CYBER502x Computer Forensics

Unit 3: Sleuthkit

Sleuth Kit/Autopsy

- http://sleuthkit.org
- http://wiki.sleuthkit.org/index.php?title=TSK_Tool_Overview
- Is a collection of file system analysis tools
- Extends the Coroner's Toolkit (TCT)
- Sleuth Kit /Autopsy supports ntfs; iso9660; hfs; ufs1,2; raw; swap; fat12 16, 32; ext2,3,4 file systems

Autopsy

- Front-end of sleuth-kit, http://www.sleuthkit.org/autopsy/
- Autopsy 3 runs on Windows while Autopsy 2 runs on Linux and OS X
- Autopsy Features:
  - File system analysis
  - Timeline analysis
  - Keyword search
  - File type sorting
  - Hash analysis

The filesystem as an abstraction

- Five “layers”
  - Physical layer
  - Data layer
  - Meta-data layer
  - File system layer
  - File name layer

Physical layer

- Magnetic hard disks (HDD)
  - Heads and Magnetic impulses
  - New data over-writes old data
- Flash-memory-based Solid State drives (SSD)
  - Transistors

Solid State drives (SSD)

- The term solid-state refers to
  - Data is stored in fixed arrangements of electronic transistors
- Rewrite requires blocks to be erased electronically before they can be used again.
  - vs ‘write-over-old-data’ property of magnetic tapes and disks
    - Garbage Collection technology built into SSD controllers
      - automatically reset the data blocks back to free space
      - Most manufacturers use TRIM, an OS-based command for cleaning up “garbage” files
    - What will affect rebuilding evidence after garbage collection has taken place?

Data layer

- data stores in this layer
- Typically in blocks (512-byte, 1024-byte, …, etc)
- Each block has an address
- Note:
  - In fat/ntfs, block is called **cluster**
  - In ufs, data is in fragments – further subdivided into **blocks**

Meta-data layer

- Contains structure and values (in inode) that define a file
  - Pointers to location(s) in data layer
  - MAC times
  - Permissions
  - Other attributes

File system layer

- Contains data that describes file system structural details
  - size of blocks
  - the address of the first inode
  - structure offsets
  - mounting info
  - …
- Usually stored in “super block” or “boot sector”

File name layer

- File name layer defines the association between a name and its inode

Sleuth Kit commands examples for each layer: 1

- File System Layer
  - **`fsstat:`** Displays the file system details
- Data Layer
  - **`blkcat:`** displays the contents of a given disk block
  - **`blkls:`** lists contents of deleted disk blocks in a raw image
  - **`blkstat:`** lists statistics associated with specific disk blocks

Sleuth Kit commands examples for each layer: 2

- Meta Data Layer
  - **`ils:`** displays inode content details
  - **`istat:`** displays information about a specific inode
  - **`icat:`** displays content of the disk blocks allocated to an given inode
  - **`ifind:`** determine the correspondent inode given a block address
- File name Layer
  - **`fls:`** list file and sub dir in a directory along with their inode content
  - **`ffind:`** determine the correspondent file give an inode

Syntax

- All commands need at least the image name
- The **`-f <FS_TYPE>`** specifies the file system type such as ext2, ext3, ntfs, fat, fat12, fat16, fat32, etc
- The **`-o imgoffset`** specifies the sector offset where the file system starts in the image

fsstat

- Displays information (for example, **block sizes**, # of inodes, **type of file system**) about the file system
- Example:
  - **`fsstat –o offset someImage.img`**
  - **`-o imgoffset`**: the sector offset in bytes where the file system starts in the image.

blkstat

- Data information (for example, allocated?) on a given data unit
- Example:
  - **`blkstat –f ext2 myImage.img 300`**
    - 300 is a block address number

blkls

- By default, it will display only the unallocated data
- It is most useful to extract unallocated data for deleted file recovery
- Example
  - **`blkls –f ext2 myImage.img > myDls`**
- Other options
  - **`e:`** list all the data
  - **`s:`** list the slack (in NTFS or FAT images)

blkcat

- Displays the content of a given data block number
- The **`-h`** flag is for a hexdump output
- To display the contents of block 200
  - **`blkcat –f ext2 myImag.img 200 | more`**

istat

- Displays inode information for a given inode number
  - Inode number
  - MAC time
  - Permission
  - Size
  - Allocation status (allocated or unallocated)
  - All allocated data block number
  - Number of links
  - All the attributes for NTFS image
- Example
  - **`istat –f ext2 myImage.img inodeNumber`**

ifind

- Map from a block number to an inode number
- When to use it
  - When you do a search and find out the data block, you need to find the meta data information
- **`ifind –f ext2 –d datablocknumber myImage.img`**

List inode info -ils

- ils
  - Can list
    - ALL inodes along with information stored in the inodes
    - Inodes of unlinked but open files
    - Inodes of deleted files
  - Often used to collect inodes for deleted files.
- http://www.sleuthkit.org/sleuthkit/man/ils.html

ils

- List inode information given a disk device or a disk image file
- By default (-r), only list the inodes of removed files
- **`-e:`** display all inodes
- **`-m:`** intermedia file mactime
- **`-o:`** open but no filename (possible data hiding)
- **`-z:`** the inodes with a zero status time change
  - The file nerve be used
- Example:
  - **`ils –rf ext2 –m /dev_hda1.img`** (+ mactime)
  - **`Wed Mar 20 2002 16:56:12 0 ..c s/srwxrwxr-x 500 500 127 <linux.dd-dead-127>`**

icat

- Cat a file content given an inode
  - **`icat –f ext2 /image 20`**
- Example:
  - To recover the deleted file
    - **`ils –rf ext2 /dev_hda1.img + mactime`**,
    - you may get the inode-number
  - **`Wed Mar 20 2002 16:56:12 0 ..c s/srwxrwxr-x 500 500 127 <linux.dd-dead-127>`**
    - Use icat to get the deleted file’s content
      - **`icat –f ext2 /dev_hda1.img 127`**

fls

- List the file names and sub-directories in a directory, including deleted files **given an inode of a directory**
- defalult it will list files in the root directory if an dir is not given
- Deleted files have a ‘*’ before them
- http://www.sleuthkit.org/sleuthkit/man/fls.html

fls usage

- fls [options] image [directory-inode-number]
- Options
  - **-a:** display the '.' and '..' also
  - **-d:** display deleted entries only
  - **-u:** display undeleted entries only
  - **-D:** display directory entries only
  - **-F:** display file entries only
  - **-r:** recursive on subdirectories
  - **-p:** display full path when process through
  - **-m output:** display in timeline format
  - **-l:** display long version information
  - **-s:** clock skew in seconds when combined with **-l** and/or **-m**

fls examples

- **# fls -r image 12**
- **# fls -f ext3 -m “/” -r images/root.dd > data/body**
- Use fls + mactime, you get …
  - **Wed Mar 20 2002 16:56:12 0 ..c s/srwxrwxr-x 500 500 127 /tmp/socket1 (deleted)**

ffind

- Finds the file name for a given inode number
- This tool processes the entire directory tree and looks for **filename that points to the given inode number**
- the deleted file name may also be found
  - if the mapping between the deleted filename and its inode is still available.

ffind usage

- **`ffind –f ext2 myImage.img inodeNumber`**
- The **`-a`** flag gets all names that point to the given inodeNumber
  - **`ffind –f ext2 –a myImage.img inodeNumber`**

If you are interested in file system (not required)

- Read the paper, “System Forensics Analysis”, by Prashant Sahu, Jayesh Plwar, …, etc.