ANALISIS FORENSE CON DISTRIBUCIONES GNU/LINUX

Autor: Manuel García Chaves
Consultor: Joaquín López Sánchez-Montañés
Tutor: Ignacio Salazar Ferrero
Plataforma GNU/LINUX
Curso: 2015-2016

INDICE

1.- Introducción

1.1 Contexto

El análisis forense es una disciplina perteneciente a la seguridad informática surgida como consecuencia de los numerosos y variados *incidentes de seguridad.* En esta rama de la seguridad informática tiene la peculiaridad de que la fase de análisis se realiza una vez que se ha producido el hecho. Para ello se tratara de reconstruir los pasos seguidos por los *atacantes* y estudiar a posteriori como se ha vulnerado o penetrado el sistema.

Actualmente los procesos de análisis de este tipo no están estandarizados ni siguen una pauta o pasos descritos en ningún documento, ni de ámbito nacional ni internacional. Aun así, distintas organizaciones en los que recogen alguna pautas y consejos para llevar a cabo este tipo de investigaciones sin que ello suponga una obligación en los procedimientos ni una garantía ante ningún tribunal. Por todo lo anterior plantearemos una guía con las mejores practicas a la hora de realizar un análisis forense.

1.2 Objetivos

El documento pretende dar una vista global del trabajo de los analistas forenses en los entornos GNU/Linux e iniciar a los administradores de sistemas en el mundo de ciencia forense informática a través de conceptos teóricos, procedimientos pre-establecidos de tratamiento de información y casos prácticos. El documento también indica la manera de montar un laboratorio forense, el equipo necesario, configuración de hardware y de software. Se dará a conocer la problemática de algunos aspectos del análisis
como congelación de la escena del crimen, preparación y análisis a través de herramientas comunes de GNU/Linux.

Debido a la gran variedad y cantidad de incidentes el alcance puede ser también muy diverso. Así pues no es lo mismo el trabajo de un colaborador que el de expertos como los forenses o las fuerzas y cuerpos de seguridad del estado los cuales cuentan con equipos íntegros, instalaciones y expertos.

No existe una pauta a seguir por eso el investigador debe tomar decisiones basándose en la experiencia. A simple vista puede que no encuentre nada anormal a la hora de analizar un sistema, pero debe buscar indicios de que el equipo ha sido vulnerado. Un trafico de datos anormal, un consumo alto de CPU de un proceso el cual no tiene asociado ningún programa en el sistema puede indicarnos que algo no va bien.

1.3 Planificación del proyecto

A continuación mostramos la planificación inicial del proyecto que nos permitirá completar todos los hitos propuestos. Este punto se ira modificando a medida que se vaya desarrollando nuestro TFC. Paralelamente se ira desarrollando la memoria del mismo incluyendo las posibles modificaciones que el consultor estime oportuno.

1.3.1 Inicio del proyecto.

Fecha inicio: 16 de septiembre
Fecha fin: 28 de septiembre (entrega borrador)

En este periodo nos dedicamos a asimilar los contenidos de la asignatura y a la lectura del plan docente. También y en paralelo empezamos a documentarnos sobre GNU/Linux, concretamos los objetivos del proyecto, definimos las fases que tendrá y elaboramos una primera planificación temporal que nos debe conducir a la realización del proyecto prevista para el día 8 de enero.

1.3.2 Introducción

Fecha inicio: 21 de octubre
Fecha fin: 12 de noviembre (entrega PAC1)

En esta fase continuamos con el proceso de documentación sobre los distintos procesos que se llevan a cabo durante el análisis forense que procesos se llevan a cabo durante el mismo así como de que tipo de documentos e informes se entregaran.

1.3.3 Aplicaciones

Fecha inicio: 13 de octubre
Fecha fin: 2 de noviembre (entrega PAC2)

En esta fase se realiza el análisis de las distintas aplicaciones que incluyen las distintas distros forenses. Estas nos servirán de ayuda a la hora de montar nuestro laboratorio forense.

1.3.4 Laboratorio Forense

Fecha inicio: 3 de noviembre
Fecha fin: 30 de noviembre (entrega PAC3)

Durante esta fase pondremos en marcha nuestro laboratorio. Para ello haremos un análisis previo de los instrumentos que necesitaremos, tanto hardware como software. Plantearemos un caso practico de como se procede ante un “incidente de seguridad”, que pasos seguiremos y como se debe documentar el mismo. Se desarrollara una serie de scripts que nos ayudaran a la automatización de tareas.

1.3.5 Ampliaciones, mejoras, contraste

Fecha inicio: 1 de diciembre
Fecha fin: 21 de diciembre (entrega PAC4)

Durante este periodo se ampliaran y modificaran algunos puntos aprovechado los conocimientos adquiridos durante el desarrollo del TFC.

1.3.6 Fin del proyecto

Fecha inicio: 22 de diciembre
Fecha fin: 8 de enero (entrega memoria y producto)

A partir de aquí debería estar el documento prácticamente listo, faltarían detalles menores como revisar el formato, la ortografía o la presentación. Se incluyen otras tareas menores como hacer la portada, o el índice, además de añadir algunos apartados más como la bibliografía, realizar el video o la valoración personal de lo que ha aportado la realización de este proyecto.

1.3.7 Filmación del video

Fecha inicio: 9 de enero
Fecha fin: 15 de enero (entrega video y presentación)

Finalizado ya el proyecto, se procederá a entregar un video y presentación que sintetice el trabajo realizado durante el cuatrimestre.

2.- Estado del Arte Forense

2.1 Introducción

En este capítulo se analizará la situación actual de la informática forense para situarnos en el contexto que nos ocupa y poder establecer unas bases para la creación de una metodología para el análisis forense.

De entrada se analizará qué es la informática forense pasando por su importancia hoy en día donde los equipos de información se hayan en cualquier punto. Se revisarán los fines de la informática forense. Finalmente, en este apartado se hará un resumen de los principios mínimos de calidad que debe cumplir la informática forense.

2.2 Aplicación

Ya vimos anteriormente que la informática forense es una ciencia encargada de asegurar, identificar, preservar, analizar y presentar un conjunto de datos de tal modo que estos puedan ser aceptados en un proceso legal. Esta ciencia y su conjunto de herramientas y técnicas permiten o facilitan, en la medida de lo posible, una reconstrucción del equipo informático afectado, el examen de los datos que se han podido recabar y la autenticación de los mismos.

Así podemos encontrar que la informática forense es de aplicación en distintos objetivos, a saber, preventivos, correctivos, probatorios y auditores. A continuación vamos a repasar brevemente de qué se trata cada uno de ellos.

Fines preventivos: Como ya se apuntaba anteriormente, la informática forense puede ayudar a prevenir posibles incidentes informáticos, formando parte del sistema de seguridad. En este caso, los responsables de seguridad de la empresa u organización pertinente, utilizarán las herramientas de la informática forense para verificar y auditar que los sistemas cumplen con los objetivos descritos en sus estándares de seguridad. Los resultados de estos estudios aportarán información valiosa de cara a mejorar los sistemas de seguridad, implementar nuevas metodologías en la organización o mantener los existentes.

Fines correctivos: En relación con el caso anterior, una organización puede detectar posibles fallos de seguridad informática, antes de cualquier incidente. Esta detección debe iniciar automáticamente una comisión que se encargue de corregir los fallos detectados e implemente las soluciones de seguridad que sean más convenientes para evitar que un usuario malintencionado pueda poner en riesgo la organización.

Fines probatorios: Esta es la finalidad de la informática forense que más nos interesa a tenor del tema que se trata en este trabajo. La informática forense con fines probatorios permite, que tras el registro de un incidente informático, se pueda recabar información sobre la intrusión en el sistema, descubrir qué daños se han producido, si ha habido robo de información o destrucción de la misma, etc. Entre otros, se puede llegar a descubrir el o los causantes del incidente, su origen, si se han comprometido más equipos o sólo el que se analiza. Finalmente, con todos los datos recabados, organizados y bien preservados de posibles manipulaciones ulteriores, se pueden presentar los datos ante un juzgado aportando una prueba con validez legal que permita la persecución y pena del hecho

delictivo.

Fines auditores: Otro campo de interés de la informática forense es la auditoría de sistemas informáticos. Relacionado con los fines correctivos, se pueden programar auditorias de seguridad, que llevan a cabo empresas especializadas, o incluso un equipo dentro de la propia organización interesada, que verifiquen periódicamente que los sistemas de información cumplen todos los requisitos de seguridad y que los usuarios mantiene unos mínimos de cuidados y prácticas de seguridad entorno a esos equipos.
Como en cualquier otra ciencia, la informática forense debe asegurar unos principios mínimos de calidad para asegurar que todos los resultados que obtiene son de calidad y no han sido manipulados en ningún momento, por ello debe asegurar que:

– Se evita la contaminación de las pruebas recogidas en el escenario. Hay que verificar que el equipo que se va a analizar no se modifica de ninguna manera ni se manipula, de igual forma con las copias de datos que se realicen. Para ello existen técnicas que prueban que una copia realizada es fiel a su original y contiene exactamente la misma información sin que haya variado ni tan siquiera un bit.
– Se debe actuar metódicamente. Hay que seguir unos pasos y seguirlos de manera correcta, se debe estar muy atento en todo momento a los detalles y no permitir que cualquier error o negligencia den al traste con todo el trabajo.
– Se debe controlar la prueba obtenida y evitar manipulaciones. Hay que realizar un registro de quién o quiénes han tenido en cada momento las pruebas y qué se ha hecho con ellas. De este modo nunca se podrá decir que las pruebas han sido manipuladas haciendo que pierdan su finalidad probativa en un proceso judicial.

Regulación

Todo lo anteriormente expuesto esta regulado legalmente tanto por normativas naciones como internacionales. Así pues en España esta regulado por:

– Ley de Enjuiciamiento Civil.
– Ley de Protección de Datos de Carácter Personal.
– Ley de Servicios de la Sociedad de la Información y comercio electrónico.
– Ley de Conservación de Datos.
– Código Penal.

En la actualidad con la nueva reforma del Código Penal (L.O 1/2015) se han endurecido las penas relativas a delitos informáticos.
Además de la regulación estatal que se ha repasado cabe destacar la regulación y propuestas de regulación a nivel europeo sobre aspectos informáticos y que son de igual interés para el profesional de este ámbito. Así pues cabe destacar dos directivas del Parlamento Europeo y del Consejo.

Directiva 2006/24/CE esta directiva trata de las obligaciones de los proveedores de servicios de comunicaciones electrónicas de acceso público o de una red pública de comunicaciones en relación con la conservación de determinados datos generados o tratados por los mismos, para garantizar que los datos estén disponibles con fines de investigación, detección y enjuiciamiento de delitos graves, tal como se definen en la legislación nacional de cada Estado miembro

Directiva 2013/40/UE establece las normas mínimas a la definición de las infracciones penales y a las sanciones aplicables en el ámbito de los ataques contra los sistemas de información.

2.3 Descripción de Herramientas.

Actualmente existen multitud de aplicaciones destinadas al análisis forense que trabajan sobre distintos aspectos de la máquina a analizar, por ejemplo, sobre las memorias, los discos de almacenamiento, los protocolos de red, las aplicaciones, etc.

También existen suites que ofrecen el análisis sobre varios de estos puntos ofreciendo herramientas verdaderamente potentes y útiles. No obstante, no existe ni la herramienta definitiva ni aquella aprobada y validada por ningún estándar. A continuación se hará un repaso de las herramientas más populares con una breve descripción y su ámbito de trabajo.

ADQUISICIÓN Y ANÁLISIS DE LA MEMORIA

Proccess Dumper - Convierte un proceso de la memoria a fichero.
FTK Imager - Permite entre otras cosas adquirir la memoria.
DumpIt - Realiza volcados de memoria a fichero.
Responder CE- Captura la memoria y permite analizarla.
Volatility - Analiza procesos y extrae información útil para el analista.
RedLine - Captura la memoria y permite analizarla. Dispone de entrono gráfico.
Memorize - Captura la RAM (Windows y OSX).

MONTAJE DE DISCOS

Utilidades para montar imágenes de disco o virtualizar unidades de forma que se tenga acceso al sistema de ficheros para posteriormente analizarla.

ImDisk - Controlador de disco virtual.
OSFMount - Permite montar imágenes de discos locales en Windows asignando una letra de unidad.
raw2vmdk - Utilidad en java que permite convertir raw/dd a .vmdk
FTK Imager - Permite realizar montaje de discos.
vhdtool - Convertidor de formato raw/dd a .vhd permitiendo el montaje desde el administrador de discos de Windows .
LiveView - Utilidad en java que crea una máquina virtual de VMware partiendo de una imagen de disco.
MountImagePro - Permite montar imágenes de discos locales en Windows asignando una letra de unidad

CARVING Y HERRAMIENTAS DE DISCO

Recuperación de datos perdidos, borrados, búsqueda de patrones y ficheros con contenido determinado como por ejemplo imágenes, vídeos. Recuperación de particiones y tratamiento de estructuras de discos.

PhotoRec - Muy útil, permite la recuperación de imágenes y vídeo.
Scalpel -Independiente del sistema de archivos. Se puede personalizar los ficheros o directorios a recuperar.

RecoverRS - Recupera urls de acceso a sitios web y ficheros. Realiza carving directamente desde una imagen de disco.
NTFS Recovery - Permite recuperar datos y discos aún habiendo formateado el disco.
Recuva - Utilidad para la recuperación de ficheros borrados.
Raid Reconstructor - Recuperar datos de un RAID roto, tanto en raid 5 o raid 0. Incluso si no conocemos los parámetros RAID.
CNWrecovery - Recupera sectores corruptos e incorpora utilidades de carving.
Restoration - Utilidad para la recuperación de ficheros borrados.
Rstudio - Recuperación de datos de cualquier sistema de disco NTFS, NTFS5, ReFS, FAT12/16/32, exFAT, HFS/HFS+ (Macintosh), Little y Big Endian en sus distintas variaciones UFS1/UFS2 (FreeBSD/OpenBSD/NetBSD/Solaris) y particiones Ext2/Ext3/Ext4 FS.
Freerecover - Utilidad para la recuperación de ficheros borrados.
DMDE - Admite FAT12/16, FAT32, NTFS, y trabaja bajo Windows 98/ME/2K/XP/Vista/7/8 (GUI y consola), DOS (consola), Linux (Terminal) e incorpora utilidades de carving.
IEF - Internet Evidence Finder Realiza carving sobre una imagen de disco buscando mas de 230 aplicaciones como chat de google, Facebook, IOS, memoria ram, memoria virtual,etc.
Bulk_extractor - Permite extraer datos desde una imagen, carpeta o ficheros.

UTILIDADES PARA EL SISTEMA DE FICHEROS

Conjunto de herramientas para el análisis de datos y ficheros esenciales en la búsqueda de un incidente.

analyzeMFT - David Kovar's utilidad en python que permite extraer la MFT
MFT Extractor- Otra utilidad para la extracción de la MFT
INDXParse - Herramienta para los indices y fichero $I30.
MFT Tools (mft2csv, LogFileParser, etc.) -Conjunto de utilidades para el acceso a la MFT
MFT_Parser - Extrae y analiza la MFT
Prefetch Parser - Extrae y analiza el directorio prefetch
Winprefectchview - Extrae y analiza el directorio prefetch
Fileassassin - Desbloquea ficheros bloqueados por los programas

HERRAMIENTAS DE RED

Todo lo relacionado con el tráfico de red, en busca de patrones anómalos, malware, conexiones sospechosas, identificación de ataques, etc.

WireShark - Herramienta para la captura y análisis de paquetes de red.
NetworkMiner - Herramienta forense para el descubrimiento de información de red.
Netwitness Investigator - Herramienta forense. La versión 'free edition' está limitado a 1GB de tráfico.
Network Appliance Forensic Toolkit - Conjunto de utilidades para la adquisición y análisis de la red.
Xplico - Extrae todo el contenido de datos de red (archivo pcap o adquisición en tiempo real). Es capaz de extraer todos los correos electrónicos que llevan los protocolos POP y SMTP, y todo el contenido realizado por el protocolo HTTP.

Snort - Detector de intrusos. Permite la captura de paquetes y su análisis.
Splunk - Es el motor para los datos y logs que generan los dispositivos, puestos y servidores. Indexa y aprovecha los datos de las generados por todos los sistemas e infraestructura de IT: ya sea física, virtual o en la nube.
AlientVault - Al igual que Splunk recolecta los datos y logs aplicándoles una capa de inteligencia para la detección de anomalías, intrusiones o fallos en la política de seguridad.

RECUPERACIÓN DE CONTRASEÑAS

Todo lo relacionado con la recuperación de contraseñas en Windows, por fuerza bruta, en formularios, en navegadores.

Ntpwedit - Es un editor de contraseña para los sistemas basados en Windows NT (como Windows 2000, XP, Vista, 7 y 8), se puede cambiar o eliminar las contraseñas de cuentas de sistema local. No valido para Active Directory.
Ntpasswd - Es un editor de contraseña para los sistemas basados en Windows, permite iniciar la utilidad desde un CD-LIVE
pwdump7 - Vuelca los hash. Se ejecuta mediante la extracción de los binarios SAM.
SAMInside / OphCrack / L0phtcrack- Hacen un volcado de los hash. Incluyen diccionarios para ataques por fuerza bruta.

2.4 Distribuciones Forenses

Aunque cualquier distro puede servirnos a la hora de poder realizar el análisis forense instalando para ello las aplicaciones necesarias existen distribuciones propiamente forenses las cuales incorporan ya dichas herramientas. Ahora veremos algunas de ellas:

2.4.1 CAINE (Computer Aided INvestigative Environment), es una distribución Live CD para realizar análisis forense informático, creada por Giancarlo Giustini es una de las mejores opciones que tenemos a la mano cuando deseamos realizar un análisis forense de algún equipo informático.CAINE se diferencia de las demás distribuciones de su tipo (Forensic Boot CD, Helix, Deft, etc..) por su facilidad de uso y que proporcionar una interfaz gráfica homogénea que guía a los investigadores digitales durante la adquisición y el análisis de las pruebas electrónicas, y ofrece un proceso semi-automático durante la documentación y generación de informes. Algunas aplicaciones instaladas en esta distro son:

Adquisición:

Grissom Analyzer (mmls, img_stat, fsstat): Es un conjunto de herramientas especializadas en el análisis de imágenes o las copias bita a bit que se le aplican a los medios de almacenamiento donde reside la evidencia.

AIR: Es una aplicación en modo gráfico para el uso del comando dd/dclfdd (Datataset Definition (dd)). Fue diseñado como una mejora en modo gráfico de todas las variantes de dd, su fácil uso permite crear imágenes forenses de discos y de particiones completas del mismo. Soporta MD5/SHAx hashes, cintas SCSI, proyección de imágenes sobre una red TCP/IP, imágenes partidas, y registro detallado de la sesión.

Guymager: Es una herramienta forense, con la capacidad de crear copias bit a bit o réplicas de imagen de disco, es bastante ágil en su funcionamiento y crea replicas en formatos dd, EWF, AFF.

DC3DD: Esta es una modificación de la herramienta de copia bit a bit dd, que incluye ciertas características que facilitan la adquisición de imágenes forenses.

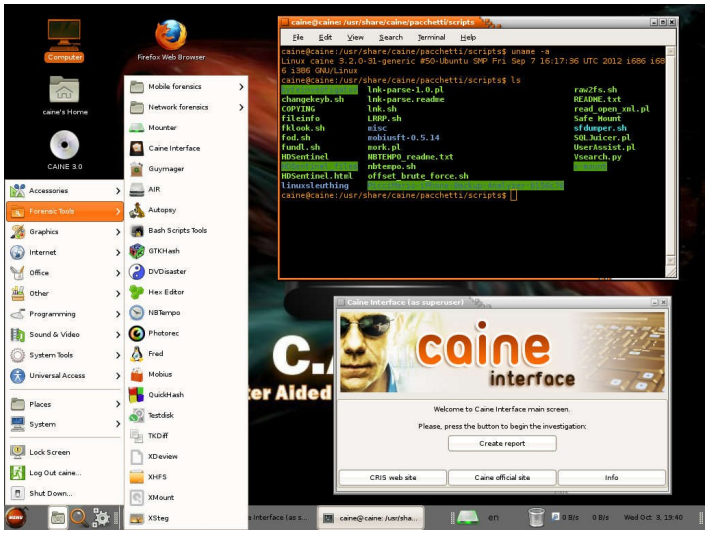
Análisis:

Autopsy: Tal vez la mejor herramienta libre que existe para el análisis de evidencia digital. Su interfaz gráfica es un browser que basado en las herramientas en línea de comandos del Sleuth Kit, permite un análisis de diversos tipos de evidencia mediante una la captura de de una imagen de disco.

The SleuthKit: Es una colección de herramientas en línea de comandos para análisis forense de archivos y volúmenes de sistema. Las herramientas del sistema de archivos permiten examinar una computadora sospechosa sin comprometerla. Debido a que las herramientas no confían en el sistema operativo para procesar el Sistema de Archivos, se muestra contenido borrado u oculto.

Foremost: es una utilidad en linea de comandos que nos permite recuperar esos archivos que hemos borrado por descuido de nuestros discos duros, memorias USB, tarjeta SD, etc…Foremost hace uso de una técnica denominada data mining que recupera los archivos basándose en sus encabezados, pies de página y estructura interna de los mismos, lo que le permite recuperar una gran variedad de formatos

Ophcrack: Utilidad para romper u obtener contraseñas de usuario en el sistema operativo Windows. Su funcionamiento se basa en el análisis de las tablas rainbow para el acceso a las claves de la SAM (Security Accounts Manager). SAM es el gestor de seguridad para cuentas de usuario, de los actuales sistemas operativos Microsoft Windows. Este servicio se emplea durante los procesos de acceso al sistema, y retiene información del usuario que se ha logeado ante el sistema.



Descarga: http://www.caine-live.net/
Herramientas: http://www.caine-live.net/page11/page11.html

2.4.2 D.E.F.T (Digital Evidence & Forensic Toolkit), es una distribución Linux basada en Xubuntu 9.10 con un kernel 2.6.31, escritorio LXDE además de una GUI con aplicaciones forenses (DEFT extra 2.0) pensada para policía, investigadores, administradores de sistemas o especialistas forenses. Entre sus opciones cabe destacar:

Analysis :Herramientas de análisis de ficheros de diferentes tipos

Antimalware :Búsqueda de rootkits, virus, malware, así como PDFs con código malicioso.

Data recovery: Software para recuperación de ficheros

Hashing :Scripts que permiten la realización de cálculo de hashes de determinados procesos (SHA1, SHA256, MD5...)

Imaging :Aplicaciones que podemos utilizar para realizar los clonados y adquisición de imágenes de discos duros u otras fuentes.

Mobile Forensics :Análisis de Blackberry, Android, iPhone, así como información sobre las típicas bases de datos de dispositivos móviles en SQLite utilizadas por las aplicaciones.

Network Forensics :Herramientas para procesamiento de información almacenada en capturas de red

OSINT :Aplicaciones que facilitan la obtención de información asociada a usuarios y su actividad.

Password recovery :Recuperación de contraseñas de BIOS, ficheros comprimidos, ofimáticos, fuerza bruta, etc.

Reporting tools :Por último, dentro de esta sección encontraremos herramientas que nos facilitarán las tareas de generación de informes y obtención de evidencias que nos servirán para documentar el análisis forense. Captura de pantalla, recopilación de notas, registro de actividad del escritorio, etc.

Actualmente se encuentra en la version 8.1 cuyos cambios mas importantes son:

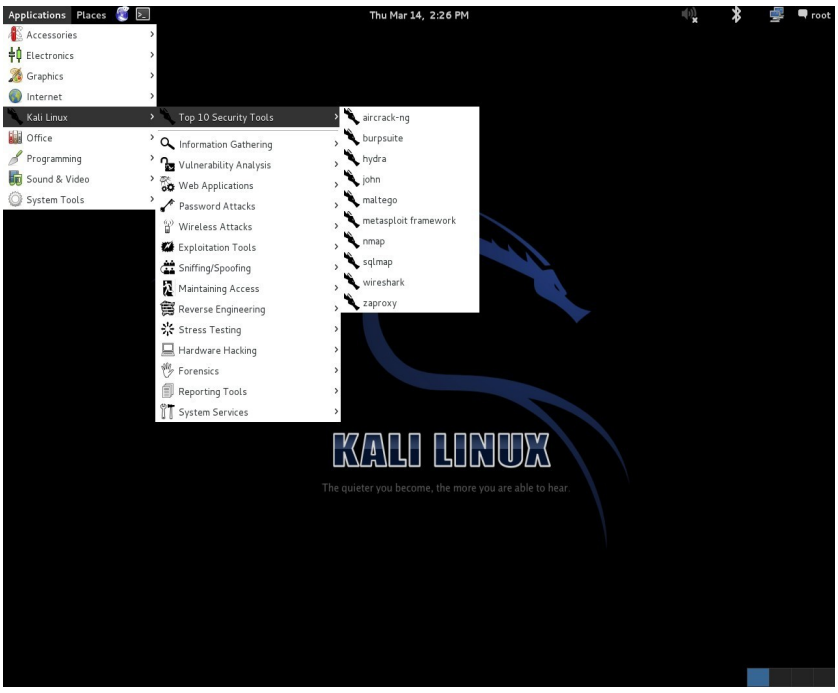
- The Sleuthkit 4.1.3
- Digital Forensics Framework 1.3
- Soporte para Android and iOS 7.1 adquisiciones lógicas (via libmobiledevice & adb)

Descarga: www.deftlinux.net
Manual: http://www.deftlinux.net/deft-manual/

2.4.3 KALI 2.0, es una distribución basada en Debian GNU/Linux diseñada principalmente para la auditoría y seguridad informática en general. Fue fundada y es mantenida por Offensive Security Ltd. Mati Aharoni and Devon Kearns, ambos pertenecientes al equipo de Offensive Security, desarrollaron la distribución a partir de la reescritura de BackTrack,que se podría denominar como la antecesora de Kali Linux. Kali puede ser usado desde un Live CD, live-usb y también puede ser instalada como sistema operativo principal.

En esta version se introdujo la opción Forensics Boot al sistema operativo y se vio continuada en BackTrack 5, existe al día de hoy en Kali Linux. Sirve para poner a trabajar las herramientas de software libre más populares en materia forense de forma rápida y sencilla. Este modo es muy popular debido a que Kali está ampliamente disponible y es fácil de conseguir; muchos usuarios potenciales ya cuentan con una Imagen ISO o un Live USB con el sistema. A su vez, Kali cuenta con el software libre forense más popular instalado y es sencillo y rápido de bootear.

Se realizaron algunos cambios importantes como que el disco duro no se utiliza en absoluto. Lo que trae como consecuencia que si existe una partición swap no va a ser usada ni se monta automáticamente ningún disco interno y se deshabilitó el auto-montado de medios removibles. Entonces, ni los pendrives ni los lectores de CD van a ser montados automáticamente.



Descarga: https://www.kali.org/
Listado de herramientas: http://tools.kali.org/tools-listing

2.4.4 HELIX, se trata de un Live CD de respuesta ante incidentes, basado en una distribución Linux denominada Knoppix (que a su vez está basada en Debian). Posee la mayoría de las herramientas necesarias para realizar un análisis forense tanto de equipos como de imágenes de discos. Ofrece dos modos de funcionamiento, tras ejecutarlo nos permitirá elegir entre arrancar un entorno MS Windows o uno tipo Linux. En el primero de ellos disponemos de un entorno con un conjunto de herramientas, que nos permitirá principalmente interactuar con sistemas "vivos", pudiendo recuperar la información volátil del sistema. En el arranque Linux, disponemos de un Sistema Operativo completo, con un núcleo modificado para conseguir una excelente detección de hardware, no realiza el montaje de particiones swap, ni ninguna otra operación sobre el disco duro del equipo sobre el que se arranque. Es ideal para el análisis de equipos "muertos", sin que se modifiquen las evidencias pues montará los discos que encuentre en el sistema en modo sólo lectura. Además de los comandos de análisis propios de los entornos UNIX/Linux, se han incorporado una lista realmente interesante de herramientas y ToolKits, alguno de ellos comentados anteriormente como el Sleuth Kit y Autopsy.

Descarga: http://www.e-fense.com/helix

Actualmente esta distribución es de pago, la ultima versión libre corresponde con Helix 2008 RC2

En resumen, hoy en día la cantidad de herramientas o scripts existentes, que realizan tareas similares, es inmenso. Claramente ciertos casos pueden ser mejores que otros, o puede tratarse de preferencias personales, pero en si el universo de herramientas es enorme. Algunas preguntas se plantean en el momento de elegir una nueva herramienta son los siguientes: ¿Es útil la herramienta en un entorno de pruebas de penetración?, ¿Contiene la herramienta las mismas funciones de otras herramientas existentes?, ¿Está permitido la libre redistribución por la licencia de la herramienta?, ¿Cuántos recursos requiere la herramienta?, ¿Funcionará en un entorno "estándar"?, etcétera. Tal y como mencionamos en puntos anteriores la destreza y a experiencia del analista forense hará que se decante por una herramienta u otra.

3.- Metodología

3.1 Introducción

Existe una gran diversidad de incidentes de seguridad pero el método de abordarlos puede resumirse en una serie de pasos que son comunes a todos ellos.

Mediante este proceso se pretende responder a las siguientes preguntas: ¿qué?, ¿dónde?, ¿cuándo?, ¿por qué?, ¿quién? y ¿cómo?

Siguiendo estos pasos contamos con varias metodologías las cuales siguen pautas similares.

--Forensic Examination of Digital Evidence: A Guide for Law Enforcement
--UNE 71506 - Metodología para el análisis forense de las evidencias electrónicas
--Good Practice Guide for Computer-Based Electronic Evidence
--RFC 3227 «Guidelines for Evidence Collection and Archiving» o Directrices para la recopilación de evidencias y su almacenamiento

A continuación, se detalla el RFC 3227 por tratarse de uno de los referentes, el cual refleja, desde un punto de vista teórico y bastante completo, el proceso de actuación y las pautas que se deben seguir a la hora de realizar un análisis de este tipo.

3.2 Directrices para la recolección de evidencias y su almacenamiento.

Los RFC «Request For Comments» son documentos que recogen propuestas de expertos en una materia concreta, con el fin de establecer por ejemplo una serie de pautas para llevar a cabo un proceso, la creación de estándares o la implantación de algún protocolo. El RFC 3227 es un documento que recoge las directrices para la recopilación de evidencias y su almacenamiento, y puede llegar a servir como estándar de facto para la recopilación de información en incidentes de seguridad.

3.2.1 Principios durante la recolección de evidencias.

- Capturar una imagen del sistema tan precisa como sea posible.
- Realizar notas detalladas, incluyendo fechas y horas indicando si se utiliza horario local o UTC.
- Minimizar los cambios en la información que se está recolectando y eliminar los

agentes externos que puedan hacerlo.

- En el caso de enfrentarse a un dilema entre recolección y análisis elegir primero recolección y después análisis.
- Recoger la información según el orden de volatilidad (de mayor a menor).
- Tener en cuenta que por cada dispositivo la recogida de información puede realizarse de distinta manera.

1. Orden de volatilidad

El orden de volatilidad hace referencia al período de tiempo en el que está accesible cierta información. Es por ello que se debe recolectar en primer lugar aquella información que vaya a estar disponible durante el menor período de tiempo, es decir, aquella cuya volatilidad sea mayor.

De acuerdo a esta escala se puede crear la siguiente lista en orden de mayor a menor volatilidad:

- Registros y contenido de la caché.
- Tabla de enrutamiento, caché ARP, tabla de procesos, estadísticas del kernel, memoria.
- Información temporal del sistema.
- Disco
- Logs del sistema.
- Configuración física y topología de la red.
- Documentos.

2. Acciones que deben evitarse

Se deben evitar las siguientes acciones con el fin de no invalidar el proceso de recolección de información ya que debe preservarse su integridad con el fin de que los resultados obtenidos puedan ser utilizados en un juicio en el caso de que sea necesario:
No apagar el ordenador hasta que se haya recopilado toda la información.
No confiar en la información proporcionada por los programas del sistema ya que pueden haberse visto comprometidos. Se debe recopilar la información mediante programas desde un medio protegido como se explicará más adelante.
No ejecutar programas que modifiquen la fecha y hora de acceso de todos los ficheros del sistema.

3.Consideraciones sobre la privacidad

Es muy importante tener en consideración las pautas de la empresa en lo que a privacidad se refiere. Es habitual solicitar una autorización por escrito de quien corresponda para poder llevar a cabo la recolección de evidencias. Este es un aspecto fundamental ya que puede darse el caso de que se trabaje con información confidencial o de vital importancia para la empresa, o que la disponibilidad de los servicios se vea afectada.

No hay que entrometerse en la privacidad de las personas sin una justificación. No se deben recopilar datos de lugares a los que normalmente no hay razón para acceder, como ficheros personales, a menos que haya suficientes indicios.

Dependiendo de si existe información con datos de carácter personal, hay que tener en cuenta la LOPD, así como su RDLOPD. De todas maneras es obvio que las leyes se deben

tener en cuenta siempre, ya que su desconocimiento no exime de su cumplimiento.

3.2.2 Procedimiento de recolección.

El procedimiento de recolección debe de ser lo más detallado posible, procurando que no sea ambiguo y reduciendo al mínimo la toma de decisiones.

1. Transparencia

   Los métodos utilizados para recolectar evidencias deben de ser transparentes y reproducibles. Se debe estar preparado para reproducir con precisión los métodos usados, y que dichos métodos hayan sido testados por expertos independientes.

2. Pasos

- ¿Dónde está la evidencia? Listar qué sistemas están involucrados en el incidente y de cuáles de ellos se deben tomar evidencias.
- Establecer qué es relevante. En caso de duda es mejor recopilar mucha información que poca.
- Fijar el orden de volatilidad para cada sistema.
- Obtener la información de acuerdo al orden establecido.
- Comprobar el grado de sincronización del reloj del sistema.
- Según se vayan realizando los pasos de recolección preguntarse qué más puede ser una evidencia.
- Documentar cada paso.
- No olvidar a la gente involucrada. Tomar notas sobre qué gente estaba allí, qué estaban haciendo, qué observaron y cómo reaccionaron.

3.2.3 El procedimiento de almacenamiento.

1. Cadena de custodia

   Debe estar claramente documentada y se deben detallar los siguientes puntos:

- ¿Dónde?, ¿cuándo? y ¿quién? descubrió y recolectó la evidencia.
- ¿Dónde?, ¿cuándo? y ¿quién? manejó la evidencia.
- ¿Quién ha custodiado la evidencia?, ¿cuánto tiempo? y ¿cómo la ha almacenado?
- En el caso de que la evidencia cambie de custodia indicar cuándo y cómo se realizó el intercambio, incluyendo número de albarán, etc.

2. Dónde y cómo almacenarlo

- Se debe almacenar la información en dispositivos cuya seguridad haya sido demostrada y que permitan detectar intentos de acceso no autorizados.

3.2.4 Herramientas necesarias

Existen una serie de pautas que deben de ser seguidas a la hora de seleccionar las herramientas con las que se va a llevar a cabo el proceso de recolección:

- Se deben utilizar herramientas ajenas al sistema ya que éstas pueden haberse visto comprometidas, principalmente en los casos de malware.
- Se debe procurar utilizar herramientas que alteren lo menos posible el escenario, evitando el uso de herramientas de interfaz gráfico y aquellas cuyo uso de memoria sea grande.
- Los programas que se vayan a utilizar para recolectar las evidencias deben estar ubicados en un dispositivo de sólo lectura (CDROM, USB, etc.).
- Se debe preparar un conjunto de utilidades adecuadas a los sistemas operativos con los que se trabaje.
- El kit de análisis debe incluir los siguientes tipos de herramientas:
  Programas para listar y examinar procesos.
  Programas para examinar el estado del sistema.
  Programas para realizar copias bit a bit.

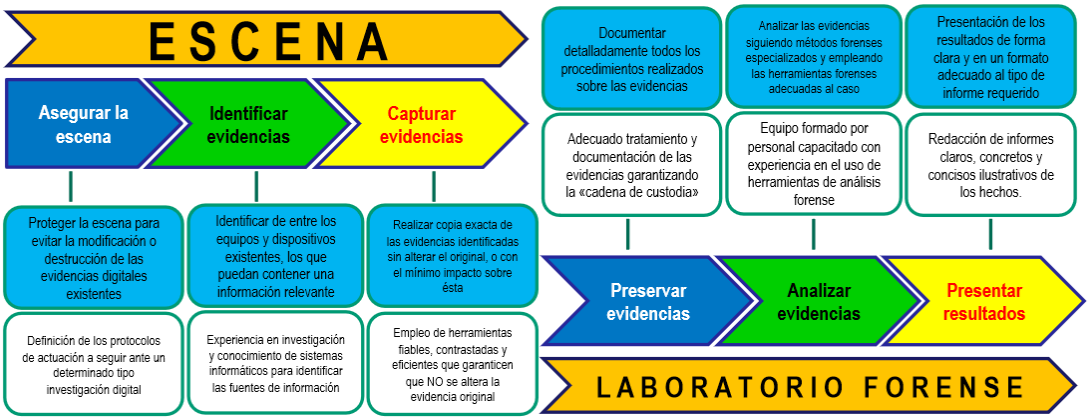
Conclusiones

A la hora de enfrentarse a un incidente de seguridad hay que tener muy claro las acciones que se deben realizar, siendo muy meticuloso y detallando en todo momento dicho proceso de manera minuciosa. Así mismo, se debe realizar el proceso procurando ser lo menos intrusivo posible con el fin de preservar el sistema en su estado original, y siguiendo las pautas indicadas en alguna de las metodologías o guías anteriormente indicadas o similares.

Finalmente, se debe tener presente que los requisitos o pautas a seguir a la hora de realizar un análisis forense digital que vaya a derivar en un proceso legal varían dependiendo del país, ya que no existe una legislación común. De todas formas, se debe tender a seguir las indicaciones establecidas en alguna metodología como el RFC 3227 con el fin de que dicho proceso sea realizado de una manera rigurosa.

4.- Fases Análisis Forense Digital

Tras un repaso de una de las metodologías utilizadas para el estudio del análisis forense digital vamos a ver cada una de las etapas mas pormenorizadamente. En el siguiente gráfico podemos observar donde se encuadran cada una de ellas y cual es su tarea principal:



Se puede crear una clasificación de tipos de análisis forense en base a qué estén orientados a analizar. Teniendo en cuenta este aspecto se pueden identificar varios tipos de análisis:

- Análisis forense de sistemas: tanto sistemas operativos Windows, como OSX, GNU/Linux, etc.
- Análisis forense de redes.
- Análisis forense de sistemas embebidos.
- Análisis forense de memoria volátil.

Si bien cada uno tiene sus propias características desde un punto de vista global son similares para cada uno de los tipos.

4.1 Adquisición.

Podemos decir que se trata de la fase mas critica dentro del análisis forense, ya que se trata del punto de partida desde el cual se realizara la posterior investigación. Como vemos en el anterior gráfico se trata de una tarea la cual se lleva a cabo dentro de lo que hemos determinado como “Escena”, es decir en el lugar en el que nos encontramos con el incidente.

Hay que tener presente que habrá pruebas ocultas con diferentes niveles de volatilidad, como los registros del procesador, estructuras de datos en la memoria RAM o memoria de tipo caché, conexiones de red activas, usuarios y procesos actuales, sistema de archivos, etc. Será muy difícil reunir toda esta información a la vez y gran parte de esta se perderá si decide apagar el equipo de la forma habitual, ya que en este proceso se realizan una serie de pasos programados para cerrar el sistema de forma limpia, pero si además el atacante ha instalado las herramientas adecuadas éste podría eliminar, modificar y sustituir ficheros a su antojo durante el apagado, y se “limpiarán” también del equipo las huellas de su

atacante. Además si el atacante sigue on-line, puede detectar su actividad y actuar con una acción evasiva o, peor aún, destructiva eliminando todo tipo de información. Pero si por la

severidad del ataque o por la importancia de los datos comprometidos se decide apagar el equipo, SE DESCONECTARA DIRECTAMENTE DE LA RED ELÉCTRICA, de esta forma perderá la información volátil de la RAM, micro, etc. pero conservará aún bastante información sobre el ataque. Así pues podemos encontrarnos con dos escenarios distintos:

1. **Equipo encendido ("vivo")**, tal y como se describe siguiendo las directrices del RFC 3227, estableceremos el siguiente orden de volatilidad y por tanto de recopilación de evidencias:
   - Registros y contenidos de la caché.
   - Contenidos de la memoria.
   - Estado de las conexiones de red, tablas de rutas.
   - Estado de los procesos en ejecución.
   - Contenido del sistema de archivos y de los discos duros.
   - Contenido de otros dispositivos de almacenamiento.

Dentro de las evidencias volátiles será de interés recuperar los siguientes datos del sistema en tiempo real:
   - Fecha y hora.
   - Procesos activos.
   - Conexiones de red.
   - Puertos TCP/UDP abiertos y aplicaciones asociadas "a la escucha".
   - Usuarios conectados remota y localmente.

Para el almacenamiento de esta información volátil utilizaremos discos externos (nunca utilizar el propio equipo comprometido para almacenar información) o memorias USB.

Para este proceso vamos a usar Volatility que es un Framework con un conjunto de herramientas desarrolladas enteramente en Python con licencia GNU. Este Framework esta pensado para extraer de una imagen de un disco los datos volátiles que estaban en memoria RAM. Estas técnicas de extracción están pensadas para que no dependan del sistema operativo del investigador, es decir podemos utilizar Windows y/o Linux.

La distribución de este framework está disponible en:
https://www.volatilesystems.com/default/volatility

La descarga de Python desde:
http://www.python.org

2. **Equipo apagado ("muerto")**, no se podrá recolectar este tipo de información por lo que pasaremos directamente al proceso de extracción de información de discos.

Tan pronto como haya obtenido toda la información volátil del sistema tendremos que recopilar la información contenida en los discos duros, teniendo en cuenta que estos dispositivos no sólo contienen las particiones, los archivos, directorios, etc. Sino que también contienen otro tipo de datos que hacen referencia a los propios archivos y a flujos de información, son los metadatos que serán de gran importancia en el análisis forense. Por ello

se procederá a realizar una copia exacta Bit a Bit en un proceso denominado clonacion. Para ello utilizaremos un LIVECD colocando el disco origen en el IDE1 como master y el disco destino en el IDE1 como exclavo.

Una de las herramientas más empleadas en entornos UNIX/Linux es dd, ésta permite crear imágenes de discos bit-a-bit, además de ofrecer otras opciones como obtención del hash MD5 de la copia, etc. En general se recomienda utilizar como salida de dd un archivo, y no otro dispositivo, para poder disponer del mismo para copiarlo al medio más adecuado.

dd if=/dev/hda of=archivo.dd conv=notrunc,noerror,sync

Las opciones utilizadas son:

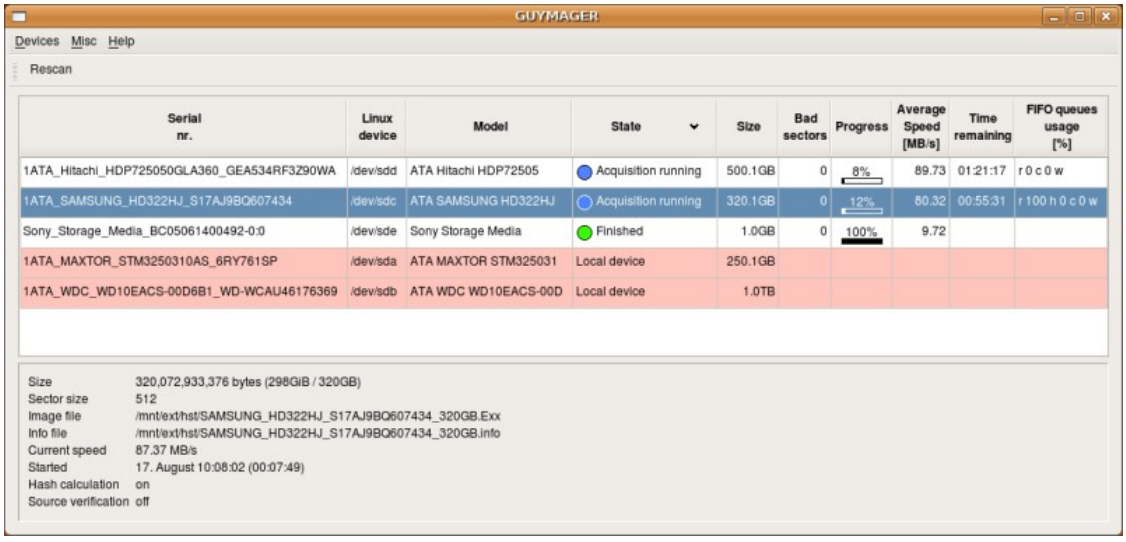
- **if=/dev/hda** : utilizada para indicar el archivo origen..
- **of=archive.dd** : utilizada para indicar el archive destino
- **conv=notrunc,noerror,sync** : utilizada para no truncar la salida en caso de error, no detener la duplicación en caso de error, y rellenar con ceros la salida en caso de error, respectivamente. Puede especificarse además el tamaño de los bloques de datos a copiar, utilizando la opción bs.

Obtendremos siempre imágenes de los discos duros para su posterior análisis y, siempre sobre medios de sólo lectura.

Para verificar que los datos obtenidos de la clonacion son validos comprobaremos la integridad de los mismos. Utilizaremos funciones HASH basadas en SHA1/MD5[1]

ssh1sum /dev/hda
ssh1sum /dev/hdb
ssh1sum archivo.dd

Dentro de la distro forense que vamos a utilizar en nuestro laboratorio cabe destacar la herramienta GUYMANAGER (http://guymager.sourceforge.net/).

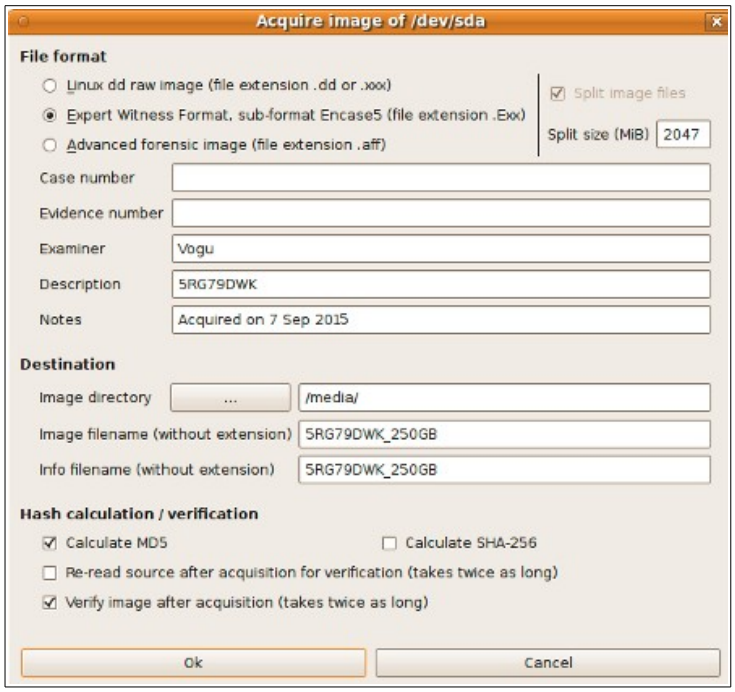


Guymanager es un generador de imágenes para la adquisición de medios físicos.

1 El uso del hash MD5, pese al alto grado de utilización, presenta el problema de que pueden surgir colisiones, es decir, puede darse el caso de que ficheros diferentes tengan el mismo MD5, por lo que puede quedar en entredicho la validez de las pruebas. Es por ello que es recomendable que vaya cayendo en desuso.
Un caso similar, aunque no igual, es el del SHA-1 por lo que se aconseja que se busquen otras alternativas como SHA-256, SHA-512, etc.

- Los dispositivos de almacenamiento conectados se enumeran en la parte superior.
- Los nuevos dispositivos se pueden conectar en cualquier momento.
- Los dispositivos marcados con el rojo son los discos duros locales, estos no pueden ser adquiridos, lo que impide adquirir discos erróneamente.
- Discos duros locales son reconocidos por sus números de serie que se pueden introducir en el archivo de configuración.

La parte inferior muestra información más detallada acerca de la adquisición seleccionado por el cursor azul.



Dialogo de adquisición por defecto GUYMANAGER de la distro CAINE.

Al igual que el dialogo anterior existe otra opción para la clonacion de disco.
Como vemos en la captura una vez finalizado el proceso se verificara que se haya realizado correctamente ademas de comprobar la integridad del mismo con el calculo del MD5 y el SHA-256.

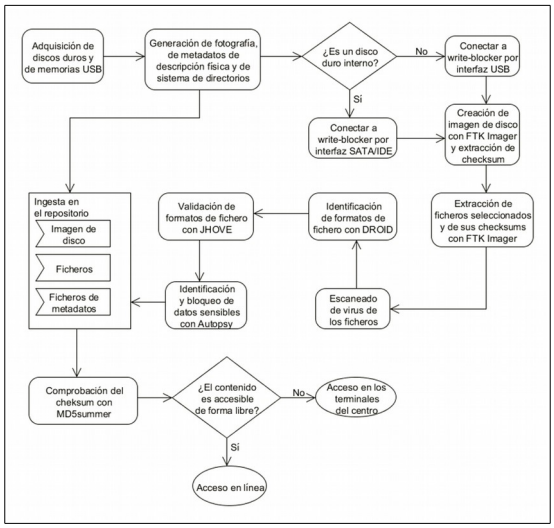


Figura: Procedimiento de trabajo para la preservación de discos.

4.2 Análisis e Investigación.

Una vez que disponemos de las evidencias digitales recopiladas y almacenadas de forma adecuada, pasemos a la fase quizás más laboriosa, el Análisis Forense propiamente dicho, cuyo objetivo es reconstruir con todos los datos disponibles la línea temporal del ataque o timeline, determinando la cadena de acontecimientos que tuvieron lugar desde el instante inmediatamente anterior al inicio del ataque, hasta el momento de su descubrimiento. Esta fase requiere el conocimiento del sistema el cual se quiere estudiar, no es lo mismo analizar un equipo con un sistema Windows o con uno Linux. Tampoco será lo mismo un caso de intrusión en el correo electrónico de alguien o un ataque de denegación de servicio. De igual forma no actuaremos con los mismos pasos en un caso de instalación de un malware que destruya información de una ubicación de disco o un malware que envíe todo lo que se teclea en un equipo.

Elementos a analizar en sistemas Windows:

- Registro del sistema.
- Contenido del sistema de ficheros cifrados (EFS).
- FAT o MTF (Tabla de Metadatos).
- Archivo BITMAP (Fichero creado durante el formateo de volúmenes NTFS)
- Papelera de reciclaje.
- Fichero de acceso directo.
- Log de visor de eventos.

Elementos a analizar en sistemas Linux:

- Listado de descriptores de ficheros.
- Ficheros SUID/SGID.
- Trabajos planificados.
- Ficheros historial Shell.

Cuando se accede a la información podemos encontrar dos tipos de análisis:

• Físico: información que no es interpretada por el sistema operativo ni por el de ficheros.

• Lógico: información que sí que es interpretada por el sistema operativo. En este nivel, por tanto, podremos obtener: estructura de directorios, ficheros que se siguen almacenando así como los que han sido eliminados, horas y fechas de creación y modificación de los ficheros, tamaños, utilización de los HASH para reconocer los tipos de archivos, contenido en los sectores libres, etc.

En un dispositivo de almacenamiento nos encontraremos con tres tipos de datos recuperados:

• Allocated: inodo[2] y nombre del fichero intactos, con lo que dispondremos del contenido integro.

• Deleted/Reallocated: inodo y nombre del fichero intactos aunque han sido

2 Es una estructura de datos propia de los sistemas de archivos tradicionalmente empleados en los sistemas operativos tipo UNIX que contiene las características (permisos, fechas, ubicación, pero NO el nombre) de un archivo regular, directorio, o cualquier otro objeto que pueda contener el sistema de ficheros.

recuperados porque habían sido borrados, con lo que dispondremos del contenido íntegro.

• Unallocated: inodo y nombre de fichero no disponibles, con lo que no tendremos el contenido integro del archivo aunque sí algunas partes. A veces, realizando una labor muy laboriosa se puede obtener parte de la información e incluso unir las partes y obtener casi toda la información del archivo.

Una de las primeras acciones que vamos a tener que efectuar es determinar la configuración horaria del sistema. Con dicha opción podremos validar las fechas y las horas que podemos identificar para que no sean cuestionadas ante otro peritaje por ejemplo.

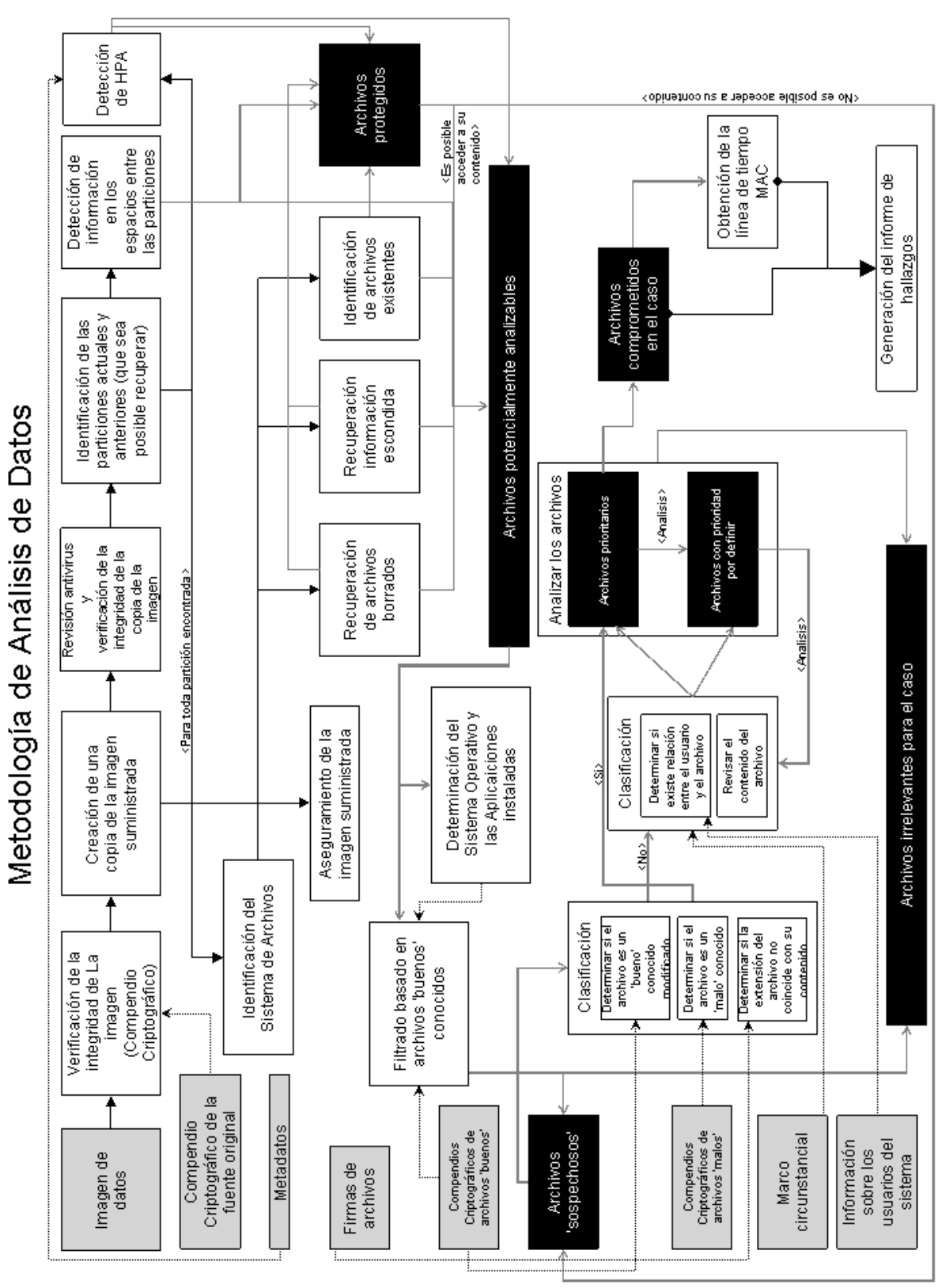
Después de identificar la configuración horaria, podremos realizar el estudio de la línea de tiempo también conocida como timeline y conocer cuáles han sido las acciones realizadas desde la instalación hasta el momento que se ha clonado el disco.

Las herramientas por excelencia para esta fase de análisis e investigación son el EnCase y el Sleunth kit & Autopsy. El EnCase es una aplicación propietaria para la realización de análisis forense mientras que Sleunth kit & Autopsy es un conjunto de herramientas de software libre creadas por Dan Farmer y Wietse Venema. Estas aplicaciones funcionan sobre Windows y GNU/Linux respectivamente, pero son capaces de analizar sistemas Unix, Linux, Mac OS X y Microsoft. En siguientes capítulos se mostrara como funciona esta herramienta.

Con la herramienta Volatility, se puede extraer información de los procesos en ejecución, sockets de red abiertos, conexión de red, DLL cargadas de cada proceso y secciones del registro de cache, IDs de los procesos y más. También tiene soporte para extraer información de los archivos de volcado de caída de Windows y archivos de hibernación.

Herramienta Volatility vemos de que tipo es la imagen capturada.

Anexo Metodología Análisis de Datos



Fuente: "Una Propuesta Metodológica y su Aplicación en The Sleuth Kit y EnCase ". Octubre de 2005. Jonathan Córdoba; Ricardo Laverde; Diego Ortiz; Diana Puentes.

4.3 Documentación y Presentación.

4.3.1 Informe Ejecutivo

Entrando más en detalle en este tipo de informes, cabe destacar que será un resumen de toda la tarea que se ha llevado a cabo con las evidencias digitales. Aunque será un documento de poca extensión, al menos comparado con el informe técnico, éste deberá contener al menos los siguientes apartados:

- Motivos de la intrusión.
  - – ¿Por qué se ha producido el incidente?
  - – ¿Qué finalidad tenía el atacante?

- Desarrollo de la intrusión.
  - – ¿Cómo lo ha logrado?
  - – ¿Qué ha realizado en los sistemas?

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

- Recomendaciones.
  - – ¿Qué pasos dar a continuación?
  - – ¿Cómo protegerse para no repetir los hechos?

4.3.2 Informe Técnico.

El informe técnico será más largo que el anterior y contendrá mucho más detalle. Se hará una exposición muy detallada de todo el análisis con profundidad en la tecnología usada y los hallazgos. En este caso se deberá redactar, al menos:

- Antecedentes del incidente.
  - – Puesta en situación de cómo se encontraba la situación anteriormente al incidente.

- Recolección de datos.
  - – ¿Cómo se ha llevado a cabo el proceso?
  - – ¿Qué se ha recolectado?

- Descripción de la evidencia.
  - – Detalles técnicos de las evidencias recolectadas, su estado, su contenido, etc.

- Entorno de trabajo del análisis.
  – ¿Qué herramientas se han usado?
  – ¿Cómo se han usado?

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

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

4.4 Soporte Legal

En este punto veremos las leyes de la legislación española que debemos tener en cuenta a la hora de realizar el análisis forense de un sistema informático en este país.

4.4.1 Ley de Enjuiciamiento Civil

La Ley de Enjuiciamiento Civil establece el marco legal, mediante el cual se regulan los procesos civiles, los tribunales y quienes ante ellos acuden e intervienen.

4.4.2 Derechos fundamentales

Los derechos fundamentales son aquellos derechos humanos garantizados con rango constitucional que se consideran como esenciales en el sistema político que la Constitución funda y que están especialmente vinculados a la dignidad de la persona humana.
La Constitución española otorga a todos los ciudadanos una serie de derechos fundamentales y libertades públicas, reguladas por el título I de la Constitución, capítulo 2, sección 1.

Los derechos se dividen fundamentalmente en 3 tipos, según el ámbito:
1) Personal.
2) Público.
3) Económico y social.

Dentro de estos derechos son de particular interés los siguientes:

• Derecho a la seguridad jurídica y tutela judicial, la cual nos garantiza un proceso penal con garantías.

• Derecho al secreto de las comunicaciones.

• Derecho a la vida privada. En este derecho se incluye el derecho a la intimidad, una vida privada, derecho al honor y la propia imagen. Asimismo se incluye la limitación del uso de la informática para proteger la intimidad.

• Derecho fundamental a la protección de datos. En el año 2000 en la sentencia 292/2000, el Tribunal Constitucional crea el derecho fundamental a la protección de datos como un derecho diferente al de intimidad.

4.4.3 Ley de Protección de Datos de Carácter Personal

LOPD son las siglas abreviadas de la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal. Esta Ley fundamentalmente tiene el objetivo de proteger a las personas físicas con respecto al tratamiento que se pueda realizar de sus datos propios por distintos sujetos, ya sean públicos o privados.

Dicha regulación pretende, fundamentalmente, establecer un control sobre quién tiene dichos datos, para qué los usa y a quién se los cede. Para ello, impone una serie de obligaciones a los responsables de dichos ficheros de datos, como son las de recabar el consentimiento de los titulares de los datos para poder tratarlos, comunicar a un registro especial la existencia de dicha base de datos y su finalidad, así como mantener unas medidas de seguridad mínimas de la misma, en función del tipo de datos recogidos. Por otro lado, la LOPD reconoce una serie de derechos al individuo sobre sus datos, como son los de información, acceso, rectificación e, incluso, de cancelación de los mismos en determinados supuestos.

Finalmente, se designa a una entidad: la Agencia de Protección de Datos, como órgano administrativo encargado de hacer cumplir la LOPD y sus reglamentos,pudiendo inspeccionar e imponer fuertes sanciones a aquellos sujetos que no cumplan con la misma.

Dentro del Reglamento de Desarrollo de la LOPD (RD 1720/2007), existen tres niveles de seguridad distintos: el básico, el medio y el alto. Para saber qué nivel debemos de aplicar, debemos referirnos al tipo de datos personales almacenados en el fichero. Para ello, estaremos a lo dispuesto en el artículo 81 del Reglamento, del que se deduce lo siguiente:

1) Nivel básico:

• Aplicable a todos los sistemas con datos personales en general.

2) Nivel medio:

• Datos de comisión de infracciones administrativas o penales.
• Datos de Hacienda pública.
• Datos de servicios financieros.
• Datos sobre solvencia patrimonial y crédito
• Conjunto de datos de carácter personal suficientes que permitan obtener una evaluación de la personalidad del individuo.

3) Nivel alto:

- Datos sobre ideología.
- Datos sobre religión.
- Datos sobre creencias.
- Datos sobre origen racial.
- Datos sobre salud o vida sexual.
- Datos recabados para fines policiales
- Datos sobre violencia de género.

4.4.4 Ley de Servicios de la Sociedad de la Información y del C. Electrónico

LSSI-CE son las siglas abreviadas de la Ley de Servicios de la Sociedad de la Información y del Comercio Electrónico aprobada el 11 de julio del 2001. Esta ley tiene el objetivo fundamental de regular y proteger a todos aquellos que intervienen en las relaciones ofrecidas por Internet. Dicha regulación pretende, fundamentalmente, establecer una normativa de Internet desde un punto de vista comercial y promocional obligando, por ejemplo a los propietarios de las webs, a incluir los datos de identificación de la empresa de modo perfectamente accesible y claro.
Además prohíbe el correo electrónico comercial no solicitado, también conocido con el nombre de spam.

4.4.5 Ley de conservación de datos relativos a las comunicaciones

Esta ley tiene como objetivo conservar los datos que pueden ser relevantes para rastrear las actividades ilícitas y así mejorar la seguridad de los ciudadanos frente a actividades terroristas. Por tanto, pretende establecer una regulación a los operadores de telecomunicaciones para retener determinados datos generados o tratados por los mismos, con el fin de posibilitar que dispongan de ellos los agentes facultados (son los miembros de los Cuerpos de Policía autorizados para ello en el marco de una investigación criminal).En su artículo 3 nos define los datos objeto de conservación dividiéndolos en diferentes tipos:

- Telefonía fija
- Telefonía móvil
- Acceso a Internet, correo electrónico y telefonía por Internet

Los datos que solicitan que sean conservados son todos los necesarios para la trazabilidad de origen a destino de cualquier comunicación telemática. El periodo de conservación de los datos impuesta cesa a los doce meses, siempre computados desde la fecha en que se haya producido la comunicación. Aunque podría haber alguna excepción, cuyo periodo mínimo deberá ser de 6 meses y máximo 2 años.

4.4.6 Código penal

El Código penal nos muestra las actitudes que se han tipificado como delito.
El concepto de delito viene descrito en el artículo 10 del Código penal (Ley Orgánica 10/1995, de 23 de noviembre) (CP): "son delitos o faltas las acciones y omisiones dolosas o imprudentes penadas por la Ley."

Por tanto, en este apartado comentaremos todas aquellas acciones que se pueden

considerar como delitos telemáticos según la LO 10/1995 y varias modificaciones posteriores:

• Corrupción de menores:

– Exhibicionismo y provocación sexual (art. 186): establece como delito la difusión, venta o exhibición entre menores de material pornográfico.

– Prostitución (art. 187 y 189.1):

• Apología del delito:

– Concepto (art. 18.1, párrafo 2.°).

– Apología del genocidio (art. 608.2).

• Delitos contra el honor (art. 211):

– Calumnias (art. 205).

– Injurias (art. 208).

"La calumnia y la injuria se reputarán hechas con publicidad cuando se propaguen por medio de la imprenta, la radiodifusión o por cualquier otro medio de eficacia semejante."

• Delitos contra la intimidad (art. 197):

1. "El que, para descubrir los secretos o vulnerar la intimidad de otro, sin su consentimiento, se apodere de sus papeles, cartas, mensajes de correo electrónico o cualesquiera otros documentos o efectos personales, o intercepte sus telecomunicaciones o utilice artificios técnicos de escucha, transmisión, grabación o reproducción del sonido o de la imagen, o de cualquier otra señal de comunicación, será castigado con las penas de prisión de uno a cuatro años y multa de doce a veinticuatro meses."

• Defraudación electrónica:

– Estafa (art. 248.2):

"También se consideran reos de estafa los que, con ánimo de lucro, y valiéndose de alguna manipulación informática o artificio semejante, consigan la transferencia no consentida de cualquier activo patrimonial en perjuicio de tercero."

• Apropiación indebida (art. 252).

• Uso ilegal de terminales (art. 256):

"El que hiciere uso de cualquier equipo terminal de telecomunicación, sin consentimiento de su titular, ocasionando a éste un perjuicio superior a cincuenta mil pesetas, será castigado con la pena de multa de tres a doce meses."

• Daños a ficheros informáticos (art. 264.2):

"La misma pena (prisión de uno a tres años y multa) se impondrá al que por cualquier medio destruya, altere, inutilice o de cualquier otro modo dañe los datos, programas o documentos electrónicos ajenos contenidos en redes, soportes o sistemas informáticos."

• Piratería informática:

– "Será castigado con la pena de prisión de seis meses a dos años y multa de 12 a 24 meses quien, con ánimo de lucro y en perjuicio de tercero, reproduzca, plagie, distribuya o comunique públicamente, en todo o en parte, una obra literaria, artística o científica, o su transformación, interpretación o ejecución artística fijada en cualquier tipo de soporte

o comunicada a través de cualquier medio, sin la autorización de los titulares de los

correspondientes derechos de propiedad intelectual o de sus cesionarios."

– Art. 270.3: "Será castigado también con la misma pena quien fabrique, importe, ponga en circulación o tenga cualquier medio específicamente destinado a facilitar la supresión no autorizada o la neutralización de cualquier dispositivo técnico que se haya utilizado para proteger programas de ordenador o cualquiera de las otras obras, interpretaciones o ejecuciones en los términos previstos en el apartado 1 de este artículo."

• Delitos documentales. En el artículo 26 del Código penal define el concepto de documento:

"A los efectos de este Código se considera documento todo soporte material que exprese o incorpore datos, hechos o narraciones con eficacia probatoria o cualquier otro tipo de relevancia jurídica."

– Falsedades documentales (del artículo 390 al 400).

– Infidelidad en la custodia (del artículo 413 al 416).

• Protección de la contraseña (art. 414.2):

"El particular que destruyere o inutilizare los medios a que se refiere el apartado anterior (los puestos para impedir el acceso no autorizado a los documentos) será castigado con la pena de multa de seis a dieciocho meses."

5.- Laboratorio

Vamos hacer uso de la distribución CAINE para realizar un análisis completo de un dispositivo, en este caso un ordenador personal, el cual ha sido utilizado para la comisión de un delito. Tras la llegada del dispositivo a nuestro laboratorio vamos a proceder al desprecintado del disco duro el cual va a ser objeto de nuestro análisis.. Como ya vimos en puntos anteriores anotaremos todos los pasos seguidos para posteriormente realizar el informe. Para la realización del análisis utilizaremos algunas de las aplicaciones que contiene esta distro forense ademas de algunos scripts de creación propia los cuales nos serán de ayuda para la automatización de algunas tareas. La peculiaridad de este proyecto es que lo vamos a realizar desde una maquina virtual con las ventajas que ello conlleva. Ademas podremos analizar desde un entorno seguro (maquina virtual) si se ha utilizado algún tipo de malware.

Una de las principales ventajas de utilizar sistemas Linux son:

· Todo, incluido el hardware se trata y representa como un fichero.

· Soporta numerosos tipos de sistemas de archivos, mucho no reconocidos por Windows

· Permite montar los sistemas de archivos; analizar un sistema en funcionamiento de forma segura y poco invasiva¸ dirigir la salida de un comando a la entrada de otros (múltiples comandos en una línea); revisar el código fuente de la mayoría de sus utilidades y generar dispositivos de arranque.

5.1 Hardware

Existen en el mercado diversos productos comerciales los cuales no solo son necesarios para investigar sino para proteger la integridad de los dispositivos, de forma que sigan siendo validos como prueba judicial. Así no podemos encontrar con dispositivos con la mas a avanzada tecnología en el campo del análisis forense, copiadoras duplicadoras, creadoras de imágenes ultra rápidas; equipos para investigación "en caliente" no intrusivos, que no dejan huella así como equipos para borrado seguro.

A continuación vamos a ver alguno de estos dispositivos:

Estaciones de trabajo

FRED (Digital Intelligence)
www.digitalintelligence.com/products/fred

Las unidades FRED están optimizadas para la adquisición y el análisis de datos y facilitan la lectura y duplicación de prácticamente cualquier dispositivo sin peligro de alterar los datos originales. Permiten la conexión directa de discos duros internos, arranque dual de sistemas operativos con la opción de instalar Linux y conexión a red por interfaz Ethernet que posibilita su uso como estación de trabajo estándar. La adquisición de una unidad incluye software de creación de imágenes de disco, antivirus y de análisis forense.

Figura: Unidad FRED. (Foto de Digital Intelligence)

Algunas de sus especificaciones:

23 3/4 "de alto, 8 3/8" de ancho, 25 4/1 "de profundidad - 80 libras
Intel Core i7-5820K CPU (Hex Core Processor), 3,3 GHz, 10 MB de Intel Smart Cache, 5 GT / s DMI
32 GB (4x8GB) PC3-17000 DDR4 Memoria 2133 MHz
1 x 256 GB de estado sólido SATA III Drive - OS Drive
1 x 128 GB de estado sólido SATA III Drive - Temporal / caché / DB Drive
1 x 2,0 TB 7200 RPM SATA III Hard Drive - datos de la unidad instalada en HotSwap Bay1
Nvidia GTX 750Ti 2GB 128 bits Tarjeta de vídeo PCI-Express GDDR5 con 1 VGA (D-Dub), 1 HDMI y 2 puertos DVI - soporta hasta 4 pantallas
22 "Widescreen LCD con altavoces integrados

Bloqueador contra escritura integrado con pantalla táctil:

IDE Drive Bloqueador
SATA Drive Bloqueador
SAS Drive Bloqueador
USB integrado 3.0 / 2.0 Bloqueador
FireWire IEEE 1394b Bloqueador

Clonadoras-Bloqueadoras

Para poder realizar el análisis forense es imprescindible asegurar que los datos de un disco duro o dispositivo USB no se alteren o modifiquen cuando se conecta a la unidad. Para ello existen los equipos write-blockers, que consiguen que los dispositivos conectados a los mismos –en especial discos duros– pasen a ser de sólo lectura, como ya lo son de origen los CD-ROM y DVD-ROM. Existen muchos modelos en el mercado, pero el más adecuado sería uno compatible con interfaces IDE, USB, SATA y opcionalmente con SCSI y FireWire (estándares antiguos de transferencia de datos).

Figura: Clonadora de discos con sistema write-bloqued. En este caso está realizando una copia fidedigna del contenido de un antiguo disco duro IDE en un nuevo disco duro SATA

Bloqueadores

Este tipo de dispositivos hardware se utilizan para conectar el dispositivo objeto de estudio a nuestra estación de trabajo impidiendo que el sistema pueda realizar cualquier tipo de escritura en el dispositivo, lo que lo invalidaría como prueba. Así mismo, las bloqueadoras también se utilizan para proteger contra escritura cualquier dispositivo de almacenamiento.

Figura:Bloqueador de escritura (Imagen ondatashop)

Estaciones de trabajo portatiles

FREDDIE (Digital Intelligence) www.digitalintelligence.com/products/freddie

Solución de procesamiento forense móvil, es una plataforma forense altamente integrado, flexible y modular diseñado desde cero, tanto para la adquisición y análisis de evidencia informática, con la ventaja añadida de ser muy portátil.

Figura: Estación de Trabajo portátil Freddie

Alguna de sus especificaciones:

– Intel Core i7-5820K CPU (Hex Core Processor), 3,3 GHz, 10 MB de Intel Smart Cache, 5 GT / s DMI
– 32 GB (4x8GB) PC3-17000 DDR4 Memoria 2133 MHz
– 1 x 256 GB de estado sólido SATA III Drive - OS Drive
– 1 x 128 GB de estado sólido SATA III Drive - Temporal / caché / DB Drive
– 1 x 2,0 TB 7200 RPM SATA III Hard Drive - datos de la unidad instalada en HotSwap Bay1
– Nvidia GTX 750Ti 2GB 128 bits Tarjeta de vídeo PCI-Express GDDR5 con 1 VGA
– (D-Dub), 1 HDMI y 2 puertos DVI - soporta hasta 4 pantallas

Bloqueador contra escritura integrado con pantalla táctil:

– IDE Drive Bloqueador
– SATA Drive Bloqueador
– SAS Drive Bloqueador
– USB integrado 3.0 / 2.0 Bloqueador
– FireWire IEEE 1394b Bloqueador

Ademas de estos dispositivos serán necesarios los cables para la conexión de los mismos. Existen gran variedad de conexiones de audio, video,alimentación y datos.

CREANDO NUESTRA ESTACION DE TRABAJO

A la hora de crear nuestra propia estación de trabajo hay que tener en cuenta algunas consideraciones previas, lugar de trabajo, condiciones ambientales, seguridad etc...En este proyecto no entraremos en este tipo de especificaciones si bien es importante tenerlo en cuenta.

Equipo Forense Base

Partiremos de un equipo básico aunque con gran capacidad de procesamiento (procesador de ultima generación así como gran cantidad de memoria RAM y Disco. Otra característica que debe poseer es el numero de conexiones, ya que en un principio no sabemos que vamos a tener que analizar.

Capacidad de proceso:
– Procesador de última generación

– 16GB de RAM

Almacenamiento:

– Sistema (>1TB)

– Trabajo (>2TB)

– Grabadora CD/DVD

Conexiones:

– IDE

– SCSI

– USB

– FireWire

– Lectores de Tarjetas de Memoria

Figura: equipo de análisis clonico

FastEthernet

Portátil

– Disco USB/FireWire

Equipo Forense Portátil.

En nuestro laboratorio vamos a disponer de un equipo portátil para extracción de de datos “in situ”. Ademas utilizaremos una clonadora de discos duros SATA de 2,5 o 3,5 pulgadas las cuales no requieren de una conexión con el ordenador host.

Figura: Clonadora de discos SATA 2,5 o 3,5

Figura:Dock Station Conexión de dos discos duros SATA III SSD/HDD de 2,5" o 3,5"

También sera necesario contar con todos los tipos de cableado para las diversas conexiones, tanto de suministro de energía como de datos.

-SATA

-USB

-FIREWIRE

-IDE

Para documentar todo el proceso nos sera de gran ayuda una cámara fotográfica. Las fotografiás realizadas podrán ser incluidas en los distintos informes. Así se podrá fotografiar la “escena” cuando el equipo analizar no se encuentre en el laboratorio (situación equipo, estado del dispositivo, aplicaciones visibles en pantalla,programas de descarga etc.).

5.2 Software.

- CAINE.(ver. 7.0)

Como software básico utilizaremos la distribución forense CAINE 7.0 .CAINE Linux es una distribución que agrega miles de paquetes de software libre y cumple con las Guías de Software Libre de Debian GNU/Linux.

- VIRTUALBOX.(ver. 5.0.10)

VirtualBox, software de virtualización para arquitecturas x86/amd64, creado originalmente por la empresa alemana innotek GmbH. Actualmente es desarrollado por Oracle Corporation como parte de su familia de productos de virtualización. Por medio de esta aplicación es posible instalar sistemas operativos adicionales, conocidos como «sistemas invitados», dentro de otro sistema operativo «anfitrión», cada uno con su propio ambiente virtual. (vía Wikipedia)

- LIBREOFFICE.(ver 5.0.3.2)

LibreOffice es un paquete de oficina libre y de código abierto desarrollado por The Document Foundation. Se creó como bifurcación de OpenOffice.org en 2010 (vía Wikipedia)

Listado de herramientas incluidas en la suite CAINE que utilizaremos:

Fmount, permite montar todos los volúmenes detectados asignados en una imagen de solo lectura en el directorio /media. Las particiones son nombradas con el formato "IMAGEN_VOLUMEN". Si la imagen está dividida, se deben especificar todos los segmentos. Las imágenes divididas son montadas primero como un disco virtual en /mnt/ antes de montarse las particiones.

GuyManager, nos permitirá capturar una imagen forense genera imágenes en bruto (dd), EWF (E01) y AFF, soportando clonación de disco.

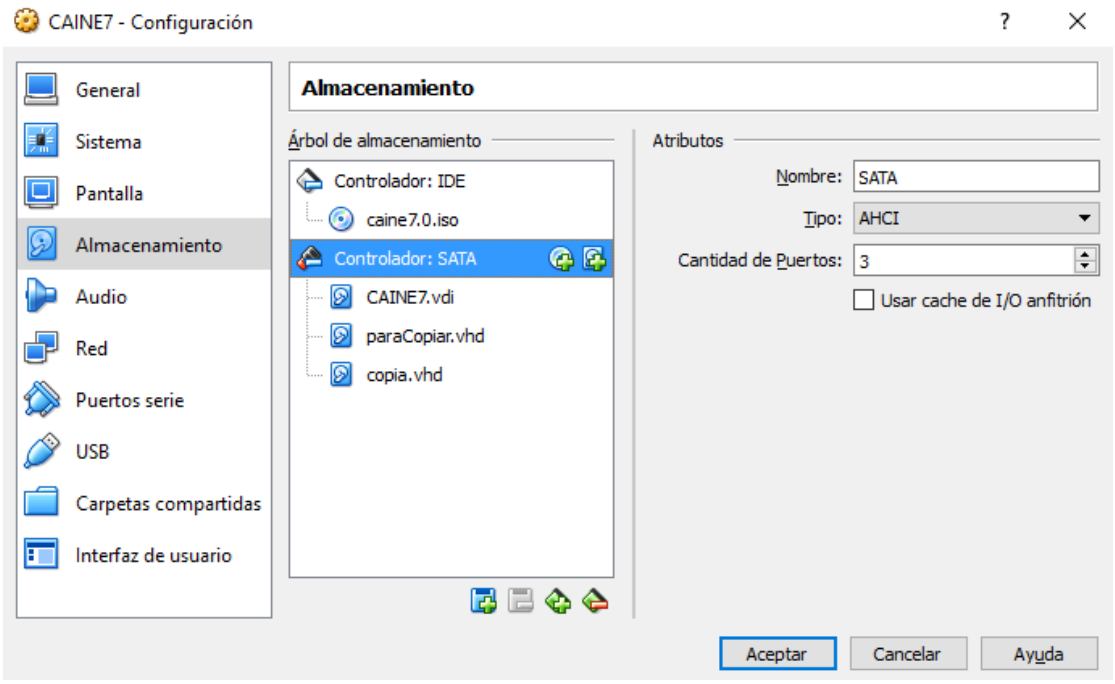
Autopsy, permite un análisis de diversos tipos de evidencia mediante una captura de una imagen de disco. Autopsy trabaja dividiendo cada investigación en casos. Cada caso puede contener uno o mas hosts, y cada uno de ellos puede a su vez contener una o varias imágenes de su sistema de ficheros. Por otra parte cada caso puede tener asignados uno o más investigadores.

Volatility, herramienta pensada para la extracción de los datos volátiles que están en una memoria RAM a partir de una imagen de un disco.

5.3 Metodología

Primero creamos una maquina virtual, con VirtualBox, y la ISO de CAINE. Antes de

encenderla iremos a su configuración (botón derecho -> configuración) y aquí vamos a la pestaña de almacenamiento. Una vez aquí añadiremos dos discos duros virtuales (VHD). El primero será el disco duro que queramos analizar, el segundo será otro en donde realizaremos la copia de seguridad que será la que analicemos (nunca se analiza el disco duro original). Como he comentado antes deberíamos hacer dos copias del original, en este caso haré solo una puesto que la segunda se haría igual.



Captura de pantalla del montaje de CAINE sobre VirtualBox.

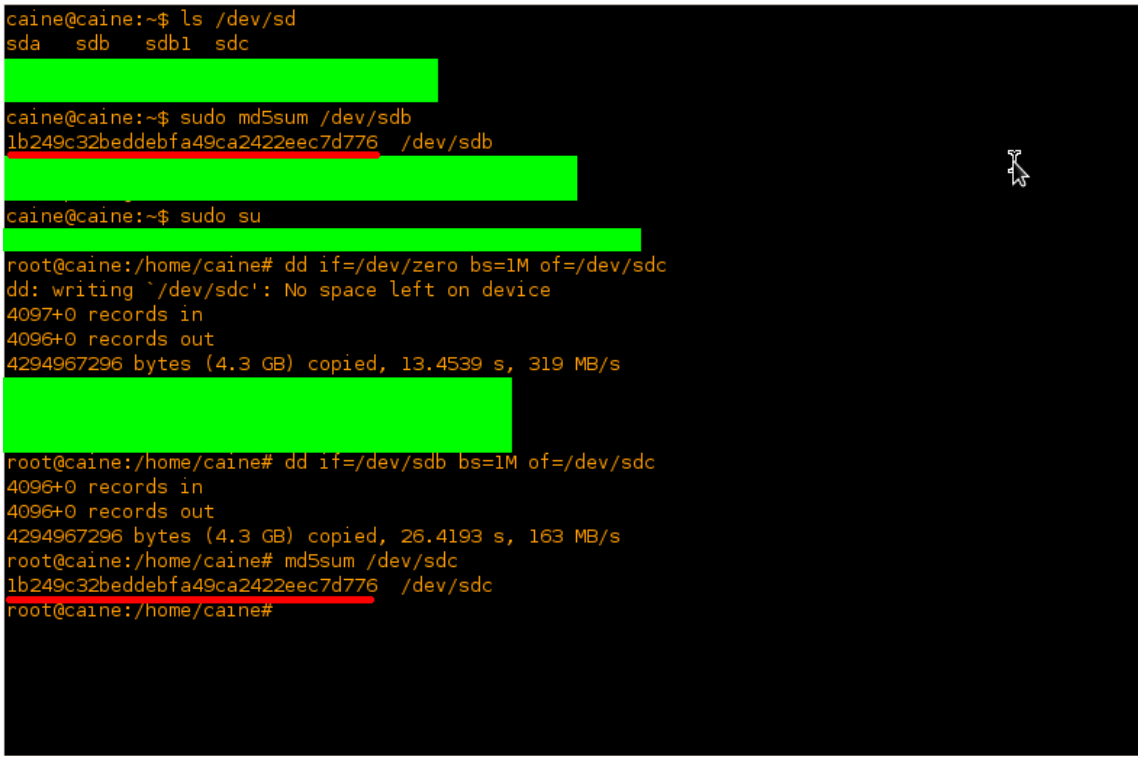
Una vez ya tenemos la maquina virtual con los 3 discos duros puestos, procederemos a encenderla. Lo primera será cambiar la distribución de teclado para ponerla en español. Para ellos en esta versión tenemos una aplicación para llevar esta tarea a cabo.

Después abriremos un terminal y con “ls /dev/sd” podremos ver los discos duros que tenemos en la maquina (sda será el propio de la distribución CAINE, sdb es el primero que hemos añadido en la configuración de almacenamiento “paraCopiar.vhd” y sdc es el segundo que hemos añadido “copia.vhd”). Lo primero que haremos será obtener un hash del disco duro sdb, para luego poder compararlo con el hash que hagamos al disco duro que contenga el clonado, si estos dos hashes no coinciden quiere decir que la copia no se ha realizado correctamente y habrá que volver a realizar la copia. Para tener más garantías de que la copia está bien hecha hay que sacar dos hashes del disco origen, otros dos del disco destino y comprobar que coinciden ambos.

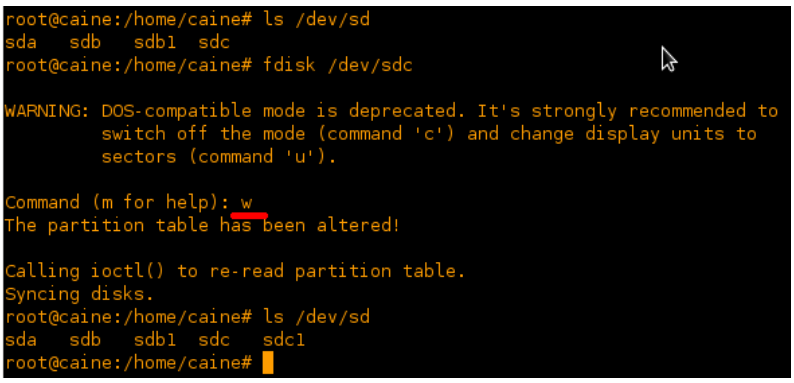
Con la herramienta DD haremos un borrado seguro del disco duro en el que se va a realizar la copia (para asegurarnos que no tiene datos de otro análisis anterior) lo que haremos será llenarlo de ceros. Para esto tenemos en linux “/dev/zero”, que lo pondremos como origen en la herramienta DD y en el destino pondremos el disco duro donde

queremos realizar la copia. Utilizaremos un block size (bs) de 1mega para que no tarde mucho la copia (el bs es el tamaño del bloque que coge la herramienta DD para ir copiando bloque a bloque). Una vez tenemos el disco destino lleno de ceros, habrá que hacer la copia del disco original que en este caso es sdb, al disco de destino que tenemos a

ceros. Por ultimo se mira el hash del disco sdc (la copia) a ver si coincide con el que obtuvimos anteriormente de sdb. En la siguiente imagen puedes ver todo el procedimiento:



Habíamos visto que sdb tiene una partición que es sdb1. Es decir que sdc debería de tener también la misma partición que sería sdc1 después de la copia. Lo que pasa que la maquina todavía no la ha reconocido, y para que esto suceda habrá que llamar a la herramienta fdisk e introducir la opción “w”. Y ya vemos como si que reconoce la maquina la partición de nuestro disco duro copia.



Por ultimo vamos a explicar como montar una partición para que puedas examinar el

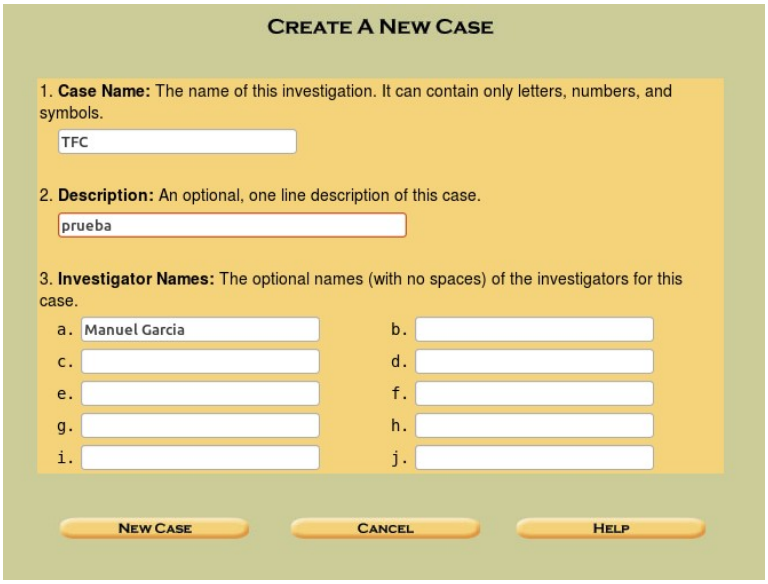
contenido del disco duro virtual que hemos clonado. En la foto he hecho la prueba con el sdb, pero sería exactamente igual para la copia sdc. Primero tendremos que crear una carpeta en donde montaremos el disco duro. A continuación el comando “mount” le indicaremos que partición queremos montar y en que carpeta queremos montarla. De esta

forma cuando entremos a la carpeta podremos ver el contenido del disco duro. Para desmontar la carpeta solo tendremos que escribir el comando "umount" seguido del nombre de la carpeta.

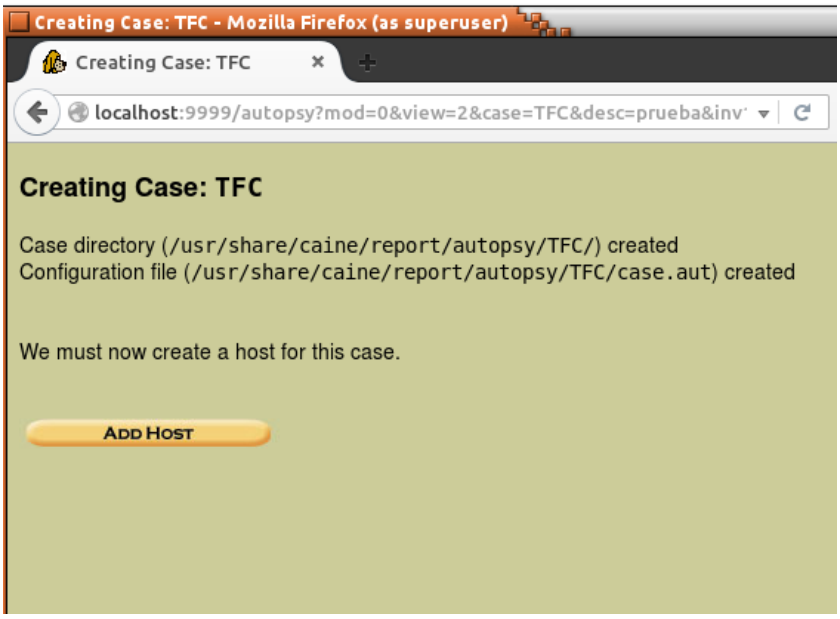
```
root@caine:/home/caine# ls /dev/sd
sda   sdb   sdb1  sdc   sdc1
root@caine:/home/caine# mkdir prueba
root@caine:/home/caine# mount /dev/sdb1 prueba/
root@caine:/home/caine# ls prueba/
131371_BDGNMRXDBJYLRTL.jpg  fhfh.txt  images2.jpeg  imag.jpeg   p
691307976_dad8d4da3f        fich.txt  images.jpeg   lost+found
root@caine:/home/caine# cd prueba/
root@caine:/home/caine/prueba# ls
131371_BDGNMRXDBJYLRTL.jpg  fhfh.txt  images2.jpeg  imag.jpeg   p
691307976_dad8d4da3f        fich.txt  images.jpeg   lost+found
root@caine:/home/caine/prueba# umount prueba/
umount: prueba/: not found
root@caine:/home/caine/prueba# cd ..
root@caine:/home/caine# umount prueba
root@caine:/home/caine#
```

Autopsy

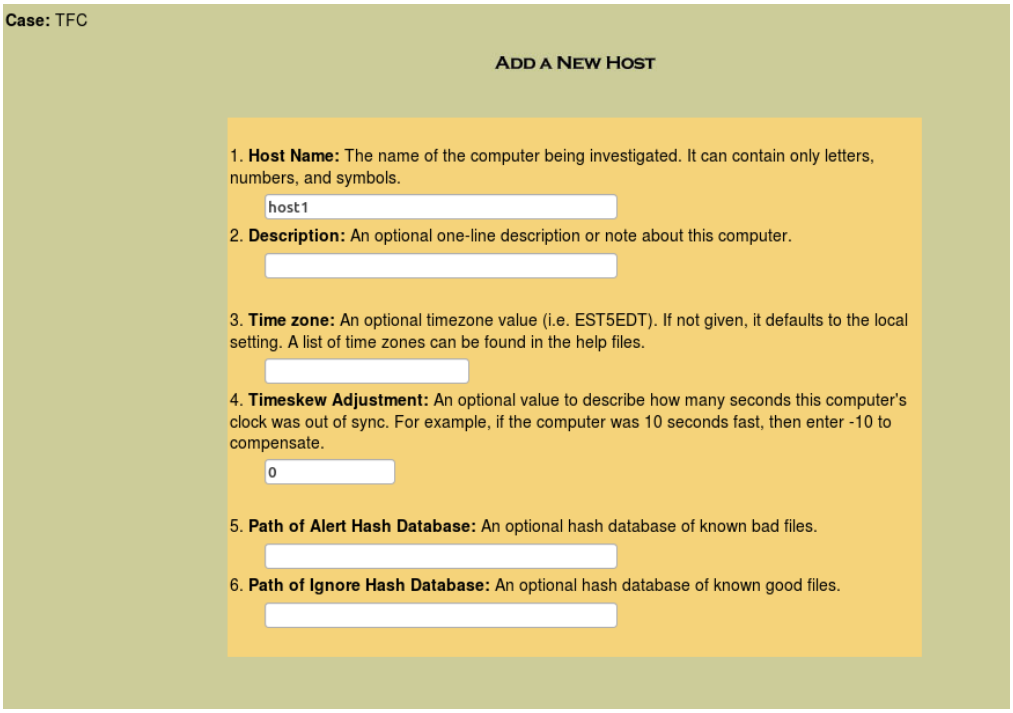
Para ello vamos a crear el primer caso pulsando sobre "New Case". Aparecerá una nueva pantalla donde introduciremos la siguiente información:



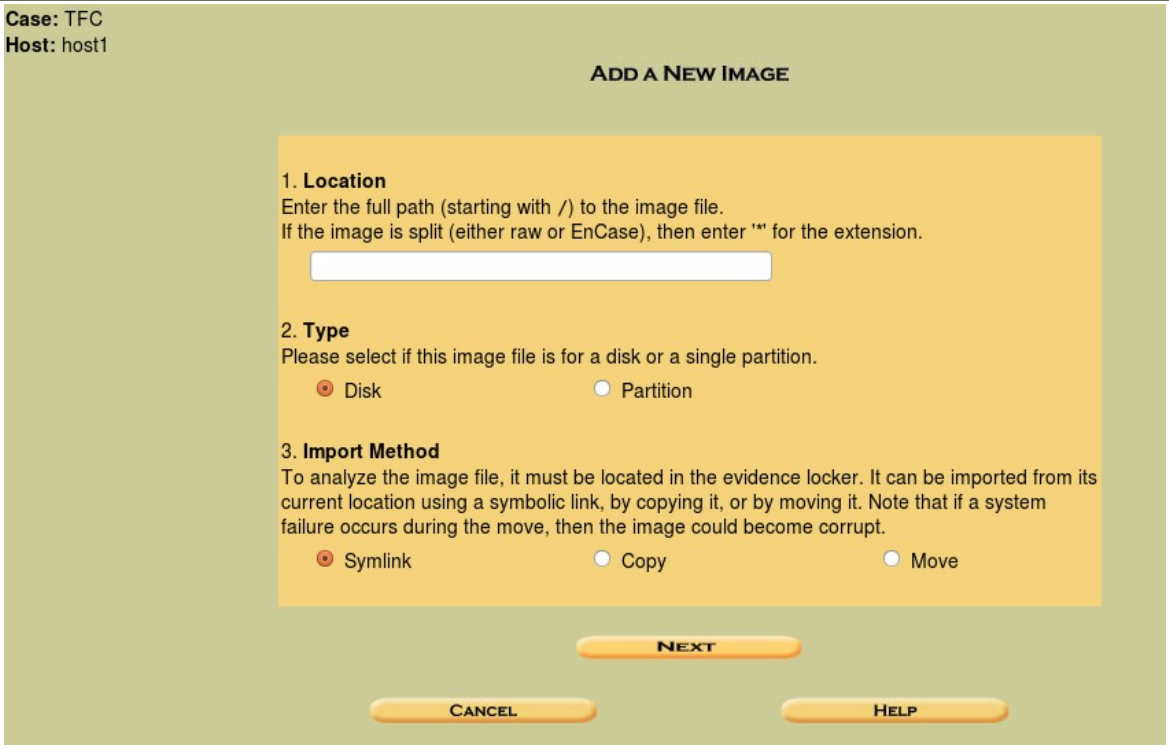
Los campos opciones como son los nombres de los investigadores los dejaremos en blanco. Una vez completada la información pulsaremos sobre "New Case". Como resultado se creará una carpeta con el nombre "prueba" en el directorio escogido durante la instalación de autopsy para almacenar las investigaciones.



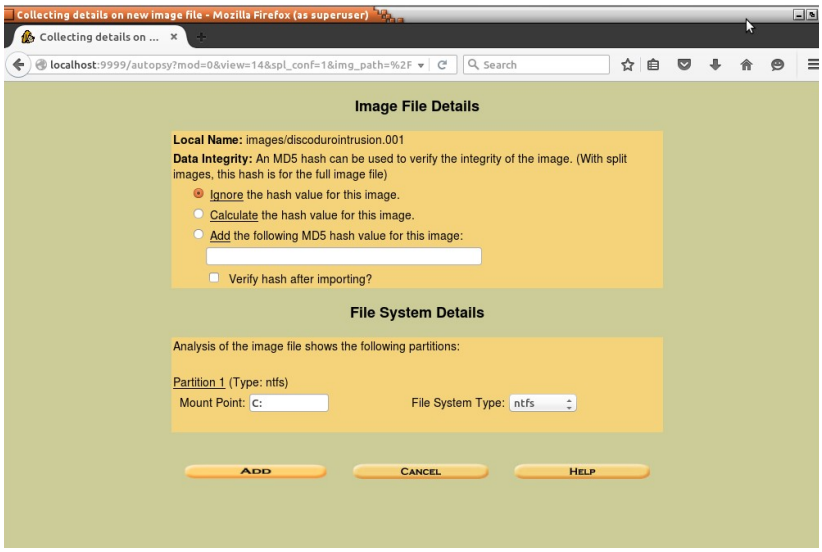
Ahora deberemos agregar al menos un host al caso. Ahora pasaremos a montar la imagen mediante el dispositivo de loopback de forma similar a como puede hacerse con una ISO. Pero primero deberemos tener claro el sistema de ficheros utilizado en la partición.



Añadimos la imagen que queremos analizar.

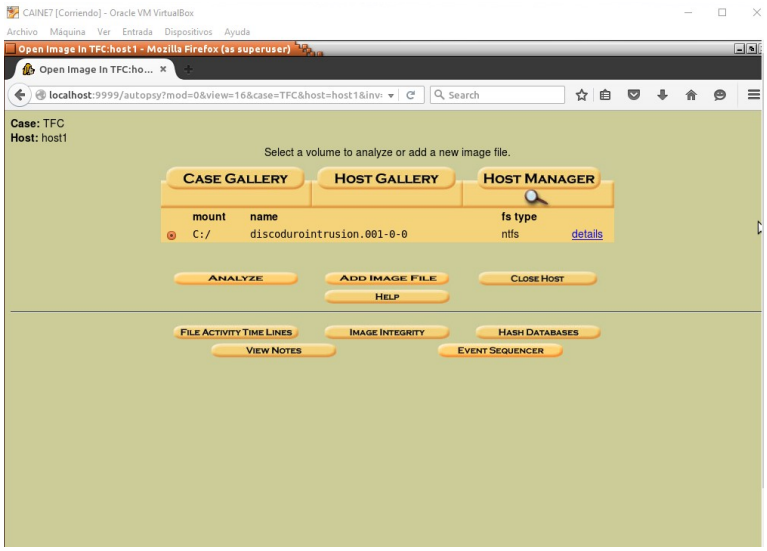


Escribiremos la ruta del disco duro que tenemos en formato .dd y le indicamos que es una partición, si fuese un disco entero pincharíamos en la opción "disk". A la hora de importar tenemos 3 métodos. Los 2 más utilizados son el primero, que trabaja directamente sobre la copia que le pasemos y el segundo que trabaja la copia que realiza la propia herramienta. Aquí podemos pedirle que nos calcule el hash MD5 del disco a analizar o ponerle nosotros uno para verificar que se ha realizado la copia correctamente. En nuestro caso el hash del original y el de la copia lo hemos comprobado anteriormente por nuestra cuenta.

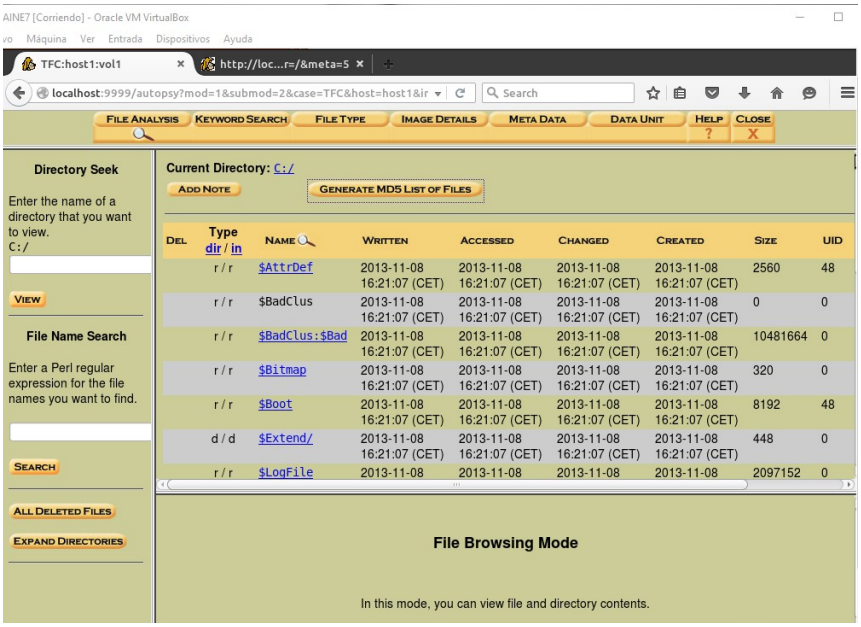


Vemos como nos ha reconocido que se trata de un sistema de ficheros NTFS y asume como punto de montaje C:

Una vez añadida la imagen se procederá analizarla. La primera de las pestañas que tiene la herramienta es la de analizar los ficheros. Aquí nos aparecen también los archivos borrados. Hay una carpeta que se llama $OrphanFiles que es donde se guarda todo lo que se elimina. La segunda pestaña es para buscar cadenas en ASCII y Unicode en toda la imagen. La tercera pestaña es para que te ordene y clasifique todos los ficheros de la imagen por su tipo. La cuarta pestaña es para ver detalles de la imagen como el tipo de archivos o la versión del sistema operativo (siempre que sea posible) entre otras cosas.
La ultima pestaña es para buscar el cluster que te interese a partir de su número.



Procedemos con la opción Analyze.



Seguidamente se deberá dar es crear una línea temporal de sucesos o timeline, para que recopile la siguiente información sobre los ficheros:

- Inodos asociados.
- Marcas de tiempo MACD (fecha y hora de modificación, acceso, creación y borrado).
- Ruta completa.
- Tamaño en bytes y tipo de fichero.
- Usuarios y grupos a quien pertenece.
- Permisos de acceso.
- Si fue borrado o no.

La idea es buscar ficheros y directorios que han sido creados, modificados o borrados recientemente, o instalaciones de programas posteriores a la del sistema operativo y que además se encuentren en rutas poco comunes. Hay que pensar que los atacantes y sus herramientas crearán directorios y descargarán sus "aplicaciones" en lugares donde no se suele mirar, como por ejemplo en los directorios temporales.

Otras opciones:

-Archivos: Permite navegar el sistema de archivos y visualizar contenidos.
-Meta Datos: Permite examinar las estructuras de metadatos.
-Unidades de Datos: Permite navegar por número de bloque.
-Búsqueda de Palabras Clave: Busca una cadena utilizando grep(1).
-Detalles de la Imagen: Detalles sobre el sistema de archivos o volumen.
-Integridad de la Imagen: Se puede verificar en cualquier momento.
-Cronología sobre Actividad de los Archivos: Cronologías en Base a tiempos (MAC) Modificado, Accedido, Cambiado (Creado en FAT/NTFS).
-Categorías por Tipo de Archivo: Ordenar archivos basado en su tipo.
-Generación de Reporte: Cada una de las técnicas permite generarlo
-Tiempo de Modificación: Existe en sistemas de archivos UNIX y NTFS. Muestra la última vez en el cual se modificó el archivo de datos. En otras palabras, cuando fueron por última vez escritos datos hacia las unidades de datos asignadas para el archivo.
-Tiempo de Escritura: Existe para sistemas de archivos FAT y es el tiempo cuando el archivo fue escrito por última vez. De los tres tiempo, este es el único valor requerido por la especificación FAT.
-Tiempo de Acceso: Contiene el tiempo del último acceso del archivo de datos. Sobre una imagen FAT, este valor es opcional y es solo preciso al día (no horas y segundos).
-Tiempo de Cambio: Existe para sistemas de archivos UNIX y NTFS. Es la última vez en el cual se cambió estado del archivo (o metadatos). Esto es diferente al tiempo de modificación, el cual trata con el archivo de datos, y este trata con los datos descriptivos en el inodo o entrada MFT.
-Tiempo de Creación: NTFS y FAT. Cuando el archivo fue creado.

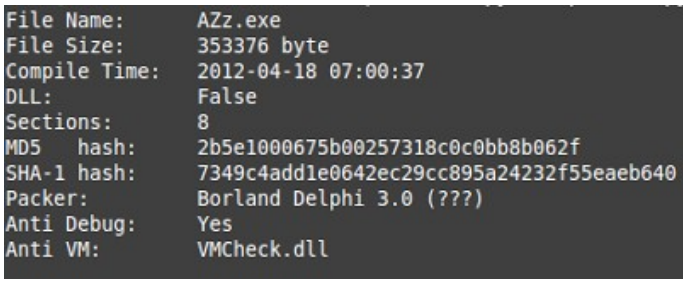
Este modo permite al investigador visualizar los detalles de las estructuras de metadatos. Las estructuras de metadatos sobre las estructuras del disco los cuales contienen los detalles del archivo, como tiempos y punteros hacia las unidades de datos asignadas. Los sistemas de archivos FFS y EXT2FS los llama estructuras inodos, los sistemas de archivos NTFS los llama entradas MFT (Master File Table) o Entradas de de Archivo, y el sistema de archivos FAT los llama entradas de directorios.

Para visualizar el contenido de una estructura únicamente se debe ingresar su dirección. También es factible visualizar el estado de asignación de estructuras de

metadatos en grupos de 500. Con esta opción podremos visualizar el contenido de unidades de datos individuales. Las unidades de datos son un termino genérico utilizado para describir las áreas del disco utilizadas para almacenar datos. Cada sistema de archivos nombra de manera diferente a una unidad de datos (Por ejemplo, Fragmentos o Clusters), es muy útil cuando se requiere recuperar y analizar datos borrados.El contenido de las unidades de datos pueden ser visualizados en formatos de cadenas, volcado hexadecimal, o ASCII.

Es factible también mostrar la dirección y nombre de archivo asignada a la unidad, encontrando su estructura de metadatos.

Un de los problemas que nos podemos encontrar a la hora de realizar un análisis con maquinas virtuales es que la muestra obtenida sea capaz de detectar un entorno virtual y no poder ejecutarla. Existen scripts que permiten analizar los binarios para saber si tienen protección de máquina virtual. Peframe (https://github.com/guelfoweb/peframe) es uno de ellos.



Como podemos observar chequea el archivo VMCheck.dll el cual detecta que se trata de una maquina virtual creada con VirtualBox.

Una posible solución es modificar ciertos aspectos de la maquina virtual para que no se pueda detectar. Crearemos una maquina virtual (para este ejemplo crearemos un MV con Windows 7). Posteriormente modificaríamos ciertos aspectos de la maquina creada

```
@reg copy HKLM\HARDWARE\ACPI\DSDT\VBOX__ HKLM\HARDWARE\ACPI\DSDT\WOOT__ /s /f
@reg delete HKLM\HARDWARE\ACPI\DSDT\VBOX__ /f
@reg copy HKLM\HARDWARE\ACPI\RSDT\VBOX__ HKLM\HARDWARE\ACPI\RSDT\WOOT__ /s /f
@reg delete HKLM\HARDWARE\ACPI\RSDT\VBOX__ /f
@reg copy HKLM\HARDWARE\ACPI\FADT\VBOX__ HKLM\HARDWARE\ACPI\FADT\WOOT__ /s /f
@reg delete HKLM\HARDWARE\ACPI\FADT\VBOX__ /f
@reg copy HKEY_LOCAL_MACHINE\HARDWARE\ACPI\DSDT\WOOT__\VBOXBIOS
HKEY_LOCAL_MACHINE\HARDWARE\ACPI\DSDT\WOOT__\WOOTBIOS /s /f
@reg delete HKEY_LOCAL_MACHINE\HARDWARE\ACPI\DSDT\WOOT__\VBOXBIOS /f
@reg copy HKEY_LOCAL_MACHINE\HARDWARE\ACPI\DSDT\WOOT__\VBOXBIOS
HKEY_LOCAL_MACHINE\HARDWARE\ACPI\DSDT\WOOT__\WOOTBIOS /s /f
@reg delete HKEY_LOCAL_MACHINE\HARDWARE\ACPI\DSDT\WOOT__\VBOXBIOS /f
@reg copy HKEY_LOCAL_MACHINE\HARDWARE\ACPI\FADT\WOOT__\VBOXFACP
HKEY_LOCAL_MACHINE\HARDWARE\ACPI\FADT\WOOT__\WOOTFACP /s /f
@reg delete HKEY_LOCAL_MACHINE\HARDWARE\ACPI\FADT\WOOT__\VBOXFACP /f
@reg copy HKEY_LOCAL_MACHINE\HARDWARE\ACPI\RSDT\WOOT__\VBOXRSDT
HKEY_LOCAL_MACHINE\HARDWARE\ACPI\RSDT\WOOT__\WOOTRSDT /s /f
@reg delete HKEY_LOCAL_MACHINE\HARDWARE\ACPI\RSDT\WOOT__\VBOXRSDT /f
@reg copy HKEY_LOCAL_MACHINE\HARDWARE\ACPI\DSDT\WOOT__\VBOXBIOS
HKEY_LOCAL_MACHINE\HARDWARE\ACPI\DSDT\WOOT__\WOOTBIOS /s /f
@reg delete HKEY_LOCAL_MACHINE\HARDWARE\ACPI\DSDT\WOOT__\VBOXBIOS /f
@reg copy HKEY_LOCAL_MACHINE\HARDWARE\ACPI\FADT\<VENDOR>\VBOXFACP
HKEY_LOCAL_MACHINE\HARDWARE\ACPI\FADT\<VENDOR>\WOOTFACP /s /f
@reg delete HKEY_LOCAL_MACHINE\HARDWARE\ACPI\FADT\<VENDOR>\VBOXFACP /f



@reg copy HKEY_LOCAL_MACHINE\HARDWARE\ACPI\RSDT\<VENDOR>\VBOXRSDT
HKEY_LOCAL_MACHINE\HARDWARE\ACPI\RSDT\<VENDOR>\WOOTRSDT /s /f
```

```
@reg delete HKEY_LOCAL_MACHINE\HARDWARE\ACPI\RSDT\<VENDOR>\VBOXRSDT /f
@reg add HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION\System /v SystemBiosVersion /t REG_MULTI_SZ /d "WOOT
-1" /f
@reg add HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION\System /v VideoBiosVersion /t REG_MULTI_SZ /d "VGA
BIOS v1.54" /f
```

Se procede apagar la maquina y ejecutar el siguiente .bat:

```
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiBoardVendor" "Fujitsu"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiBoardProduct" "Fujitsu"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiBoardVersion" "1.12"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiBoardSerial" "D461561561>"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiBoardAssetTag" "Fujitsu"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiBoardLocInChass" "Fujitsu"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiChassisVendor" "Fujitsu"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiChassisVersion" "Fujitsu"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiChassisSerial" "D44445115"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiChassisAssetTag" "Fujitsu"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSVendor" "<FUJITSU // Phoenix
Technologies Ltd.>"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSVersion" "<1.18>"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSReleaseDate" "<03/12/2012>"
#VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSReleaseMajor" <03/09/2012>
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSReleaseMinor" <03/06/2012>
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSFirmwareMajor" <03/11/2012>
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSFirmwareMinor" <03/04/2012>
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiSystemVendor" "<Fujitsu>"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiSystemProduct" "<Fujitsu>"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiSystemVersion" "<Fujitsu>"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiSystemSerial" "<DSBW011977>"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiSystemSKU" "DSBW014878"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiSystemFamily" "<"2.25">"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/pcbios/0/Config/DmiSystemUuid" "2sfsdfsdfC-FsfsdfA8-Esfsdf1-
8B14-5C9AD8sfsdfsdfsdfE0"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/piix3ide/0/Config/PrimaryMaster/SerialNumber" "Dsdfsdfsdfsdf8"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/piix3ide/0/Config/PrimaryMaster/FirmwareRevision" "2.50"
VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/piix3ide/0/Config/PrimaryMaster/ModelNumber" "Fujitsu"
```

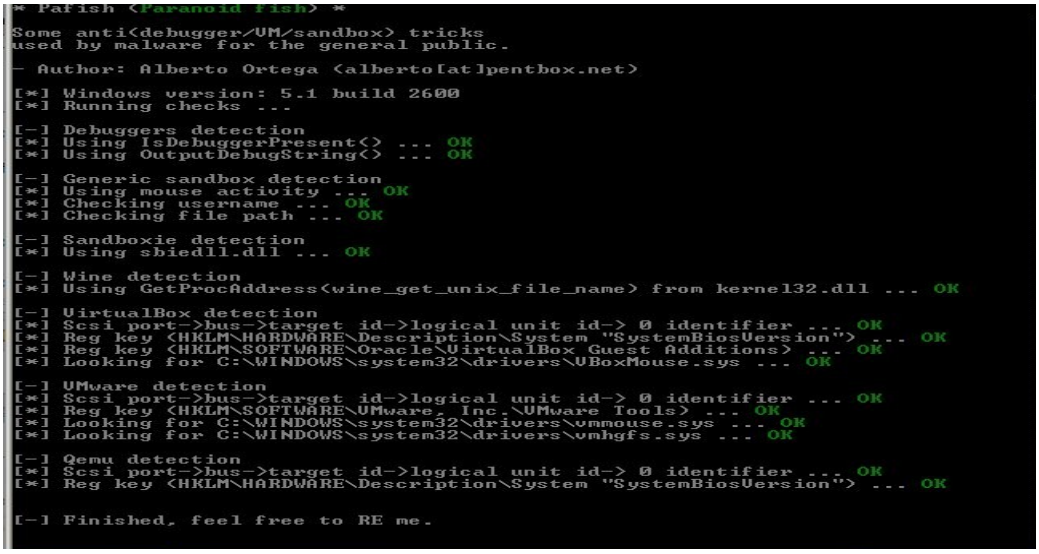
En lugar de "XP-VM" ha de ser el nombre que le hayamos dado a la máquina virtual.

Ahora tendremos que hacer unos últimos pasos:

```
caine@caine ~ $ sudo dd if=/sys/firmware/acpi/tables/SLIC of=SLIC.bin
caine@caine ~ $ mv SLIC.bin /home/marc/VirtualBox\ VMs/XP-VM/
caine@caine ~ $ VBoxManage setextradata "XP-VM" "VBoxInternal/Devices/acpi/0/Config/CustomTable" /home/marc/VirtualBox\
VMs/XP-VM/SLIC.bin
```

A continuacion ejecutamos el scritp desarrollado en pyhton *peframe*[3]:

3 Peframe nos permite analizar un ejecutable PE y sacar información útil como los 'strings' del binario o la lista de DLLs que importa y las funciones que llama.



HONEYPOTS

Un honeypot no es más que una aplicación, servicio o sistema que simula ser lo que no es. No tiene un valor productivo para quien lo implanta y está preparado para ser sondeado, atacado y comprometido. Es básicamente un señuelo con el objeto de engañar al atacante que pretenda amenazar nuestros sistemas, al mismo tiempo que ayuda a entender las técnicas de ataque utilizadas.

Por estas razones, es lógico pensar que cualquier actividad que se genere desde o hacia un honeypot, será muy probablemente una actividad ilegítima o no autorizada. Existen diversas formas de clasificar los honeypots. Aquí lo haremos basándonos en dos de sus propiedades principales: la localización concreta dentro de una red y la interacción que permite con el atacante.

Según la LOCALIZACIÓN, pueden situarse en un entorno de:

Producción: El objetivo que se pretende alcanzar al implantar un honeypot en una red en producción no es otro que la obtención de información sobre las técnicas empleadas para tratar de vulnerar los sistemas que componen dicha infraestructura.
El abanico de posibilidades que nos ofrece un honeypot en una red en producción es muy amplio. Desde la posibilidad de ubicar el honeypot en el segmento de la red de servidores internos de la compañía, con el objetivo de detectar posibles accesos por parte de usuarios internos a recursos críticos de la organización (por ejemplo al fichero de nóminas), hasta la publicación de un servicio web con idéntica configuración y diseño que el mismo servicio que está en producción o preproducción.

El mayor inconveniente que supone esta elección es el peligro que supone para los sistemas organizativos el permitir (incluso provocar) que el tráfico malintencionado conviva con el legítimo.

Investigación: En este caso, el principal objetivo es la recopilación de la mayor

cantidad de información que permita al investigador poder analizar las nuevas tendencias en los métodos de ataque, así como los principales objetivos perseguidos y los distintos orígenes de los ataques. El resultado de este análisis es recogido en informes cuyo

objetivo es respaldar la toma de decisiones en la implantación de las medidas de seguridad preventivas.

La principal ventaja de situar el honeypot en una red independiente, dedicada únicamente a la investigación, es la separación del sistema vulnerable del resto de sistemas productivos y evitar así la posibilidad de sufrir un ataque a través del propio honeypot. Por el contrario, el inconveniente es la cantidad de recursos necesarios.

Otro método de clasificación de los "tarros de miel" es el que define la INTERACCIÓN con el atacante. En este caso, los honeypots se agrupan en dos tipos:

1. <u>Baja Interacción</u>: El honeypot emula un servicio, una aplicación o un sistema vulnerable. Sus características principales son su sencilla instalación y configuración, junto con lo limitado de su capacidad para obtener diferentes tipos de datos. Unos ejemplos de honeypots de este tipo son:

Honeyd: Quizás uno de los honeypots más sencillos y populares. Es un demonio que crea hosts virtuales en una red. Los anfitriones pueden ser configurados para ejecutar servicios arbitrarios, y su comportamiento puede ser adaptado para que simule estar en ejecución en ciertos sistemas operativos.

HoneyC: El objetivo de este honeypot es la identificación de servidores web maliciosos en la red. Para ello emula varios clientes y recaba la mayor cantidad posible de información de las respuestas de los servidores cuando estos contestan a sus solicitudes de conexión. HoneyC es ampliable de diversas formas: pueden utilizarse diferentes clientes, sistemas de búsqueda y algoritmos de análisis.

Nephentes: Honeypot de baja interacción que pretende emular vulnerabilidades conocidas para recopilar información sobre posibles ataques. Nepenthes está diseñado para emular vulnerabilidades que los gusanos utilizan para propagarse y cuando estos intentan aprovecharlas, captura su código para su posterior análisis.

Honeytrap: Este honeypot está destinado a la observación de ataques contra servicios de red. En contraste con otros honeypots, que se suelen centrar en la recogida de malware, el objetivo de Honeytrap es la captura de exploits.

Glastopf: Emula miles de vulnerabilidades para recopilar datos de los ataques contra aplicaciones web. La base para la recolección de información es la respuesta correcta que se le ofrece al atacante cuando intenta explotar la aplicación web. Es fácil de configurar y una vez indexado por los buscadores, los intentos de explotación de sus vulnerabilidades se multiplican.

2. <u>Alta Interacción</u>: En este caso el honeypot es una aplicación con la cual se puede interactuar y que responde como se espera, con la diferencia de que su diseño está orientado a realizar un registro exhaustivo de la actividad que se lleva a cabo sobre ella y de que la información que contiene no es relevante en ningún caso.

HI-HAT (High Interaction Honeypot Analysis Toolkit): Herramienta que transforma aplicaciones php en aplicaciones honeypot de alta interacción. Además ofrece una interfaz web que permite consultar y monitorizar los datos registrados.

HoneyBow: Herramienta de recopilación de malware que puede integrarse con el honeypot de baja interacción Nephentes para crear una herramienta de recolección mucho más completa.

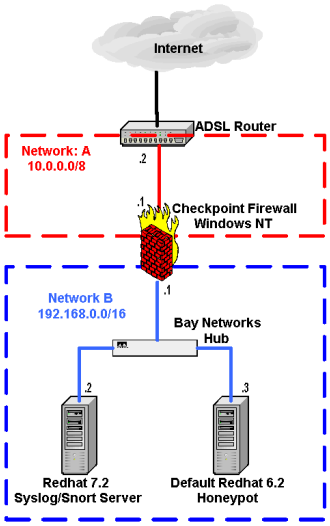
Sebek: Funciona como un HIDS (Host-based Intrusion Detection System) permitiendo capturar una gran variedad de información sobre la actividad en un sistema ya que actúa a muy bajo nivel. Es una arquitectura cliente-servidor, con capacidad

multiplataforma, que permite desplegar honeypots cliente en sistemas Windows, Linux, Solaris, *BSD, etc., que se encargan de la captura y el envío de la actividad recopilada hacia el servidor Sebek. Podríamos decir que forma parte de una tercera generación de honeypots.

Capture-HPC: Del tipo cliente, como HoneyC, identifica servidores potencialmente maliciosos interactuando con ellos, utilizando una máquina virtual dedicada y observando cambios de sistema no previstos o autorizados.

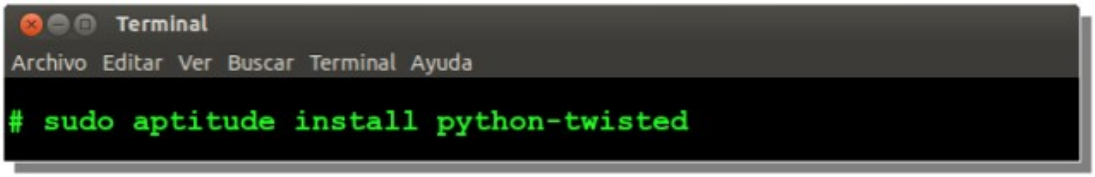
Otra opción a destacar en cuanto a la elección de un honeypot es la de la web Project HoneyPot (se desarrollara en el siguiente punto). Se trata de un portal que facilita un recurso web para usarlo como honeypot. Este genera una página con un código script, la cual utiliza diversas técnicas para la recopilación de información (IPs, logins usados en ataques de fuerza bruta, spammers, etc…).

Para nuestro caso vamos a emular un servidor SSH que se despliega con la intención de recopilar información de los diversos ataques que existen contra este servicio. Un honeypot emula un servicio vulnerable, en caso de Kippo el de SSH pero los hay también para otros servicios como FTP o web, con el fin de registrar la interacción del atacante. De esta manera, se puede tener constancia de la técnica y el tipo de ataques que se llevan a cabo. El honeypot puede ser de baja interacción, si emula un servicio no existente, o de alta interacción, si trabaja sobre un servicio real. Kippo es de los primeros.

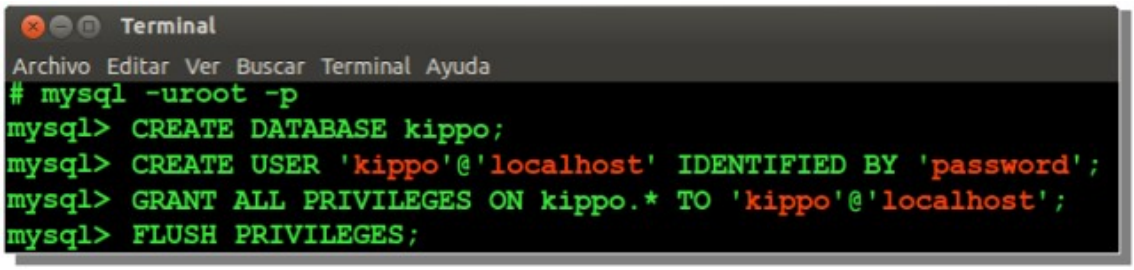


Instalación y configuración

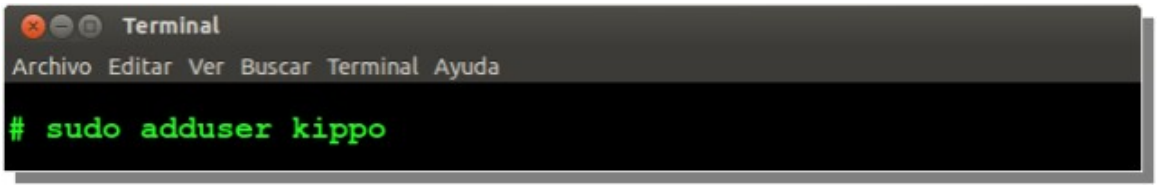
Primero instalaremos las dependencias:



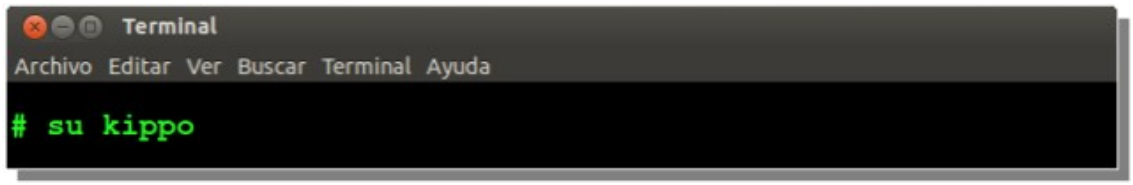
Creamos un usuario y una base de datos en MySQL para guardar los ataques:



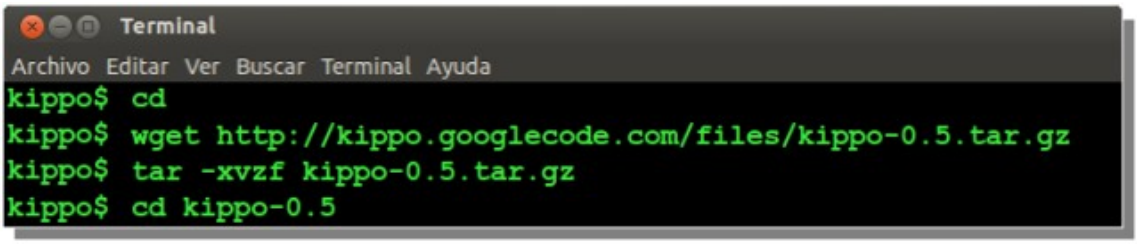
Crearemos un usuario sin privilegios en el sistema para ejecutar el honeypot:



Cambiamos de usuario:



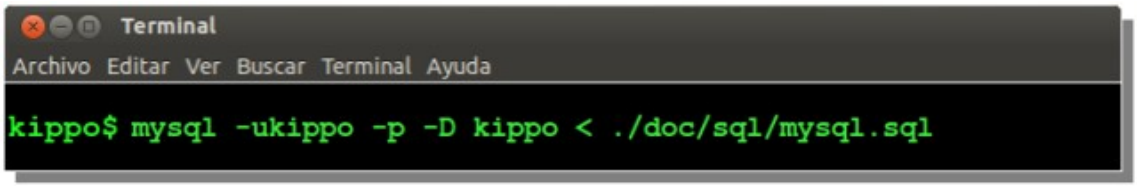
Descargamos el código y lo descomprimimos:



En este directorio podemos encontrar:

-*dl/*: Donde guardaremos los ficheros descargados mediante wget
-*log/kippo.log*: Se guarda la información de uso y depuración.
-*log/tty/*: logs de las sesiones.
-*utils/playlog.py*: herramienta para reproducir los logs de sesión.
-*utils/createfs.py*: utilizado para crear fs.pickle.
-*fs.pickle*: falso sistema de ficheros.
-*honeyfs/*: contenido del falso sistema de ficheros. Aquí podemos poner una copia de un sistema real.

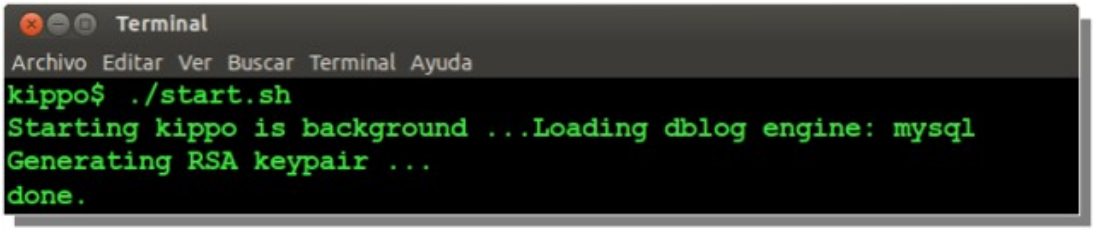
Creamos la estructura de la base de datos mediante script proporcionado:



Añadimos la configuración de MYSQL al final del archivo de configuración de Kippo, *kippo.cfg*:

```
[database_mysql]
host = localhost
database = kippo
username = kippo
password = password
```

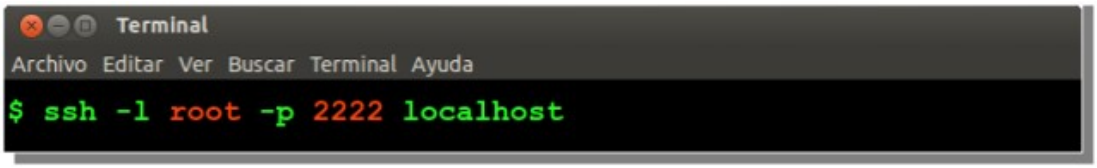
Para arrancar el honeypot:



Controlando la actividad

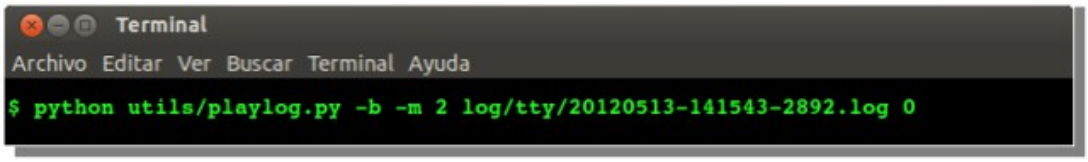
Podemos comprobar que el honeypot está a la escucha ejecutando:

Podemos hacer las primeras pruebas desde la máquina local. El usuario es root y la contraseña 123456:



Una opción interesante es reproducir la sesión de un usuario mediante el script *playlong.py*

Por ejemplo:

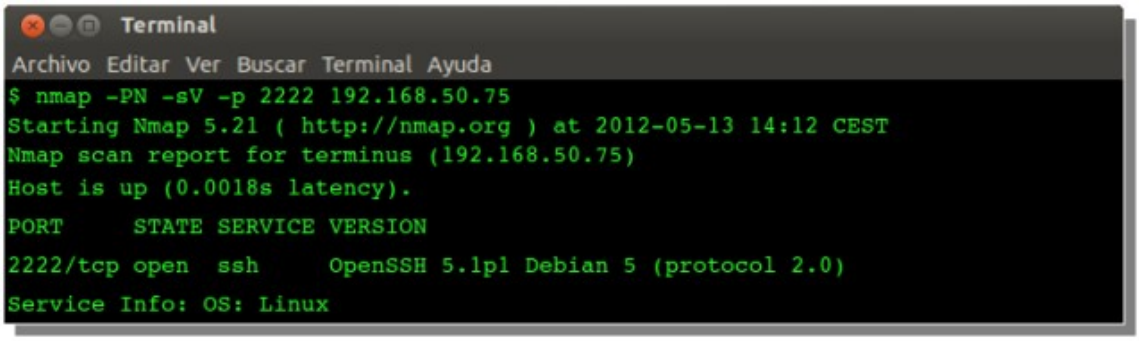


Acceso desde el exterior

El honeypot se ejecuta en el puerto 2222, por defecto, por lo que deberemos crear una redirección desde el puerto 22 (para que se ejecutase en el puerto 22 debería tener privilegios de administrador, y esto es algo que no queremos). Para redirigir el puerto podemos utilizar la NAT del router, o utilizar iptables si queremos que a redirección se lleve a cabo en el propio equipo:

Si estamos utilizando algún tipo de cortafuegos, por ejemplo *ufw,* deberemos crear una regla para permitir el acceso:

Para comprobar que se puede establecer la conexión podemos utilizar *nmap*:



DISTRIBUCION HONEYDRIVE

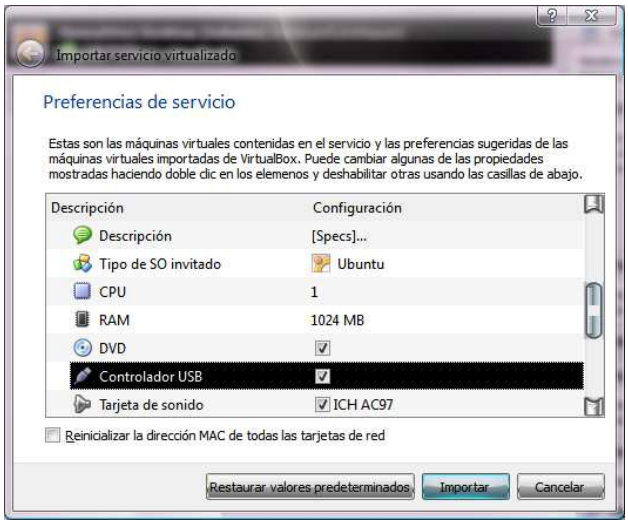
Para nuestro proyecto vamos a utilizar una distribución linux llamada HONEYDRIVE en una maquina virtual ya preparada para funcionar con VIRTUALBOX.

Una vez descargado lo instalamos, no creo que haga falta explicar cómo instalar VirtualBox en nuestra computadora.

Una vez instalado nos vamos a bajar la máquina virtual Honeydrive que alberga nuestros Honeypot: *http://bruteforce.gr/honeydrive*

Entre sus características podemos encontrar:
· Corre con Xubuntu Desktop 12.04 32 bits
· Servidor LAMP con PHPMyAdmin
· Kippo SSH, Kippo-Graph y Kippo2MySQL
· Dionea Malware + phpLiteAdmin
· Honeyd + Honeyd2MySQL y Honeyd-Viz
· LaBrea, Tiny Honeypot, IIS Emulator, INetSim y SimH
· Varias utilidades para el análisis de malware, PDFs, etc

La imagen ocupa más de 2 Gb y procederemos a su instalación bajo Virtual Box (también pude correr bajo VMWare). Una vez descargado todo lo necesario lo pondremos en marcha con la siguiente configuración:



Ahora importará la imagen de la máquina virtual. Una vez importada se debe configurar de red y poner la placa de red en modo promiscuo. Puede que informe sobre que la tarjeta de red no es la misma. En este caso simplemente se debe aceptar y poner la que tenga el sistema.

Le damos a iniciar la máquina virtual y esperamos a que cargue la imagen de Honeydrive. Cuando el proceso termine se debe ver esta pantalla.

La contraseña es honeydrive. Luego de ingresar recomiendo cambiar la contraseña a través del comando.

Lo primero que vamos a realizar es abrir el fichero *readme.txt*. En este archivo nos encontramos una aclaración de los honeypots que podemos utilizar y la ubicación de los archivos de configuración de cada uno de ellos. Esto va a ayudar mucho en nuestra labor de configuración.

README.txt

File Edit Search Options Help

```
[Specs]
OS: Xubuntu Desktop 12.04 32-bit
HDD: VDI 16GB (dynamically allocated)
Localization: English (United Kingdom)
Timezone: Europe/London (GMT)
Keyboard layout: English (United States)

[System]
Connectivity: DHCP
Hostname: honeydrive
User: HoneyDrive
Username/Password: honeydrive/honeydrive

[LAMP]
Apache 2, + support: PHP, Perl, Python, Ruby
MySQL root password: honeydrive

[Kippo]
Script: /opt/kippo/start.sh
Downloads: /opt/kippo/dl/
TTY logs: /opt/kippo/log/tty/
Credentials: /opt/kippo/data/userdb.txt
MySQL database: kippo
MySQL user/password: root/honeydrive

[Kippo-Graph]
Location: /var/www/kippo-graph/
Config: /var/www/kippo-graph/config.php
URL: http://local-or-remote-IP-address/kippo-graph/
MySQL database: kippo
MySQL user/password: root/honeydrive
```

Nosotros vamos a poner en marcha dos honeypots, Dionaea y Kippo. Creo que con estos dos tenemos más que suficiente y cubre nuestras necesidades. Un consejo es apuntar dónde están ubicadas las rutas de los archivos de configuración.

Honeypot

Debido a la cantidad de ataques que sufrimos en Internet desde hace años las empresas, estamentos gubernamentales, usuarios avanzados, etc. han tenido que adoptar medidas para incrementar la seguridad de los sistemas. Una de las maneras de implementar esta seguridad es la utilización de sistemas trampa para observar el comportamiento de un ciberataque y analizar la intrusión y el método utilizado.
Estos sistemas simulan ser equipos vulnerables y que son perceptibles de ser atacados. Los especialistas en seguridad tienen una gran ventaja al utilizar estos métodos ya que les da la posibilidad de aprender métodos de intrusión, captura de malware entre los que puede haber un zero-day y por supuesto analizar con tiempo la intrusión y poder aplicar fórmulas de seguridad que permitan detectar y rechazar estos ataques.

Estos sistemas trampa muchas veces nos proporcionan los programas que han utilizado los ciberdelincuentes y como todos sabemos eso es oro en nuestro trabajo. Una vez obtenidos estos programas podemos realizar un análisis en profundidad para observar el comportamiento y tomar las medidas oportunas. En algunos casos son ataques con mucho ingenio ayudados por creaciones de malware que te dan una idea de la magnitud y dedicación de estas personas al desarrollar sus armas de ataque.

Las funciones principales de un Honeypot son:
· Desviar la atención del atacante con este tipo de sistemas para salvar el sistema principal y en muchos casos en el que el atacante pueda determinar que se trata de un honeypot poder disuadirle de seguir adelante con la intrusión o ganar un tiempo muy valioso que nos permita reaccionar y tomar las medidas oportunas para frenar el ataque.
· Capturar nuevo tipo de malware para el estudio del mismo.
· Poder obtener una base de datos con direcciones de atacantes y métodos de ataques desconocidos.
· Una de las más importantes. Poder conocer nuevas vulnerabilidades y de esta manera poder aplicar medidas para que no afecten a la seguridad de nuestros sistemas.

Clasificación

Para poder implementar debidamente un honeypot debemos tener clara la idea de qué tipo de información queremos obtener. Una vez tengamos claro este concepto solo nos queda llevar a cabo la idea. El típico honeypot es aquel que nos permite analizar el comportamiento de un atacante y las herramientas que está utilizando para poder descubrir nuevos tipos de ataque y vulnerabilidades en el sistema. Es el que más nos puede interesar de cara a si somos administradores de sistemas o simplemente usuarios con inquietudes y ganas de aprender.

Según lo anterior, se debe remarcar que un honeypot no es una medida de seguridad en un sistema. Es una implementación en nuestro sistema de seguridad, es decir que por sí solo no sirve para poder frenar un ataque y asegurar de manera efectiva el sistema que tenemos que administrar.

Una de las clasificaciones que podemos llevar a cabo al implementar un honeypot es según el nivel de interacción que un atacante puede tener con él.

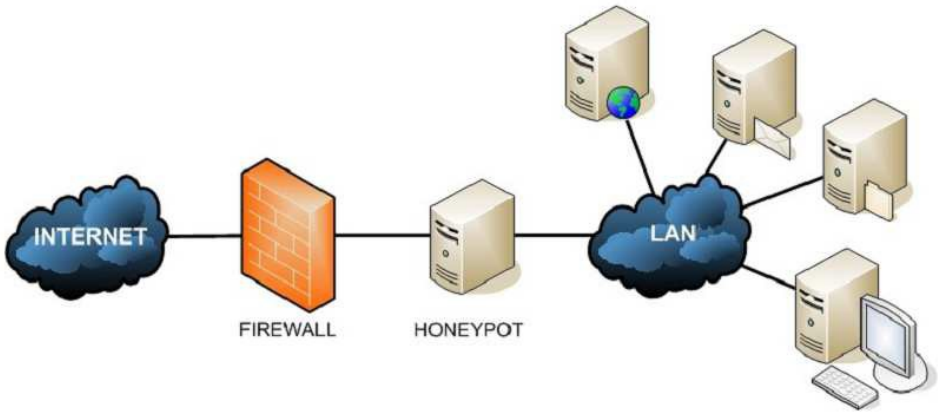
Lo importante de nuestro honeypot es que el atacante no se dé cuenta de que es un sistema trampa y que genere mucha actividad dentro del mismo. De esta manera estará

facilitándonos información que puede resultar muy valiosa y aprenderemos mucho sobre sus métodos de ataque y el tipo de herramientas que ha utilizado.

De todas maneras, cuanta más interacción tenga el atacante con el honeypot más expuestos estaremos a que logre tener acceso a otros equipos "reales" de la red.
Según el nivel de interacción podemos clasificarlos.

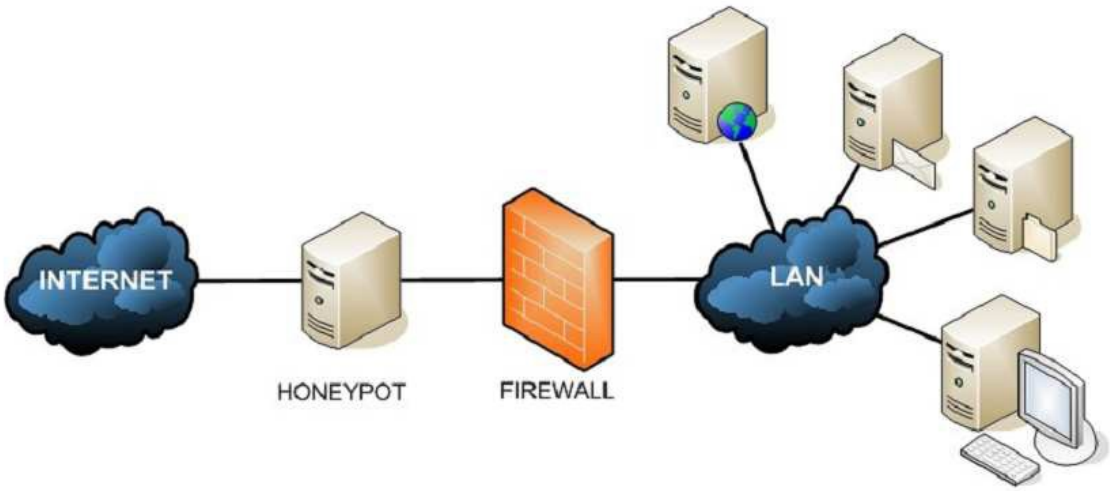
Ubicación de un honeypots

Antes del firewall (Front of firewall): esta localización permite evitar el incremento del riesgo inherente a la instalación del honeypot. Como este se encuentra fuera de la zona protegida por el firewall, puede ser atacado sin ningún tipo de peligro para el resto de la red.



Esta configuración evitara las alarmas de otros sistemas de seguridad de la red (IDS) al recibir ataques en el honeypot. Sin embargo, existe el peligro de generar mucho tráfico debido precisamente a la facilidad que ofrece el honeypot para ser atacado.

Detrás del firewall (Behind the firewall): en esta posición, el Honeypot queda afectado por las reglas de filtrado del firewall. Por un lado se tiene que modificar las reglas para permitir algún tipo de acceso al honeypot por posibles atacantes externos, y por el otro lado, al introducir un elemento potencialmente peligroso dentro de la red se puede permitir a un atacante que gane acceso al honeypot y a la red.



Cualquier atacante externo será lo primero que encuentra y esto generará un gran

consumo de ancho de banda y espacio en los ficheros de log. Por otro lado, esta ubicación evita la detección de atacantes internos.

La ubicación tras el firewall permite la detección de atacantes internos así como firewalls mal configurados, máquinas infectadas por gusanos o virus e incluso atacantes externos.

Sin embargo las contrapartidas más destacables son la gran cantidad de alertas de seguridad que generarán otros sistemas de seguridad de la red (Firewalls, IDS). Al recibir ataques el honeypot se ve la necesidad de asegurar el resto de nuestra red contra el honeypot mediante el uso de firewalls extras o sistemas de bloqueo de acceso, ya que si un atacante logra comprometer el sistema tendrá vía libre en su ataque a toda la red.

Hay varias circunstancias que obligan a este tipo de arquitectura, como por ejemplo la detección de atacantes internos o la imposibilidad de utilizar una dirección IP externa para el honeypot.

En la zona desmilitarizada (DMZ): la ubicación en la zona desmilitarizada permite por un lado juntar en el mismo segmento a los servidores de producción con el honeypot y por el otro controlar el peligro que añade su uso, ya que tiene un firewall que lo aísla de resto de la red local.

Además, se elimina las alarmas de los sistemas internos de seguridad y el peligro que supone para la red al no estar en contacto directo con esta. La detección de atacantes internos se ve algo debilitada, puesto que al no compartir el mismo segmento de red que la LAN, un atacante local no accederá al Honeypot.

Sin embargo, desde la red local sí es posible acceder al Honeypot, con lo que un atacante interno que intente atacar a los servidores públicos u otros sistemas externos, por ejemplo un gusano, muy probablemente acabe siendo detectado.
Conociendo todas las posibilidades, hemos decidido colocar este sistema trampa en una DMZ creada exclusivamente para su uso.

Instalación de los honeypots

Es importante el tema de la configuración de red. Aconsejo editar el fichero de configuración de Honeydrive y poner una IP estática para no tener cambiar la configuración de las interfaces cada vez que arranque el sistema. El archivo en cuestión se encuentra en la ruta:
/etc/network/interfaces.

Bien echa estas aclaraciones vamos a empezar a instalar los honeypots.
Ante todo un consejo: cuando pongan en marcha estos honeypot hay que practicar con ellos para ver su funcionamiento, siempre monitorizando el host que alberga la máquina virtual.

Kippo

Aquí presento a Kippo, un honeypot de media interacción:*http://code.google.com/p/kippo/*

Este tipo de honeypot nos va a permitir emular un servicio SSH con un login y pass

poco seguras, reproducir un sistema de ficheros en el que incluso podremos integrar un sistema operativo, poner documentos y ejecutar comandos.
En este tipo de honeypot aprenderemos mucho sobre un atacante ya que quedan registrados los comandos utilizado por un atacante y se pueden reproducir de manera automática la sesión que se ha establecido. Nos muestra estadísticas, localización GeoIP, un top de comandos introducidos, comandos "curiosos", etc.

Directorios importantes
· *dl/*: donde se guarda los ficheros descargados mediante wget
· *log/kippo.log*: donde se guarda información de uso y depuración
· *log/tty/*: logs de las sesiones
· *utils/playlog.py:* herramienta para reproducir los logs de sesión
· *utils/createfs.py*: utilizado para crear fs.pickle
· *fs.pickle*: falso sistema de ficheros
· *honeyfs/*: contenido del falso sistema de ficheros. Aquí podemos poner una copia de un sistema real.

[Kippo]
Script: /opt/kippo/start.sh
Downloads: /opt/kippo/dl/
TTY logs: /opt/kippo/log/tty/
Credentials: /opt/kippo/data/userdb.txt
MySQL database: kippo
MySQL user/password: root/honeydrive

[Kippo-Graph]
Location: /var/www/kippo-graph/
Config: /var/www/kippo-graph/config.php
URL: http://local-or-remote-IP-address/kippo-graph/
MySQL database: kippo
MySQL user/password: root/honeydrive

[Kippo2MySQL]
Location: /opt/kippo2mysql/
MySQL database: kippo2mysql
MySQL user/password: root/honeydrive

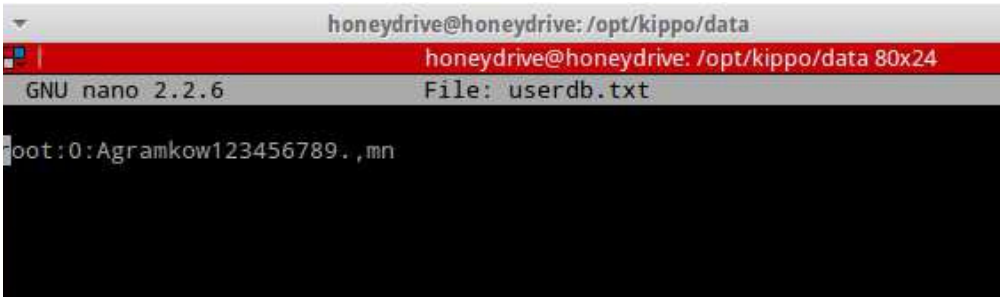
[Kippo-Scripts]
Location: /opt/kippo-scripts/
+ kippo-sessions
+ kippo-stats
+ kippo2wordlist

Vamos a ponerlo en marcha. Abrimos el Terminal que hay en el escritorio llamado "terminator". Lo primero que vamos a realizar es cambiar la contraseña por defecto de SSH para no dejar la proporcionada por la distribución y que el atacante no entre a las primeras de cambio. Para eso tecleamos lo siguiente para ir a la ruta donde se encuentra el fichero de configuración.

cd /opt/kippo/data

Como vemos tenemos un fichero llamado userdb.txt. Este lo vamos a editar para poder cambiar el pass por defecto. El nombre de usuario lo voy a dejar cómo lo tengo por defecto para no poner las cosas tampoco muy complicadas. Con una buena contraseña es suficiente siempre y cuando no pongan en marcha la honey de manera continua. En caso contrario cambiar tanto el login como el pass.

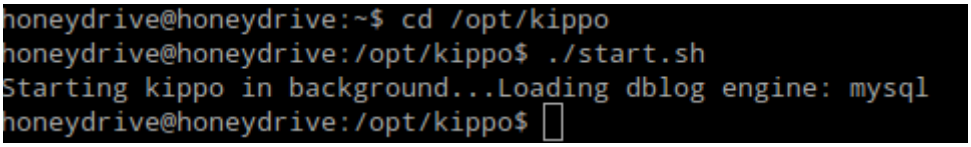
nano userdb.txt



Bien eso que veis es el login “root” y el pass”123456” es el que viene por defecto. Una vez terminado de editar pulsamos Crt + X, presionamos Y para salvarlo.
Ahora solo nos queda ponerlo en marcha. Nuevamente en la consola vamos a la ruta donde ejecutaremos el script de Kippo para ponerlo en funcionamiento y a la escucha.

cd /opt/kippo/
./start.sh

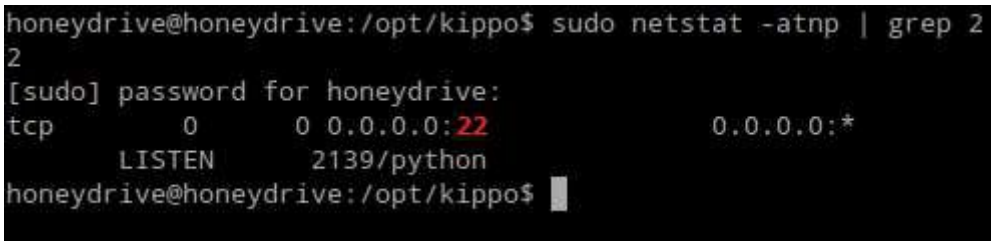
Si todo ha ido de manera correcta ya tenemos kippo corriendo y con el servicio SSH a la escucha para posibles atacantes.



Comprobamos que el servicio está a la escucha con el comando:

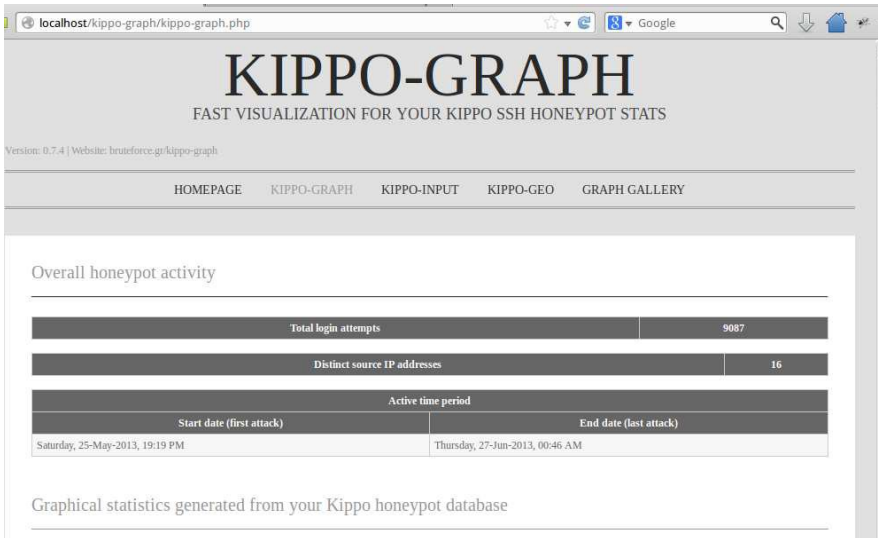
sudo netstat -atnp | grep 22

Si todo esta correcto nos saldrá lo siguiente:

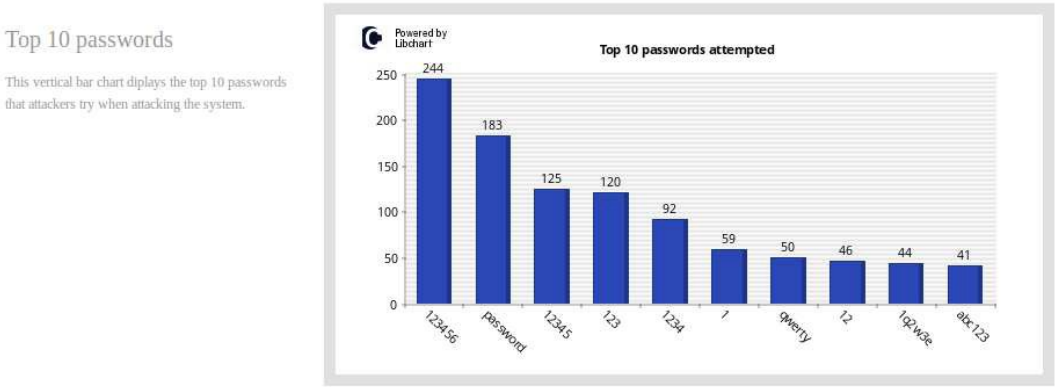


Ahora ya lo tenemos en marcha y hay que esperar a posibles ataques. Normalmente no

tardan mucho en producirse pero hay que tener paciencia. Podemos consultar la base de datos para mirar los ataques, comandos introducidos, etc. O bien podemos echar mano de lo fácil y bonito, el entorno gráfico de Kippo.
Para poder arrancarlo, simplemente se debe ingresar al navegador, según nos indica el fichero de configuración "readme.txt":
http://local-or-remote-IP-address/kippo-graph/



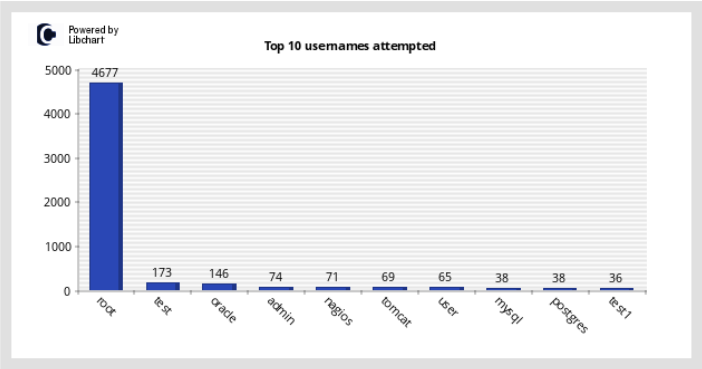
Si abrimos KIPPO-GRAPH veremos si hemos sufrido ataques, cuando se han producido, Ips del ataque, gráficos, comandos, etc. Os dejo unas capturas de pantalla para que lo veáis. Vemos el número total de logs que se ha producido, las diferentes IP que han intentado entrar. También observamos la fecha y hora del primer ataque registrado y del último registrado. Aquí vemos el top 10 de los passwords introducidos para poder entrar en mi servicio SSH.



En este gráfico vemos el top 10 de los login realizados.

Top 10 usernames

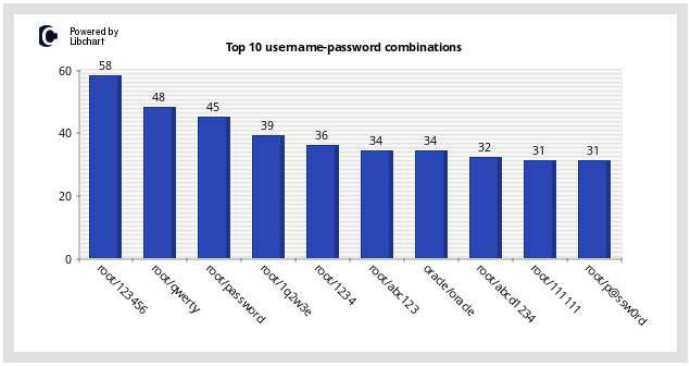
This vertical bar chart diplays the top 10 usernames that attackers try when attacking the system.



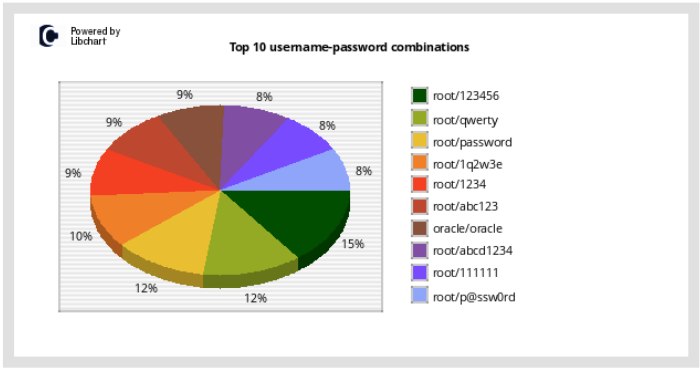
En esta otra captura vemos la combinación de login y pass utilizados.

Top 10 user-pass combos

This vertical bar chart diplays the top 10 username and password combinations that attackers try when attacking the system.



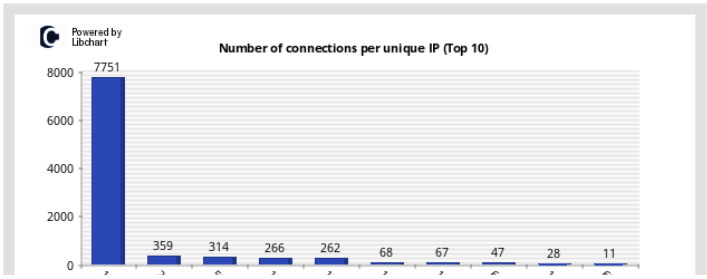
This pie chart diplays the top 10 username and password combinations that attackers try when attacking the system.



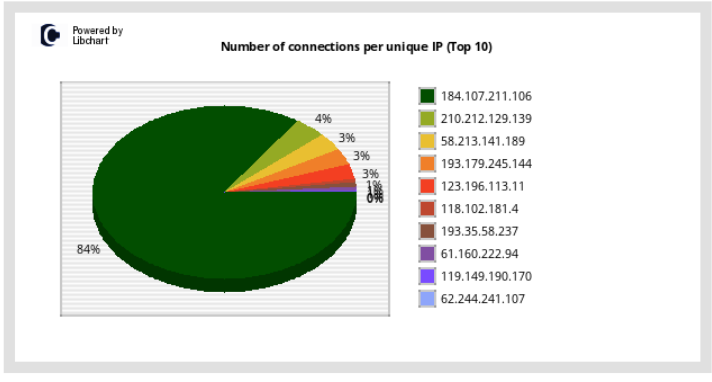
Ahora veremos las IPs que se han intentado conectar por SSH a nuestro honeypot.

Connections per IP

This vertical bar chart diplays the top 10 unique IPs ordered by the number of overall connections to the system.



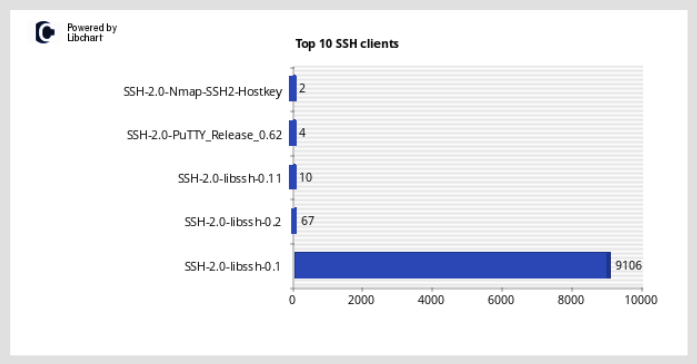
This pie chart diplays the top 10 unique IPs ordered by the number of overall connections to the system.



En esta otra vemos el cliente que han utilizado para intentar la conexión.

Top 10 SSH clients

This vertical bar chart diplays the top 10 SSH clients used by attackers during their hacking attempts.



En la pestaña KIPPO-INPUT input podemos ver lo siguiente:

Overall post-compromise activity

Post-compromise human activity	
Total number of commands	**Distinct number of commands**
11	10

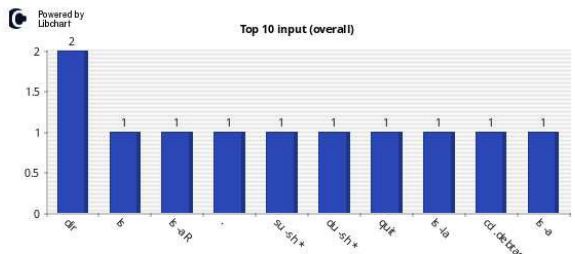
Downloaded files	
Total number of downloads	**Distinct number of downloads**
0	0

Top 10 input (overall)

The following table diplays the top 10 commands (overall) entered by attackers in the honeypot system.

ID	Input	Count
1	dir	2
2	ls	1
3	ls -aR	1
4	.	1
5	su -sh *	1
6	du -sh *	1
7	quit	1
8	ls -la	1
9	cd .debtage	1
10	ls -a	1

This vertical bar chart visualizes the top 10 commands (overall) entered by attackers in the honeypot system.



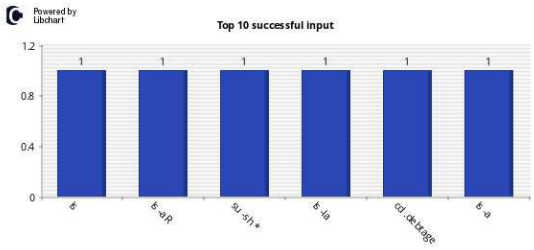
Entradas correctas

Top 10 successful input

The following table diplays the top 10 successful commands entered by attackers in the honeypot system.

ID	Input (success)	Count
1	ls	1
2	ls -aR	1
3	su -sh *	1
4	ls -la	1
5	cd .debtage	1
6	ls -a	1

This vertical bar chart visualizes the top 10 successful commands entered by attackers in the honeypot system.



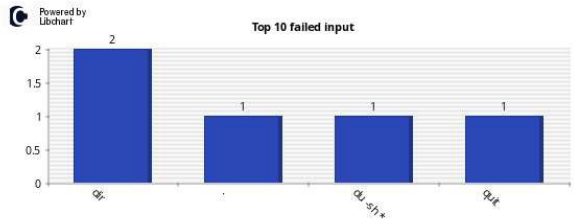
Entradas fallidas

Top 10 failed input

The following table diplays the top 10 failed commands entered by attackers in the honeypot system.

ID	Input (fail)	Count
1	dir	2
2	.	1
3	du -sh *	1
4	quit	1

This vertical bar chart visualizes the top 10 failed commands entered by attackers in the honeypot system.



En la pestaña KIPPO-GEO tenemos la localización de procedencia de las IP y algunos

datos más, muy interesante esta pantalla que además nos brinda de herramientas de localización.

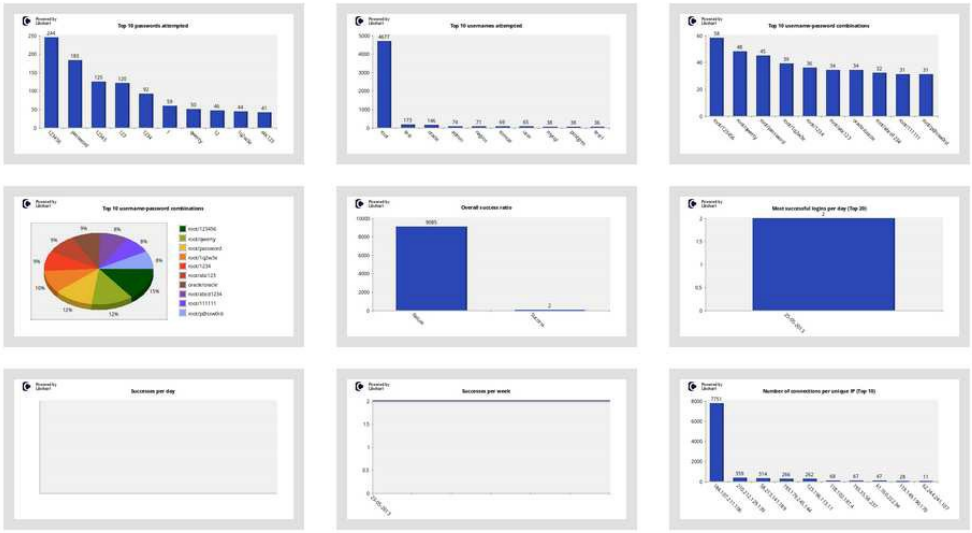
Geolocation information gathered from the top 10 IP addresses probing the system

The following table displays the top 10 IP addresses connected to the system (ordered by volume of connections).

ID	IP Address	Probes	City	Region	Country Name	Code	Latitude	Longitude	Hostname	Lookup
1	184.107.211.106	7751		QC	Canada	CA	45.5	-73.583298	host.ondawebserver.com.br	
2	210.212.129.139	359	Bharuch	Gujarāt	India	IN	21.700001	72.966698	210.212.129.139	
3	58.213.141.189	314	Nanjing	Jiangsu	China	CN	32.061699	118.777802	58.213.141.189	
4	193.179.245.144	266			Czech Republic	CZ	49.75	15.5	193.179.245.144	
5	123.196.113.11	262	Beijing	Beijing	China	CN	39.928902	116.388298	123.196.113.11	
6	118.102.181.4	68	Puri	Orissa	India	IN	19.799999	85.849998	abs-static-4.181.102.118.aircel.co.in	
7	193.35.58.237	67			United Kingdom	GB	51.5	-0.13	193.35.58.237	
8	61.160.222.94	47	Beijing	Beijing	China	CN	39.928902	116.388298	61.160.222.94	
9	119.149.190.170	28	Seoul	Seoul	Korea, Republic of	KR	37.598499	126.978302	119.149.190.170	
10	62.244.241.107	11			Turkey	TR	39	35	62.244.241.107	

The following Intensity Map shows the volume of attacks per country by summarising probes originating from the same nation, using the same IP or not.

Por último tenemos la pestaña GRAPH GALLERY que nos muestra todos los datos en una sola ventana y podemos acceder a ellas pulsando en las mismas.



Si

queremos ver los logs generados podemos ir a /opt/kippo/log y dentro del directorio tty tenemos logs generados que nos van a servir para poder simular lo que ha realizado un atacante mediante el script playlog.py que se encuentra en la ruta /opt/kippo/utils. Nos fijamos en el nombre de los logs y vamos a la ruta mencionada para ejecutar el script.

sudo python playlog.py -b -m 2
/opt/kippo/log/tty/20130525-214559-6881.log 0

En último lugar, podemos ver el log que se genera en tiempo real. Para esto nos situamos en el directorio donde se guardan los logs:

tail -f kippo.log

Dionaea

Dionaea está diseñado principalmente para recoger malware a través de las vulnerabilidades de seguridad que ofrece. Gracias a esto un atacante nos descargará malware pensando que de esa manera el ordenador está bajo su poder y utilizarlo para el fin que el tenga pensado. Dioanea está compuesto de módulos que emulan a protocolos. Prácticamente se pueden emular todos los que queramos aunque ya viene bien configurado para este aspecto. Por ejemplo el protocolo SMB está activo para que sufra ataques. Otro de los módulos activos es el de SIP que es capaz de establecer sesiones. Otra de las características de Dionaea es su capacidad de escuchar en varias interfaces de red y recoge información de muchas IPs de forma simultánea.

Gracias a sus características y a las vulnerabilidades expuestas podemos observar los exploits que han utilizado, analizar el código introducido en una shell, recuperación de binarios y por supuesto analizar los logs que nos van a revelar mucha información.

Tenemos el registro detallado de los ataques y esto incluye cualquier acción que se puedan realizar en el sistema. El fin último para lo que está programado este honeypot tan fantástico es el de la obtención de malware para su posterior análisis.

Entre los servicios vulnerables que mantiene a la espera para ser atacado nos encontramos con SMB, SIP, MYSQL, Ftpd, epmapper, etc. Utiliza para almacenar todos estos resultados una base de datos SQLite. Otra de sus características es que es capaz de almacenar el login y pass que el atacante ha utilizado para intentar obtener un acceso. Ante todo el archivo de configuración:

[Dionaea]
Location: /opt/dionaea/
Bin: /opt/dionaea/bin/dionaea
Config: /opt/dionaea/etc/dionaea/dionaea.conf
Logs: /opt/dionaea/var/log/
SQLite database: /opt/dionaea/var/dionaea/logsql.sqlite
Malware samples: /opt/dionaea/var/dionaea/binaries/
+ phpLiteAdmin: /var/www/phpliteadmin,
+ password: honeydrive,
+ URL: http://local-or-remote-IPaddress/
phpliteadmin/phpliteadmin.php

[DionaeaFR]
Location: /opt/dionaeaFR/
Script: /opt/dionaeaFR/manage.py

[Dionaea-Scripts]
Location: /opt/dionaea-scripts/
+ mimic-nepstats
+ dionaea-sqlquery

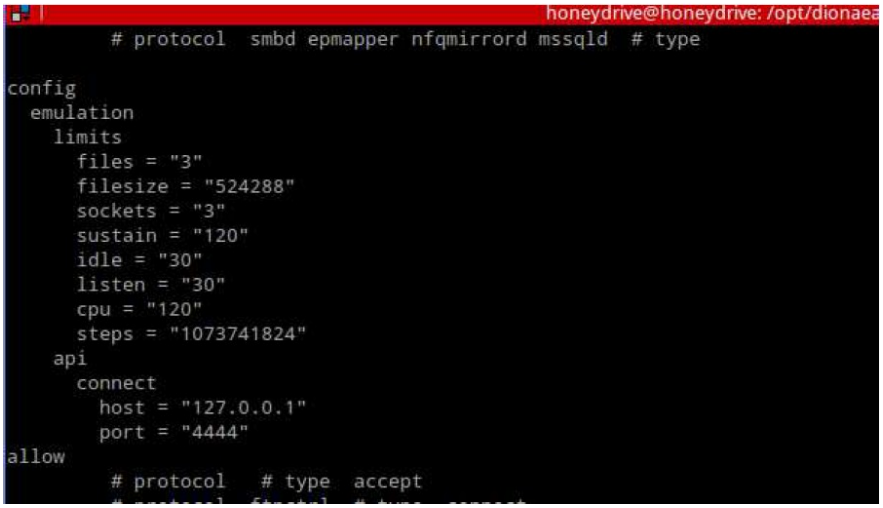
Para ponerlo en marcha vamos a una shell en nuestra distribución. Vamos a ir a la ruta donde se encuentra el binario para poner en marcha el honeypot.

/opt/dionaea/bin

Una vez en este directorio pasamos a ejecutar el script para cazar todo tipo de ataques y malware.

./dionaea -l all,-debug -L ''*

Si todo ha funcionado bien, se ejecuta el script y se verá cómo va cargando el fichero de configuración y abriendo los puertos que emulan los servicios.



Ya tenemos nuestro honeypot corriendo. Ahora pondremos en marcha el entorno gráfico llamado *DionaeaFR*. Para eso vamos a otra shell y nos vamos a la ruta donde está el script que lanza este entorno y así poder verlo desde un navegador.

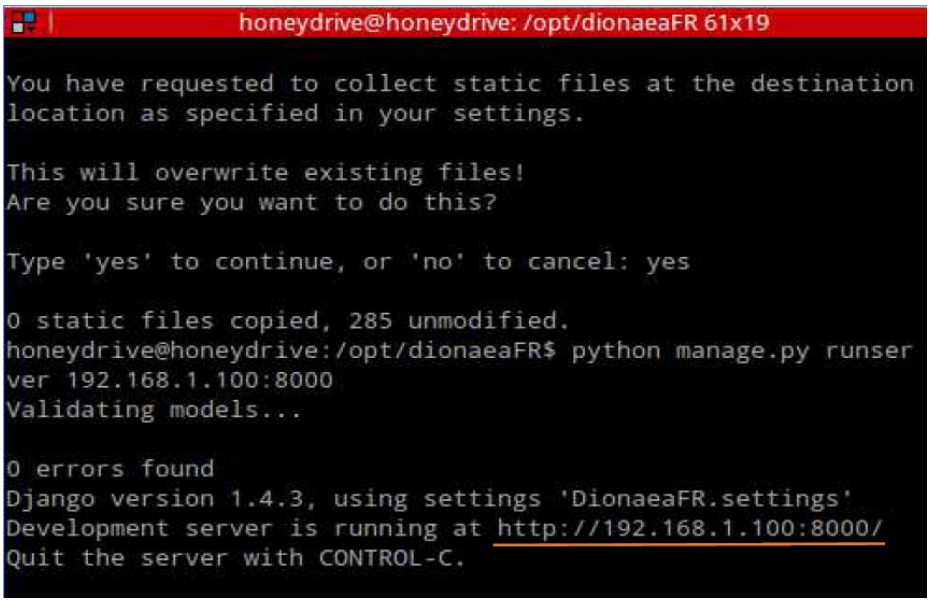
/opt/dionaeaFR

Ahora ejecutamos los siguientes scripts para poder ponerlo en marcha.

python manage.py collectstatic

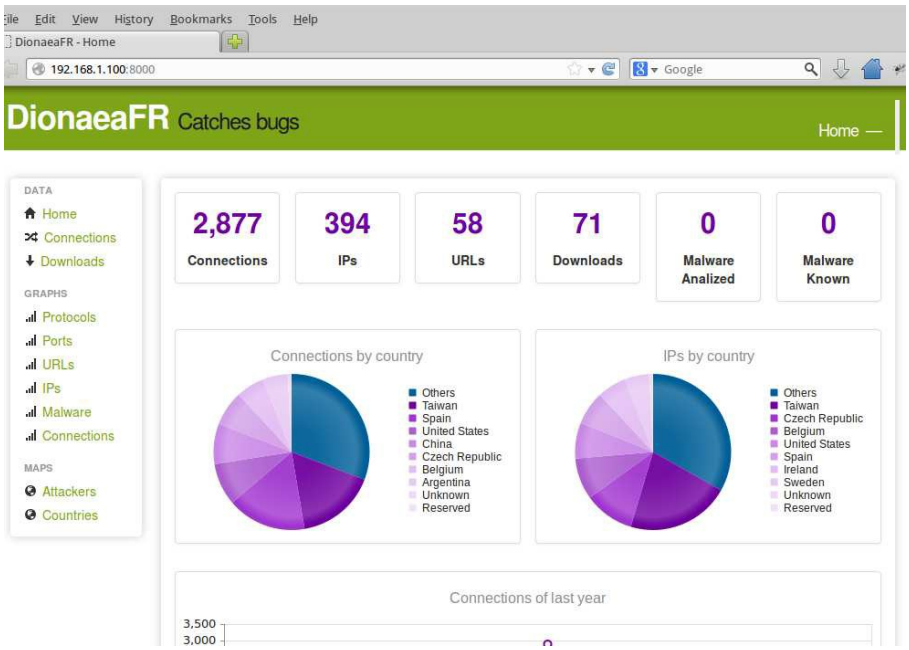
Nos pedirá que confirmemos con "Yes" para importar todos los registros. Ahora ejecutamos el siguiente script que nos va a lanzar el entorno gráfico y la ruta con el puerto que queramos.

python manage.py runserver 192.168.1.100:8000



Ya tenemos nuestro honeypot en marcha, ahora abrimos en el navegador la dirección que hemos puesto con ese puerto y se nos abrirá la visualización:

http://192.168.1.100:8000

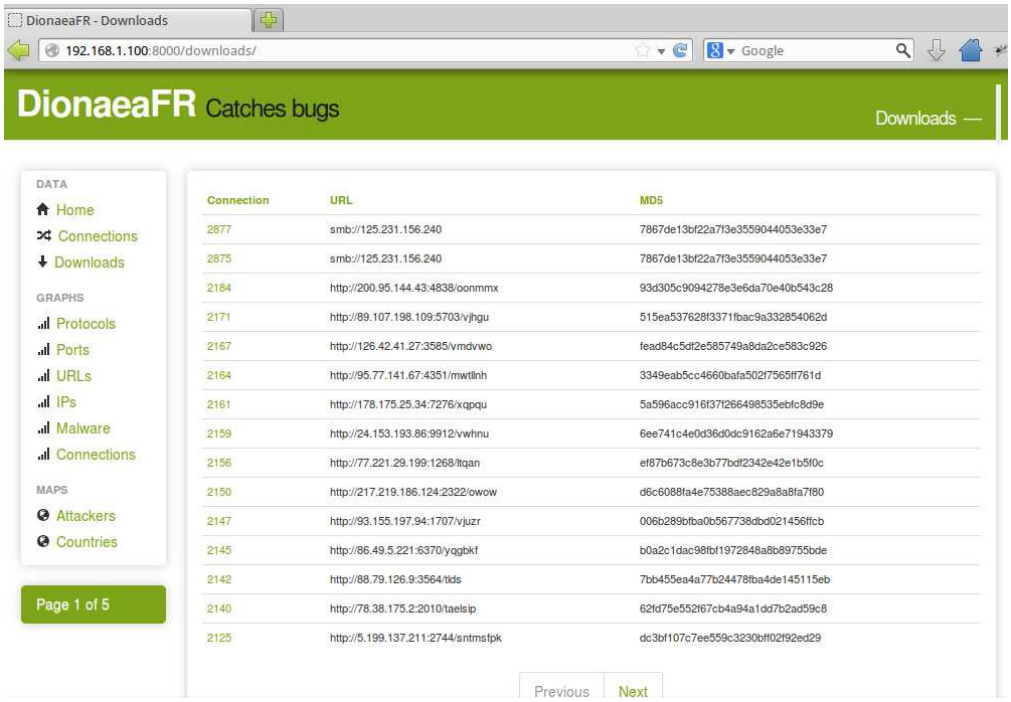


En esta pantalla, de manera intuitiva y rápida podemos ver:

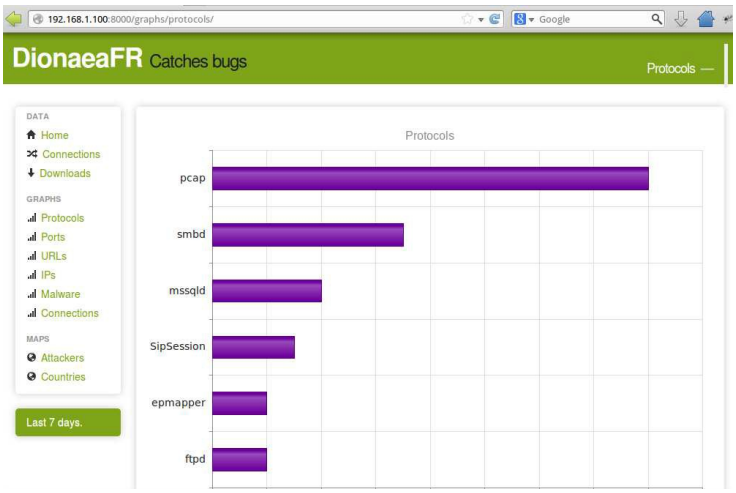
· Conexiones realizadas a nuestro honeypot.
· IPs diferentes que han realizado esas conexiones.
· URLs que nos han atacado.
· Binarios que nos han descargado y hemos capturado.

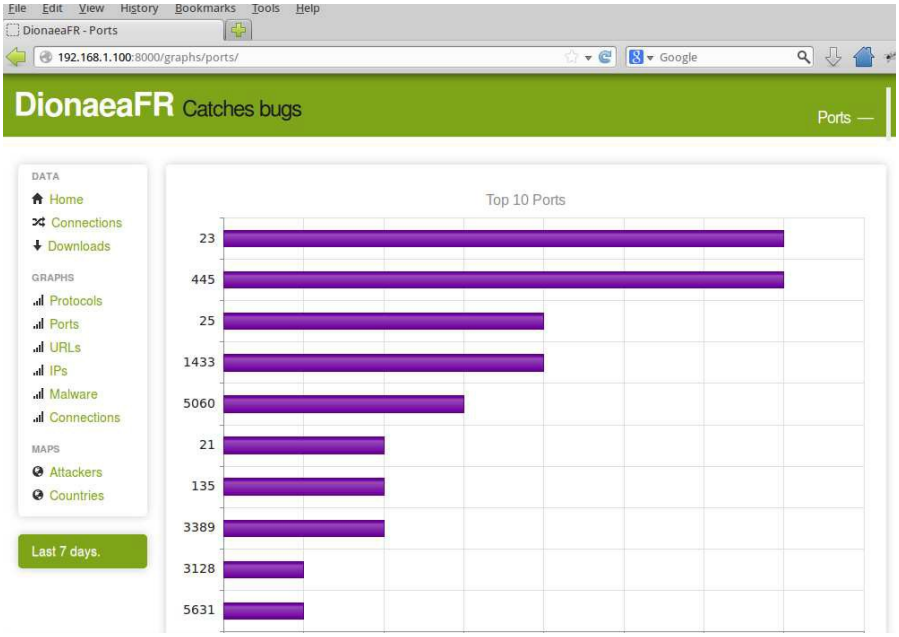
· Malware analizado.
· Malware desconocido.
· Conexiones por país = 0
· IPs por país = 0

Estos dos últimos valores que se ven con valor cero están así de manera predeterminada y no he utilizado API para subir las muestras a Virustotal para analizarlos. Ahora, vamos a seguir viendo resultados obtenidos con este magnífico honeypot. Creo que para entender esta pantalla no hace falta ninguna explicación.

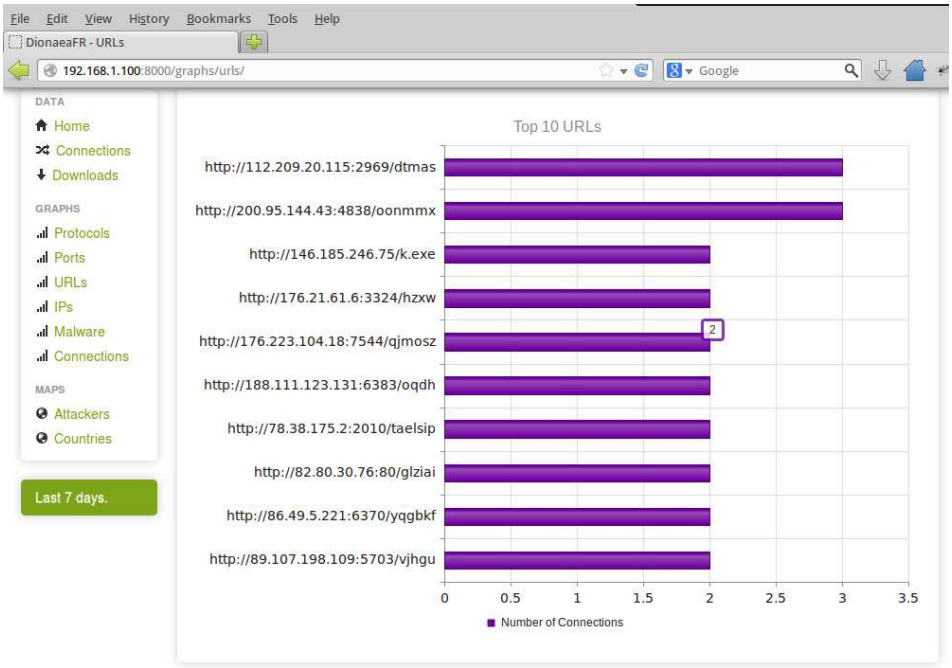


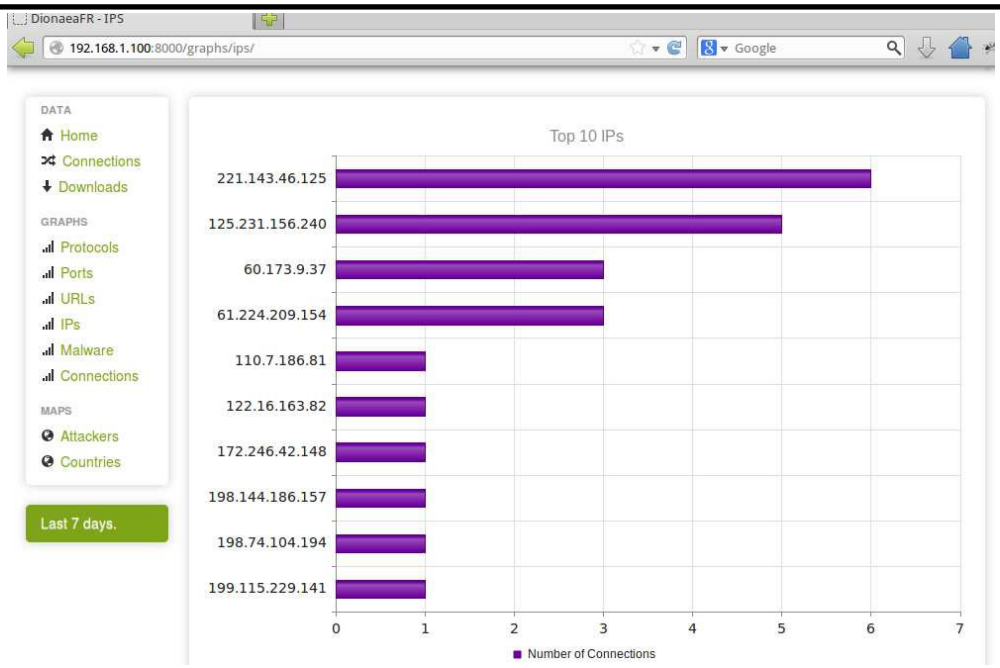
Ahora vemos los protocolos y los puertos que más se han utilizado a la hora de atacar el honeypot.



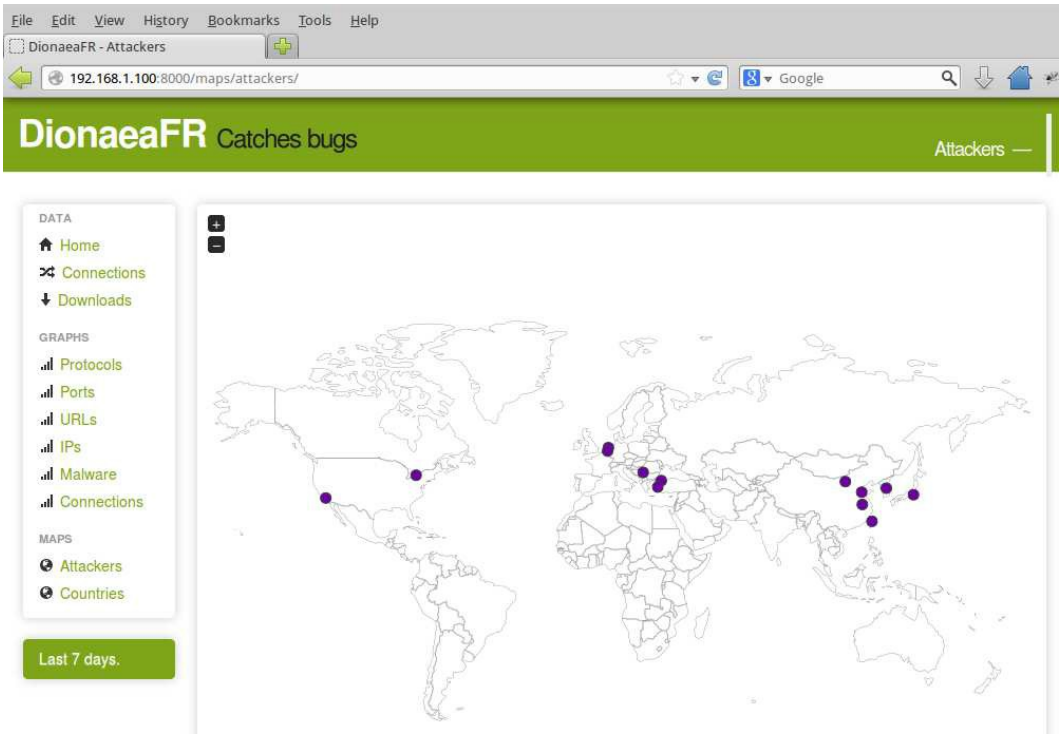


A continuación aparecen las URL e IPs de los ataques:



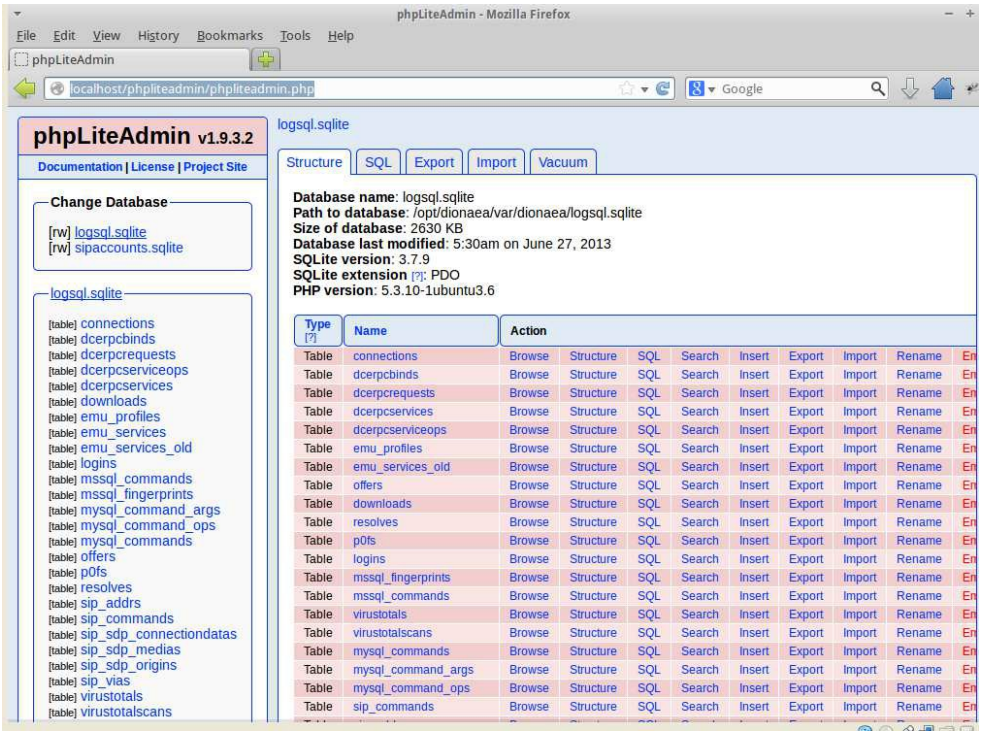


La siguiente pantalla nos informa la localización de los ataques. El mapa no está tan logrado cómo en Kippo pero es más que suficiente.



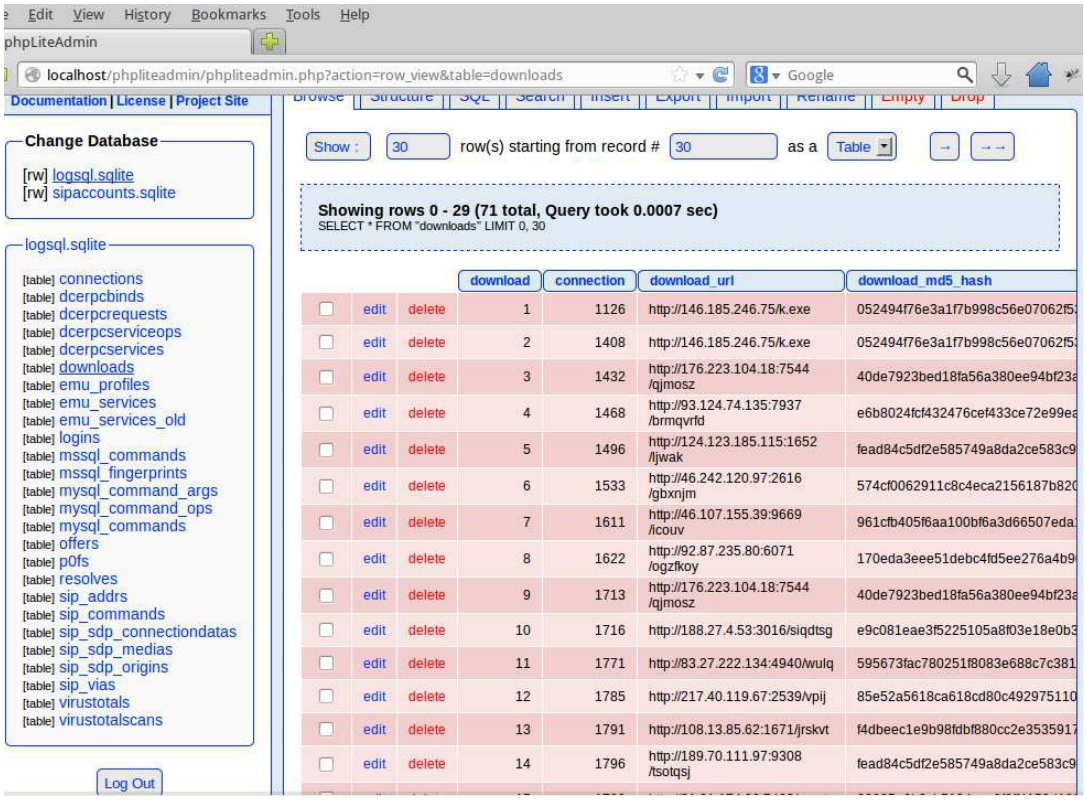
En cuanto al entorno gráfico ya hemos visto todo lo que nos ofrece. Como podemos ver es muy detallado y no hay problemas para interpretar los datos que nos muestra.
Si hacemos un listado de lo que tenemos dentro veremos los binarios descargados:

http://localhost/phpliteadmin/phpliteadmin.php

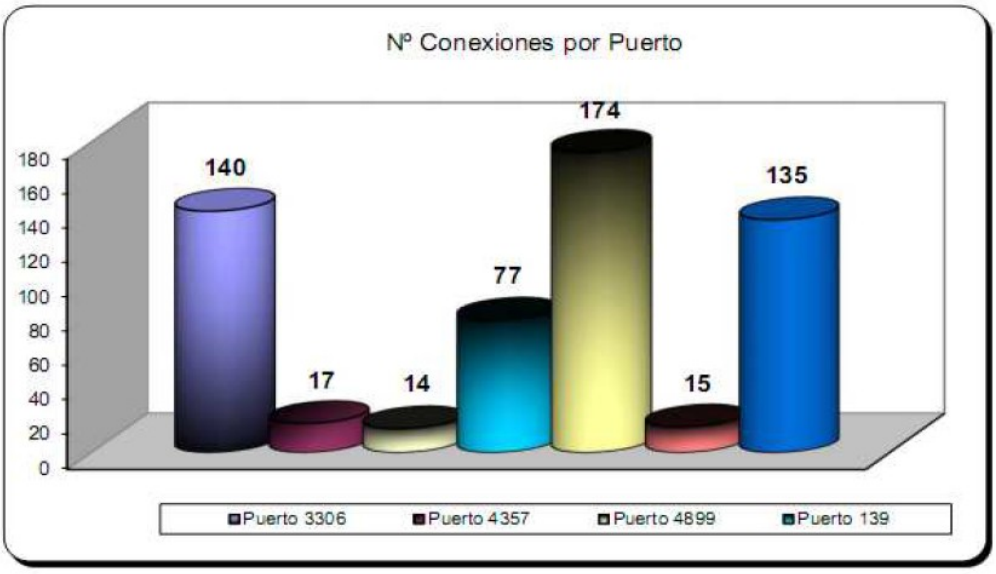


Aquí tenemos por ejemplo la tabla donde se almacenan las conexiones que luego DioaneaFr nos muestra.

Aquí la tabla que contiene el malware recogido.



Ahora muestro las estadísticas que estoy realizando en Excel para tener un mejor control de todos los datos que recogemos y tener una mejor visión a la hora de su análisis.



6.- Anti-Forense

Como definimos anteriormente, el análisis forense digital se ocupa del estudio de la adquisición, preservación y presentación de evidencias electrónicas para ser procesadas y conservadas de tal forma que puedan utilizarse como prueba legal. Se trata de entender y contestar preguntas referidas a cómo, cuándo y desde dónde se produjo el incidente, así como cuál fue su impacto y a qué afectó.

Para los analistas forenses, se presentan retos cada día más exigentes en cuanto al rastreo y detección de un atacante, dada la sofisticación de los intrusos en sus técnicas de evasión. El reconocimiento de las vulnerabilidades en las herramientas utilizadas por procedimientos de esta materia ha generado la aparición de las llamadas técnicas anti forenses, que son definidas como metodologías para comprometer la disponibilidad de la evidencia en un proceso forense. De esta forma, se intenta manipular el material de una pericia destruyendo, ocultando, eliminando y/o falsificando la evidencia.

Si bien existen muchas técnicas, principalmente pueden ser agrupadas en cuatro clasificaciones, que veremos a continuación.

A) Técnicas de borrado o destrucción de la información

El fin de esta técnica es imposibilitar la recuperación de las evidencias ya sea por el borrado de archivos o particiones del disco. Existen muchas aplicaciones que realizan estas tareas y son capaces de ejecutarse en la mayoría de los sistemas operativos.

Algunas se basan en el Algoritmo de Gutmann, el cual detalla las operaciones para que un archivo o directorio sea borrado en forma segura; estas aplicaciones son generalmente de escaso tamaño y portables, es decir que no precisan una instalación.

En sistemas Linux o Unix, los códigos más utilizados no contienen una interfaz gráfica sino que se ejecutan a través de un terminal o consola como es el caso de Wipe. Para probarlo, se puede instalar con el siguiente comando:

usuario@maquina:~$ sudo apt-get install wipe

Posteriormente, el comando man wipe será de gran ayuda para entender cómo funciona esta aplicación.

En el caso de la información no sea eliminada por este tipo aplicaciones, entrarán en escena diferentes técnicas como el file carving o slack space, las cuales intentarán recuperar la información borrada. En algunos casos pueden servir para recuperar archivos afectados por algunas familias particulares de ransomware.

En este grupo, también entran diversas técnicas vinculadas a la desactivación de logs, es decir, cuando el sistema tiene por defecto la creación de registros por eventos e incidentes, los atacantes intentarán desactivarlos para ocultar sus huellas.

B) Técnicas de ocultación de la información

Este método tiene como principal objetivo hacer invisible la evidencia para el analista. De este modo, los atacantes ocultan mensajes u objetos dentro de otros archivos, de tal forma que no se perciba su existencia.

Esta técnica, llamada esteganografía, puede llegar a ser muy eficiente de ser bien ejecutada, pero conlleva muchos riesgos para el atacante o intruso. Al no modificar la evidencia, de ser encontrada puede ser válida en una investigación formal y por lo tanto

servir para la incriminación e identificación del autor de dicho ataque.

C) Técnicas de sobreescritura de metadatos

Este grupo de técnicas tiene como fin engañar, creando falsas pruebas para cubrir al verdadero autor, incriminando a terceros y por consiguiente desviando la investigación.
Si el analista descubre cuándo un atacante tuvo acceso a un sistema Windows, Mac o Unix, con frecuencia es posible determinar a qué archivos o directorios accedió. Utilizando una línea de tiempo para indicar las acciones y clasificándolas en orden cronológico, el analista tendría un esquema de lo que fue ocurriendo en el sistema.

Aunque un atacante podría borrar los contenidos de los medios de comunicación, esta acción podría atraer aún más la atención. Otra estrategia bastante utilizada es cuando el atacante oculta sus pistas al sobrescribir los propios tiempos de acceso, de manera que la línea de tiempo no pueda construirse de forma fiable.

Existen varias herramientas comúnmente utilizadas con este fin, como ExifTool o Metasploit que en conjunto con Meterpeter permiten borrar o cambiar estos parámetros.

De este modo, afectando a los archivos claves sería casi imposible generar una línea de tiempo con los incidentes ocurridos en un orden cronológico para los analistas.
Este tipo de técnicas no necesariamente implica actividad maliciosa; por ejemplo, en algunos casos se pueden utilizar estos comandos para esconder información cambiando los atributos, debido a que es muy improbable que al atacante que haya ingresado a un sistema le interese un archivo antiguo –sobre todo si no ha sido abierto y ha sido creado algunos años atrás.

D) Técnicas de cifrado de la información

Estas técnicas tienen como objetivo dificultar la lectura de datos para los analistas. Esta información puede estar vinculada a un directorio, una aplicación, una partición o inclusive una comunicación. Veamos algunos ejemplos de estas técnicas en los diversos escenarios.

Algunas herramientas muy utilizadas para cifrar archivos o particiones son Truecrypt, Bitlocker o DiskCryptor. Estas herramientas permiten mantener almacenada de una forma segura la información que ha sido cifrada, y en teoría solo se podría tener acceso a ella mediante la clave que, como es lógico, el analista no tendría.

Cuando hablamos de códigos maliciosos, normalmente los malware se encuentran empaquetados o comprimidos con el fin de dificultar el estudio y comprensión de su funcionamiento. Existen muchas aplicaciones utilizadas con tal fin, como es el caso de UPX, PEcompact o Themida; por otra parte, también existen mecanismos antiVM que se encargan de matar la aplicación en caso de que sea ejecutada en un entorno virtualizado con el objetivo de su estudio.

E) Otras técnicas

Diversas técnicas anti forenses también se observan en las comunicaciones; de este modo se utilizan proxies con el objetivo de enmascarar la IP de un atacante. Un ejemplo muy activo de estas técnicas lo vemos en la mayoría de los ataques e inclusive en los códigos maliciosos como CTB-Locker, que se comunican a través de Tor con su C&C.

Por otra parte, siendo aún más meticulosos, los atacantes suelen utilizar herramientas

como exploits dirigidos a las herramientas de análisis forense más utilizadas entorpeciendo aún más la investigación, es por esto que es conveniente utilizar varias herramientas distintas pero con el mismo objetivo para luego comparar resultados.

Está claro que irán apareciendo nuevas técnicas anti forenses, aunque tener conocimiento de las aquí mencionadas contribuye a generar un panorama más claro para entender cabos sueltos que puedan generarse en una investigación forense.

Podemos hacer una clasificación de los métodos mas utilizados a la hora de poner trabas a un análisis forense:

- Ocultación de información, ofuscación y cifrado.
- Falsificación de datos.
- Borrado de datos y destrucción física.
- Prevención de análisis.
- Anonimizadores de conexión.
- Explotación de bugs en herramientas forenses.

Una de las ventajas del atacante es que posee mas tiempo que el analista forense para recoger y analizar objetos digitales.

A continuacion vamos a ver algunas de las medidas y contramedidas que se pueden llevar a cabo entre un Atacante (A) y un Forense (F).

- A utiliza ordenador con HDD dummie; trabaja con distro Live; sus archivos guardados en Cloud/remoto; simula uso normal de HDD → Actividad reciente en HDD hará pensar a F que era unidad de trabajo.

  MEDIDAS:
  - Comprobar presencia de USBs.
  - Monitorizar tráfico de red para detectar uso de almacenamiento Cloud/remoto.

- A adopta medidas anti IP tracing (proxy, reverse-proxy, túnel SSH, VPN, TOR) → F no puede determinar IP origen.

  MEDIDAS:
  - Solicitar registros de conexión a ISP.
  - Tener “pinchada” TOR.
  - Obtener información del proveedor VPN.

- A cifra un dispositivo/archivo mediante algoritmo criptográfico robusto → F no puede acceder al contenido en claro.

  MEDIDAS:
  - Dispositivo apagado → Recuperación por fuerza bruta.
  - Dispositivo hibernado → hiberfil.sys contiene dump de memoria → ¿pass en claro?.
  - Realizar análisis en vivo del dispositivo.
  - Test de entropía para determinar si algoritmo de cifrado es conocido.
  - Explotar vulnerabilidades en algoritmos cifrado custom.

- A aplica transfiguración a ficheros → F no identifica correctamente el tipo de fichero analizado.

MEDIDAS:

- Utilizar Hex editor para examinar contenido de ficheros.
- Utilizar fuzzy hashing para detectar similitud entre ficheros.
- Analizar Recent Files para detectar anomalías.

• A utiliza esteganografía para ocultar información dentro de un archivo → El visualizado/reproducción del archivo no permite a F descubrir la existencia de información oculta.

MEDIDAS:

- Emplear automatización en busca de esteganografiados.

• A utiliza rootkit para enmascarar procesos en memoria → F no identifica proceso malicioso en ejecución.

MEDIDAS:

- Volcado de memoria y conexiones de red.
- Emplear herramientas AV/AR.
- Análisis estático del dispositivo.

• A oculta información en espacio no utilizado del MBR (62 sectores libres) → F no saca una imagen de unidad completa solo de SO y datos, perdiendo dicha información solo de particiones información.

MEDIDAS:

- Realizar copia bit a bit completa de la unidad e inspeccionar esos a 62 sectores.

• A oculta información en HPA → F utiliza una aplicación forense que no recupera información de ese área.

MEDIDAS:

- Utilizar aplicaciones como EnCase, FTK, TSK.

• A oculta información en área DCO → F utiliza una aplicación forense que no recupera información de ese área.

MEDIDAS:

- Utilizar herramientas como The ATA Forensic Tool.

• A utiliza slack space (file slack, partition slack o falsos bad blocks) para ocultar información → F realiza extracción lógica de unidad y no recupera la información oculta.

MEDIDAS:

- Realizar copia bit a bit de la unidad.

• A utiliza Alternate Data Stream (ADS) o Extended Attribute (xattr) para asociar más de un stream de datos a un archivo y ocultar información → F no detecta esta información oculta.

MEDIDAS:

- Correlar resultados de herramientas forenses que analicen los espacios de almacenamiento menos utilizados.
- Comprobar parámetros de hardware del dispositivo.
- Análisis estadístico del slack space.

• A utiliza gran número de dispositivos → F obligado analizar todos.

MEDIDAS:

- Paralelizar la obtención de evidencias.

• A utiliza dispositivos con conectores no estándar → F necesita conectores específicos.

MEDIDAS:

- Adaptar el material a nuevas tendencias criminales.
- Capacidad NAND chip-off, tarjetas especiales lectura HDD, reparación HDD…

• A utiliza configuración no estándar de RAID → F obligado a probar gran número de configuraciones.

MEDIDAS:

- Recombinar RAID en equipo de A.

• A utiliza dispositivos con medidas anti-tampering → F dañará la evidencia al manipularlo.

MEDIDAS:

- Identificación visual del dispositivo y proceder de manera alternativa.

• A daña físicamente los dispositivos → F no podrá utilizar equipo de extracción de evidencias habitual.

• A aplica NSRL scrubbing: Modificar todos los archivos del SO y aplicaciones instaladas (Para poder seguir ejecutando EXEs y DLLs recalcular su CRC) → F emplea De-NISTing y no coincide nada.

MEDIDAS:

- Utilizar una política de whitelisting para buscar ficheros que coincidan.
- Utilizar histogramas para detectar fechas con actividad por encima de la media.

• A modifica los timestamps (MACE times) de todos los archivos; aleatorizar periódicamente la hora de la BIOS; desactivar en Registro la actualizacion de LastAccess → F no prodra hacer un análisis temporal.

MEDIDAS:

- Ignorar los timestamps en los metadatos.
- logs secuenciales pueden ayudar a identificar líneas temporales.

• A modifica los timestamps de manera que aparentan consistencia → F no podrá saber si el timestamp de los ficheros fue modificado.

MEDIDAS:

- Informe “Según el log ocurrió esto a esta hora”.

• A utiliza nombres de archivo restringidos (CON, PRN, AUX…) → F tardará en detectar esta situación.

MEDIDAS:

- Nunca exportar archivos con sus nombres nativos.
- Exportar ficheros con nombre generado automáticamente.

• A utiliza referencias circulares (carpetas anidadas en NTFS máx 255 caracteres) para ocultar información delictiva → Aplicación de análisis de F entra en un loop infinito o arroja una excepción al detectar ruta mayor de 255 caracteres. No podrá realizar recogida selectiva o remota de evidencias.

MEDIDAS:

- Obtener una imagen completa del dispositivo.
- Siempre trabajar desde una imagen.
- Emplear herramientas como EnCase y FTK.

• A manipula los logs (broken logs). Introducir el magic number de .EVT [eLfL (0x654c664c)] en cuerpo de un evento; introduce caracteres UNICODE extendidos → Herramienta análisis de logs de F interpreta comienzo de nueva entrada; difícil de parsear.

MEDIDAS:

- Estimar si el log es estrictamente necesario.
- Parsear únicamente los registros necesarios manualmente.
- Programar un parser adecuado al log a analizar.

• A añade información dummie a archivo a exfiltrar hasta conseguir hash MD5 coincidente con archivo legítimo del SO (ej.rundll.dll) → De-NISTing de F considera correcto el archivo.

MEDIDAS:

- Utilizar funciones hash con menos colisiones (SHA-256).
- No confiar únicamente en hashes para determinar si un archivo es correcto.

ANEXO I

Bibliografía.

[1] Instituto Nacional de Ciberseguridad https://www.incibe.es/
[2] Análisis Forense de Sistemas Informáticos y Respuesta http://neosysforensics.blogspot.com.es/
[3] Proyecto para la creación de laboratorio forense http://digitalfire.ucd.ie/?page_id=1011
[4] Estaciones portátiles http://acmeportable.de/en/solutions/sherlock-lite
[5] Guía para montar laboratorio forense http://highsec.es/2013/09/analisis-forense-parte-i-como-montar-un-laboratorio-forense-y-clonar-con-dd/
[6]Juan Luis García Rambla, Un Forense llevado a Juicio http://www.sidertia.com/
[7]Análisis Forense de Sistemas Informático (UOC).

Webgrafia.

[8] http://www.securitybydefault.com
[9] http://conexioninversa.blogspot.com.es/
[10] http://www.elladodelmal.com/
[11] http://www.hackplayers.com/search/label/forense
[12] http://digital-forensics.sans.org/blog
[13] http://www.flu-project.com/
[14] https://github.com/search?utf8=%E2%9C%93&q=FORENSIC
[15] http://stackoverflow.com/
[16] https://github.com/
[17] http://scripts4cf.sourceforge.net/tools.html
[18] http://wiki.yobi.be/wiki/RAM_analysis
[19] https://github.com/sleuthkit/autopsy
[20] https://github.com/dloss/python-pentest-tools

Lecturas Relacionadas

http://conexioninversa.blogspot.com.es/2013/03/lo-que-la-mentira-esconde-el-caso-de.html
https://www.elhacker.net/InfoForenseWindows.html
https://www.elhacker.net/InfoForenseWindows2.html
https://www.elhacker.net/InfoForense3.html

ANEXO II (Comandos UNIX)

SYSTEM

```
uname -a              =>Displaylinux system information
uname -r              =>Display kernel release information
uptime                =>Show how long the system has been running + load
hostname              =>Show system host name
hostname -i           =>Display the IP address of the host
last reboot           =>Show system reboot history
date                  =>Show the current date and time
cal                   =>Show this month calendar
w                     =>Display who is online
whoami                =>Who you are logged in as
finger user           =>Display information about user
```

HARDWARE

```
dmesg                 =>Detected hardware and boot messages
cat /proc/cpuinfo     =>CPU model
cat /proc/meminfo     =>Hardware memory
cat /proc/interrupts  =>Lists the number of interrupts per CPU per I/O device
lshw                  =>Displays information on hardware configuration of
                        the system
lsblk                 =>Displays block device related information in Linux
free -m               =>Used and free memory (-m for MB)
lspci -tv             =>Show PCI devices
lsusb -tv             =>Show USB devices
dmidecode             =>Show hardware info from the BIOS
hdparm -i /dev/sda      =>Show info about disk sda
hdparm -tT /dev/sda     =>Do a read speed test on disk sda
badblocks -s /dev/sda   =>Test for unreadable blocks on disk sda
```

USERS

```
id                    =>Show the active user id with login and group
last                  =>Show last logins on the system
who                   =>Show who is logged on the system
groupadd admin        =>Add group "admin"
useradd -c "Sam Tomshi"    =>g admin -m sam #Create user "sam"
userdel sam           =>Delete user sam
adduser sam           =>Add user "sam"
usermod               =>Modify user information
```

FILE COMMANDS

```
ls –al                =>Display all information about files/ directories
pwd                   =>Show the path of current directory
mkdir directory-name    =>Create a directory
rm file-name            =>Delete file
rm -r directory-nam     =>Delete directory recursively
rm -f file-name         =>Forcefully remove file
rm -rf directory-name   =>Forcefully remove directory recursively
cp file1 file2        =>Copy file1 to file2
cp -r dir1 dir2       =>Copy dir1 to dir2, create dir2 if it doesn't exist
mv file1 file2        =>Rename source to dest / move source to directory
ln –s /path/to/file-name link-name      #Create symbolic link to file-name
touch file            =>Create or update file
cat > file            =>Place standard input into file
more file             =>Output contents of file
head file             =>Output first 10 lines of file
tail file             =>Output last 10 lines of file
tail -f file          =>Output contents of file as it grows starting with the
                        last 10 lines
gpg -c file           =>Encrypt file
gpg file.gpg          =>Decrypt file
wc                    =>print the number of bytes, words, and lines in files
xargs                 =>Execute command lines from standard input
```

PROCESS RELATED

```
ps                    =>Display your currently active processes
ps aux | grep 'telnet'    =>Find all process id related to telnet process
pmap                  =>Memory map of process
top                   =>Display all running processes
kill pid              =>Kill process with mentioned pid id
killall proc          =>Kill all processes named proc
pkill process-name    =>Send signal to a process with its name
bg                    =>Resumes suspended jobs without bringing them to
                        foreground
fg                    =>Brings the most recent job to foreground
fg n                  =>Brings job n to the foreground
```

FILE PERMISSION RELATED

```
chmod octal file-name       =>Change the permissions of file to octal
Example
chmod 777 /data/test.c      =>Set rwx permission for owner,group,world
chmod 755 /data/test.c      =>Set rwx permission for owner,rx for group
                              and world
chown owner-user file       =>Change owner of the file
chown owner-user:owner-group file-name      =>Change owner and group
                                              owner of the file
chown owner-user:owner-group directory      =>Change owner and group
                                              owner of the directory
```

NETWORK

```
ip addr show          =>Display all network interfaces and ip address
                        (a iproute2 command,powerful than ifconfig)
ip address add 192.168.0.1 dev eth0     =>Set ip address
ethtool eth0          =>Linux tool to show ethernet status
mii-tool eth0         =>Linux tool to show ethernet status
ping host             =>Send echo request to test connection
whois domain          =>Get who is information for domain
dig domain            =>Get DNS information for domain
dig -x host           =>Reverse lookup host
host google.com       =>Lookup DNS ip address for the name
hostname –i           =>Lookup local ip address
wget file             =>Download file
netstat -tupl         =>Listing all active listening ports
```

COMPRESSION / ARCHIVES

```
tar cf home.tar home      =>Create tar named home.tar containing home/
tar xf file.tar           =>Extract the files from file.tar
tar czf file.tar.gz files =>Create a tar with gzip compression
gzip file                 =>Compress file and renames it to file.gz
```

INSTALL PACKAGE

```
rpm -i pkgname.rpm        =>Install rpm based package
rpm -e pkgname            =>Remove package
```

INSTALL FROM SOURCE

```
./configure
make
make install
```

SEARCH

```
grep pattern files        =>Search for pattern in files
grep -r pattern dir       =>Search recursively for pattern in dir
locate file               =>Find all instances of file
find /home/tom -name 'index*'     =>Find files names that start with "index"
find /home -size +10000k          =>Find files larger than 10000k in /home
```

LOGIN (SSH AND TELNET)

```
ssh user@host             =>Connect to host as user
ssh -p port user@host     =>Connect to host using specific port
telnet host               =>Connect to the system using telnet port
```

FILE TRANSFER

```
scp
scp file.txt server2:/tmp       =>Secure copy file.txt to remote host /tmp folder
rsync
rsync -a /home/apps /backup/    =>Synchronize source to destination
```

DISK USAGE

```
df –h                 =>Show free space on mounted filesystems
df -i                 =>Show free inodes on mounted filesystems
fdisk -l              =>Show disks partitions sizes and types
du -ah                =>Display disk usage in human readable form
du -sh                =>Display total disk usage on the current directory
findmnt               =>Displays target mount point for all filesystem
mount device-path mount-point    =>Mount a device
```

DIRECTORY TRAVERSE

```
cd ..                 =>To go up one level of the directory tree
cd                    =>Go to $HOME directory
cd /test              =>Change to /test directory
```