

SLEUTHKIT INSTALLATION & USE

Taken From the Sleuthkit Documentation (Brian Carrier)

Introduction

The Sleuth Kit is an open source forensic toolkit for analyzing Microsoft and UNIX file systems and disks. The Sleuth Kit enables investigators to identify and recover evidence from images acquired during incident response or from live systems. The Sleuth Kit is open source, which allows investigators to verify the actions of the tool or customize it to specific needs.

The Sleuth Kit uses code from the file system analysis tools of The Coroner's Toolkit (TCT) by Wietse Venema and Dan Farmer. The TCT code was modified for platform independence. In addition, support was added for the NTFS and FAT file systems. Previously, The Sleuth Kit was called The @stake Sleuth Kit (TASK). The Sleuth Kit is now independent of any commercial or academic organizations.

It is recommended that these command line tools can be used with the Autopsy Forensic Browser. Autopsy, (<http://www.sleuthkit.org/autopsy>), is a graphical interface to the tools of The Sleuth Kit and automates many of the procedures and provides features such as image searching and MD5 image integrity checks.

As with any investigation tool, any results found with The Sleuth Kit should be recreated with a second tool to verify the data.

Installation

If you are planning on using the Autopsy Forensic Browser with The Sleuth Kit, then it is recommended that you make a symbolic link from the specific install directory (i.e. sleuthkit-1.00) to a generic directory (i.e. sleuthkit) and use that location for the Autopsy configuration file. For example: ***ln -s /usr/local/sleuthkit-1.00 /usr/local/sleuthkit***

The Sleuth Kit uses basically the same installation scripts that were written for TCT. The process should happen automatically by typing: ***make***

The script attempts to identify if certain items on the system. If the script says that it can not find something, then it does not necessarily mean that it is an error. If you get a 'gcc: not found' error, then the compiler can not be found. If you only have 'cc' installed, use the following line: ***make CC=cc***

Or if you have gcc installed, but not you your path, use something like: ***make CC=/usr/local/bin/gcc***

If you do not have Perl, then you can use: ***make no-perl***

If you only want the mactime tool, then you can use: ***make mactime***

All tools will be compiled into the 'bin' directory. All manual pages are located in the 'man' directory. To always have access to the manual pages, add the directory to your MANPATH

SLEUTHKIT INSTALLATION & USE

Taken From the Sleuthkit Documentation (Brian Carrier)

environment variable. If you would like the binaries to be placed in a common directory, such as `/usr/local/bin`, then it must be done manually.

Toolkit Description

The Sleuth Kit allows one to analyze a disk or file system image created by 'dd', or a similar application that creates a raw image. These tools are low-level and each performs a single task. When used together, they can perform a full analysis. The tools are briefly described in a file system layered approach. Each tool name begins with a letter that is assigned to the layer.

File System Layer:

A disk contains one or more partitions (or slices). Each of these partitions contain a file system. Examples of file systems include the Berkeley Fast File System (FFS), Extended 2 File System (EXT2FS), File Allocation Table (FAT), and New Technologies File System (NTFS).

The ***fsstat*** tool displays file system details in an ASCII format. Examples of data in this display include volume name, last mounting time, and the details about each "group" in UNIX file systems.

Content Layer (data):

The content layer of a file system contains the actual file content, or data. Data is stored in large chunks, with names such as blocks, fragments, and clusters. All tools in this layer begin with the letter 'd'.

The ***dcat*** tool can be used to display the contents of a specific unit of the file system (similar to what 'dd' can do with a few arguments). The unit size is file system dependent. The ***dls*** tool displays the contents of all unallocated units of a file system, resulting in a stream of bytes of deleted content. The output can be searched for deleted file content. The ***dcalc*** program allows one to identify the unit location in the original image of a unit in the ***dls*** generated image.

A new feature of The Sleuth Kit from TCT is the '-l' argument to ***dls*** (or `unrm` in TCT). This argument lists the details for data units, similar to the ***ils*** command. The ***dstat*** tool displays the statistics of a specific data unit (including allocation status and group number).

Metadata Layer (inode):

The metadata layer describes a file or directory. This layer contains descriptive data such as dates and size as well as the addresses of the data units. This layer describes the file in terms that the computer can process efficiently. The structures that the data is stored in have names such as inode and directory entry. All tools in this layer begin with an 'i'.

The `ils` program lists some values of the metadata structures. By default, it will only list the unallocated ones. The ***istat*** displays metadata information in an ASCII format about a specific structure. New to The Sleuth Kit is that 'istat' will display the destination of symbolic

SLEUTHKIT INSTALLATION & USE

Taken From the Sleuthkit Documentation (Brian Carrier)

links. The *icat* function displays the contents of the data units allocated to the metadata structure (similar to the UNIX cat command). The *ifind* tool will identify which metadata structure has allocated a given content unit or file name.

Human Interface Layer (file):

The human interface layer allows one to interact with files in a manner that is more convenient than directly with the metadata layer. In some operating systems there are separate structures for the metadata and human interface layers while others combine them. All tools in this layer begin with the letter 'f'.

The 'fls' program lists file and directory names. This tool will display the names of deleted files as well. The *ffind* program will identify the name of the file that has allocated a given metadata structure. With some file systems, deleted files will be identified.

Time Line Generation

Time lines are useful to quickly get a picture of file activity. Using The Sleuth Kit a time line of file MAC times can be easily made. The mactime (TCT) program takes as input the 'body' file that was generated by fls and ils. To get data on allocated and unallocated file names, use 'fls -rm dir' and for unallocated inodes use 'ils -m'. Note that the behavior of these tools are different than in TCT.

Hash Databases

Hash databases are used to quickly identify if a file is known. The MD5 or SHA-1 hash of a file is taken and a database is used to identify if it has been seen before. This allows identification to occur even if a file has been renamed. The Sleuth Kit includes the 'md5' and 'sha1' tools to generate hashes of files and other data.

Also included is the *hfind* tool. The 'hfind' tool allows one to create an index of a hash database and perform quick lookups using a binary search algorithm. The 'hfind' tool can perform lookups on the NIST National Software Reference Library (NSRL) (www.nsrll.nist.gov) and files created from the 'md5' or 'md5sum' command.

File Type Categories

Different types of files typically have different internal structure. The 'file' command comes with most versions of UNIX and a copy is also distributed with The Sleuth Kit. This is used to identify the type of file or other data regardless of its name and extension. It can even be used on a given data unit to help identify what file used that unit for storage. Note that the 'file' command typically uses data in the first bytes of a file so it may not be able to identify a file type based on the middle blocks or clusters.

The *sorter* program in The Sleuth Kit will use other Sleuth Kit tools to sort the files in a file system image into categories. The categories are based on rule sets in configuration files. The 'sorter' tool will also use hash databases to flag known bad files and ignore known good files.