GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials: Network, Endpoint, and Cloud (Security 401)"
at http://www.giac.org/registration/gsec

The Coroners Toolkit: A Handy Suite of Utilities

Mike Wagner

13 December 2000

The Coroner's Toolkit (TCT) is a great suite of utilities for forensic investigators and security types alike. Whether looking for a spreadsheet that was accidentally deleted, or investigating an intrusion into a Department of the Defense computer system, TCT is of great assistance in recovery, collection, and analysis. Two well-known computer security professionals, Dan Farmer and Weitze Venema are the men who authored these utilities. They note that although the TCT was designed for Unix systems, the principles apply to many operating systems. TCT is freely available at: http://www.fish.com/tct.

What utilities are included in the TCT?

The TCT toolkit includes various utilites for examining and collecting data from a computer. The major parts of the kit, which this paper focuses on, are: grave-robber, the main data-gathering program; lazarus, the data reconstruction program; and mactime, the M, A, C time file system reporter. Other programs included are:

- reconfig - tries to find all the appropriate files on the current system
- file - attempts to determine the content type of a file
- icat - copies (cat(1)) a file by inode number.
- ils - list file system inode information.
- lastcomm - a portable lastcomm command
- major_minor - used internally by the TCT, it emits perl code
- md5 - the RSA MD5 digital signature tool.
- pcat - copies the address space of a running process.
- strip_tct_home - used internally by the TCT, it strips out a variable from various files

Grave-robber: The main data-gathering program.

Grave-robber is a file walking tool that runs various Perl modules, in the lib directory, collecting the basic system information and saving various files necessary to interpret that data. This data includes configuration files, log files, process information, network states, and other critical data necessary for a well-versed forensic exam. Although the tool does not require root privileges to run, it should be run by root in order to retrieve root and other processes for further analysis. Furthermore, the session is logged with all the shell commands, what times they were run, and an md5 checksum is generated.

Grave-robber can be run against any directory of interest, including the / directory. However, this may take some time. Optionally, grave-robber could just be run with the –f flag, enabling a fast scan. It is important to point out that the tool collects information in regard to order of volatility which allows the memory to be collected before disk information and so on.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

All captured data is saved in a subdirectory of the data directory. The name of the directory created is dependent upon the system name and time the tool was run. A symbolic link, data/hostname, is also created to the actual data directory. Included in the newly created data/hostname subdirectory are the following:

- command_out – keeps the output of the programs within grave-robber that are run.
- string_out – keeps the output of the strings command on every directory the file walker found.
- body – the mactime database
- body.S – keeps file attributes of all SUID files.
- coroner.log – keeps the date and time all the programs were run.
- error.log – keeps all the errors created by grave-robber.
- deleted_files – keeps all deleted files that were still open or running when the tools ran.
- pcat – keeps the images of the processes the pcat recovered. Pcat copies the address space of a running process.
- conf_vault – keeps an archive of all the files of interest

Grave-robber uses the configuration file grave-robber.cf. Most of the configuration file parameters do not need to be changed and are well explained in the file itself.

unrm & lazarus: the data reconstruction program

Lazarus is a handy utility that reconstructs deleted files or data from raw data, such as that found on unallocated portions of the file system or other data such as system memory and swap. Its purpose is to identify and recover lost or hidden data. The program consists of two portions, one that grabs the input and the other that reads, analyzes, and reports its findings. Like grave-robber, the program should be run as root.

The program unrm, which uncovers unallocated blocks from a raw Unix file system, is packaged with TCT and can be used in conjunction with lazarus. In this fashion, the output from unrm, which would be data from disk space previously used and released, could be the input for lazarus. It should be noted that lazarus is time consuming and requires patience. However, it is useful for finding "lost" data, especially deleted system logs and intruder source code.

If decided that unrm and lazarus are to be used in the forensic analysis you first need to determine the file system the lost file was on.

```
# df /directory/of/lost/file
```

This will return something like:

```
Filesystem          1k-blocks    Used    Available Use% Mounted on
/dev/hda1            1008872    246788    710836  26%   /
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Next, you should save the unallocated blocks of /dev/hda1 using unrm. Note: do not save the recovered data to the same file system it was lost on. Also ensure you have about 2.25 times the amount you want to recover on your working copy disk. This is necessary for both unrm and lazarus. Assuming that you will save and analyze on hda2 you would enter the command from the tct/bin directory:

./unrm /dev/hda1 > /TCT/path/unrm_ouput *(on hda2)*

Now you are ready to run Lazarus on the output of unrm.

./lazarus –h unrm_output *(h flag generates output in html)*

This creates two directories, one of which is called "www" and the other "blocks" which contains the many files recovered.

Now it is time for the analysis, a tedious duty. If you are looking for lost emails, a program included called rip-mail is included. This program attempts to recover mail for bits and pieces of mangled data. If searching for text then the *grep* utility would work. Images could be recoved with *xv*, a Unix image viewer. *sort* and *uniq* are also important utilities helpful to the forensic examiner.

Another noteworthy detail is that the documentation provided with TCT is very good and supporting documentation on the web is getting better every day. Many of the sites have command examples from which to choose.

MACtime: the M, A, C time file system reporter

MACtime is a program that attempts to see what files have been created, accessed, or modified during a given timeframe. It can be run by itself but it is more often useful with the grave-robber utility, more specifically the database generated by grave-robber's output. Atimes are changed when a file is accessed, Mtimes change by modifying a file's contents, and Ctimes keep track of when the meta-information about the file has changed, such as the owner, group, file permission, etc.

MACtime incorporates the *lstat* command to gather the information of a particular file. *lstat* like the *stat* command prints out the contents of an inode, except that it avoids getting all the information about links and sticks to the particular file.

Also, it is important to note that every command typed changes a time somewhere. So before mucking around on the system, either immediately do a *dd* backup or run grave-robber. Grave-robber should not disturb the system in any way other than the process walking through the file system with *lstat*. Then run the mactime program. Grave-robber can be run to just collect the mactime information with the following command:

./graverobber –m /directory

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

then run mactime including the time parameter:

./mactime 12/13/2000

MACtime is a powerful program and when combined with other types of information gathering techniques such as program profiling, process accounting, logs, and other auditing methods, it can produce results and solve the mystery.

Conclusion

There are many ways to fully utilize the TCT. The programs above are just three of the many packaged in the suite. The other C and Perl utilities included make the TCT the best forensic examiner's toolkit that money can buy (its free). In addition to the TCT tools, it would be smart to have *lsof*, *ls*, *dd*, *ps*, *netstat*, and other vital programs on a floppy for the particular operating system you are examining in case of trojanized software. Try it out.

References:

Basic Steps in Forensic Analysis of Unix Systems. [Online] Available: http://staff.Washington.edu/dittrich/misc/forensics

Farmer, Dan. What are MACtimes? Dr. Dobbs Journal. October 2000. [Online] Available: http://www.ddj.com/articles/2000/0010/0010f/0010f.htm

Farmer, Dan. FAQ Webpage. [Online] Available: http://www.fish.com/tct/FAQ.html

Farmer, Dan. Help! How do I recover that important file? [Online] Available: http://www.fish.com/tct/help-recovering-file

Farmer and Venema. Computer Forensic Class Handouts. [Online] Available: http://www.porcupine.org/forensics/handouts.html

Venema, Wietse. File Recovery Techniques. Dr Dobbs Journal. December 2000. [Online] Available: http://www.ddj.com/articles/2000/0012/0012h/0012h.htm

TCT documentation and man pages

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.