

Sleuthkit, la herramienta forense digital

La Huella

Sleuthkit realiza análisis forenses en archivos del sistema de Microsoft y Unix aplicando sus habilidades como detective desde la línea de comandos para identificar evidencias, recuperar archivos borrados o reconstruir escenarios. Todo esto es esencial en el arte del trabajo de los forenses digitales. En este incipiente campo es mejor prepararse y practicar para los problemas que nos acecharán en el futuro.

POR RALF SPENNEBERG



Los cortafuegos y los sistemas de detección de intrusión (IDS) alertan a los administradores y proporcionan protección contra la mayoría de los ataques. Pero los intrusos siguen encontrando agujeros y atacando máquinas. El administrador entonces debe actuar como un forense realizando pruebas en el sistema aún vivo (encendido) o ya muerto (apagado) para salvar evidencias vitales.

Herramientas como TCT, la clásica utilidad forense (el paquete de herramientas Coroner [4] o el más nuevo Sleuthkit [1], objeto de este artículo), ayudan a resolver estos problemas. Tras una intrusión, hay una serie de difíciles preguntas que responder como:

- ¿Cómo ocurrió la intrusión?
- ¿Qué provocó que el cortafuegos no parase el ataque?
- ¿Por qué el IDS no detectó el ataque (hasta que fue demasiado tarde)?
- ¿Qué sistemas han sido afectados?

- ¿Ha sido modificada alguna información almacenada en los sistemas modificada? ¿Cuál?
- ¿Qué pretendía el intruso?

Habitualmente es posible responder a estas preguntas tras examinar los protocolos del cortafuegos y del IDS... si previamente el administrador ha implementado un sistema de seguridad y acceso adecuado. Pero la monitorización integral de sistemas y redes normalmente va en contra de las políticas de protección de datos personales.

El análisis forense de los sistemas afectados tras el evento es la única forma de obtener evidencias claras. Los procedimientos exactos son complicados y dependen de numerosos factores. Si estamos buscando una introducción completa en la materia hay unos pocos de documentos (algunos disponibles en la Red) que pueden ayudar [5]. El propósito de cualquier investigación forense es averiguar los eventos y la secuencia en la que sucedieron: la serie de actividades que llevaron a la ocurrencia y cualquier

acción que el intruso ejecutara en la máquina de la víctima. Si los resultados van a ser usados posteriormente como evidencias hay requerimientos de cara a documentación ambigua de los pasos individuales de la investigación.

Dispositivo de circuito cerrado mejorado

Una vez que el controlador de circuito cerrado mejorado y el comando *losetup* modificado han sido instalados hemos creado los dispositivos adecuados usando el comando *createdev*. Ahora simplemente podemos montar la imagen de un disco duro completo usando los siguientes comandos:

```
# losetup -r /dev/loopa host_2
hda.dd
# fdisk -l /dev/loopa
Ahora podemos acceder a las particiones individuales usando /dev/loopa1,/dev/loopa2, etc. Puesto que la opción -r establece el dispositivo de circuito cerrado como solo lectura, incluso los sistemas de archivos de publicación no pueden escribir.
```

TCT y Sleuthkit

El primer paquete de herramientas para el análisis forense de sistemas UNIX fue escrito por Wietse Venema y Dan Farmer en 1999 [5]. El paquete Coroner (TCT) es una colección de comandos que los administradores pueden usar para realizar tareas forenses tras un incidente en ordenadores UNIX. Wietse Venema aún mantiene este paquete de herramientas y ofrece regularmente actualizaciones en su página Web [4]. TCT puede analizar los siguientes sistemas operativos: Sun Solaris, SunOS, Linux, FreeBSD, OpenBSD y BSD/OS. Brian Carrier de Atstake (típicamente dele-

treado @stake [3]) comenzó añadiendo herramientas adicionales a TCT al principio, lanzando estos como utilidades de TCT. Estas utilidades ofrecen una serie de funciones de sistemas de archivos añadidos a los que aparecen en TCT.

Analizando sistemas de archivos no UNIX.

Al principio del 2002 Brian Carrier implementó el soporte a FAT y NTFS para proporcionar capacidad analítica a sistemas de archivos adicionales. También trabajo en los códigos fuente de TCT, lo que fue razón suficiente para cambiar el nombre a TASK,

The Atstake Sleuth Kit. Brian lanzó adicionalmente el visor Autopsy Forensic. Al margen de su nombre es un visor Web que los administradores pueden encontrar útil como un interfaz GUI para comandos TASK. En abril del 2003 Brian Carrier cambió de nuevo el nombre del proyecto a "The Sleuthkit" para marcar el lanzamiento de la versión actual, la 1.61. Esto enfatiza la naturaleza Código Abierto del proyecto y su independencia de la corporación Atstake. Los archivos del proyecto están disponible en Sourceforge [1,2].

Ejecutar los pasos individuales de la investigación en un orden determinado asegurará que los resultados son reproducibles: "Asegurar y aislar la escena. Grabar la escena, proseguir con una búsqueda sistemática de evidencias". [6]

Asegurar y Aislar.

El primer paso en el análisis forense es un asunto muy controvertido. La mayoría de expertos recomiendan desconectar el sistema de la red activa inmediatamente. Esto previene que el intruso se de cuenta del análisis y borre sus pasos.

Algunos expertos no están de acuerdo con esto. El intruso puede haber instalado un código oculto que automáticamente borra el sistema si este es desconectado de la red. Es difícil hacer una recomendación general, pero yo suelo recomendar y usar el primer método.

Grabando el escenario

Analizamos un ordenador comprometido, el experto forense primero debe procurar salvar la información volátil antes de proceder a salvar la no-volátil.

Las siguientes son volátiles:

- Memoria principal.
- Mensajes del Kernel.



Figura 1: Cuando el IDS anuncia un intruso (en este caso Short está usando Win-Popup para alertar una explotación IMAP) es tarea del administrador el iniciar los procedimientos de respuesta ante incidentes. Sleuthkit nos puede ayudar a analizar los sistemas de archivos.

- Fecha y hora.
- Procesos activos.
- Archivos abiertos.
- Configuración de red.
- Conexiones de red.

Esta información se pierde cuando el sistema es apagado, pero puede ser extremadamente útil. Típicamente solo necesitamos comandos de Linux normales para asegurar estos datos, y el comando *grave-robber* de tct es muy útil. El comando no es parte de Sleuthkit. Es importante usar programas de fuentes muy fiables cuando aseguramos una evidencia. Por ejemplo, comandos compilados estáticamente en un CD o en un disquete protegido contra escritura.

Los datos no-volátiles es lo siguiente. Estos significa los archivos del sistema y la memoria intercambio. Adicionalmente debemos grabar cualquier documento cercano y la apariencia general del sistema. Una cámara puede ser muy útil por motivos de documentación.

Simple Copia de Seguridad

Una simple copia de seguridad del sistema no es suficiente. La copia de los archivos del sistema deben ser idénticos a los originales, y podemos perder archivos borrados por el intruso. La copia de seguridad debe incluir espacio de sobra. La herramienta *dd* de UNIX puede ejecutar esta tarea. La versión Windows está disponible en [7]. EL administrador debe usar *dd* para hacer copias de seguridad del disco duro completo y no solo de particiones individuales. Al hacerlo es importante no almacenar ningún dato en el disco original. En su lugar debemos usar un disco externo usando la herramienta Netcat, su

homólogo criptográfico Crycat o un túnel SSU para transferir los datos al sistema forense en el que comprobaremos y analizaremos los datos.

Migrando datos

El Netcat debe estar a la escucha en la máquina de destino si usamos Netcat para transferir el contenido del disco:

```
nc -l -p 3000 > host_hda.dd
```

Y entonces pasamos los datos del disco duro desde el sistema que tenemos que copiar hasta Netcat:

```
dd if=/dev/hda | nc Server 3000
```

Los investigadores forenses saben muy bien que deben calcular la suma de control del disco imagen inmediata-

Listado 1: Particiones para el primer ejemplo.

```
01 Disk /dev/hda: 0 heads, 0
sectors, 0 cylinders
02 Units = sectors of 1 * 512
bytes
03 Device Boot Start End Blocks
Id System
04 /dev/hda1 63 15119999 7559968+
83 Linux
05 /dev/hda2 15120000 80408159
32644080 5 Extended
06 /dev/hda5 15120063 16178399
529168+ 82 Linux swap
07 /dev/hda6 16178463 24373439
4097488+ 83 Linux
08 /dev/hda7 24373503 32568479
4097488+ 83 Linux
09 /dev/hda8 32568543 80408159
23919808+ 83 Linux
```

mente después de la copia. Esta suma proporciona después evidencia de que la copia de seguridad no ha sido manipulada.

```
md5sum host_hda.dd
```

La siguiente tarea es dividir el disco duro en particiones puesto que Sleuthkit solo puede manejar particiones. El comando `fdisk -lu host_hda.dd` lista las particiones (ver listado 1).

Podemos usar de nuevo `dd` para extraer particiones individuales. Para hacer esto necesitamos calcular el tamaño de estas usando los cilindros del inicio y del final. En el caso de `hd1` esto supone $15\ 119\ 999 - 63 + 1 = 15\ 119\ 937$. `dd` puede entonces usar esta información para guardar la primera partición en un archivo propio:

```
dd if=host_hda.dd of=host_hda1.  
.dd bs=512 skip=63 count=2  
15119937
```

Si estos pasos parecen muy complicados podemos hacer copia de seguridad de las particiones individuales, pero esto supone correr el riesgo de perder información necesaria para el análisis forense si ésta está almacenada en áreas no particionadas. El controlador mejorado de circuito cerrado [8], que es capaz de acceder al disco duro completo como un dispositivo de circuito cerrado, es más práctico que la separación manual de imágenes individuales del disco.

Búsqueda Sistemática

Tras guardar los archivos del sistema del equipo comprometido, crearemos las sumas de comprobación y almacenaremos una copia en un lugar seguro, el investigador puede proceder a analizar una copia de los archivos del sistema dejando el original como una evidencia no tocada. Sleuthkit proporciona una serie de comandos para este paso. Veremos algunas de las más importantes herramientas en las siguientes secciones.

La segunda parte de esta serie describe unas sencillas investigaciones usando Autopsy.

Su instalación es extremadamente sencilla: descargamos el paquete desde su página Web, lo extraemos y ejecutamos `make`. Esto sitúa las herramientas en el subdirectorio `bin/`. El visualizador forense Autopsy es muy fácil de compilar, sin embargo, puede que prefiramos usar los paquetes RPM.

Las herramientas Sleuthkit están divididas en 4 categorías. La primera muestra información de sistemas de archivos completos y solo contiene el comando `fsstat`. El segundo grupo comienza por `d` y nos permite acceder a los datos almacenados en archivos como: `dcalc`, `dcat`, `dls`, and `dstat`. Sleuthkit nos proporciona las siguientes herramientas para información meta almacenada en índodos: `icat`, `ifind`, `ils`, and `istat`. Los comandos en el 4º grupo comienzan con `f` y están diseñados para tareas de nivel de

Despertando de la Muerte

Restaurar archivos borrados no es para nada una tarea trivial en sistemas de archivos UNIX. No conozco ningún comando `unerase` para restaurar datos sin peligro de pérdidas independiente del sistema operativo UNIX que usemos. Por tanto ese esencial disponer de algunos conocimientos previos de cómo guarda y borra datos el sistema de archivos para usar herramientas como TCT o Sleuthkit y para afrontar los incidentes que puedan surgir al restaurar archivos borrados. El segundo sistema de archivos extendidos (EXT 2) es un útil ejemplo de un índodo heredado basado en sistemas de archivos UNIX. Cada archivo se representa por una estructura especial, su índodo. El índodo almacena la información meta que pertenece al archivo. Adicionalmente, se requieren los bloques de datos para almacenar el payload. Los directorios son simple archivos especiales que de nuevo comprometen el índodo. Sus bloques de datos almacenan las listas de directorio que contienen nombres de archivos y enlaces a los índodos apropiados.

El índodo en EXT2

Los índodos almacenan la información meta para un archivo excepto su nombre. Esto incluye su tamaño, tipo, permisos, propi-

etario y grupo, el contador de referencia y 3 sellos de tiempo UNIX (`ctime`, `atime` y `mtime`). El índodo también contiene 12 referencias directas a bloques de datos que son las direcciones de los primeros 12 bloques de datos en el archivo. Los índodos usan adicionalmente referencias indirectas, apuntando la primera a un bloque de datos que contiene referencias directas a bloques de datos. Las 2 últimas referencias apuntan a un bloque indirecto doble o triple (ver figura 2). Cuando el sistema de archivos borra un archivo simplemente marca el acceso del directorio y el índodo como borrado y libera el espacio ocupado por el índodo y los bloques

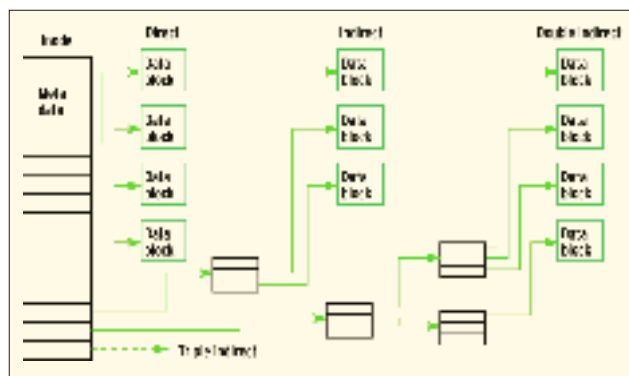


Figura 2: El índodo (izquierda) guarda la información meta para un archivo. Referencia los bloques de datos con los contenidos del archivo. Los archivos más grandes pueden requerir el uso de referencias indirectas.

de datos. Las referencias a los bloques de datos y el enlace entre el nombre del archivo y el índodo tiende a no cambiar. Por tanto es posible usar herramientas como `ils` para mostrar índodos borrados e `icat` para restaurar los contenidos de los archivos. No obstante esto solo funciona si el sistema de archivos no ha reasignado el índodo o los bloques de datos a otro archivo. El comando `fs` muestra los nombres de los archivos borrados.

LAS IMPLEMENTACIONES MODERNAS BORRAN MÁS.

Muchas distribuciones modernas de Linux borran archivos de una forma que previenen de forma efectiva que sean restauradas.

Desafortunadamente esta opción de seguridad tiende a impedir el análisis forense con herramientas como `ils` y `icat`. En este caso, el administrador que realice la investigación se ve forzado normalmente a restaurar usando el comando Sleuthkit `dls`, que lee el espacio descuidado del disco duro. El investigador puede entonces usar `grep` o la herramienta `sorter` para investigar los resultados. Autopsy es un interfaz elegante que ayuda con estas investigaciones más complejas.

Listado 2: Archivos Borrados

```
01 class|host|device|start_time
02 ils|kermit.spenneberg.de|honeybot.hda5.dd|1052056153
03 st_ino|st_alloc|st_uid|st_gid|st_mtime|st_atime|st_ctime|st_dtime|st_mode|st
04 _nlink|st_size|st_block0|st_block1
05 1|a|0|0|973385737|973385737|973385737|0|0|0|0|0|0
06 1890|f|17275|20|973695367|973695367|973695367|973695367|40755|0|0|6172|0
07 1891|f|17275|20|846630170|973695173|973695367|973695367|100644|0|1082|6173|0
08 1892|f|17275|20|973695367|973695367|973695367|973695367|40755|0|0|6174|0
09 1893|f|17275|20|846630171|973695173|973695367|973695367|100644|0|1458|6175|0
10 1894|f|17275|20|846630171|973695173|973695367|973695367|100644|0|1362|6176|0
11 2243|f|17275|20|846630171|973695173|973695367|973695367|100644|0|1465|6177|0
```

archivos como la creación de listas y búsqueda de archivos.

El comando *mactime* muestra los eventos cronológicamente en un sistema. Los comandos *file* y *sorter* clasifican los archivos en la imagen del disco duro por tipo.

Archivos de Sistema Soportados

Sleuthkit analiza los tipos de archivos en el sistema mostrados en la tabla 1. Desafortunadamente no soporta archivos de sistema típicos de Linux como Reiser FS, JFS o XFS. El cuadro “Despertando de la Muerte” describe como funcionan los archivos del sistema UNIX, indicando

la información hacia la que las herramientas están orientadas y mostrando sus limitaciones.

EXT2 y EXT3 pueden producir algunos efectos secundarios desagradables en algunas distribuciones modernas. Dependiendo de los parches aplicados, estos archivos de sistema pueden borrar la referencia a los bloques de datos del inódo y esto impide a herramientas sencillas como *icat* restaurar el contenido de un inódo borrado.

Sugerencias Prácticas: Reanimación.

Ahora es momento de analizar las imágenes del disco duro de las que hicimos

copias de seguridad. Nuestro ejemplo usa el sistema de archivos Forensic Challenge. Esta competición fue organizada por el proyecto HoneyNet en enero del 2001. Un archivo tar que contiene datos está disponible para ser descargado desde varios sitios [9]. Puesto que las particiones han sido separadas previamente no podemos usar estos archivos para ilustrar los primeros pasos de la copia de seguridad del disco duro.

Tras la descarga, el forense novato debe extraer los datos y verificar la suma de comprobación. Esta última está localizado en el archivo *readme*. El siguiente comando extrae la suma MD5 y las verifica con las imágenes individuales:

```
# tail +12 readme | head -6 | md5sum -c
honeybot.hda1.dd: 0k
honeybot.hda5.dd: 0k
honeybot.hda6.dd: 0k
honeybot.hda7.dd: 0k
honeybot.hda8.dd: 0k
honeybot.hda9.dd: 0k
```

La función de las particiones individuales se proporciona para suplementar la investigación (ver tabla 2). Montaremos las particiones como solo lectura usando el dispositivo de circuito cerrado que proporciona un vistazo inicial. Pero esto puede ser peligroso porque puede modificar los datos. Ver el cuadro “Solo lectura, escritura a veces”.

Los comandos Sleuthkit proporcionan un vistazo más profundo. Los archivos borrados son el primer punto de interés (si el atacante ha intentado borrar sus huellas) este es normalmente el mejor camino para descubrirlos.

Índos de archivos borrados

El comando *ils* muestra información inódo de un archivo del sistema; el parámetro *-r* indica al parámetro que se concentre en inódos borrados (por defecto). El listado 2 muestra los resultados para *ils -f linux-ext2 -r honeybot.hda5.dd* El resultado está en formato de tabla, conteniendo la línea 1 la cabecera y la línea 2 los datos correspondientes. Evidentemente el comando se ejecutó en

Listado 3: Descubriendo Datos Interesantes

```
01 # ils -f linux-ext2 -r honeybot.hda5.dd | tail +4 |
02 > awk -F '|' '{ $11 > 0 {print $1} }' |
03 > while read in; do
04 > icat honeybot.hda5.dd $in > data/hda5.icat/$in
05 > done
06 # file data/hda5.icat/* | egrep "tar|gzip|RPM|ELF"
07 data/hda5.icat/109791: GNU tar archive
08 data/hda5.icat/109861: GNU tar archive
09 data/hda5.icat/109865: RPM v3 bin i386 nfs-utils-0.1.9.1-1
10 data/hda5.icat/109866: RPM v3 bin i386 wu-ftpd-2.6.0-14.6x
11 data/hda5.icat/109943: ELF 32-bit LSB relocatable, Intel 80386,
    version
12 1 (SYSV), not stripped
13 data/hda5.icat/109944: ELF 32-bit LSB relocatable, Intel 80386,
    version
14 1 (SYSV), not stripped
15 # tar tf data/hda5.icat/109791
16 ssh-1.2.27/
17 ssh-1.2.27/COPYING
18 ssh-1.2.27/README
19 ssh-1.2.27/README.SECURID
20 [...]
```

Solo lectura, Escritura a Veces

Cuando realicemos análisis forenses es normalmente útil montar las particiones del disco duro para buscar archivos. Esto es particularmente cierto para Reiser FS, puesto que Sleuthkit no soporta el sistema de archivos de Reiser. Para garantizar la integridad de los datos, los investigadores deben generar sumas e comprobación para todos los sistemas de archivo y montarlas como solo lectura. Desafortunadamente, el modo solo lectura no siempre se puede considerar como válido: los sistemas de no publicación realmente no cambian nada si son montados como solo lectura. Sin embargo, los sistemas de archivos de publicación como EXT3 o reiser FS tienden a actualizar la evento de la publicación de opciones montadas como de solo

lectura. Por supuesto, esto cambia la suma de comprobación MD5, hay una simple prueba para esto: necesitaremos una imagen de prueba y su suma de comprobación para el ensayo:

```
# dd if=/dev/zero
of=/tmp/testimage.orig
bs=1024k count=50
# losetup /dev/loop0
/tmp/testimage.orig
# mkreiserfs /dev/loop0
# md5sum /tmp/testimage.orig
2ceed9e819bf4348a33a21f7697149c8
/tmp/testimage.orig
Tras montar en solo lectura e inmediatamente desmontar, la suma MD5 ha cambia-
```

do:

```
# mount -o ro -t reiserfs
/dev/loop0 /mnt
umount /mnt
# md5sum /tmp/testimage.orig
c6ffc8a13a6821cd327d1db9ee351faf
/tmp/testimage.orig
```

Solo hay 3 formas de evitar esto:

- Modificar los controladores EXT3 o Reiser FS.
- Salvar el sistema de archivos en un CD-ROM.
- Usar el controlador mejorado de circuito cerrado [8].

el puesto *kermit spenneberg.de* para un archivo llamado *honeypot hda5.dd*. Las cosas comienzan a ser más interesantes en la línea 3. Ésta es la segunda cabecera de la tabla y se utiliza para cualquier línea restante. La primera columna, *st_ino* contiene el número inodo, y la segunda columna (*st_alloc*) indica si el inodo esta libre (*f*) u ocupado (*a*). Hay muchos caminos para descubrir archivos interesantes. El primer camino supone cualquier archivo con tamaño superior a cero (columna 11, *st_size*). Estos archivos se investigan usando el comando *file* con el objeto de catalogar el contenido del archivo. Para hacer esto la herramienta accede a una gran base de datos que contiene casi todos los formatos de archivos conocidos (*magic**). La versión Sleuthkit de *file* no es diferente de la versión incluida con muchas distribuciones, si bien la base de datos puede ser mayor.

Archivos Interesantes

El listado 3 muestra las sintaxis posibles. La línea 1 quita las primeras 4 líneas de la información inodo del comando *ils* y pasa el resto al comando *awk* (línea 2). Esto encuentra cualquier inodo con tamaño superior a cero y pasa su número al comando *icat*. Este último extrae los archivos de la imagen y los guarda en un subdirectorio llamado *data/hda5.icat/* usando como nombre del archivo el número del inodo.

El comando *filees* entonces usado para garantizar el tipo de archivo y *grep* filtra los archivos interesantes (línea 7). La

investigación en el primer archivo tar (línea 15) muestra que hemos descubrier- to SSH Versión 1.27.

Diario de Abordo

El segundo método para analizar archivos borrados implica que el investigador forense cree primero un diario cronológico de las operaciones de los archivos en el sistema. El diario cubrirá cualquier archivo en lugar de solo archivos borrados. También tiene sentido cubrir todos los archivos del sistema en el análisis.

El administrador primero recoge datos de los archivos del sistema (ver figura 3). Al igual que el

familiar comando *ils*, este implica usar *fls*. Este comando recoge información de cualquier archivo que aún existe en el sistema de archivos. La opción *-m* crea un resultado que luego es procesado por el comando *mactime*. También se muestra el punto de montaje. Se requiere el parámetro *-m* para *fls*, no para el punto de montaje.

El archivo utilizado para recoger la mayoría de los datos se llama *body* por

Listado 4: Sumario de Mactime

```
01 ed Nov 08 2000 14:51:53 17969 .a. -/-rwxr-xr-x 1010 100 109832
/usr/man/.Ci/scan/x/x
02 1760 .a. -/-rwxr-xr-x 1010 100 109829 /usr/man/.Ci/scan/bind/ibind.sh
03 15092 .a. -/-rwxr-xr-x 1010 100 109836 /usr/man/.Ci/scan/x/pscan
04 4096 .a. d/drwxr-xr-x 1010 100 109841 /usr/man/.Ci/scan/port/strobe
05 1259 .a. -/-rwxr-xr-x 1010 100 109834 /usr/man/.Ci/scan/x/xfil
06 4096 .a. d/drwxr-xr-x 1010 100 109831 /usr/man/.Ci/scan/x
07 [...]
08 Wed Nov 08 2000 14:52:09 9 m.c 1/1rwxrwxrwx 0 0 46636
/root/.bash_history -> /dev/null
09 9 m.c 1/1rwxrwxrwx 0 0 23 /.bash_history -> /dev/null
```

Tabla 1: Sistemas de archivos en Sleuthkit

Sistema de Archivos	Opción de Línea de Comandos
BSDi FFS	bsdi
FAT Filesystem FAT12	fat12
FAT Filesystem FAT16	fat16
FAT Filesystem FAT32	fat32
FreeBSD FFS	freebsd
Linux Filesystem EXT2	linux-ext2
Linux Filesystem EXT3	linux-ext3
NTFS	ntfs
OpenBSD FFS	openbsd
Solaris FFS	solaris

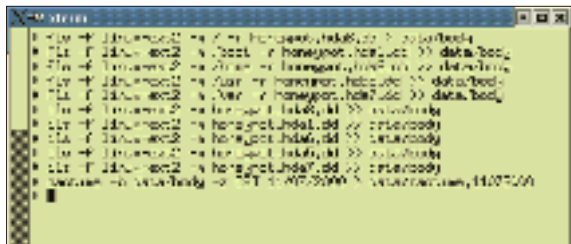


Figura 3: Para analizar el orden de las modificaciones de múltiples sistemas de archivos el investigador necesita cotejar la información del archivo y del inodo en el archivo *body*.

motivos históricos. Ejecutar *mactime* para este archivo genera un diario de abordo de modificaciones del sistema de archivos. Podemos restringir el resultado al punto en el que estemos interesados. En nuestro ejemplo cogido de Forensic Challenge podemos suponer que el ataque se produjo el 7 de noviembre de 2000.

Monitorización del Tiempo Consistente

Para proveer monitorización del tiempo constante debemos usar el parámetro *-z* para especificar la zona horaria usada por el sistema que crea los datos. En este caso usamos CTS (horario central estándar - GMT-0600). El resultado muestra actividades no usuales a las 14:51 el 8 de noviembre (ver listado 4). El directorio */usr/man/.Ci/* no debería estar allí en realidad.

Los archivos que contienen cambios también son de interés. Estos archivos son fácilmente reconocibles por la *m*

en la tercera columna. Una *a* indica acceso al archivo y una *c* indica que ha sido modificada información meta (permisos, propietarios, ...). Podemos ver donde se han llamado a los compiladores y librerías asociadas, lo que nos permite reconstruir los pasos del atacante.

Hay una contrapartida a este camino. Si el atacante ha modificado el archivo más de una vez el inodo solo salvará la fecha de la última modificación. La herramienta *mactime* solo mostrará la última modificación.

GUI al rescate

Los comandos Sleuthkit pueden ser muy difíciles de usar y eso hace complicado prestar atención a los resultados. La búsqueda de códigos específicos en todos los sistemas de archivos y analizar los contenidos de archivos borrado puede ser tedioso y requerir mucha programación.

En estas circunstancias, GUI puede ser muy útil cuando realizamos análisis forenses. Proporciona una capa abstracta para los comandos y presenta solo los resultados, lo que significa que los investigadores forenses no necesitan consultar las páginas principales para obtener resultados.

RECURSOS

- [1] Sleuthkit: <http://www.sleuthkit.org>
- [2] Autopsy Forensic Browser: <http://autopsy.sf.net>
- [3] Atstake: <http://www.atstake.com>
- [4] The Coroner's Toolkit: <http://www.porcupine.org/forensics/tct.html>
- [5] Dan Farmer and Wietse Venema, "Computer Forensic Analysis": <http://www.porcupine.org/forensics/>
- [6] Richard E.Saferstein, "Criminalistics: An Introduction to Forensic Science", Prentice Hall
- [7] dd for Windows - Unx-Utills: <http://unxutils.sf.net/>
- [8] Enhanced Loopback: ftp://ftp.hq.nasa.gov/pub/ig/ccd/enhanced_loopback/
- [9] Forensic Challenge files: <http://project.honeynet.org/challenge/images.html>

El visor forense Autopsy (ver figura 4) proporciona un GUI de este tipo. También permite al investigador documentar y comentar los datos y resultados. Bajo algunas circunstancias puede tener sentido o ser necesario usar la línea de comandos, pero los investigadores serán capaces de realizar los pasos más importantes dentro del GUI.

La segunda parte de esta serie revisará el visor forense Autopsy Forensic Browser usando los ejemplos de la Forensic Challenge. ■

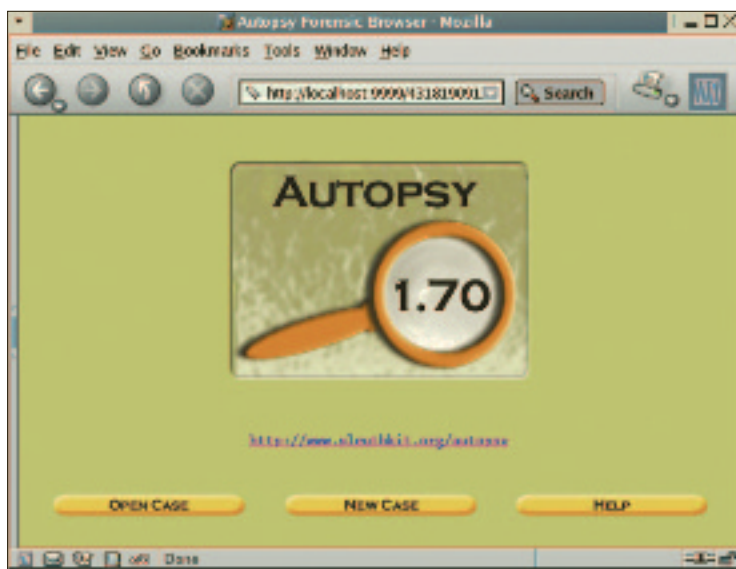


Figura 4: El visor forense Autopsy provee una apariencia Web a Sleuthkit y facilita el análisis de sistemas de archivo.

Tabla 2: Particiones del Desafío

Partición	Sistema de archivos
/dev/hda8	/
/dev/hda1	/boot
/dev/hda6	/home
/dev/hda5	/usr
/dev/hda7	/var
/dev/hda9	swap

EL AUTOR

Ralf Spenneberg es instructor y autor de Unix/Linux. El año pasado vió publicado su primer libro. "Intrusion Detection Systems for Linux Servers" (Sistemas de Detección de Intrusión para Usuarios de Linux). Ralf también ha desarrollado diverso material formativo.

