IJoFCS (2006) **1**, 41-44

e-forensic PRESS

The International Journal of FORENSIC COMPUTER SCIENCE

www.IJoFCS.org

Computer Forensics with The Sleuth Kit and The Autopsy Forensic Browser

Ricardo Kléber Martins Galvão

Abstract - Computer invasions, with the purpose of extinguishing data, are on the rise. To retrieve erased data system audits, a computer must recover and identify the extinguished data content. For these types of inquiries, UNIX-like machines can utilize The Coroner's Toolkit (TCT). However, because this solution has limits regarding auditing partitions, UNIX appears to necessitate similar tools that investigate other types of archives systems. The set of tools presented in this article supplants this lack of recognizing and investigating partitions NTFS, FAT, UFS, EX2 and EXT3, generating reports detailed in a browser, beyond the resources already implemented for the TCT.

Key Words: Computer Forensics, Network Security.

I.INTRODUCTION

The use of TCT tools (**unrm** + **Lazarus**) for investigating free blocks in EXT2 partitions is desirable for visualizing the data browser with *hyperlinks*, indicating potential recovered file types with labels (letters → file type), as described in Figure 1.

Letra	Descrição	Letra	Descrição
A	Arquivo	Q	Mailq
C	Código C	R	Removido
E	ELF	S	LISP
F	Sniffers	T	Texto
H	HTML	U	Uuencoded
I	Imagem	W	Arquivo passwd
L	Logs	X	Exe
M	Mail	Z	Comprimido
O	Null	.	Binário
P	Programas	!	Som

Figura: 1 Labels of the report presented by TCT/Lazarus tool.

Data retrieval, conducted with the unrm tool, is a straightforward process. The device under investigation must be informed about the file to which will be generated the image of the non-allocated blocks, as in Figure 2. This example illustrates an IDE hard disk that is mounted as slave and its first partition (hdb1) is investigated.

```
unrm /dev/hdb1 >> imagem_hd.out
```

Fig. 2. Example: Non-allocated blocks retrieved with the TCT/unrm tool.

The resulting image is then submitted to the Lazarus tool (Figure 3) which interprets it to generate the HTML files to be viewed by any *browser*, as shown in Figure 5.

```
lazarus -h -D . -H . -w . \
imagem_disquete.out
```

Figura 3 Example: Using the TCT/Lazarus tool to generate visualization for the browser.

Main parameters used by Lazarus are described in Figure 4.

-h creates a HTML document (viewed by any *browser*);
-D <dir> send the blocks to a specific directory;
-H <dir> send the main HTML files to a specific directory;
-w <dir> send other HTML outputs to a specific directory.

Figura 4: Main parameters used by TCT/Lazarus

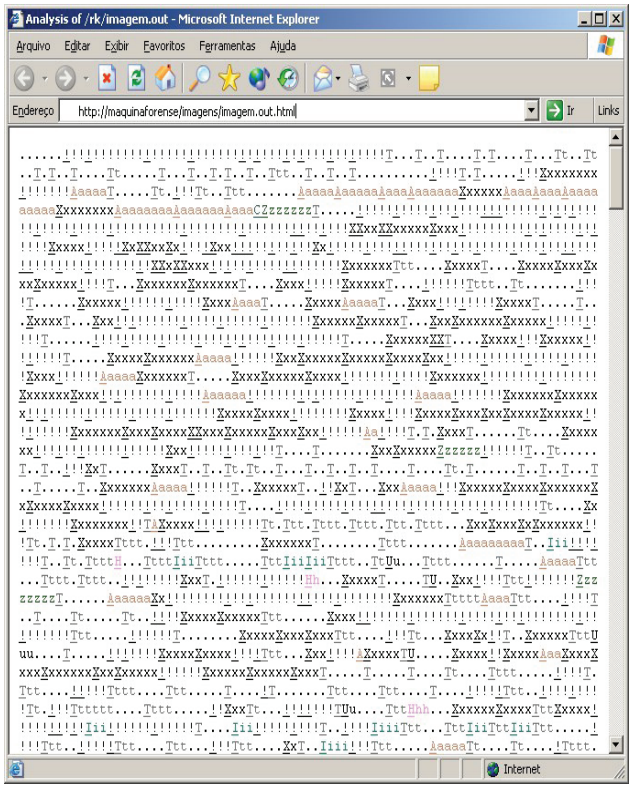

Figura 5: Example of browser visualization of a result generated by TCT/Lazarus

II. TCT ANALYSIS

By critically analyzing TCT and its **unrm** and **Lazarus** tools, we identified two negative factors, with regard to efficiency, that should be considered when adopting them as a main solution for uncovering erased data system audits. Specifically, these include:

a) The investigated partition type is limited

The TCT tools do not recognize NTFS, FAT or EXT3 partitions, making them of little use when performing forensic investigations in machines with Microsoft Windows and/or Linux operating systems with EXT3 file systems.

The new EXT3 features, when compared to EXT2, have caused system administrators to switch to this file system, thus reducing the number of computers that are amenable to investigation with TCT tools.

Despite the great increase of Linux operating system adoption, especially on servers, the number of Microsoft Windows machines is still too high among servers, and even higher when desktops are considered.

Investigating Windows (FAT) partitions with TCT is only possible with a conversion to EXT2 format, demanding alterations on the i-nodes table of the investigated partition. This activity is not always possible with data analysis.

b) The web interface is not user-friendly

Despite labels and colors used to identify the block types identified by Lazarus (Figure 5), the web interface lacks familiarity and ease-of-use due to assuming label knowledge and not providing additional information about the data.

III. Sleuth Kit + The Autopsy Forensic Browser

3.1 Sleuth Kit[1]

The Sleuth Kit *open source* tool kit for digital forensics developed by Brian Carrier to be used in UNIX systems (Linux, OS X, FreeBSD, OpenBSD and Solaris) is capable of analyzing NTFS, FAT, UFS, EXT2 and EXT3 file systems.

In its first version, the Sleuth Kit was called The @stake Sleuth Kit (TASK). TASK consists of a collection of commands based on the TCT command line.

With this kit, the user can examine the computer file systems through a non-intrusive approach that is not dependent on the investigated machine operating system to process the file system, deleted and hidden from files DOS, BSD, Mac, Sun and Linux partitions.

The results generated by Sleuth Kit tools are used by another tool – The Autopsy Forensic Browser[2] – which presents such details as image integrity, keyword searches and other automatized operations about the investigated partition through a graphical interface.

1 http://www.sleuthkit.org
2 http://www.sleuthkit.org/autopsy

3.1.1 Data Input Features

1. The Sleuth Kit analyzes the file system images generated by the **dd** command – found on every UNIX and available for Microsoft Windows – in a non-proprietary format;
2. The data format of the investigated partition (NTFS, FAT, UFS, EXT2 or EXT3) does not depend on the operating system of the machine on which the Sleuth Kit is run;
3. The Sleuth Kit can be run from a UNIX system during an incident response, showing files that might be hidden by running rootkits without modifying the files (not even A-Time) under investigation.

3.1.2 Search Techniques

- Deleting and allocating file names listings;
- Showing details and content of all NTFS attributed (including Alternate Data Streams);
- Showing file system details and meta-data structure;
- Creating file activity timelines that can also be imported by spreadsheets for creating graphics and reports;
- Visualizing hash files in a hash database, customizing databases that can be created with the md5sum tool;
- Filetype-based organizing (grouping, for example, executables, images and documents). Can also generate thumbnails of the images that were found for a quicker analysis.

3.1.3 General Features

The Sleuth Kit was written in C and Perl and uses an aspect of the TCT code. It has been tested under Linux, Mac OS X, Open & FreeBSD, Solaris and CIGWIN platforms. The information written herein refers to version 1.70 of July/02/2004. There are add-ons on the project's website that can be applied as patches to improve certain Sleuth Kit features. fFor example, showing Unicode names in NTFS partitions and image indexing by name.

3.1.4 Details on Sleuth Kit tools

File system tools: These conduct the general processing of data from file systems, including layout and allocation structures.

- **fstat**[3]**:** Shows details of the file system and statistics, including layout, size and labels.

File Names tools: Conduct the processing of the file names structure, typically found in parent directories.

- **ffind**[4]**:** Searches for file names allocated and non-allocated from a point in the metadata structure;
- **fls**[5]**:** Lists named file and directories from an analyzed image.

Metadata tools: Conduct the processing of the metadata structure, storing details about the files.

- **icat**[6]**:** Extracts data units from a file specified by a metadata address (i-node number);
- **ifind**[7]**:** Searches for a metadata structure;
- **ils**[8]**:** Lists a metadata structure and its content in a pipe-delimited format;
- **istat**[9]**:** Shows the statistics and details about a metadata structure (i-nodes) in a human-reading format.

Data Unit tools: Conducts the processing of the data units in which the file contents are saved (FAT and NTFS clusters, and UFS, EXT2 and EXT3 blocks/ fragments).

- **dcat**[10]**:** Extracts the content of a file unit;
- **dls**[11]**:** Lists details about data units and can extract the non-allocated space from a file system;
- **dstat**[12]**:** Shows statistics about a file unit in human-reading format;
- **dcalc**[13]**:** Calculates where the data of an image of non-allocated spaces (generated from a DLL) are in the original image. This tool is used when evidence is found in a non-allocated space.

Media management tools: These tools receive a disk image as input and analyze the management structure on which they are organized.

- **mmls**[14]**:** Shows a disk layout, including non-allocated spaces. The output identifies a partition type and its size, in order to ease the use of **dd** to extract partitions. The output is classified based on a boot sector in order to ease identification on layout.

Other tools:

- **hfind**[15]**:** Uses a binary classification algorithm to find hashes;
- **mactime**[16]**:** Uses fsl and isl tools output as input to create a timeline of a file activity;

3 http://www.sleuthkit.org/sleuthkit/man/fsstat.html

4 http://www.sleuthkit.org/sleuthkit/man/ffind.html
5 http://www.sleuthkit.org/sleuthkit/man/fls.html
6 http://www.sleuthkit.org/sleuthkit/man/icat.html
7 http://www.sleuthkit.org/sleuthkit/man/ifind.html
8 http://www.sleuthkit.org/sleuthkit/man/ils.html
9 http://www.sleuthkit.org/sleuthkit/man/istat.html
10 http://www.sleuthkit.org/sleuthkit/man/dcat.html
11 http://www.sleuthkit.org/sleuthkit/man/dls.html
12 http://www.sleuthkit.org/sleuthkit/man/dstat.html
13 http://www.sleuthkit.org/sleuthkit/man/dcalc.html
14 http://www.sleuthkit.org/sleuthkit/man/mmls.html
15 http://www.sleuthkit.org/sleuthkit/man/hfind.html
16 http://www.sleuthkit.org/sleuthkit/man/mactime.html

- **sorter**[17]**:** Classifies files based on their type and executes extension checks and hash database checks.

3.2 The Autopsy Forensic Browser

This *Open Source* tool written in Perl provides a HTML-based graphical interface for Sleuth Kit that is similar to a file manager, showing details about deleted data and file system structures, with results that can be accessed using a HTML browser.

Distinct from Lazarus, Autopsy does not require any tool to be executed previously. It can work directly over mounted partitions or over image files generated by the **dd** command.

Autopsy can be considered an interface for Sleuth Kit, as everything done through its interface generates Sleuth Kit commands which are interpreted and shown again by Autopsy.

Running Autopsy is simple; after its installed, to the user runs the **autopsy** binary that will indicate the address/port to be accessed by a browser. These data can be customized in an Autopsy config file.

3.2.1 Additional Features

When being run, Autopsy asks for the creation of a new **Case** or to open a previous **Case**. Each created **Case** is saved as a directory to make it easier to search for audits that it created.

One or more **Hosts** must be specified inside each **Case**, and these hosts will be sub-directories of **Cases** which specify, for example, more than one machine being audited in the same process.

After this, all menus present Sleuth Kit functions to be invoked at each request by the web interface.

17 http://www.sleuthkit.org/sleuthkit/man/sorter.html

IV. Running Sleuth Kit + The Autopsy Forensic Browser from a CD

Most auditing procedures of suspect and/or compromised machines must be conducted without removing the hard drive. As you cannot trust the analyzed machine operating system, it is recommended that the user implements a live-CD prepared for this task, which contains basic UNIX tools that are as long as Sleuth Kit, Autopsy and other auditing tools.

Some distributions have CD ISO images with this goal, making it easier for the user to conduct forensic jobs. Two of these distributions are presented below:

- Professional Hackers Linux Assault Kit (http://www.phlak.org)
  Morphix-based, created by Alex de Landgraaf.
- Knoppix security tools distribution (http://www.knoppix-std.org)
  Knoppix-based, with light window managers; ideal for forensics use in older machines.

References

[1] CARRIER, Brian. File Activity Timelines. Available online on September/2006 at URL http://www.sleuthkit.org/sleuthkit/docs/ref_timeline.html.

[2] CARRIER, Brian. The FAT File System – Sleuth Kit Implementation Notes (SKINs). Available online on September/2006 at URL http://www.sleuthkit.org/sleuthkit/docs/skins_fat.html.

[3] CARRIER, Brian. The NTFS File System – Sleuth Kit Implementation Notes (SKINs). Available online on September/2006 at URL http://www.sleuthkit.org/sleuthkit/docs/skins_ntfs.html.

[4] CARRIER, Brian. The Sleuth Kit Informer – Issue #13 - UNIX Incident Verification with Autopsy. Available online on September/2006 at URL http://www.sleuthkit.org/informer/sleuthkit-informer-13.txt.

[5] LUCAS, Charles. Running Sleuthkit and Autopsy Under Windows. Available online on September/2006 at URL http://www.memophage.net/Running_Sleuthkit_and_Autopsy_Under_Windows.pdf.