Recogiendo datos para Analisis Forense de sistemas Windows

"Cuando señalas con el dedo índice a una persona, no olvides que en ese momento hay tres dedos de tu mano que te señalan a ti mismo"

Indice

~

"Bienvenido a mi morada. Entre libremente, por su propia voluntad, y deje parte de la felicidad que trae"
Drácula

1.- Introduccion

Tradicionalmente los forenses vienen dividiendo los datos en **2** tipos:

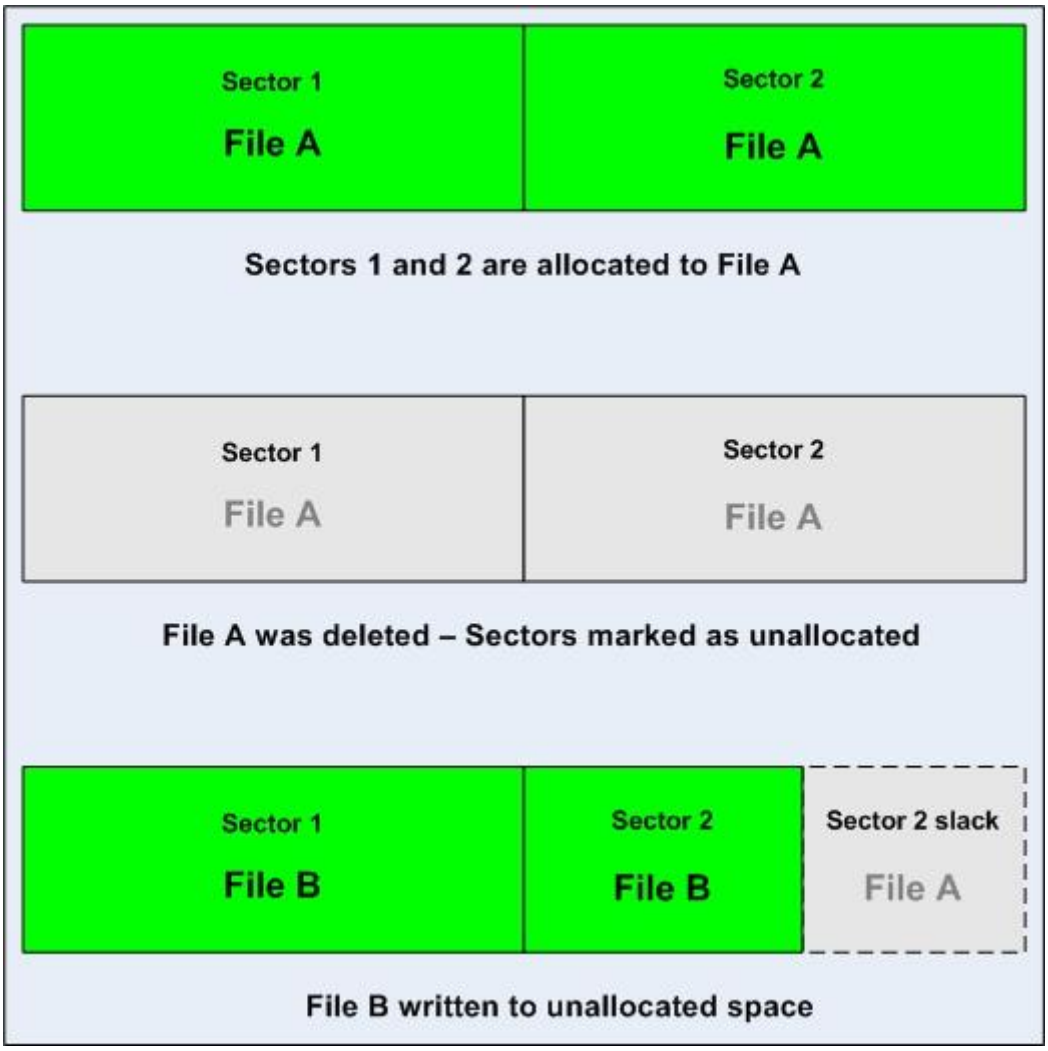
-Datos **volatiles**.
-Datos **no volatiles**.

Los datos volatiles son aquellos que al apagar la maquina a analizar se pierden. Esto no es del todo exacto, pues tal y como debatimos **aqui** es posible recuperar datos almacenados en RAM **_tras apagar_ el equipo**.

Los datos no volatiles por el contrario son aquellos que permanecen en el disco duro tras apagar la maquina.

Tipicamente, un forense adquiriria una imagen del disco mediante herramientas tipo **dd** o cualquiera de sus evoluciones. Nunca con herramientas tipo **Ghost**, pues perderiamos la información a mas bajo nivel como en que **clusters** o **sectores** estaba almacenada la informacion, espacio particionado y libre (***unallocated***), **espacio slack** o espacio sin particionar.

Y como una imagen vale mas que mil palabras, como dice el refran, a continuacion muestro una imagen donde se ve mas claro lo que son estos conceptos:



Sobre el espacio slack me gustaria hacer un inciso, enlazando un video algo viejo pero interesante titulado ***Look for deleted data on the slack space of a disk***.

Tambien comentar la existencia de la genial utilidad **slacker**, sobre la que hablamos, quizas muy por encima para lo que se merece, en un hilo llamado **Anti-Forensics**.

Asi que los que querais iniciaros en el uso de dicha herramienta, podeis consultar un paper bautizado con el atrayente titulo de ***catch me, if you can...*** presentado en la Blackhat 2005 por ***James Foster*** y el crack ***Vinnie Liu***, que no se si a dia de hoy sigue implicado en el ***Proyecto Metasploit***.

Y por ultimo, para los que querais profundizar sobre sistemas de archivos a bajo nivel, os recomiendo el _excelente_ libro ***File System Forensic Analysis***, de **Brian Carrier**; ademas de los talleres de **Vic_Thor** sobre **FATxx** y **NTFS**.

~

"Cualquier contacto, deja un trazo"
Locard's Exchange Principle

2.- Obteniendo datos volatiles

Sobre todo lo que comentare, decir que lo ideal es utilizar siempre binarios llevados por nosotros a ser posible en un CD personalizado por nosotros mismos con binarios limpios y DLLs limpias, _nunca_ usar los binarios del sistema por razones evidentes.

Tal y como comentan **Steve Anson** y **Steve Bunting** en su libro ***Mastering Windows Network Forensics and Investigation***, lo ideal es disponer de un CD para cada S.O. que tengamos que analizar. Tambien explican el proceso de creacion de CDs personalizados.

Lo primero que obtendremos en la maquina presuntamente comprometida sera un volcado de la memoria fisica del sistema, mediante herramientas como **dd**, **windd** o **mdd**.

Mediante **dd**, familiar para los que venimos de UNIX, podemos obtener un completo volcado de la memoria fisica, mediante una sintaxis tal que asi:

_Código:

```
dd if=\\.\PhysicalMemory of=%DIR%\fecha.bin bs=4096 -localwrt
```

Nota: El 4096 viene de multiplicar 1024*4 (4KB).

Actualmente este **dd** es algo obsoleto, por los problemas que paso a enumerar a continuacion.

La version disponible de **dd** en el momento de escribir esto con las **Forensic Acquisition Utilities** no incluye soporte para utilizar el objeto **\Device\PhysicalMemory**, ya que su autor, **George M. Garner** de **GMG Systems** se ha concentrado en el desarrollo de **KnTTools**, cuya version *basic* esta disponible, previa solicitud, para personal acreditado en el sector; tal y como se explica en la *licensing fee*.

Ademas **dd** se basaba en la existencia del objeto **\Device\PhysicalMemory**, el cual a partir de **Windows Server 2003 SP1** y **Vista**, tal y como se menciona en este **articulo de technet**, no es accesible desde modo usuario, asi que en esos casos ya no podemos especificar el pseudodispositivo **\\.\PhysicalMemory** como archivo de entrada a **dd**.

Existen alternativas libres como alguna de las mencionadas anteriormente, ademas de **Autodump+** (AKA **ADplus**), que basicamente es un script **VBS** que se apoya en **Cdb.exe**.

Sobre la adquisicion de volcados de memoria por SW, comentar el principal problema es que al hacerlos en caliente es bastante probable que sean inconsistentes, por lo que existen evoluciones de estas utilidades como el mencionado **ADplus** que puede operar en modo *crash* o modo *hang*.

Podemos encontrar info sobre esta herramienta por ejemplo en el KB de Microsoft titulado **Como utilizar ADPlus para solucionar bloqueos**.

Tambien disponemos de la utilidad **Dumpchk**, de la que podemos encontrar mas informacion en un KB de Microsoft titulado ***How to use Dumpchk.exe to check a memory dump file***.

Es por todos estos problemas, que existen aparatos especificos que permiten adquirir volcados de memoria por HW.

El primero que me consta es uno que bautizaron con el nombre de **Tribble**, anunciado en 2004 en el siguiente paper:

A Hardware-Based Memory Acquisition Procedure for Digital Investigations, Carrier & Grand

De todas formas, aunque se sale un poco del tema, comentar que la idea no es nueva, pues en plataformas SPARC (I'm lovin' it) podemos pasar a la Openboot mediante un **init 0** o pulsando la combinacion de teclas **STOP** + **A**.

Una vez el firmware nos presenta el prompt **ok**, señal de que ya estamos en la linea de comandos de la Openboot, mediante el comando **sync** sincronizariamos los sistemas de archivos y el sistema escribiria un *crash dump* a disco para finalmente reiniciar la maquina.

Y termino el inciso en UNIX recomendadoos un libro relativamente viejo (1995) titulado ***PANIC! UNIX System Crash Dump Analysis Handbook***, de **Chris Drake** y **Kimberley Brown**.

No me entretendre mas sobre la adquisicion de volcados de memoria, pues nuestro compañero **neofito** hablo de ello en este **hilo**.

Un tema relacionado con esto, tambien por **neofito** **aqui**, donde se habla de obtener hashes a partir de un volcado de memoria.

Tambien añadir, y con esto si que realmente termino, si no hay mas sugerencias claro, que via **Conexion Inversa** me acabo de enterar de una interesante manera de obtener un **volcado de memoria de una maquina sin tener acceso _fisico_ a ella**.

Una practica comun entre forenses es definir una variable de entorno _externa_ al disco duro de la maquina comprometida. Recordemos que debemos ser lo menos intrusivos posible.

En los ejemplos siguientes, se ha definido la variable de entorno **%DIR%**, apuntando, a ser posible a la disketera del equipo, ya que debemos ser lo menos intrusivos posible, y la insercion de una llave USB "hace saltar" el servicio de deteccion de HW de windows. El problema es que hoy en dia cada vez es menos comun que los equipos dispongan de una disketera, lo cual dificulta en cierto modo nuestro trabajo. Asi que comentar que a ser posible, la llave USB deberia llevar una proteccion contra escritura, bien mediante la tipica pestañita o habiendo copiado nuestro kit de herramientas en una particion **CDFS** (solo lectura) en la llave USB.

Bien, podemos empezar nuestra adquisicion de datos estableciendo un *listener* _en una maquina remota_ que vaya capturando paquetes enviados por la maquina sospechosa a traves de la red, mediante la mitica utilidad **netcat**:

_Código:
```
nc -L -p 7777 >\datos_listener.txt
```

Tras establecer el listener en la maquina remota, podemos pasarle informacion a traves de la red desde la maquina sospechosa de haber sido comprometida:

_Código:
```
tlist.exe -c |nc <ip_remota> 7777 -w 5
tlist.exe -t |nc <ip_remota> 7777 -w 5
tlist.exe -s |nc <ip_remota> 7777 -w 5
tcpvcon -can |nc <ip_remota> 7777 -w 5
netstat -naob |nc <ip_remota> 7777 -w 5
```

Sobre los 2 ultimos comandos, comentar que resulta de gran ayuda en nuestras investigaciones comparar la salida de dichos comandos.

Tambien comentar que **tcpconv**, con los parametros que hemos especificado, obtiene info en formato **.csv** sobre conexiones TCP/UDP sin resolucion inversa (IPs a nombres), lo cual aumenta su velocidad.

Bien, pasemos a obtener mas datos.

Fecha y hora actual del sistema:

_Código:
```
((date /t) & (time /t)) >%DIR%\SystemTime.txt
```

Uptime de la maquina:

_Código:
```
(systeminfo | find "Boot Time") >%DIR%\uptime.txt
```

Tambien es habitual recoger parametros de red que puedan mostrar evidencias de botnets para realizar por ejemplo SPAM, etc.

_Código:
```
ipconfig /all >%DIR%\ipconfigNICs.txt
netstat -rn >%DIR%\TablaEnrutamiento.txt
nbtstat -c >%DIR%\CacheNombresNetbios.txt
```

Deteccion de NICs en modo promiscuo, señal de que se ha instalado un sniffer en la maquina:

_Código:

```
promqry >%DIR%\promiscuo.txt
```

Esta utilidad puede descargarse del sitio web de Microsoft.

Tambien disponemos de la utilidad **PromiscDetect**

Tambien pueden sernos utiles utilidades que muestren los usuarios logeados en el sistema al estilo de who en UNIX, o conocer los intentos OK/KO de conexiones al sistema. Los genios de SysInternals tienen utilidades CLI de este estilo llamadas **psloggedon** y **logonsession**.

Info sobre procesos en ejecución:

Podemos usar **tasklist /svc** , que creo recordar que recoge los datos de **svchost.exe** , pero yo personalmente prefiero usar **psservice** de SysInternals, que ofrece resultados mas detallados, y que me consta que recoge los datos basandose en el registro de Win. Lo ideal es usar las 2 herramientas para poder comparar. Nunca sabemos a priori que puede haber sido troyanizado.

Arbol de procesos en ejecución (SysInternals):

_Código:
```
pslist -t >%DIR%\ArbolProcesos.txt
```

Descubrir DLLs maliciosas cargadas por procesos en ejecucion, por ejemplo un keylogger:

_Código:
```
listdlls >%DIR%\DLLs.txt
```

De SysInternals para variar. De estos cracks tambien disponemos de la utilidad **handle**, muy similar a la maravillosa y ya mitica utilidad **lsof** de *nix.

_Código:
```
handle -a >%DIR%\handles.txt
```

Con el parametro **-a** conseguimos obtener toda la información que esta utilidad nos puede ofrecer sobre todos los recursos que un proceso dado tiene abiertos: archivos, claves de registro, puertos, etc.

Una practica comun del malware en sistemas windows es ingenierselas para que el malware se ejecute al arrancar el S.O., o cuando un usuario inicia sesion en el sistema. **Mark Russinovich** y **Bryce Cogswell**, de SysInternals tienen una utilidad llamada **autorunsc** que crea un informe exhaustivo de ejecutables que se cargan al inicio:

_Código:
```
autorunsc -a >%DIR%\autorun.txt
```

Decir que disponemos asi mismo de los switches **-m** y **-v**, que sirven, respectivamente, para hacer caso omiso de los ejecutables firmados por fabricantes reconocidos por Microsoft y para verificar firmas digitales.

Comenta **Carvey** que dicha herramienta tambien es util para chequear ciertas areas del sistema como tareas programadas ejecutadas bajo entorno SYSTEM, que como sabreis, dispone de mas privilegios que el usuario Administrador.

Tambien disponemos de utilidades de serie en win para obtener informacion sobre tareas programadas.

_Código:
```
at >%DIR%\at.txt
schtasks /query >%DIR%\tasks.txt
```

Para los que quieran profundizar sobre el analisis de tareas programadas, os recomendaria la entrada ***Windows Scheduler (at job) Forensics***, publicada recientemente en ***SANS Computer Forensics***.

Contenidos del Portapapeles:

Podemos llegar a encontrar info interesante previamente copiada o cortada.

Existe una utilidad llamada **pclip** que nos puede servir para este fin en scripts por ejemplo.

Podemos encontrarla en **http://unxutils.sourceforge.net**.

Por otra parte, os animo a que os leais el siguiente KB en support de Microsoft:

How to prevent web sites from obtaining access to your contents of your windows clipboard

Es aplicable a **IE**, desde las versiones **4** a la **6** al menos.

Historial de comandos:

_Código:
```
doskey /history >%DIR%\historial.txt
```

Puede sernos util enumerar los recursos compartidos, ademas de los archivos que han sido abiertos de forma remota.

_Código:
```
net share >%DIR%\shares.txt
psfile >%DIR%\remote_open_files.txt
```

Metodicamente, lo siguiente que suelen hacer los forenses, es un escaneo de puertos de la maquina sospechosa de haber sido comprometida, desde una maquina limpia.

_Código:
```
nmap -sS <IP_maquina_comprometida>
```

Para los que quieran profundizar sobre el uso de nmap, os recomiendo el libro sobre dicha herramienta que recientemente publico su autor, **Fyodor**, titulado ***nmap Network Scanning***

Tambien os animo a que os leais el resumen que sobre dicho libro hizo nuestro compañero **Sor_Zitroën** en su **blog**, ademas de en este **hilo**, donde podreis encontrar tambien el **enlace** al esquema que se preparo para una charla que dio.

Comentar que existe un interesante proyecto denominado ***The Volatility Framework: Volatile memory artifact extraction utility framework***, especializado precisamente en obtener datos volatiles.

Una vez recolectada toda esta informacion, es _muy importante_ desconectar el cable de red inmediatamente.

~

-¡Mira! una estrella fugaz, pide un deseo cariño.
-Quiero ser invisible mamá.

3.- Obteniendo datos no volatiles

Una vez desconectado el cable de red, podemos estar seguros de que no hay ninguna **_persona_** interfiriendo en la maquina sospechosa de haber sido comprometida.

Fijaos que subrayo *persona*, pues es posible, y puede que probable, que la maquina se encuentre infectada de backdoors, troyanos, rootkits, etc.

Ha llegado a mis manos el libro ***Windows Forensic Analysis***, de **Harlan Carvey**. Paso a comentar algunos retoques que el autor recomienda hacer sobre el registro de Windows antes de comenzar a obtener informacion **no volatil**.

El primer valor se llama ***ClearPageFileAtShutdown***, autodescriptivo por si mismo.

Comenta **Carvey** que si el archivo de paginacion (similar a la *swap* para los que vengais de UNIX) se borra al apagar la maquina, evidentemente perdemos informacion que pueda contener, como por ejemplo passwords en texto plano, o partes de conversaciones de programas de IM; lo cual dificultaria nuestra investigacion.

Podemos encontrar mas informacion al respecto en los siguientes KB de support de microsoft, segun el S.O. sobre el que estemos trabajando:

-**Windows 2000**

-**Windows XP**

Otro valor sobre el que hace hincapie **Carvey** es ***DisableLastAccess***, que podemos encontrarlo en **HKLM\System\CurrentControlSet\Control\FileSystem\NtfsDisableLastAccessUpadate**.

Este valor se encuentra deshabilitado por defecto en **Vista**.

En **Win2k3** este valor por defecto no existe, asi que deberia ser creado.

En **XP** y **Win2k3** podemos consultar el valor de esta clave de registro mediante la siguiente sintaxis:

Código:
```
fsutil behavior query disablelastaccess
```

Bien, pasemos a obtener informacion relevante sobre datos no volatiles.

Obtener info del SW instalado (**psinfo** de SysInternals) y de los parches instalados (**systeminfo**) en la maquina:

Código:
```
psinfo -s >%DIR\sw.txt
systeminfo >%DIR%\parches.txt
```

Una practica comun que suele hacer un atacante es crearse un usuario en el sistema.

Debemos ser muy cuidadosos con esto, pues un atacante inteligente crearia un usuario similar a alguno de los que existen en el sistema por defecto. Por ejemplo, crear un usuario llamado ***TsInternetUser*** puede pasarnos desapercibido haciendonos creer que se trata de un usuario por defecto utilizado por los servicios de **Terminal Server**.

Enumeremos usuarios y grupos.

He aqui un script en **VBS** que enumera las cuentas de **usuario** existentes en el sistema:

Código:
```
Set objNetwork = CreateObject("Wscript.Network")
strComputer = objNetwork.ComputerName

Set colAccounts = GetObject("WinNT://" & strComputer & "")
colAccounts.Filter = Array("user")

For Each objUser In colAccounts
  Wscript.echo objUser.Name
Next
```

Y a continuacion otro script, en **VBS** tambien, que enumera los **grupos** que existen en el sistema:

_Código:
```
strComputer = "."
Set objWMIService =
GetObject("winmgmts:{impersonationLevel=impersonate}!\\" &
strComputer & "\root\cimv2")
Set colItems = objWMIService.ExecQuery ("SELECT * FROM
Win32_Group WHERE LocalAccount = True")

For Each objItem In colItems
  Wscript.Echo "Caption: " & objItem.Caption
  Wscript.Echo "Description: " & objItem.Description
  Wscript.Echo "Domain: " & objItem.Domain
  Wscript.Echo "Local Account: " & objItem.LocalAccount
  Wscript.Echo "Name: " & objItem.Name
  Wscript.Echo "SID: " & objItem.SID
  Wscript.Echo "SID Type: " & objItem.SIDType
  Wscript.Echo "Status: " & objItem.Status
  Wscript.Echo
Next
```

Ambos scripts los he sacado del libro ***The Rootkit Arsenal. Escape and Evasion in the Dark Corners of the System***, de **Bill Blunden**.

Bien, ejecutemos los scripts. Al primero lo hemos llamado **users.vbs** y al segundo **groups.vbs**

_Código:
```
cscript //H:cscript
cscript /nologo users.vbs >%DIR%\users.txt
cscript /nologo groups.vbs >%DIR%\groups.txt
```

A continuacion, ejecutamos el siguiente bucle

_Código:
```
for /F "delims=" %%i in (users.txt) do net user "%%i"
>%DIR%\users_detalis.txt
```

Lo que hace basicamente es recorrer todos los usuarios que previamente el script **users.vbs** guardo en el archivo **users.txt**, y para cada uno de ellos ejecuta el comando **net user**, lo cual nos ofrece una informacion mas exhaustiva, por ejemplo la ultima vez que un usuario dado se logueo en el sistema.

Obtener info sobre ***Event logs***:

_Código:
```
auditpol /get /category:* >%DIR%\Logs_auditpol.txt
wevutil el >%DIR%\Logs_event_names.txt
```

Obtener info del **registro** de win:

Mediante los siguientes comandos obtenemos una copia de las partes mas relevantes del registro, tanto en texto plano como en binario.

_Código:
```
reg save HKLM\SYSTEM system.dat
reg save HKLM\SOFTWARE software.dat
reg save HKLM\SECURITY security.dat
reg save HKLM\SAM sam.dat
reg save HKLM\COMPONENTS components.dat
reg save HKLM\BCD00000000 bcd.dat

reg export HKLM hklm.reg
reg export HKU hku.reg
```

Disponemos en nuestro foro de un **borrador sobre el analisis del registro de windows**, publicado por **neofito**, y que debo reconocer que aun no me he leido (sorry no he tenido tiempo  , pero tengo ganas de ponerme a ello, visto el excelente feedback que produjo)

Un libro al que siempre he visto que hacen referencia practicamente todos los documentos sobre el registro de Win es la 2a Edicion del ***Microsoft Windows Registry Guide***, de **Jerry Honeycutt**, publicado en 2005 por Microsoft Press. No obstante debo decir que no he tenido la oportunidad de leerlo ya que no lo tengo.

Obteniendo marcas de tiempo ***(timestamps)***:

Podriamos obtener un listado de todos los archivos de la unidad **C:** con sus respectivos ***timestamps*** de la siguiente forma:

_Código:
```
dir c:\ /a /o:d /t:w /s
```

En el siguiente **KB** de support de Microsoft podemos encontrar mas informacion al respecto.

Tambien en una entrada titulada con el interesante titulo de **Ensayo acerca del tiempo**, en el **blog** de **neofito**.

Pero quizas, el recurso mas importante, o por lo menos mas extenso sobre *timestamps* sea **Time Forensics. Information about forensic analysis of digital timestamps**, con gran cantidad de enlaces sobre el tema.

Ah, decir que para un analisis minucioso de windows a bajo nivel, ademas de otros muchos aspectos (es un buen "tocho" ), os recomiendo por ejemplo la **5a Edicion** del libro ***Windows Internals***, apasionante para los que gustamos de llegar al mas bajo nivel posible del sistema.

Es bastante reciente, y cubre hasta Windows Server 2008 o Vista.

Y, para "nostalgicos", la 5a Edicion del mitico **PC Interno**, de **M. Tischer** y **B. Jennrich** con mas de 1000 paginas llenas de puro y bello **ASM**. Eso si, se centra sobre todo en DOS, aunque toca tambien hasta Windows 95.

~

"I can clone a human being"
Dr Zavos

"This affair shows a complete lack of responsibility. If true, Zavos has again failed to observe the universally-accepted ban on human cloning,"
Professor Surani

4.- Clonando el disco

No sabia muy bien si llegar hasta aqui, pues ciertamente, todos los autores que he leido vienen separando esta parte de las anteriores.

Sin embargo al final he decidido incluirlo, pues es un aspecto estrechamente relacionado con la adquisicion de datos tanto volatiles como no volatiles.

Como dije en la introduccion, **_nunca_** debemos obtener un clon del disco mediante herramientas como **Ghost** o **Acronis**. Cuidado, no es que sean malas herramientas, de hecho son geniales en otros ambitos. Es solo que simplemente no han sido diseñadas para el Analisis Forense, pues no almacenan la informacion a mas bajo nivel, que es precisamente lo que buscan los forenses.

Existen diversas posibilidades. Podemos usar aparatos diseñados especificamente para clonar discos. Podemos usar **dd** o cualquiera de sus evoluciones. Podemos usar SW como **EnCase** arrancando por ejemplo desde una Live-CD.

Decir que para los que quieran iniciarse en el manejo de **EnCase**, os recomiendo el excelente **Taller de FATxx** por nuestro compañero **Vic_Thor**.

Decir que, a pesar de que la clonacion de un disco esta relacionado con el titulo de este documento, lo cierto es que se sale un poco del tema, tal y como comente antes.

Es por esto que, para no repetir lo mismo que puede encontrarse en la red, he estado buscando info, y de entre todos los documentos, el que mas claro y practico me ha parecido, es el siguiente:

**[url=http://mgtclass.mgt.unm.edu/Alex/MGMT639Spring09/Digital%20Forensic%201%2817%29.pdf]Lab 17 – Digital Forensics (Part 1)
Disk Imaging and Cloning[/url]**, por **Regis Cassidy**, Sandia National Laboratories College Cyber Defenders.

Tambien me gustaria citar aqui un mensaje enviado a la lista Forensics de SecurityFocus:

Folks,

one of my cases has been going on for almost half a year now. Unfortunately, I now do have the need to transfer the forensic dd image to another target disk which will then become the case disk.

Is there anything other than the usual stuff (securely erase the 'new' disk, create checksums before copying, transfer the image using dcfldd, create checksums after copying, securely erase the 'old' disk, document the whole process) which needs to be done?

Am I missing anything important here?

Cheers,

Stefan.

Una vez hemos recolectado la informacion, es cuando llega la hora de realizar realmente un **Analisis** Forense en el laboratorio. Y es ahi donde en mi opinion empieza lo realmente apasionante, esto es, la labor de investigacion.

"El mundo está lleno de problemas fascinantes que esperan ser resueltos"
Eric S. Raymond