		Monday (1)	Leetch High School (C)
24		Tuesday (2)	Birard High School (D)

El documento recuperado indica que el proveedor de JOE JACOBS es Jimmy Jungle, y su residencia es 626 Jungle Ave, Apt 2, Jungle, NY 11111. Así mismo la cadena "PW=GOODTIMES" es fundamental dentro de la investigación pues permite extraer el listado que muestra las otras escuelas donde se distribuye marihuana.

Anexo 4. Reto forense 2 del hacker.net

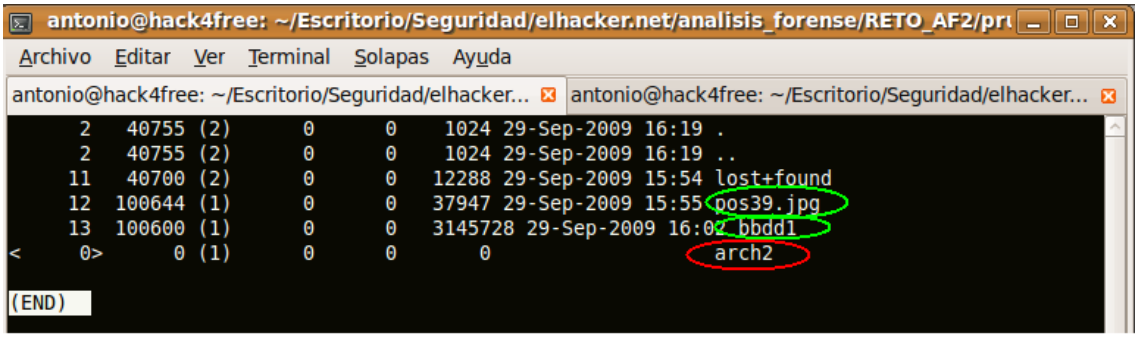
Lo primero que hacemos es descomprimir el zip y comparar que coinciden los checksum:



A continuación procederemos a ver que hay dentro de este sistema de ficheros, no hace falta montarlo, con *debugfs* podemos averiguarlo fácilmente con el siguiente comando:

 # debugfs -w imagen.dd

Y una vez dentro, aplicando el comando *ls -ld*:

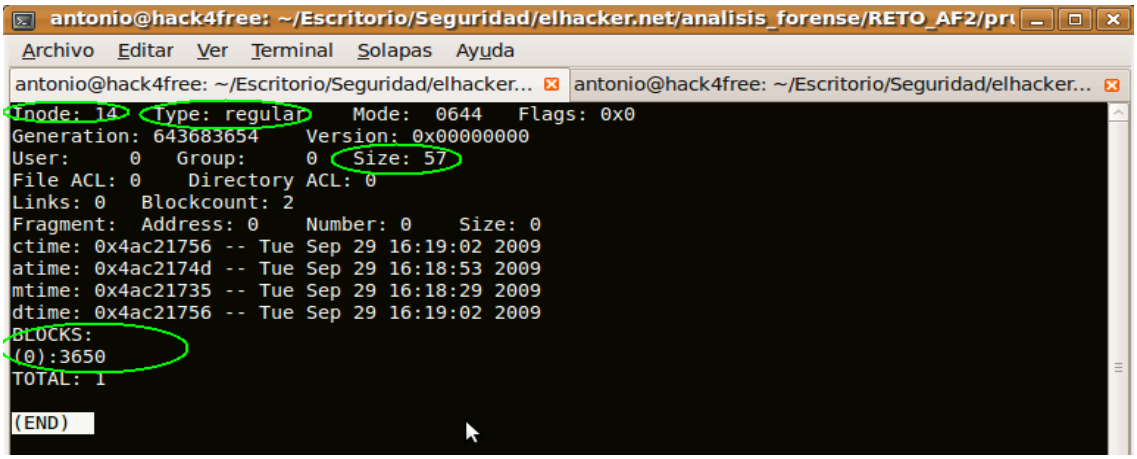


Hay un archivo que en primera instancia parece una imagen jpg

Un archivo que por el nombre parece una base de datos

Un fichero que ha sido borrado y que procederemos a rescatar

El número de i-nodos va en orden: 11, 12, 13,... El archivo borrado esta linkado con el i-nodo 14, así que probemos a listar la información de este i-nodo, de nuevo con ayuda de debugfs:



Se ve que estábamos en lo cierto, el i-nodo nos indica que este archivo, si no ha sido sobreescrito, se encuentra en el bloque 3650. Con toda esta información que hemos recopilado, y con ayuda del mismo *debugfs* podemos recuperar los archivos, inclusive el borrado, os dejo de vuestra mano que investiguéis como se lleva esto a cabo.

Una vez que tenemos los 3 archivos en nuestra carpeta de trabajo, echémosle un vistazo por orden.

El archivo borrado, contiene la siguiente frase:

"Hay que ser precavido y Encriptar las cosas de Verdad..."

Como bien indicaron en el foro, si pensamos un poco y le buscamos las vueltas, esto probablemente indique que en algún momento dado, se ha utilizado TrueCrypt, así que nos lo apuntamos por si más adelante nos pudiera ser de utilidad.

El siguiente archivo es una imagen, más en concreto un problema de ajedrez.
Lo más probable es que esta esconda algo, mediante técnicas de Steganografía. Vamos a intentar ver qué pasa si usamos uno de los programas más conocidos bajo Linux para estos menesteres.

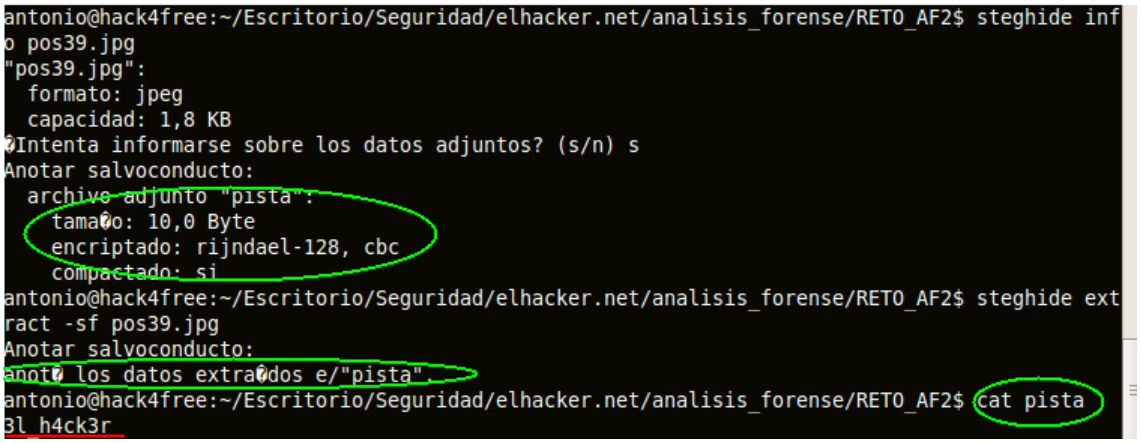
```
antonio@hack4free:~/Escritorio/Seguridad/elhacker.net/analisis_forense/RETO_AF2$ steghide inf
o pos39.jpg
"pos39.jpg":
  formato: jpeg
  capacidad: 1,8 KB
¿Intenta informarse sobre los datos adjuntos? (s/n) s
Anotar salvoconducto:
steghide: ¿no pude extraer ningún dato con ese salvoconducto!
antonio@hack4free:~/Escritorio/Seguridad/elhacker.net/analisis_forense/RETO_AF2$
```

Vemos que nos pide un salvoconducto, introducimos uno cualquiera y nos dice que naranjas de la china..

Pero claro, no iba a ser tan fácil, no?. Si pensamos un poco, la imagen es un problema de ajedrez, si hacemos una búsqueda del nombre del archivo tal cual, podemos llegar fácilmente a la página de donde ha sido sacado, y en la que si buscamos un poco, encontraremos la solución al problema de ajedrez (Dxc4, muy bonita por cierto), en caso de que no lo encontrásemos tenemos 2 opciones:
Resolverlo nosotros mismos

Usar un motor de ajedrez, que en un periquete nos dirá la jugada

Probemos ahora con este salvoconducto...



Vemos que había escondido un archivo llamado pista, y que contiene una cadena de text: **3l_h4ck3r**

El tercer archivo, parece un montón de bits sin sentido, si le hacemos un *file*, no nos aporta mucho, así que, si unimos todos los cabos sueltos... seguramente tengamos razón y esto no sea más que un contenedor de un volumen cifrado con TrueCrypt, será la cadena que conseguimos antes la contraseña para abrir este volumen?