NAME

sorter − Sort files in an image into categories based on file type

SYNOPSIS

[-b *size* **] [-e] [-E] [-h] [-l] [-md5] [-s] [-sha1] [-U] [-v] [-V] [-a** *hash_alert* **] [-c** *config* **] [-C** *config* **] [-d** *dir* **] [-m** *mnt* **] [-n** *nsrl_db* **] [-x** *hash_exclude* **] [-i imgtype] [-o imgoffset] [-f fstype] image [image] [meta_addr]**

DESCRIPTION

sorter is a Perl script that analyzes a file system to organize the allocated and unallocated files by file type. It runs the 'file' command on each file and organizes the files according to the rules in configuration files. Extension mismatching is also done to identify 'hidden' files. One can also provide hash databases for files that are known to be good and can be ignored and files that are known to be bad and should be alerted.

By default, the program uses the configuration files in the directory where The Sleuth Kit was installed. Those can be overruled with run-time options. There is a standard configuration file for all file system types and then a specific one for a given operating system.

ARGUMENTS

The required arguments are as follows. This will analyze one or more images and either save the results in the '−d' directory or list the results to STDOUT (if '−l' is given).

-d dir Specify the location of where all files should be written. This includes the index files and subdirectories if the '−s' flag is given. This MUST be given, unless the '−l' list flag is given.

-l List information to STDOUT (no files are ever written). This is useful for Incident Response, with the use of 'netcat'. This cannot be used if '−d' is used.

image [images]
The disk or partition image to read, whose format is given with '−i'. Multiple image file names can be given if the image is split into multiple segments. If only one image file is given, and its name is the first in a sequence (e.g., as indicated by ending in '.001'), subsequent image segments will be included automatically.

The options are as follows:

-f fstype
Specify the file system type of the image(s). This is the same type that The Sleuth Kit uses.

-i imgtype
Specify the image type in which the file system is located. This is the same type that The Sleuth Kit uses.

-o imgoffset
Specify the sector offset from the beginning of the image to the start of the file system.

-b size Specify the minimum size of file to process. All files less than this size will be ignored.

-c config
Specify the location of an additional configuration file. This file will be loaded in addition to the standard ones in the install directory. These settings will have priority over the standard files.

-C config
Specify the location of the ONLY configuration file. The standard config files will not be loaded if this option is given. For example, in the 'share/sort' directory there is a file called 'images.sort'.

This file contains only rules about graphic images. If it is specified with −C, then only images will be saved about the image.

-m mnt Specify the mounting point of the image being analyzed. This is only for cosmetic reasons. When the entries in the output files are written, the files will have a the full path instead of just the relative path. If this is given, then only one image can be given.

-a hash_alert
Specify the location a hash database with entries of known 'bad' files. If any file is found with an MD5 hash value in this database, it will be placed in a special alert file. This database must have been indexed for MD5 using 'hfind' in The Sleuth Kit before it is used by sorter.

-n nsrl_db
Specify the location of the NIST National Software Reference Library (NSRL) database (www.nsrl.nist.org). Any file found in the NSRL will be ignored and not placed into a category. The database must be indexed for MD5 with 'hfind' in The Sleuth Kit before it is used by sorter. The database file is currently called 'NSRLFile.txt'.

-x hash_exclude
Specify the location a hash database with entries of known 'good' files. If any file is found with an MD5 hash value in this database, it will be ignored and not processed or saved to the category files. This database must have been indexed for MD5 using 'hfind' in The Sleuth Kit before it is used by sorter.

-e Perform extension mismatch checks on (no category index files are generated)

-U Do no save data about unknown file types. By default, an 'unknown' file is created for files where the 'file' output is not known. This allows one to refine their configuration. If this is not desired, use this flag.

-h Create category files in HTML

-md5 Calculate the MD5 value for each file and save it in the category file. This will be done automatically when any of the databases are given.

-sha1 Calculate the SHA-1 value for each file and save it in the category file.

-s Save the actual file content to sub-directories in the directory specified by '−d'. For example, all JPG and GIF files would actually be saved in the 'images' directory. If '−h' is also given, thumbnails of graphic images are also created.

-v Display verbose information

-V Display version.

[meta_addr]
The meta data address of the directory to start with. By default, the root directory is used. If this is given, then only one image can be given.

HIGH-LEVEL OVERVIEW OF PROCESS

sorter is a Perl script that interacts with other The Sleuth Kit tools. It starts by reading the configuration files from the installation directory. There is a general configuration file and a specific one for each operating system. The specific one is determined from the '−f' flag. Each configuration file contains rules for processing the output of the 'file' command. One type of line identifies which category (i.e. 'images') a given 'file' output belongs to (i.e. 'image data') (using regular expressions). Another rule shows the file extensions (i.e. .txt) that belong to a 'file' output (i.e. ASCII(.*?)text). See the Rules section below.

The program then runs the 'fls' tool in The Sleuth Kit to identify the files in the file system image. Each identified file is viewed using the 'icat' tool. If a hash database is given, the hash of the file is calculated and looked up. If it is found in an 'alert' database, then it is added to a special 'alert.txt' file. If it is found in the NSRL or 'exclude' database, then it is ignored as a known good file. Excluded files are recorded in an 'exclude' file for future reference but it is not saved in the category files.

The ’file’ command is then run to identify the file type (based on header information). The configuration file rules are used to identify which category it belongs to. An entry is added to the corresponding category file (in the ’−d dir’ directory). If the ’−s’ flag is given, then a copy of the file is saved in a subdirectory of the same name as the category. If the HTML format is used, then hyper-links will allow one to easily view saved files and view what is in each category.

Files that do not have a category are recorded in the ’unknown’ category and the ’data’ category. ’data’ is for files with a structure that ’file’ does not know and ’unknown’ is for files with a structure that ’file’ knows about. These are saved for future reference, but the unknown category can be ignored by using the ’−U’ flag.

A copy of the files can be saved by using the ’−s’ flag. If so, then the files are saved in a subdirectory that is named with the category name. Each file is named using the file system image name followed by the meta data address and the original file extension. The category index file can be used to translate the actual name to the saved name. The HTML format makes viewing easier as there are links to each file from the category index file.

The program will also consult the rules about the file extension. If the file has an extension at the end of it (anything after a '.'), it will be compared to the rules. If the extension is not found in the rules as a valid extension for the file type, it will be added to the file of ’mismatch’. If the file does not have an extension it will not be entered even if the file type has valid extensions. This check is done even if the file is found in one of the known good hash databases. If it is found in one of those, it will be added to a special file. Files of type ’data’ have no extension checks done by default (as they have an unknown structure).

The program repeats the above procedures using the output of the ’ils’ command as well. This allows ’sorter’ to examine the contents of unallocated files that still have pointers to the data units (not all file systems will produce data from this step).

CONFIGURATION FILES

Configuration files are used to define what file types belong in which categories and what extensions belong to what file types. Configuration files are distributed with the ’sorter’ tool and are located in the installation directory in the ’share/sorter’ directory.

The ’default.sort’ file is used by any file system type. It contains entries for common file types. A specific operating system file also exists, which is useful for extensions that are specific to a given OS. By default, the default file and the OS specific one will be used. Using the ’−c’ flag, an additional file can be used. If the ’−C’ flag is used, then only the supplied configuration file is used.

There are two rule types in the configuration files. Each rule starts with a header that specifies which rule type it is (category or ext). Both rule types have two additional columns that can be separated by any white space.

The category rule has the category name as the second column and a Perl regular expression in the third column. The category name can not have any spaces in it and can only be letters and numbers. The regular expression is used to examine the output of ’file’. The regular expression will be used case insensitive. More than one rule can exist for a category, but only one category can exist for a given file output. For example:

This saves all file output with ’image data’ anywhere in it to the 'images' category:

```
category    images    image data
```

This saves all file output that has 'ASCII' followed by anything and then 'text' to be saved to the 'text' category:

```
category    text    ASCII(.*?)text
```

This saves all file output that is just 'data' to the 'data' category (the ^ and $ define the boundaries in Perl). The 'data' value is common in the output of file for unknown binary data.

```
category    data    ^data?
```

There is a special category of 'ignore' that is used to skip over files of this type. This is mainly a time and space saver.

The extension rule is similar except that the second column has the value extensions for the file output. Multiple rules can exist for the same file type. The comparison will be done case insensitive. If no extension is valid for the file type, a rule does not need to be made. That is already assumed.

For example, the ASCII is used for several file extensions so the following rules could exist:

```
ext    txt,log     ASCII(.*?)text
ext    c,cpp,h,js  ASCII(.*?)text
```

Please email me any rules that you find useful for standard investigations and I will incorporate them into future releases (carrier at sleuthkit dot org).

EXAMPLES

To run sorter with no hash databases, the following can be used:

```
# sorter -f ntfs -d data/sorter images/hda1.dd
# sorter -d data/sorter images/hda1.dd

# sorter -i raw -f ntfs -o 63 -d data/sorter images/hda.dd
```

To include the NSRL, an exclude, and an alert hash database:

```
# sorter -f ntfs -d data/sorter -a /usr/hash/rootkit.db -x /usr/hash/win2k.db -n /usr/hash/nsrl/NSRLFile.txt images/hda1.dd
```

To just identify images using the supplied 'images.sort' file:

```
# sorter -f ntfs -C /usr/local/sleuthkit/share/sort/images.sort -d data/sorter -h -s images/hda1.dd
```

REQUIREMENTS

The NIST National Software Reference Library (NSRL) can be found at www.nsrl.nist.gov.

LICENSE

AUTHOR

Brian Carrier <carrier at sleuthkit dogt org>

Send documentation updates to <doc-updates at sleuthkit dot org>