

Freeware Forensics Tools for Unix

Derek Cheng 2001-11-01

Freeware Forensics Tools for Unix

by *Derek Cheng* CISSP, GCIH, Prociinct Security

last updated November 1, 2001

You are a security specialist brought in to investigate the suspected security compromise of a Unix machine. You are expected to gather as much information as possible without altering or contaminating the evidence. The data you collect must be good enough to determine whether a compromise has actually occurred on the system.

During the analysis of data, you will need to create a detailed time-based reconstruction of the attack and compromise. You must also answer questions such as: when and where did the compromise occur, how did the compromise occur, how many systems were affected, and what files were affected. This information is critical in determining who attacked your system, how they gained access, and whether prosecution is justified.

You may need to rely on forensic tools to perform these tasks. Unfortunately, your company cannot afford to purchase expensive commercial forensics tools. Fortunately for you, there are sophisticated forensics tools that can help you accomplish these tasks for free. This article will discuss three popular freeware forensics tools for the Unix platform: The Coroner's Toolkit (TCT), TCTUtils, and Autopsy Forensic Browser. These tools, when used together, offer a comprehensive solution for forensic data gathering.

(While an indepth discussion of forensics is beyond the scope of this article, readers may want to check out Timothy E. Wright's [Field Guide for Investigating Computer Crime](#).)

In this article, you will learn basic steps to take when gathering data using TCT, TCTUtils, and Autopsy. The following is an overview of these freeware forensic tools, which will then be followed by step-by-step instructions. It is essential to run all of these tools from a source that is known to be good, preferably either from a CD or an analysis test system. Be sure to review the documentation and instructions before using these tools on your system. Readers should note that these instructions are for Linux and may not work with other flavors of Unix.

Overview of Popular Freeware Forensics Tools for Unix

[The Coroner's Toolkit \(TCT\)](#)

TCT is the leader on the Unix operating system. Designed by Dan Farmer and Wietse Venema, TCT is a collection of four distinct groups of tools that, when used together, provide powerful techniques for collection and analysis of forensic data with the goal of reconstructing past events. TCT has the ability to analyze activities on a live system and capture current state information that would be difficult to capture manually. It is important to mention that TCT was not designed to collect evidence that would be admissible in court; it was designed to help determine what happened on a compromised machine. TCT includes the following programs:

- **grave-robber**: Captures various types of data quickly via order of volatility and creates MD5 hashes of the evidence to preserve its integrity. The optimum way to run grave-robber is to collect the volatile data on a live system, shut down the system, image the drive, and then use grave-robber's -f option against a copy of the file systems.
- **pcat, ils, icat, file**: Records and analyzes processes and inode data. Pcat copies process memory from a live system. Ils lists inode information. Icat copies files by inode number. File classifies files into various types.
- **unrm and lazarus**: Recovers and analyzes the unallocated disk blocks on a file system. Unrm collects information

in unallocated portions of the file system. Lazarus analyzes raw data from unrm and attempts to classify what type of data it contains.

- **mactime**: Helps create a chronological timeline of when files have been Modified, Accessed, or Changed (MAC) for each inode, along with their associated filenames.

TCTUtils

Written by Brian Carrier, TCTUtils is a collection of utilities that adds functionality to TCT. The following programs are included with TCTUtils.

- **bcats**: Displays the contents of a disk block to stdout.
- **blockcalc**: Maps between dd images and unrm results.
- **fls**: Displays file and directory entries that have been deleted in a directory module. Using fls with the -d option can list the names of all of the deleted files on the image.
- **find_file**: Determines which file has allocated an inode in an image.
- **find_inode**: Determines which inode has allocated a block in an image.
- **istat**: Displays information about an inode.
- **mac_merge**: Merges the output from 'fls-m' with the output from TCT mactime to create one large timeline.

Autopsy Forensic Browser

The Autopsy Forensic Browser is an easy-to-use browser-based GUI for TCT and TCTUtils. It allows an investigator to browse and analyze forensic images at the file, block, and inode level. It also provides a convenient interface for searching for key words in an image.

Step-by-Step Instructions

For these step-by-step instructions, we will be using the Linux file system and a disk image file (dev_hda1.img) created by the Unix dd command.

Throughout these instructions, we will use the following file names and directories:

dev_hda1.img	Name of the image file (/root partition)
/image	Directory where the image is copied to
/mnt/forensics/root	Directory for where the image file is mounted
/usr/local/tct-1.07	Directory where TCT is stored
/usr/local/tctutils-1.01	Directory where TCTUtils is stored
/usr/local/autopsy-1.01	Directory where Autopsy Forensic Browser is stored
/tmp/deleted	Directory to store recovered deleted files

Before running any of these tools, we need to perform the following preparation steps:

Create the following directories:

```
# mkdir /mnt/forensics
# mkdir /mnt/forensics/hack
# mkdir /image
```

Copy the image file, usually from a CD-ROM, to the /image directory:

```
# mount /mnt/cdrom
```

```
# cp /mnt/cdrom/dev_hda1.img /image
```

Mount the image file as a loop device to the /mnt/forensics/root directory, making sure that it is mounted read-only.

```
# mount -o ro,loop,nodev,noexec,nosuid,noatime /image/dev_hda1.img /
mnt/forensics/root
```

Running TCT version 1.07

TCT is a collection of four distinct groups of tools that, when used together, provide powerful techniques for collection and analysis of forensic data

Working on the live system:

1. Use grave-robber to gather information such as process, host, and network information.

```
# cd /usr/local/tct-1.07/bin
# ./grave-robber -lPstv

-l:    Collects inode information before gathering data using lstat()
-P:    Collects running information using ps and icat
-s:    Gathers host and network information using netstat and df
-t:    Gathers trust information like hosts.equiv, .rhosts, and xhost
-v:    Performs in verbose mode
```

The results are stored in /usr/local/tct-1.07/data/host.domain, where host.domain is the fully qualified domain name of your system.

2. Use pcat to copy the process memory. Note: pcat will not copy its own process memory.

Search for an interesting process using ps, netstat, or lsof

```
# cd /usr/local/tct-1.07/bin
# ./pcat [process_id] | strings | less
```

3. Use ils to find possible data hiding (open but unlinked files).

```
# cd /usr/local/tct-1.07/bin
# mount (to find the root (/) partition)
# ./ils -of extf2 /dev/[drive with root partition]
```

To find unused files:

```
# cd /usr/local/tct-1.07/bin
# ./ils -zf ext2ds /dev/[drive with root partition]
```

Working with the Disk Image:

1. Use grave-robber to collect information from the disk image.

```
# cd /usr/local/tct-1.07/bin
# ./grave-robber -c /mnt/forensics/root -o LINUX2 -MivVt
```

```
-M: Performs an MD5sum of all files and collects inode MACTimes
-i: Collects inode information from unallocated portions of the image
-v: Performs in verbose mode
-V: Gathers the major and minor numbers from /dev
-t: Gathers trust information
```

The results are stored in /usr/local/tct-1.07/data/host.domain, where host.domain is the fully qualified domain name of your system.

2. Use ils to list inode information and collect inodes of deleted files.

```
# cd /usr/local/tct-1.07/bin
# ./ils -rf ext2fs /image/dev_hda1.img
```

```
-r: Lists inode numbers of deleted files
-f ext2fs: Declares you are working with the Linux file system
```

3. Use icat to copy files by inode number (particularly the inode number of a deleted file.)

```
# cd /usr/local/tct-1.07/bin
# ./icat -hf extfs /image/dev_hda1.img [an inode number from the ils -rf command in Step 4]
```

```
-h: Skips over any holes in the file
-f ext2fs: Declares that you are working with the Linux file system
```

To recover all of the deleted files on the image into /tmp/deleted

```
# mkdir /tmp/deleted
# cd /usr/local/tct-1.07/bin
# ./ils -rf ext2fs /image/dev_hda1.img | awk -F '|' '($2=="f") {print $1}' |
while read i; do /usr/local/tct-1.07/bin/icat /image/dev_hda1.img $i > /tmp/deleted/$i; done
```

(Credit for this code goes to [Thomas Roessler](#))

4. Use file to classify and determine the file types of recovered files.

```
# cd /usr/local/tct-1.07/bin
# ./file /tmp/deleted/*
```

5. Use unrm to collect all of the unallocated disk space of a partition. Note: Never collect the output on the same file system that you are analyzing, otherwise you will write over your own data.

```
# cd /usr/local/tct-1.07/bin
# ./unrm /image/dev_hda1.img > hda1_unrm.results
```

- Use lazarus to analyze the raw data collected from unrm and attempt to classify what type of data it contains.
Note: Depending on the size of the file, lazarus can take many hours to complete.

```
# cd /usr/local/tct-1.07/bin
# ./lazarus -h /image/hda_unrm.results
```

-h: Produces an HTML report

The output is stored in two directories, a www directory and a blocks directory. The HTML file created is called hda_unrm.results.frame.html.

- Using mactime, we can determine which files have been Modified, Accessed, or Created (MAC) during a specified time window.

To create a file system timeline activity report for all files starting from the MACtime of 4/01/2001:

```
# cd /usr/local/tct-1.07/bin
# ./grave-robber -m /mnt/forensics/root
# ./mactime 4/01/2001 | less
```

-m: Collects MACtimes

Running TCTUtils version 1.01

Together with TCT, TCTUtils adds enhanced functionality such as image analysis at the file, block, and inode level.

After installing TCTUtils, do not forget to edit the Makefile and change the TCT_DIR to point to the directory where TCT is stored.

Working with the disk image:

- To create a better timeline that includes deleted files showing when they were modified, accessed, and deleted:

```
# cd /usr/local/tct-1.07/bin
# ./grave-robber -m /mnt/forensics/root
# cd /usr/local/tctutils-1.01/bin
# ./fls -m -/mnt/forensics/root/" /image/dev_hda1.img 2 >>
/usr/local/tct-1.07/data/[name of directory (localhost.localdomain)]/body
# cd /usr/local/tct-1.07/bin
# ./mactime 4/01/2001 > /tmp/full_mactime.report
```

Autopsy Forensic Browser

Autopsy merges the information from TCT and TCTUtils into a simple point and click GUI.

After installing Autopsy, run the 'configure' program and make sure that you have the correct directories for TCT, TCTUtils, and the Morgue (or image) directory. Next you must edit the 'fsmorgue' file under the /image directory to let Autopsy know how to mount the imaged partitions.

Working with the disk image:

2. To start the Autopsy Forensic Browser:

```
# cd /usr/local/autopsy-1.01/bin  
# ./autopsy [any port number] localhost &
```

Autopsy will give you a URL to enter into your browser.

Conclusion

TCT, TCTUtils, and Autopsy Forensic Browser are extremely valuable to investigators because sophisticated and comprehensive investigations can be conducted without having to purchase expensive commercial tools. These freeware tools do an excellent job of gathering data. However, the daunting task of data analysis is left to the investigator. Data must still be analyzed manually and formal investigation reports and data sets must be created regardless of whether you wish to prosecute the attacker.

[Derek Cheng](#), CISSP, GCIH, is a Senior Security Engineer at [Procinct Security](#) where he consults with clients on emerging security issues. He has extensive knowledge and expertise on numerous security products, technologies, and architectures. Derek's main areas of expertise include security risk assessments, vulnerability testing, and security product analysis and research.

Relevant Links

[Basic Steps in Forensic Analysis of Unix Systems](#)

Dave Dittrich

[TCT Homepage \(1\)](#)

[TCT Homepage \(2\)](#)

[The SecurityFocus Forensics Mailing List](#)

[DDJ Computer Forensics Column Index](#)

[Computer Forensics Analysis Class Handouts](#)

[What To Do If You've Just Been Broken Into, FAQ for using TCT](#)

[Interpreting Network Traffic: An Intrusion Detector's Look at Suspicious Events](#)

[Interpreting Network Traffic: An Intrusion Detector's Look at Suspicious Events \(PDF\)](#)

[Privacy Statement](#)

Copyright 2006, SecurityFocus