



HERLITZCORP
DIVISION DE TECNOLOGIAS

Análisis Forense en Sistemas Linux

por

HEINZ HERLITZ GATICA

Agosto 2007

INDICE DE CONTENIDOS

Resumen	2
1. Capitulo.....	3
1.1 Introducción.....	3
1.2 Objetivo General.....	4
1.3 Objetivos Específicos.....	4
1.4 Justificación.....	4
2. Capitulo II	5
2.1 ¿Que es GNU/Linux?	5
2.2 Principales características de Linux	5
2.2 Distribuciones de Linux forenses	6
3. Capitulo III	7
3.1 Análisis forense en Linux	7
3.1.1 Montaje del dispositivo	7
3.1.2 Verificación de particiones	8
3.1.3 Suma de verificación	9
3.1.4 Creación de imágenes	10
3.2 Información de archivos	11
3.3 Búsqueda de archivos en Linux	13
3.4 Búsqueda y recuperación de datos en Linux	16
3.4.1 Búsqueda de datos	16
3.4.2 Recuperación de datos	17
3.5 Principales log de auditoria en Linux	18
3.6 Scripts para la automatización forense	20
4. Conclusión	22
5. Referencias	23

RESUMEN

El análisis forense informático es una metodología de estudio para la revisión de evidencias digitales, estas evidencias digitales deben ser extraídas, conservadas, identificadas, documentadas, interpretadas y presentadas, de manera que sean legalmente válidas. Usualmente se encuentran en unidades de almacenamiento¹ internas o externas, como por ejemplo: discos duros, disquetes, CDs, DVDs, pendrives, tarjetas de memoria, entre otros.

La revisión de estas evidencias se realiza con el objetivo de encontrar, reconstruir o recuperar información vital que permita descubrir pautas, acciones y procesos realizados a través de medios computacionales.

Generalmente estas revisiones se hacen sobre y desde sistemas operativos Microsoft Windows utilizando herramientas forenses existentes en el mercado como Encase y WinHex, que se obtienen adquiriendo una licencia.

En el presente documento se abordan técnicas de forensia bajo sistemas GNU/Linux, utilizado comandos tradicionales del sistema. Logrando así a través del software libre resultados similares en cuanto a análisis forense se refiere.

¹ Dispositivo de almacenamiento: Es un dispositivo que se utiliza para grabar la información digitalmente.

Capítulo I

1.1 Introducción

El uso de sistemas Linux se ha masificado con el tiempo, las grandes empresas lo utilizan buscando robustez y estabilidad. En los hogares ya lo comienzan a utilizar como alternativa para el PC de escritorio.

Así como se masifica su uso, también se masifican los ataques, el robo de información y otros delitos realizados a/y desde estos sistemas. Por esto es necesario dar a conocer alternativas a las típicas herramientas forenses informáticas y mostrar técnicas diferentes, utilizando los comandos propios del sistema operativo, sin utilizar software adicional que permita encontrar, recuperar o reconstruir la información buscada en una investigación.

El enfoque de este documento es describir y ejemplificar análisis forense en sistemas Linux, sin embargo se puede adaptar fácilmente a otros sistemas operativos que funcionen bajo el esquema de la familia Unix, como lo son: Solaris, OpenBSD, FreeBSD, etc.

1.2 Objetivo General

Describir un esquema de funcionamiento general y los procedimientos necesarios para realizar un completo análisis forense sobre sistemas Linux, utilizando los comandos del sistema y ejemplificando cada proceso.

1.3 Objetivos Específicos

- Descripción general de distribuciones Linux forenses.
- Describir los comandos utilizados para análisis forense.
- Búsquedas de archivos avanzadas.
- Recuperación de datos.
- Describir los principales logs de auditoría.
- Creación de scripts para la automatización de los procesos forenses.

1.4 Justificación

La poca información existente en Internet sobre los pasos a seguir para un determinado peritaje en los sistemas informáticos en general, y sobre todo en sistemas operativos basados en Unix, crea la necesidad de desarrollar este documento. Además el hecho de que con comandos nativos del sistema operativo Unix se pueda realizar análisis forense, posiciona al software libre como una alternativa para desarrollar peritajes informáticos.

Capítulo II

2.1 Que es GNU/Linux

GNU/Linux es la denominación que se da a los sistemas operativos que utilizan el kernel Linux en conjunto con las aplicaciones de sistema creadas por el proyecto GNU. Comúnmente a estos sistemas operativos se les denomina Linux. Fue creado por Linus Torvalds en 1991, quien utilizó minix² como base para su código. Al poco tiempo comenzó a recibir contribuciones de muchos programadores de diversas partes del mundo, lo que ayudo a la consolidación de un sistema robusto y confiable para Equipos PC y servidores. Por ser un sistema de libre distribución posee archivos y programas en diversos lugares de Internet, es por este motivo que diferentes empresas toman el kernel del Sistema Operativo y personalizan sus propias distribuciones de Linux, las cuales poseen ciertas características y sistemas de instalación. Gracias a esto existe una gran cantidad de distribuciones, dentro de las mas conocidas tenemos Debian, Red Hat, Fedora, SuSe, entre otros.

2.2 Principales características de Linux

- **Multitarea:** Describe la capacidad de ejecutar varios procesos al mismo tiempo. Linux utiliza la llamada multitarea preventiva, la cual asegura que todos los procesos que se están ejecutando en un momento dado serán ejecutados, siendo el sistema operativo el encargado de ceder tiempo de microprocesador a cada proceso.
- **Multiusuario:** Muchos usuarios usando la misma maquina al mismo tiempo.
- **Multiplataforma:** Las plataformas en las que se puede utilizar Linux son x86. Amiga y Atari, también existen versiones para su utilización en otras plataformas, como Alpha, ARM, MIPS, PowerPC y SPARC, AMD64, x64.
- **Multiprocesador:** Soporte para sistemas con mas de un procesador esta disponible para AMD, Intel y SPARC, AMD64, x64.

² Minix: Sistema operativo basado en Unix

2.3 Distribuciones Linux orientadas al análisis forense

Así como existen distribuciones para servidores y PC de escritorio, también tenemos distribuciones Linux orientadas a la informática forense, como lo son L.A.S, Helix, F.I.R.E e Insert. Estas distribuciones se pueden conseguir desde la dirección <http://www.securedvd.org/>, donde es posible descargar una imagen de DVD.

Dentro de estos LiveCd existen una gran cantidad de aplicaciones para una variedad de situaciones como por ejemplo el soporte de múltiples sistemas de archivos, soporte para adaptadores WLAN, herramientas para el análisis de red, aplicaciones para recuperación de desastres, antivirus, aplicaciones de forensia informática OpenSource, herramientas de Internet, etc.

 Helix	http://www.e-fense.com/helix/ Es una distribución cuyo fin es proveer un ambiente inmediato para desarrollar el análisis forense, respuesta de incidentes, recuperación de datos, escaneo de virus y evaluación de vulnerabilidad.
 F.I.R.E.	http://biatchux.dmzs.com/ Es una distribución cuya meta es proveer un ambiente inmediato para desarrollar el análisis forense, respuesta de incidentes, recuperación de datos, escaneo de virus y evaluación de vulnerabilidad.
 INSERT	http://www.inside-security.de/insert_en.html Insert es una distribución que está orientado al análisis de red, recuperación de daños, antivirus, análisis forense, etc.
 LOCAL AREA SECURITY	http://www.localareasecurity.com/ Es una distribución LiveCD con gran énfasis en herramientas de seguridad y huellas.

CAPITULO III

3.1 Análisis forense en Linux

Dentro de un esquema general de forensia bajo un sistema Linux, se identificarán procesos típicos a la hora de periciar algún dispositivo ya sea un disco duro, un pendrive, un CD y cualquier dispositivo de almacenamiento.

Para ello se mostrarán los comandos nativos del sistema que permiten realizar estas operaciones. Los pasos a seguir deberían tener una estructura aproximada a la siguiente:

- 1.- Montaje del dispositivo.
- 2.- Obtener las particiones del dispositivo.
- 3.- Realizar suma de verificación.
- 4.- Creación de una imagen para trabajar sobre ella.
- 5.- Realizar un análisis general, recopilando horas de acceso, paquetes instalados, archivos ocultos, información de archivos, entre otros.
- 7.- Generación de un archivo con todas las actividades realizadas en el disco en forma cronológica. (Esto se puede realizar usando el comando *script*, que guarda todos los comandos ejecutados y sus resultados en un archivo *script -a* registro.txt).
- 8.- recopilar la información obtenida.

3.1.1 Montaje del dispositivo

Lo primero que se hace frente a un análisis forense de una evidencia en un sistema Linux es lograr tener acceso a los datos sin la modificación de estos, para esto se montará el dispositivo a periciar con el comando *mount*. El principal cuidado que hay que tener al realizar este proceso es activar la opción de montaje de solo lectura (*ro*), no ejecución (*noexec*) y no dispositivo (*nodev*). Una de las ventajas que tiene la ejecución de este comando es que se puede definir el tipo de sistema de archivos a montar, de lo contrario se puede intentar auto detectar el sistema de archivos. A continuación se describirá la sintaxis de uso de este comando para montar diversos dispositivos en Linux:

```
# mount [-t <tipo>] <dispositivo> <punto_de_lectura> [-o <opciones>]
```

Montajes de dispositivos de solo lectura y no ejecución:

```
[heinz@goku]/root>mount -t ext2 /dev/fd0 /mnt/floppy -o ro, noexec,
nodev
[heinz@goku]/root>mount -t raiserfs /dev/hdb0 /mnt/disco -o ro, noexec,
nodev
[heinz@goku]/root>mount /dev/sda1 /mnt/usb -o ro, noexec, nodev
```

Para el montaje de imágenes de discos ópticos se crea un dispositivo del tipo loopback:

```
[heinz@goku]/root>mount -t iso9660 -o loop /opt/imagen.iso /mnt/imagenes
```

3.1.2 Verificación de particiones

Lo primero es identificar las particiones existentes en el disco duro, para esto se usa el comando *fdisk* con el argumento list (*-l*), que listará la tabla de particiones para el dispositivo especificado. Luego de ser ejecutado se tendrá una salida similar a la siguiente:

```
[heinz@goku]/root>/sbin/fdisk -l

Disk /dev/hda: 40.0 GB, 40020664320 bytes
255 heads, 63 sectors/track, 4865 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1    *           1          13        104391   83  Linux
/dev/hda2                14         4865       38973690   8e  Linux LVM
```

Esta acción se realiza para detectar si existe en el dispositivo alguna partición oculta a la cual no se pueda acceder directamente desde la consola.

3.1.3 Suma de verificación

Una vez montado el dispositivo lo primero que debe realizarse es la suma de verificación³, que es una de las técnicas más simples de para verificar datos. Esto se realiza con el fin de comprobar que los datos no sean modificados o alterados una vez que se tenga acceso a ellos. Para realizar esta suma de verificación se utilizaran comandos como *md5sum* y *sha1sum*, la sintaxis de esta instrucción es:

```
#md5sum <punto de montaje> <archivo>
```

Ejemplo:

```
[heinz@goku]/root>md5sum /dev/hda1
7b8af7b2224f0497da808414272e7af4 /dev/hda1

[heinz@goku]/root>sha1sum DVD-pirate.iso
d7ba5c12f43cb93856a855b147b5eb3c2b087835 DVD-pirate.iso
```

El valor entregado por la suma de verificación debe ser guardada para hacer una comparación una vez que se realice la copia sobre la cual se trabajará.

³ Suma de verificación: Forma de control de redundancia para proteger la integridad de datos.

3.1.4 Creación de Imágenes de los sistemas de archivos

a) Usando dd

Para realizar una copia exacta del dispositivo tenemos varias opciones, pero una de las más típicas y confiables es el uso del comando **dd** (duplicate disk), con los argumentos **if** (input file) y **of** (output file).

La sintaxis es la siguiente:

```
# dd if=<origen> of=<destino>
```

Entonces para duplicar la partición que se encuentra montada en /mnt/disco se tendrá:

```
[heinz@goku]/root>dd if=/mnt/disco of=/opt/destino
```

Lo mejor sería realizar esta copia bit a bit en un nuevo disco duro que tenga espacio suficiente como para mantener la partición completa a verificar, una vez realizada la copia completa se compara la nueva suma de verificación con el resultado anterior, si ambos son iguales se puede proceder a periciar el medio con los comandos tradicionales de Linux. Si por el contrario no se pudiera realizar una copia bit a bit por problemas de lectura en el disco duro, lo que se procede a hacer es realizar una copia que continúe aunque se produzcan errores, esto se realiza con la opción (noerror).

```
[heinz@goku]/root>dd conv=noerror if=/dev/hda1 of=/mnt/nuevodisco
```

b) Usando tar y gzip

Otra manera de realizar copias de seguridad de los archivos es usando los comandos **tar** y **gzip**, el primero deja todo en un solo paquete de archivos y el segundo lo comprime. Los argumentos que recibe son las opciones, el paquete final y el directorio a respaldar.

```
# tar <opciones> <archivo> <directorio>
```

El uso de **gzip** es pasado como argumento al **tar**. Por ejemplo si se quiere respaldar el directorio /mnt/disco/info:

```
[heinz@goku]/root>tar czfv Respaldo.tgz /mnt/disco/info
```

Dentro de las principales características que posee este comando, es que realiza el respaldo manteniendo el dueño de los archivos y la fecha de creación de los mismos, esto también ocurre cuando se realiza la tradicional copia de archivos con el comando *cp* con el argumento *-p*.

Para extraer los archivos que se respaldaron se tendrá que ejecutar la siguiente sentencia:

```
[heinz@goku]/root>tar xzfv Respaldo.tgz /opt/directorio
```

Con esto se tendrán los archivos con nombre, permisos, dueño y fecha original. Luego de esto los archivos estarán listos para ser revisados.

3.2 Información de los archivos

Para obtener información detallada de un archivo como el nombre, tamaño, bloques que ocupa, permisos, dueño, fechas de acceso y modificación existe el comando *stat* el que presenta la siguiente información:

```
[heinz@goku]/home/heinz>stat lib.sh
  File: `lib.sh'
  Size: 219             Blocks: 16             IO Block: 4096   regular file
Device: fd00h/64768d   Inode: 4784411         Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 500/   heinz)   Gid: ( 500/   heinz)
Access: 2007-08-17 08:52:12.000000000 -0400
Modify: 2007-07-27 19:41:42.000000000 -0400
Change: 2007-08-09 12:08:59.000000000 -0400
```

Esta información puede ser muy importante a la hora de realizar un peritaje, ya que presenta datos como fecha de último acceso y modificación, además del usuario dueño del archivo.

Otro comando muy útil a la hora de trabajar con archivos es el comando *file*, el cual determina el tipo de archivo por la cabecera que éste posee. Por ejemplo se tiene el siguiente conjunto de archivos:

```
[heinz@goku]/home/heinz/jail>ls -laht
total 1.8M
drwxr-xr-x  2 root  root  4.0K Aug 17 10:53 .
-rwxr-xr-x  1 heinz heinz  787 Aug 17 10:52 ejecutatodo
-rwxr-xr-x  1 heinz heinz  6.2K Aug 17 10:51 buscarbar
-rw-r--r--  1 heinz heinz  1.3K Aug 17 10:51 buscarbarco
drwxr-xr-x 41 heinz heinz  4.0K Aug 17 10:51 ..
-rw-r--r--  1 heinz heinz 267K Aug 17 10:51 chkrootkit
-rw-r--r--  1 heinz heinz  38K Aug 17 10:51 instalador
-rw-r--r--  1 heinz heinz  1.4M Aug 17 10:51 basura
-rw-r--r--  1 heinz heinz  219 Aug 17 10:50 libreria
```

Como en Linux no es necesario que los archivos posean extensión, no sabemos a que tipo de archivo se refiere el listado anterior, para solucionar esto usamos *file*, que entregará información detallada del tipo de archivo:

```
[heinz@goku]/home/heinz/jail>file *
basura:      MPEG ADTS, layer III, v2,  56 kBits, 22.05 kHz, JntStereo
buscarbar:   ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
for GNU/Linux 2.6.9, dynamically linked (uses shared libs), for
GNU/Linux 2.6.9, not stripped
buscarbarco: ASCII C program text
chkrootkit:  gzip compressed data, from Unix, last modified: Fri Aug 17
10:00:25 2007
ejecutatodo: Bourne-Again shell script text executable
instalador:  RPM v3 bin createrepo-0.4.4-0.2
libreria:    ASCII text
```

Con esto tendremos identificado el tipo de archivo lo que será importante a la hora de buscar archivos como imágenes, documentos, videos, etc.

3.3 Búsquedas de archivos en Linux

Para realizar búsqueda de archivos podemos buscar por extensión, que es un método rápido de búsqueda o se puede buscar por cabecera utilizando el comando *file* anteriormente mencionado conjunto con *find*. El uso de *find* es el siguiente:

```
# find <ruta> <expresión>
```

```
[heinz@goku]/home/heinz/jail>find /mnt/disco -name *.gif
```

Esto mostrará todos los archivos que contengan la extensión de imagen .gif, lo que quiere decir que sea una imagen, solo posee su extensión. Si se desea buscar múltiples extensiones de archivos simultáneamente se tendrá que ejecutar algo como:

```
[heinz@goku]/home/heinz/jail>find / \( -name *.png -o -name *.jpg -o -name *.gif \)
```

Para buscar por encabezado de archivo se usará una integración de comandos a través de tuberías o pipes (|), para aprovechar la redirección de la salida de cada comando conjunto con dos comandos como lo son *xargs* que sirve para pasar parámetros a otros comandos y el comando *grep* que sirve para buscar cadenas de texto.

```
[heinz@goku]/home/heinz>find /mnt/disco | xargs file | grep 'GIF image data'
```

```
/mnt/disco/.home/cli/cabana:  GIF image data,version 89a, 510 x 283  
/mnt/disco/.home/cli/lolita:  GIF image data,version 89a, 557 x 349  
/mnt/disco/.home/cli/terminal: GIF image data,version 89a, 548 x 443  
/mnt/disco/.home/cli/siguiente:GIF image data,version 89a, 548 x 443
```

El comando anterior realiza una búsqueda en /mnt/disco, de todos los archivos que sean del tipo gif pero a través del resultado que nos entregue el comando *file*, con lo que se tendrá un listado de todos los archivos que realmente son imágenes.

Ahora si se tiene que periciar un disco para identificar que archivos fueron modificados en las últimas 24 horas, se ejecuta:

```
[heinz@goku]/home/heinz>find /home/ -daystart -type f -mtime 1
```

Con el argumento *-daystart* le pasamos *-mtime 1* para indicarle que es desde el día actual hasta un día anterior. Más aún si se quisiera una búsqueda mas detallada como por ejemplo los archivos que fueron modificados en un rango de minutos se pasará a través del argumento *-amin*. Por Ejemplo:

```
[heinz@goku]/home/heinz>find /home -amin +2 -amin -60
```

Tendremos un listado de todos los archivos que fueron modificados entre 2 y 60 minutos atrás.

Usando *find* y cualquier otro comando se pueden generar reportes a través de la redirección de las salidas. Si se necesita extraer la primera línea de un archivo para ver si tiene o no relación con alguna situación puntual, se realiza una búsqueda en toda una partición y se ejecuta el comando *head* para extraer solo 1 línea.

```
[heinz@goku]/home/heinz>find /home/heinz -name *.txt -exec head -n 1 -v {} \; > reporte.txt
```

Con la opción **-v** del comando **head** imprime la cabecera del nombre del archivo. El reporte generado se vería así:

```
[heinz@goku]/home/heinz>cat report.txt
==> ./jailkit-2.4/README.txt <==
ABOUT
==> ./jailkit-2.4/INSTALL.txt <==
#QUICK INSTALL
==> ./mp3/sal.txt <==
/usr/sbin/showmount -e
==> ./c/pl/Log.txt <==
Thu Aug 9 14:01:27 CLT 2007
==> ./c/nueva.txt <==
Recopilando info
```

3.4 Búsqueda y recuperación de datos en Linux

3.4.1 Búsqueda de datos

Para buscar una cadena de texto dentro de múltiples archivos dentro de un directorio se usará el comando **grep**, que busca cadenas dentro de archivos o dispositivos.

El modo de uso es el siguiente:

```
# grep <opciones> <patrones> <archivo>
```

```
[heinz@goku]/home/heinz/>grep -i 'JUAN JOSE' *.TXT
Personas.doc:11817465      ,9,210,BARNAO,ROJAS      ,JUAN JOSE
Personas.doc:11817675      ,9,210,BORNDO,ROJAS      ,JUAN JOSE
```

Nos arrojará como resultados todos los archivos que posean la cadena “JUAN JOSE” dentro de todos los archivos con extensión TXT.

Para obtener información de algunos archivos que no son de texto lo que se hace es ejecutar un comando que muestra solo las cadenas de texto imprimibles del archivo, este comando se llama **strings** y puede servir para saber que acción o con que esta relacionado cierto archivo como por ejemplo un ejecutable:

```
[heinz@goku]/home/heinz/reparacion>strings modinfo
/lib/ld-linux.so.2
_Jv_RegisterClasses
__gmon_start__
libc.so.6
fopen
fputs
GLIBC_2.1
GLIBC_2.0
./fix [archivo]
comas.txt
UPDATE Personas SET PerApePat='%s', PerApeMat='%s', PerNombre='%s' WHERE
PerNroDoc='%s' AND TipDocCod='%s' AND PerPaiEmiDoc='%s'
Commit
```

Con esta información se puede deducir que el ejecutable “modinfo” es una aplicación que modifica una tabla personas cambiando los apellidos y nombres.

Dentro de los procesos de búsqueda de datos es usual visualizar los datos de un archivo a través de algún editor hexadecimal⁴. Este se realiza para hacer seguimiento byte a byte del contenido exacto de un archivo, o sea el contrario de la interpretación que otros editores le puedan dar. Para Linux existe un comando llamado *xxd*, que realiza un dump hexadecimal de los datos.

```
[heinz@goku]/home/heinz/reparacion>xxd /home/Heinz/mail.sh
0000000: 2321 2f62 696e 2f62 6173 680a 666f 7220  #!/bin/bash.for
0000010: 6920 696e 2060 7365 7120 3120 3130 603b  i in `seq 1 10`;
0000020: 0a64 6f0a 6d61 696c 202d 7320 2248 6f6c  .do.mail -s "Hol
0000030: 6122 2068 6865 726c 6974 7a40 616c 752e  a" hherlitz@alu.
0000040: 7563 742e 636c 203c 202f 686f 6d65 2f68  uct.cl < /home/h
0000050: 6865 726c 6974 7a2f 6461 746f 732e 7478  herlitz/datos.tx
0000060: 740a 736c 6565 7020 302c 350a 6563 686f  t.sleep 0,5.echo
0000070: 2022 636f 7272 656f 2065 6e76 6961 646f  "correo enviado
0000080: 220a 646f 6e65 0a0a                                     ".done..
```

También *xxd* hace un dump de dispositivos o particiones completas, además de poder transformar un dump hexadecimal a su forma binaria original.

3.4.2 Recuperación de datos

Para realizar una recuperación con los comandos tradicionales, lo que se busca es una cadena de texto que contenga algún texto relacionado al hecho en cuestión, para esto utilizaremos el comando *grep* en conjunto con *strings*, para mostrar solamente los caracteres imprimibles de cualquier archivo, como por ejemplo:

```
[heinz@goku]/home/heinz/reparacion>strings /dev/hda1 | grep '07646-7364-939'
```

Con eso buscamos en toda la partición las la cadena “07646-7364-939”.

La recuperación de archivos en Linux es complicada, esto es por que los sistemas de archivos en Linux funcionan por el medio de desfragmentación en vivo, lo que hace muy

⁴ Editor Hexadecimal: Programa que permite leer o modificar archivos binarios.

difícil la recuperación de los archivos. Existe un comando llamado *e2undel*, que es una extensión de ext2fsprogs, que permite determinar cuales son archivos eliminados, cuales han sido sobrescritos y cuales podrían ser recuperables.

```
[heinz@goku]/home/heinz>e2undel -d /dev/hda1 -s ~/heinz/eliminados
```

La diferencia de recuperación entre ext2 y ext3 es que en ext3 se rellena con ceros el bloque de punteros del inodo, donde ext2 marca el bloque como no usado ext3 marca el inodo como eliminado y deja solo el bloque de punteros.

3.5 Principales logs de auditoria en Linux

Los logs son registros de datos que se almacenan en ciertos archivos definidos por el sistema operativo o por las aplicaciones. Estos registros contienen diversos datos, como usuarios, comandos, direcciones IP, mensajes, etc. Linux posee una gran cantidad de logs que proveen al administrador de sistemas una gran cantidad de información. A continuación se describen las más importantes:

Las instrucciones ejecutadas se pueden revisar a través del log *.bash_history*, si es que la cuenta del usuario tiene por defecto a *bash*, este log se encuentra localizado en:

```
[heinz@goku]/home/heinz>ls -la /home/heinz/.bash_history
-rw----- 1 heinz heinz 10125 2007-08-06 07:18
.bash_history
```

Otro log importante a la hora de revisar un sistema es el de *messages*, este contiene en su configuración por defecto los errores, mensajes de entrada/salida, trabajo en redes, y otros datos generales de sistema:

```
[heinz@goku]/var/log>ls -la /var/log/messages
-rw----- 1 root root 614 Aug 14 10:36 /var/log/messages
```

Si el sistema revisado corre el servidor web Apache, entonces existen dos logs importantes que revisar, estos son *access.log* y *error.log*.

```
[heinz@goku]/var/log>ls -la apache2/*.log
-rw-r----- 1 apache apache 3120 2007-08-13 06:27 access.log
-rw-r----- 1 apache apache 1726 2007-08-13 06:28 error.log
```

Estos contienen los accesos al sitio Web existente en el disco duro y los errores generados por algún tipo de conexión, también registra intentos de conexión a páginas no autorizadas del sitio.

El log de inicio del sistema es importante al momento de ver los dispositivos que se iniciaron por ultima vez después de apagado el equipo, así como también errores al inicio del sistema.

```
[heinz@goku]/var/log>ls -lah /var/log/dmesg
-rw-r--r-- 1 root root 18K Jul 24 18:32 /var/log/dmesg
```

Además si el sistema revisado posee algún motor de base de datos como postgresql, se debería revisar *.psql_history*, el cual es el log que contiene la historia de los comandos ejecutados en el cliente postgresql.

```
[heinz@goku]/root>ls -lash .psql_history
8.0K -rw----- 1 root root 351 Jul 26 19:12 .psql_history
```

wtmp almacena la información de las conexiones y desconexiones al equipo, Contiene datos como el nombre del usuario, por donde accede, la dirección IP de origen, la fecha y hora del acceso.

```
[heinz@goku]/var/log>ls -lash /var/log/wtmp
36K -rw-rw-r-- 1 root utmp 29K Aug 17 09:32 /var/log/wtmp
```

Que puede ser leído a través del comando *last*, con la opción *-f*

```
[heinz@goku]/root>last -f /var/log/wtmp
```

3.6 Scripts para la automatización de procesos forenses

El siguiente Script inicia una búsqueda en todo un dispositivo, buscando archivos de imágenes ('image data') por cabecera de archivo gracias al comando *file* y que además contenga las palabras clave: porno, child, sex, xxx.

```
#!/bin/bash
# palabras clave a buscar
b1="porno"
b2="child"
b3="sex"
b4="xxx"
destino="/home/heinz/peritaje/"

echo 'hora de inicio : ' $(date) >> Imagen.txt
echo 'usuario : ' $(whoami) >> Imagen.txt
echo 'Pto de Montaje'
read pto

echo 'Montaje :' $pto >> Imagen.txt

find $pto | awk '{ print "\"" $line "\"" }' | \
  xargs file | grep 'image data' | \
  awk ' { split ($line,nombre,":"); print "cp -p \"" \
    nombre[1] "\"" " $destino" }'
grep '$b1\|$b2\|$b3\|$b4' | bash

# fin de la copia
echo 'hora de termino :' $(date) >> Imagen.txt
```

Los archivos encontrados son copiados al directorio **/home/heinz/peritaje**. Manteniendo los permisos, fecha y el dueño de los archivos.

El script descrito a continuación realiza una búsqueda de una cadena en el punto de montaje, ambos definidos por el usuario. La salida es un archivo llamado `reporte.txt`, que contiene la ruta de todos los archivos que posean esa cadena.

```
#!/bin/bash
clear
echo "Hora de inicio:" $(date) >> reporte.txt
echo " " >> reporte.txt
echo 'Cadena a buscar:'
read busca
echo 'Pto de Montaje:'
read pto

echo "Buscando $a en $pto" >> reporte.txt
echo " " >> reporte.txt

find $pto | awk '{ print "\"" $line "\"" }' | \
    xargs grep $busca $rep >> reporte.txt

echo " " >> reporte.txt
echo 'Hora de termino' $(date) >> reporte.txt
```

Para finalizar se describe un script que genera una imagen iso con el comando *mkisofs*, de un archivo o dispositivo ingresado por el usuario, una vez creada la imagen se monta con el comando *mount* de solo lectura para poder trabajar sobre el directorio montado.

```
#!/bin/bash
destino="/mnt/cdrom"
echo 'Generar backup iso'
echo 'Hora: ' $(date)
echo 'Nombre Imagen:'
read nombre
echo 'Ruta a copiar'
read ruta
mkisofs -R -J -T -o $nombre.iso $ruta
mount -o loop $nombre.iso $destino -o ro,noexec,nodev
cd $destino
echo 'Hora' $(date)
```

CONCLUSION

Linux se presenta como un entorno ideal para poder realizar tareas relacionadas al análisis digital forense, ya que se encuentra dotado de una gran cantidad de comandos y herramientas que facilitan un análisis completo del sistema. Además provee de un entorno seguro con una gran cantidad de archivos log que permiten al administrador de sistemas o al perito informático, encontrar la información necesaria para determinar los hechos ocurridos bajo un incidente. Las características nativas que posee son ventajas frente a otros S.O., como el soporte de múltiples sistemas de archivos, el tratamiento de todo como un archivo, el montaje de múltiples dispositivos, entre otras.

REFERENCIAS

- 1.- **Helix Linux (e-fense)**
<http://www.e-fense.com/helix/>
- 2.- **Fire Linux**
<http://biatchux.dmzs.com/>
- 3.- **Insert Linux**
http://www.inside-security.de/insert_en.html
- 4.- **L.A.S. Linux**
<http://www.localareasecurity.com/>
- 5.- **Análisis forense de sistemas GNU/Linux, Unix**
David Dittrich, Ervin Sarkisov
- 6.- **Análisis forense de sistemas Linux**
Juan Manuel Canelada Oset