

By SHELDON TEERLINK *and* ROBERT F. ERBACHER

Improving the Computer Forensic Analysis Process through VISUALIZATION

The ability to display mountains of data in a graphical manner significantly enhances the time needed to locate and analyze suspicious files.

Computer forensics is the preservation, analysis, and interpretation of computer data [1]. There is a need for software that aids investigators in locating data on hard drives left by persons committing illegal activities. These software tools must reduce the tedious efforts of forensic examiners, especially when searching very large hard drives.

A great deal of time is wasted by analysts trying to interpret massive amounts of data that isn't correlated or meaningful without high levels of patience and tolerance for error. Visualization techniques can greatly aid forensic specialists to direct their search to suspicious files; in effect aiding the interpretation process. Essentially we are relying on the old adage: "A picture is worth a thousand words;" a saying derived from the fact that humans have the ability to visually interpret and comprehend pictures, video, and charts much faster than reading a textual description of the same. Indeed, humans perceive graphical images perceptually while text is perceived serially.

A GREAT DEAL OF TIME IS WASTED BY ANALYSTS TRYING TO INTERPRET MASSIVE AMOUNTS OF DATA THAT ISN'T CORRELATED OR MEANINGFUL WITHOUT HIGH LEVELS OF PATIENCE AND TOLERANCE FOR ERROR. VISUALIZATION TECHNIQUES CAN GREATLY AID FORENSIC SPECIALISTS TO DIRECT THEIR SEARCH TO SUSPICIOUS FILES

Using this concept of visual perception, we have developed a GUI and associated visualizations that display file information in a graphical manner. These visualization techniques reduce the time examiners need to analyze data and greatly increase the probability of locating criminal evidence. The user is able to query a specific directory to see statistics such as file size, access date, creation date, modification date, owner, and file type. Requests for more information about a suspect file can be filled by clicking on the display and walking various menus.

Viewing information about multiple files or understanding the relationship between them is also helpful. The incorporated UI allows file searching, pattern matching, and display of file contents. Each of these options allows a deeper analysis of the data stored on the hard drive and results in a flexible and customizable tool for locating criminal evidence. This set of graphical representations and interactive capabilities greatly aids the computer forensic process by reducing the time required to identify suspicious files and increasing the probability of locating criminal evidence.

This article discusses the applied visualization techniques as well as the results of user studies to measure the effectiveness of the techniques in contrast with

traditional techniques. The visualization environment incorporates two techniques based upon two fundamentally different paradigms. The first is based on providing a non-hierarchical visualization—directory information that is thrown out—while the second provides a hierarchical representation. It is these advanced visualization techniques that differentiate such capabilities from traditional forensic tools such as EnCase (www.encase.com/) and Helix (www.e-fense.com/helix/). Such traditional tools provide GUI front ends, but lack advanced visualization techniques to aid data correlation and analysis.

NONHIERARCHICAL VISUALIZATION TECHNIQUES

Nonhierarchical views of file statistics show all files in a directory and its subdirectories without any consideration given to the relationship between said files and directories. A very simple way to draw a nonhierarchical view is to have a single square block represent a file, where the gray tone or color of the block is a measurement of the file, such as size. In essence this is similar to the typical defragment view of a hard drive. Light and darker colored blocks represent large and smaller files respectively. If the size of a file contrasts greatly with other files under examination, it is easily spotted because it stands out against a sea of darker colored blocks. This creates a form of automatic visual clustering and anomaly detection. When a time attribute is used for filtering, lighter colored blocks represent files with more

recent activity. Ultimately, filtering may be applied according to any available file attributes. Different scenarios will require examination of different file attributes for location of target files.

These square block diagrams are simple but present the investigator with a wealth of information. For instance, the visualization will change dynamically based on the selected file attribute, such as access time, modification time, creation time, and size. The user can view in parallel the relative time or size for thousands of files in one glance. Compare this with reading a long list of file names and other attributes where it is easy to forget information and difficult to convert the text into a relational model, as seen with typical file system browsers.

HIERARCHICAL VISUALIZATION TECHNIQUES

Hierarchical views of file statistics show the relationship of files as they exist in the directory structure. Graphs and space-filling forms such as tree maps [2] maintain the relationship of files to one another. They are designed for human visualization by displaying the entire tree structure in one screen. With tree maps in particular, each file is represented by a shaded box that adheres to a chosen coloring scheme that highlights file and directory boundaries. Box size is determined by two parameters—the size of the user-selected display region and percentage of the selected directory the file occupies. Subdirectories are likewise displayed, subdividing each region until individual files alone are represented.

We incorporate extensions to tree maps in the form of filtered tree maps. The concept behind these maps is that instead of displaying available size data in the hierarchy as is traditionally done, we are able to construct tree maps based on other file attributes such as time; thus we are filtering based on the file attributes.

The filtered tree maps we propose offer more flexibility to forensic examiners than a tree map that simply represents the size of files. Additionally, they provide better use of screen real estate than the use of graphs. Traditional tree maps are typically only used to visualize the size of files with, at most, an option to view only those files, for example, that have access times in a certain range. The visualization is slightly altered by what files meet the filtering criteria, but

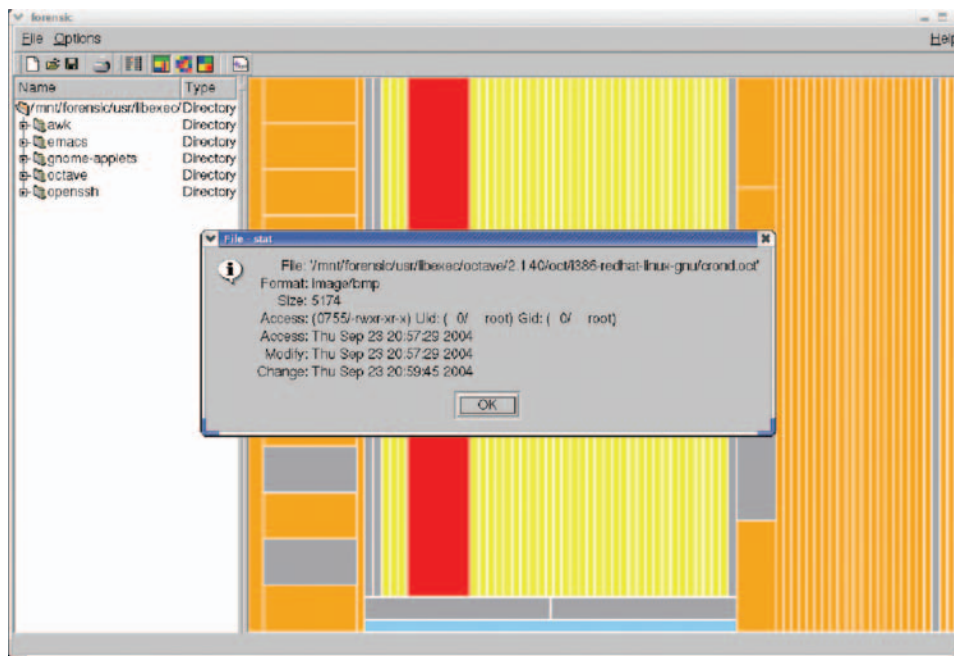


Figure 1. The filtered tree map diagram shows a large red rectangle surrounded by smaller yellow rectangles. The red rectangle represents an image file accessed more recently than the surrounding yellow files. The message box has popped up after selection of the red rectangle and shows the file's detailed information. Notice the discrepancy between the file name and format, indicative of a simple attempt to hide informational content.

each square in the tree map still represents the size of the file. Our version of filtered tree maps draws the tree map based on a selectable attribute of the file, rather than only on the size of the file. Time filters create tree maps where larger squares represent files with recent activity. Since recently accessed files are more likely to yield evidence of a crime, it makes sense to highlight these files by rendering them larger and more visibly. Drawing the filtered tree map based on criteria other than size makes it much easier to understand the relationship between files, especially based on more relevant attributes such as time or type.

In addition, each tree map square representing a file is colored based on a custom coloring scheme. The coloring scheme is designed so files containing different magic keys, indicative of the file's type, are colored according to the user's choice. This improves identification of stor-

age anomalies. For example, a red colored file representing an image would stand out against a multitude of yellow colored system binaries. One goal with such an analysis would be to identify out-of-place files; for example, files marked as system binaries inappropriately or with additional information content appended to them. The complete forensic analysis process can be very complex when a savvy criminal attempts to hide their information.

It is the tree map view that aids resolution of scalability issues. Given a 300GB+ hard drive it is impossible to display all files simultaneously. The tree maps essentially organize the information and allow subregions to be examined in greater detail through either focused tree map views or through the nonhierarchical view.

EVALUATION RESULTS

In order to provide an effective evaluation of the visualization environment for computer forensic analysis, two different data sets are used for searching; corresponding to the two different analysis paradigms. The first analysis paradigm is the traditional Unix shell-based command search. The second is a search based on our advanced visualization techniques. Such a Unix command shell provides similar capabilities as with EnCase's analysis tools, but without the GUI front end. EnCase also provides many additional search capabilities tailored for the forensic process. The chosen scenarios were designed to avoid the need for such capabilities.

We evaluated the effectiveness of the developed techniques through a controlled, human-computer interaction experiment, where a human subject was tasked with looking for three altered or hidden files on a hard drive using the two specified paradigms. When the subjects began each

method they were told they were looking for an unknown number of files, some of which are related to drug trafficking. The first method is to use traditional Linux commands such as `ls`, `cd`, `grep`, `file`, `md5sum`, `stat`, and `find`. Our second method used the developed visualization techniques.

During the study, each subject recorded four pieces of information: the time the study began, the discovery time, the name of each suspect file, and the time the study ended. In gathering this information we hoped to determine if one method proved better at finding more files and in less time.

Ultimately, each subject using the forensic visualization techniques was able to locate a greater or equal number of files compared with using the Linux-based command search. Only one tester located the same number of files using both methods. All other subjects located one more file using the forensic visualization techniques than they did using the traditional search. The results show that on average 53% more files were located using the forensic visualization techniques, supporting our claim that organizing information in a way that supports clustering and outlier detection increases the probability of discovering suspect files.

Moreover, a 35% reduction in time was realized using the forensic visualization techniques. In regard to time, another supporting statistic shows the time to locate the first file was 57% faster using the forensic visualization techniques. This shows the subjects were able to use the visualization techniques with ease and achieve results in just a few minutes. Figure 2 shows the relationship between the number of files found over time assuming all the subjects were searching simultaneously. This plot shows that at any given time during the study more files were identified using our visualization techniques than using the traditional search method.

An interesting point made amid all the data regarding the speed and success of the forensic visualization techniques is that a renamed media file was never located using the Linux-based command search. The file `/lib/libdth.so.420` was a JPEG file hiding among a sea of shared libraries. It could have been detected

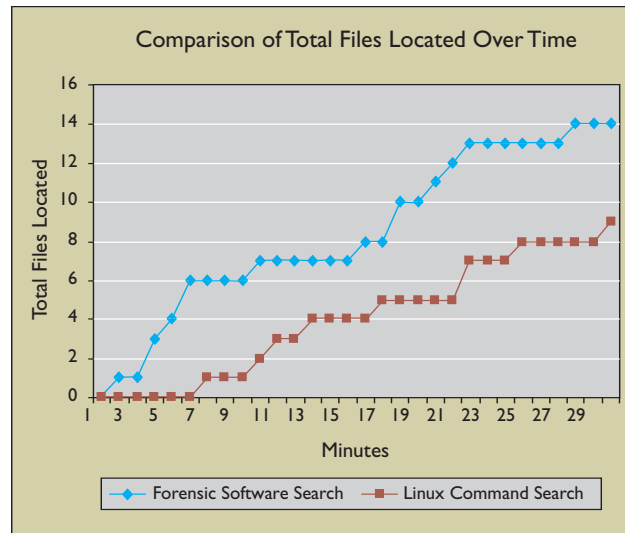


Figure 2. This plot shows the number of files located over time assuming all subjects were searching simultaneously. At all points our tool shows significantly more files located.

THE CONCEPT OF FILTERED TREE MAPS HAS PROVEN CRITICAL FOR THE ANALYSIS OF DIRECTORY HIERARCHIES IN CONJUNCTION WITH BLOCK VIEWS OF FILES.

using the command 'file /lib/*', which would have listed out of each file type in the directory. A search of the output would yield a single line stating the file was really a JPEG image and not a shared library as its extension implied. This file was easily located by many of the subjects using the visualization approach.

CONCLUSION

We have developed several novel techniques for the forensic analysis of hard drives in association with the needed interactive metaphors. In particular, the concept of filtered tree maps has proven critical for the analysis of directory hierarchies in conjunction with block views of files for zoomed or more detailed analysis of individual directories or large numbers of files. A test environment was developed, incorporating the two developed visualization techniques, as well as sufficiently extensive interaction metaphors to make the developed techniques useful for exploration.

In addition, a set of user experiments revealed the effectiveness of the techniques. Compared to Linux-based shell commands, the visualization techniques have proven far more effective. The evaluations performed were not meant to be complete, but rather to validate the effectiveness of the techniques on standard scenarios. Given the level of success with these test cases the visualization techniques will likely prove enormously valuable for the more complex scenarios.

The visualization techniques currently provide a novel component for forensic analysis. Our goal is not to develop a complete environment in competition with the likes of EnCase, but to ultimately integrate the output of EnCase's search capabilities as an additional input metaphor for the visualization environment. This additional input source would greatly aid in analysis. Given the ability to automate searches or execute frequently used but complex searches, highlighting identified files within the visualization will allow the files to be rapidly analyzed, as well as allow additional files to be located through application of the visualization techniques. ■

REFERENCES

1. Kruse, W.G. and Heiser, J.G. *Computer Forensics Incident Response Essentials*. Addison Wesley, Boston, MA, 2002.
2. Schneiderman, B. Tree visualization with tree maps: 2D space-filling approach. *ACM Trans. on Graphics* 11, 1 (Jan. 1992), 92–99.

SHELDON TEERLINK (steerlink@accessdata.com) is a software engineer at AccessData, a computer forensics software and training firm in Lindon, UT.

ROBERT F. ERBACHER (Robert.Erbacher@usu.edu) is an assistant professor in the computer science department of Utah State University, Logan, UT.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
