5-2009

Using Visual Capabilities to Improve Efficiency in Computer Forensic Analysis

Karen A A. Forcht
North Carolina A & T State University

Joan C. Hubbard
University of North Texas

Using Visual Capabilities to Improve Efficiency in Computer Forensic Analysis

Dr. Karen A. Forcht, North Carolina A & T State University
Dr. Joan C. Hubbard, University of North Texas

ABSTRACT

Computer forensics is the preservation, analysis, and interpretation of computer data. Computer forensics is dependent on the availability of software tools and applications. Such tools are critical components in law enforcement investigations. Due to the diversity of cyber crime and cyber assisted crime, advanced software tools are essential apparatus for typical law enforcement investigators, national security analysts, corporate emergency response teams, civil lawyers, risk management personnel, etc.

Typical tools available to investigators are text-based, which are sorely inadequate given the volume of data needing analysis in today's environment. Many modern tools essentially provide simple GUIs to simplify access to typical text-based commands but the capabilities are essentially the same. For simplicity we continue to refer to these as text-based and command-based in constrast to the visualization tools and associated direct manipulation interfaces we are attempting to develop. The reading of such large volumes of textual information is extremely time-consuming in contrast with the interpretation of images through which the user can interpret large amounts of information simultaneously. Forensic analysts have a growing need for new capabilities to aid in locating files holding evidence of criminal activity. Such capabilities must improve both the efficiency of the analysis process and the identification of additionally hidden files.

This paper discusses visualization research that more perceptually and intuitively represents file characteristics. Additionally, we integrate interaction capabilities for more complete exploration, significantly improving analysis efficiency. Finally, we discuss the results of an applied user study designed specifically to measure the efficacy of the developed visualization capabilities in the analysis of computer forensic related data.

KEYWORDS: Computer Forensics, Visualization, User Interfaces, Digital Evidence, Cyber-Forensics, Digital Forensics Information Security

1 INTRODUCTION

Computer forensics is the preservation, analysis, and interpretation of computer data [1]. Given the rapidly increasing number of crimes in which computers are involved, as well as the increasing diversity of the crimes, the large volume of data needing analysis is overwhelming investigators. This is further complicated by the increasing size of hard drive storage capabilities. Thus, there is a need for new capabilities and techniques in order to assist investigators. Investigators need these and other more advanced analysis techniques to more readily detect files hidden by increasingly more sophisticated methods. Such methods include: relocation, deletion, renaming, encryption, compression, etc.

"The recently enactment of the Sarbanes-Oxley Act requires publicly traded corporations to institute effective information security controls that adhere to the control framework set by the Committee of Sponsoring Organizations (COSO). This has had an impact on the collection of forensic data that is now emerging as an essential part of ensuring corporate compliance to the new governance environment set by government [2, 3]. Development of a responsive forensic team to perform data collection activities and mitigate legal liability is imperative. This is especially necessary in association with insider abuse, fraud, asset misappropriation, corporate espionage, and pornographic material that may create a hostile work environment [4, 5]. At the same time, organizations must ensure the costs associated with the formation of these teams and their associated activities are appropriate for the identified risk.

Therefore, the creation of highly usable forensic tools is emerging as a crucial need [6]. If these tools do not appear easy to use and helpful in detecting and analyzing hidden files, end users will not adopt them [7]. It is also imperative that these tools improve the efficiency of forensic examiners to justify acquisition costs. Such tools can be costly due to their small market; however, if they can eliminate the need for hiring an additional analyst then the tradeoff would be well worth it. One way to meet this demand is by increasing the accuracy of their results, in terms of false positives and false negatives.

Visualization techniques may aid investigators in the identification of suspicious files through the graphical display of file information. Such techniques greatly reduce the amount of time required for the analysis of large amounts of data. This is exemplified by the age-old adage, "a picture is worth a thousand words" [40]. In essence, we are relying on the human visual system that is able to interpret imagery data perceptually in parallel [9], in contrast to text that must be interpreted perceptually in serial [10]. The human visual system essentially performs some visual processing

early in the chain of handling visual input without conscious thought [8]. In other words, humans must process each character of text individually while and image can be interpreted en masse. Similarly, the human visual system can preprocess images before conscious thought to identify patterns and anomalies, greatly aiding identification of elements of interest before the application of conscious analysis.

Through the application of visual perception concepts we have developed visualization capabilities that facilitate the display of hard drive information for forensic analysis. Through this visualization display, and the associated graphical user interface, the user is able to query and then display characteristics of files and subdirectories in a selected sub-region of a hard drive. The visualization display graphically represents file characteristics using color, intensity, and size. Additional interaction allows specific file details to be displayed in addition to the abstract representation. Ultimately, the expectation is that the developed capabilities will significantly improve the computer forensic analysis process. In addition to research with the described visualization and interaction techniques, we implemented a prototype system implementing these techniques. This system was used in a set of controlled user studies to measure the effectiveness of the developed techniques.

2 TARGET USERS

The computer forensic process has two phases: the collection of data from partitions or disks; and the analysis of the collected data. The visualization techniques discussed here, fall into the analysis step of the forensic process. Once a hard drive has been imaged, these techniques may be used to search for evidence contained within the captured image. The target users for the visualization techniques fall into three main groups: law enforcement, security professionals, and forensic services groups. These groups may report their findings to prosecutors, national security analysts; to corporate legal, human resource, and IT departments. Within these groups, the skill levels and technical understandings may vary greatly among investigators. Some police officers, detectives, and special agents, who are interested in computer forensics, attend single or multiple, short courses on investigating computers. While their background is not always technical in nature, they have the benefit of understanding the forensic process, and the importance of preserving and properly documenting evidence. However, only 12.3% of sworn law enforcement officers have any training in computer forensics, and only 6.8% of them have any training in computer science [11].

Security professionals usually have a well-developed practical knowledge of computer and network architecture, but often overlook the necessity of the forensic process. The forensic process is essentially the identification and analysis of data relevant to a crime while maintaining chain of custody, data integrity, and analysis steps undertaken. Corporate emergency response teams rarely have training in the forensic process, and are traditionally concerned with restoring services rather than preserving evidence. A small police department may not have an in-house trained staff to perform the investigation and may need to outsource this work to external specialists. There are forensic service groups that offer computer forensic services to companies and individuals for payment. They also cater to companies who are not large enough to warrant specialized computer security professionals.

A user of the system presented here should have a working knowledge of file systems, Linux, Windows, directory traversal, file formats (such as .bmp, .jpg, .dll, .so, etc.), file attributes, compression, and regular expressions. With these skills, users are able to commence searching for hidden files or altered system files that obscure the presence of suspect files. The next section (section 2.1) describes the significance of hidden files and the skills needed to identify them and evaluate their evidentiary value.

2.1 Data Hiding and Concealment

Most hard drives requiring forensic analysis possess hidden data [12]. The data may be password protected, encrypted, compressed, renamed, placed in an unusual location, appended to another file, or may fail to show up in a directory listing because system programs have been altered. Today's typical hard drives are enormous, and when full, contain hundreds of thousands of files. An average size hard drive today is about 100GB. For servers and non-traditional systems the amount of storage is easily 500GB to 1+TB [30]. This clearly becomes daunting to analyze without sophisticated tools.

Employee abuse of computing resources is a major threat to corporations. However, without an effective and efficient forensic investigation, it may be impossible to take action against employees who have violated company policies, or engaged in illegal activity at work. Having an effective forensic process is a key risk management tool for larger corporations in today's highly regulated and litigious environment [31].

National security officials routinely examine computers and other electronic devices seized from suspected and known terrorists [32]. Though specific cases are generally classified, it has been reported in the media that data recovered from computers seized from raids in Iraq and Afghanistan have led to arrests. Thus, forensic examiners must analyze systems protected by dedicated opponents, and may examine files in languages they do not read or speak. This highlights the need for visualization tools as one means to aid in the detection of altered or anomalous files.

Knowing the type and number of files stored on a hard drive may ensure that the search process is quicker yet less tedious. This knowledge helps an investigator determine how effective string matching [33] will be or what applications he will need to open certain files. In a typical system, the majority of the consumed storage space is in data files. These often include such files as: .m3a, .mdf, .mp3, .dbx, .zip, .mhk, .mdb, .jpg, and .hxs. The next most common files include the shared libraries (.dll and .lib), followed by executables (.exe), files with no extension, and dump files

(.dmp). Many of the unfamiliar file endings like .drs and .mhk are application specific. For example, .mhk files are data files for Riven, the predecessor to the popular game Myst. Knowledge of file descriptions and their related extensions may speed up the analysis process.

If the vast amount of data that may be held in large hard drive storage space is not enough to intimidate an investigator, there are also many ways to hide a file. One method is the use steganography applications that camouflage information inside innocuous file. Messages, for example, may be embedded with pictures in such a way that the images do not look any different from the original. Niel Provos [13] discusses how steganography works and suggests methods for identifying it during forensic investigation. A number of detection tools are available to forensic analysts including: Provos' *StegDetect* [14], Wetstone Technologies' *Gargoyle* [15] and *Stego Watch* [16], AccessData's *Forensic Toolkit* [17], Guidance Software's *Encase* [18], etc. The extent of visualization for these tools essentially includes file explorer type views and simple graphs and charts. While advancements in these detection tools have reduced much of the challenge for modern forensic analysts in detecting this form of data hiding, new steganography technologies with improved data disguising capabilities are being developed. Additionally, larger hard drives can easily overwhelm the ability for tools to analyze such drives.

Suspects, including those who know they may be investigated by law enforcement agencies, may opt to delete incriminating files from their hard drive, thereby evading detection and prosecution. A range of software tools used by investigators that are able to retrieve deleted data overcome attempts by suspect to hide data, provided it has not been overwritten by a wipe utility [19, 20]. Garfinkle [19], in his study, described the methods used to retrieve information off discarded or second hand drives. The techniques used to retrieve deleted data are applicable in any computer forensic analysis. A paper published by Gutmann [20] covers some of the methods available to retrieve erased or deleted data even in cases where it has been overwritten 10-15 times. He suggests a set of 22 different write patterns to minimize the probability of an unauthorized person from recovering erased data. Gutmann [20] later reports, that newer generation of hard drives are denser and have less slack space where data can hide. As a result, fewer overwrite passes can be made to minimize data recovery [21]. As a result, recovering deleted data may be an increasingly significant challenge for computer forensic analysts.

Another hurdle, for even the best investigator, involves locating suspect files, whether or not they have been encrypted [37, 38]. Kruse and Heiser [1] discuss general methods an investigator might use when analyzing a hard drive. Much of an investigator's ability to locate suspect files relies on their understanding of the operating system and the intricacies of file hiding. For example, computers running the Windows NT kernel have a feature called abstract data streams (ADS) that allows multiple file formats in a single file. When a program or other data is stored to a file's abstract data stream, it remains undetected by Explorer. The new file size is not reported and it appears as though it is a single stream file [22]. Without knowledge of such system intricacies, it becomes difficult to search the hard drive for hidden or less obvious evidence.

Security organizations, such as the SysAdmin, Audit, Network, and Security (SANS) Institute and the Computer Emergency Response Team (CERT) Coordination Center, offer guidelines and background information that investigators may use to initiate the forensic analysis of a hard drive [23, 24]. After anomalous files are located, they may be examined in text editors, hex editors, or specific applications. If the file is encrypted, a password cracking program, such as L0phtCrack (LC 5), may be used to retrieve the keys [25]. While criminals may be clever at concealing their activities on a target computer system, they often overestimate the capability of the methods they use to hide digital evidence of their attacks. This is partly due to the fact that they must access criminally relevant data frequently; i.e. a bookie will access spreadsheets of bets and odds almost continuously. Even if techniques that are more sophisticated are used to hide the data then it is simply an issue of how long it will take the analyst to locate the hidden data**.** While commercial tools exist, such as the Forensic Tool Kit, these tools require the analyst to search directories of recovered files or look at source code for files. Though powerful, these tools are often found to be very time-consuming and frustrating to examiners who are unfamiliar with the data hiding techniques or file formats used in a particular case [36].

The goal of this work is to develop visualization techniques that will improve efficiency and effectiveness for the expert analyst and the less experienced analyst new to the discipline. Our research looks at a means to achieve this and this is described in the following section.

3 VISUALIZATION OF DATA

Our goal is to aid the analysis of hard drives by aiding analysts in examining, correlating, and analyzing critical file attributes. Relevant file attributes would include: file size, file type, file type vs. extension mismatches, file access time, file modification time, file creation time, file path, and file neighbor types. These attributes will aid identification of files a user may have attempted to hide, for instance:

- A file hidden in a random path will not have the same type as its neighbors and may have had its extension changed.
- As the owner of the system will need to access the documents they are trying to hide, access times and modification times will be more recent and aid identification.

- The documents we are looking for will likely not be the larger files on the system. For instance, a spreadsheet of bets and odds will be fairly small compared to many files and the available space. Thus, size can be used to rapidly focus in on more relevant files.

To help represent file attributes, the software developed for this research used visualization techniques. One method of displaying the relationship of files visually in two dimensions is a Treemap [26]. In particular, we describe modifications to Treemaps to make them more effective for forensic analyst needs. Treemaps are particularly valuable due to their representation of not only the attributes mentioned previously but also of the underlying hierarchical information lost with most techniques. Treemaps attempt to remove the scrolling required by traditional node and edge tree views used by many tools, such as Microsoft Windows Explorer. Traditional Treemaps allow for rapid identification of large files as well as clusters of files. It is effective even when files are deeply nested in the directory structure

Schneiderman [26] explains that Treemaps are a two dimensional space-filling algorithm for complex tree structures. Treemaps are designed to display tree structures in their entirety on a single display for rapid analysis and interpretation. With Treemap each file is represented by a colored box spatially positioned to be representative of its relationship to other files and directories. The box is colored based on a chosen color scheme, generally with color representing a user chosen file attribute. The box size is determined by the size of the display region and a percentage relating the size of the file to the size of the entire directory in which the file is placed. Subdirectories are treated similarly to files but are subdivided to represent all of the files and directories contained therein. Essentially, this amounts to a recursive algorithm.

Treemaps are primarily designed to identify large files. However, Schneiderman [26] points out that a user may drag a mouse over the display and click on displayed boxes to identify the file name or additional information. Such detailed feedback is critical in analysis to identify why a file is anomalous and identify the specific file for segregation, further analysis, and inclusion in any formal action. Treemaps still have weaknesses as applied to computer forensics. For instance, small files and directories are hidden within the morass of larger files. Given a reasonable 100+ GB size hard drive, a single file will be hidden by the large number of directories, subdirectories, and larger files [27]. Thus, traditional Treemaps do not meet the needs of forensic analysis without additional filtering and display options. With these added features, Treemaps may provide groundwork for developing new tools for computer forensics.

The visualization techniques, and more importantly, the modifications to the visualization techniques for forensic analysis, are presented in the next section.

4 METHODS

We developed two visualization techniques to assist analysts in the investigation and analysis of hard drives; a non-hierarchical techniques and the Treemap-based hierarchical technique discussed previously. Each technique was designed around a different metaphor. The first technique, a non-hierarchical technique, assumes that the investigation environment with relate to hard drives as flat file systems without incorporation of any relationships or hierarchical information. The location information is essentially thrown out. This technique is effective for examining individual directories, as the lack of hierarchy information may make it difficult to identify anomalies or relationships critical to effective analysis.

The second technique, a hierarchical technique, is based on a visualization method designed to incorporate the hierarchical information critical to file locations and relationships on a hard drive. Thus, the positioning information thrown away with the first technique is incorporated at a fundamental level with this technique. The next section presents the non-hierarchical technique followed by the hierarchical technique.

4.1 Non-Hierarchical

Non-hierarchical representations of files are a classic representation familiar to most individuals. For instance, most disk defragmentation programs will use such a view [35]. The key characteristics of such techniques are the lack of any relationship between files and directories; i.e., any association of a file with location on the hard drive is lost. The disk defragmentation view is a very simple representation and essentially consists of representing files as simple, square blocks with color, intensity, or block size, used to represent file characteristics, such as file size. In this case, lighter and darker colored blocks represent larger and smaller files respectively. An investigator could easily discover a file size that contrasts greatly with the other files under investigation since it would stand out among a sea of differently colored blocks. Yet another display, using time as a filter, would still contain a mix of lighter and darker colored squares, but the meaning would be slightly different from the meaning associated with filtering by size. Using a time attribute for filtering, lighter colored blocks represent files with activity that is more recent. While this method explores the visualization and interaction techniques, it also explores differing parameters that are available and their effectiveness for differentiating and locating files of interest.

Such simple square block representations still provide enormous amounts of information very concisely and visually interpretable [39, 40]. Four different diagrams may be constructed from the files in any given directory and its subdirectories. One diagram is created for each file attribute: access time, modification time, creation time, and size. With these visualization techniques, the user may examine simultaneously the file size for thousands of files by visually

interpreting a single display. In contrast, reading the same data in textual format would be prohibitive and very difficult to identify relationships, inconsistencies, or anomalies. Each square block diagram is drawn so the individual blocks remain square and are as large as possible for the given screen size. The individual blocks automatically resize as the user increases or decreases the size of the GUI window. Having all the relevant information in one easy to view chart, graph, or picture aids in decision-making; this is a novel feature for forensic analysis. When an individual must make increased connections between facts over time and space, it reduces said individual's ability to decipher relevant information. This square block diagram gives users the ability to extract information relevant to the search without struggling with what the square blocks mean.

Time is also a critical piece of information in the forensic process. Knowing when a file was created, modified, or accessed can reveal behavior patterns leading to the discovery of evidence or successful prosecution. Files that show recent activity may be singled out as possibly being more relevant to the investigation than older files that have not been modified or accessed in months or even years. An assumption made in this research project, using the square block diagrams, is that files with recent activity have a higher probability of being suspect. Drawing on that assumption, the block diagrams were rendered to draw visual attention to the more recently accessed files by rendering them as lighter colored squares.

The use of square blocks in the manner proposed has not been examined. However, such non-hierarchical techniques are limited in their inability to show relationships within a directory structure effectively. The hierarchical technique described in the next section is designed to show these relationships.

4.2 Hierarchical

The key failing of the square block visualization technique is that since it provides only a flat view of the files it loses all relationship information that is critical to identifying whether a file is anomalous, i.e. out of place [26]. Treemaps as a hierarchical visualization technique do not lose this relationship information. In the case of forensics, the hierarchical representations maintain the directory relationships critical to identifying anomalies in terms of file locations. We extend Treemap's capabilities with the concept of a filtered Treemap, which can display other file characteristics of interest and additionally control the selectively filter the range of elements to be displayed. The use of filtered Treemaps in computer forensics, as proposed here, has not previously been examined. This provides a novel contribution and offers many advantages over traditional Treemaps. For instance, a filtered Treemap may correlate the graphical element's position and dimensions with modification time, rather than file size. This is important since Treemaps merely represent file size as a space-filling two dimensional graph. With the ability to filter the file information that is graphically represented (for example, displaying only files with access times falling within a specified range) the needs of the analysts are better met. Only the files meeting the filtering requirements are used to generate the graphical display; the size of the graphical elements are still be controlled by the file size [28].

These filtered Treemaps provide far more flexibility than traditional Treemaps. The filtered Treemap essentially filters the file attributes; much as the regular Treemap filters files based on size. Each of these hierarchical methods has strengths and weaknesses that affect the visual display of information and the resources necessary to manage the display. For example, graphs consume large amounts of screen real estate, Treemaps emphasize large files, and filtered Treemaps are dependent on the parameters used for filtering for their informative contribution.

Filtered Treemaps are interactive in that the user may dynamically select the file attribute mapped to the Treemap display and thus map it to the associated display element sizes. In essence, we are merely changing the analyst's *view* of the data and not changing the source data. When the mapping is changed, by the user selection of either access time or modification time as the representation parameter, the Treemap is redrawn based on the newly selected attribute such that the size of each display element is representative of how recently the file was accessed or modified. This simply creates Treemap in which larger display elements represent more recent activity and smaller display elements represent older or less recent activity, i.e. less recent access or modification of the file.

The idea behind this application of the filtered Treemap is that more recently accessed files are more likely to hold data related to a crime; i.e. a bookie would be accessing their spreadsheet of bets and odds constantly. Such files, i.e. evidence, would not go un-accessed for any length of time. Alternatively, by specifying a time range of interest older activities could be identified. Thus making the more relevant files larger and more visible, based on more relevant attributes, draws attention to them such that they may be analyzed more efficaciously.

We employ a coloring scheme in addition to the visual scale of the elements. This coloring scheme identifies files of known types according to a user specified table. This will clearly identify the types of files present, particularly when an anomaly is present and a file is out of place. Additionally, this will aid identification of misnamed files in which the file type does not match the extension of the file.

Specifically, we may consider the default coloring scheme. In this scenario, an image file would stand out against a directory full of system binaries or shared libraries. Here, the image file would stand out as red against a background of blue. This is an example of one of the test files presented to the subjects in our user study as will be described later. This paradigm will allow an investigator to rapidly identify anomalous and irregular placement of files for further analysis in order to identify actual evidence of criminal activity; i.e., through visual clustering and outlier detection.

4.3 Test System

Here we describe the characteristics of the test system employed for the user study. The primary goal of the test system is to provide a model for the described visualization techniques. However, interaction techniques are critical for any analysis environment. Therefore, we also present the interaction capabilities incorporated to make the visualization techniques effective. This system not only uses visualization to represent a file system, but also is specifically designed around the forensics process.

The developed system will visually render the data from a user-selected sub-region of a hard drive. Such a sub-region will typically consist of a multitude of hierarchically organized directors, subdirectories, and files. As both described visualization techniques, hierarchical and non-hierarchical, are implemented, the user is able to rapidly switch between the two techniques.

Additionally, the visualization display itself is responsive to user interaction. The most important use for this is to garner specific details of a file, through a popup window. These details are accessed simply by clicking on the visual element representative of the file. The popup window includes the following details: file type, file name, permissions, owner, group, access time, modification time, and creation time. Additional characteristics may easily be added should they be deemed appropriate in the future. An example of the file details is shown in Figure 1.

Furthermore, the environment allows files to be opened directly from within the test system. Thus, should analysis identify a file deemed anomalous then it may be opened fore detailed analysis. Files may be opened with an application of choice, such as a traditional application or a hex viewer.

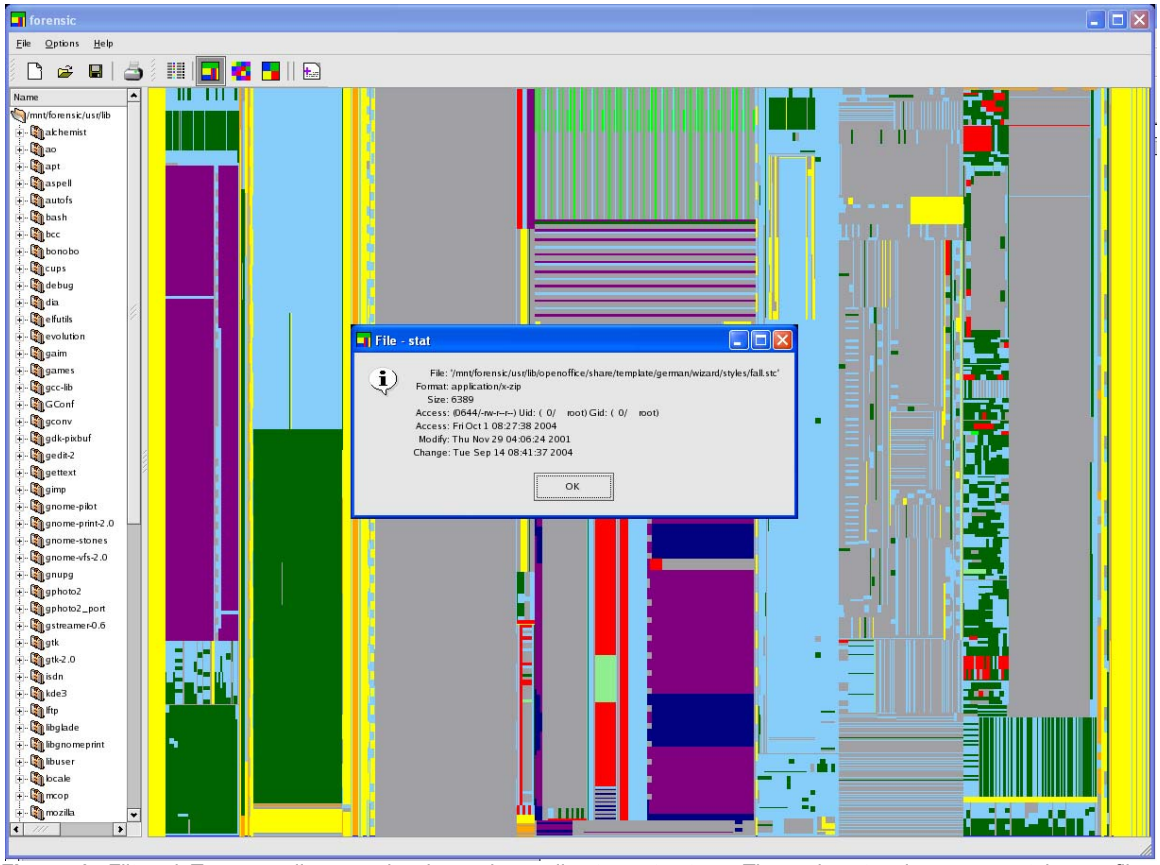

Figure 1: Filtered Treemap diagram showing a large directory structure. The red rectangles represent image files accessed recently. A message box is displayed after selection of one of the red rectangles and shows the selected file's detailed information. Notice the discrepancy between the file name and format. (use of arrows and a more readable diagram will enhance greatly the visual impact of this paper and help emphasise your diagram better.

The ability to view the contents of files has been extended to allow for the viewing of the contents of compressed and archived files within the test system. Thus, when a file is identified as compressed or archived in an initial visualization view, likely through the coloring of the file, the file may be selected and essentially zoomed into to identify what files are contained within the archive or to view the contents of the compressed file without the need for external tools, i.e. archive file zooming. This allows for rapid analysis, even against trivial techniques such as compressing or archiving files that have been used to files that may contain evidence of criminal activity.

One final capability, associated with the color scheme presented earlier, is the ability to highlight files identified as

having been modified. This follows the paradigm put forward by tools such as Tripwire [29]. The basic idea here is that the user would create a baseline Message-Digest algorithm 5 (MD5) digest for select files within the system. This digest can then be compared with the current MD5 value for a file to identify unexpectedly modified files. Clearly, this approach may not be applied without prior preparation, and managing the MD5 digest may be a nuisance to investigators when the system is frequently updated. However, this may aid in the identification of hidden information.

In terms of a specific example, the MD5 digest of the *ls* system command for the Redhat 9 OS running kernel version 2.4.20-8 is 'dbc1a18b2e447e0e0f7c139b1cc79454'. Should this 128 bit value from the digest not match the actual MD5 value of the *ls* command on the system then the mismatch will be noted and the visual representation of the file noted by coloring the file in a hatch pattern as shown in Figure 2. This will again identify files needing further investigation as such modified files could be indicative of an attempt to hide criminal activity. For instance, modification of the *ls* command is often done in order to prevent the *ls* command from listing out key files related to criminal activity.

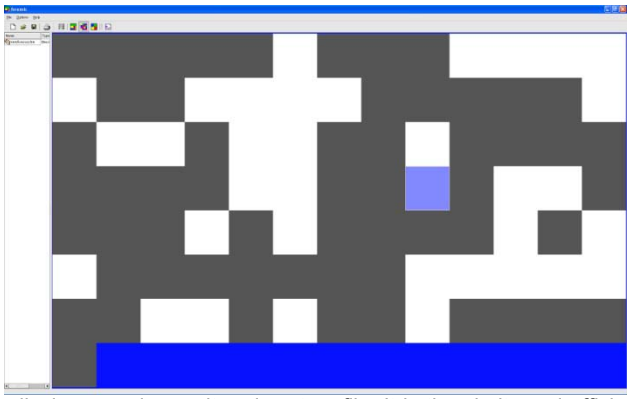

Figure 2: The square block display reveals an altered system file. It is the clarity and efficiency of this revelation this is critical, particularly when considering the analysis problem at hand.

4.4 Data

Two different datasets are used for searching, one for each analysis method. The first is a traditional UNIX shell-based command search which acted as our control. The second is a search based on the researcher's advanced visualization techniques.

The two datasets each amount to approximately 2GB file systems. 2GB was chosen so as to be a representative subset of a much larger dataset, providing a substantial search space, without being unreasonable to manage with respect to the user studies themselves. For instance, we expect to save the datasets for future evaluations and for any needed validation. A much larger dataset would become unreasonable to store effectively.

In terms of much larger hard drives a majority of the disk space would likely increase the number of similar files, e.g., MP3 files, video files, or large-data files; most of which would be trivially rejected or accepted for further analysis. Thus, this would not significantly impact the investigative process but make handling the data more unmanageable. Additionally, there would not be a significant increase in the diversity of files.

A Linux directory structure was chosen as it matches the development environment and avoided any unknown conflicts that would inappropriately impact the user studies. However, any other file system could by used, including NTFS from a Microsoft Windows system. The dataset was a scaled down version of a default Linux file system. Directories used in the test data include: /bin, /sbin, /lost+found, /usr, /lib, /root, /dev, /home, and /tftpboot.

Creation of the test datasets was done by performing a new operating system install, creating four new user accounts, and filling the user accounts with typical data found in such accounts. Creation of typical data was done by logging in as each of the newly created users and playing with the system for some time. Finally, the anomalous files were created within the directory structure with the location, names, and characteristics of the inserted files maintained for future reference. The anomalous files were created and accessed in order to be representative of documents holding evidence with an attempt at being hidden. This includes changing their names, relocating them, and compressing them. It should be noted that this test was not intended to be overly complicated or all inclusive but to be indicative of the capabilities of a *typical* criminal, relatively unsophisticated in their capability, and capable of allowing the hidden evidence to be identified by the hypothetical investigator through reasonable investigative diligence.

Given the created file systems, a file system image was created for each case that could be copied to the testing machine and made mountable. Once copied to the testing machine the test image was mounted read-only such that the subjects would be able to explore the data without modifying the test data and impacting the remaining user studies.

Once the new file system is mounted read only, it may be analyzed without modifying any of the data. It is noted here that the forensic process step of creating an exact image of the hard drive was not used. Since the purpose of the software in this research project is to identify existing files rather than deleted files it was not necessary to move hard drives between machines and create an image. However, in a real forensic examination, the target drive would always

be imaged and the image examined on a forensically sterile machine. Otherwise, the defense may argue successfully that the "evidence" has been tampered with.

The final two test cases were similar such that they contained the same number of files and the files were similarly, though differently hidden. This was to ensure that there would be no bias based on the test dataset itself and no transferable knowledge. Each test case contained an altered system file, a renamed media file, and a renamed office document. In preference to altering the system files, the MD5 cryptographic hash in the database was changed for the targeted commands, thereby simulating the modification of a file by making it appear as if the current copy was in fact a modification.

In terms of the selection of files for analysis, it was assumed that the background of the scenario would not be valuable. The rationale for this assumption rests in the fact that computers are often found at crime scenes and the investigators do not know what data may be on the computers (e.g., data associated with an unrelated crime). In addition, since data may be hidden in relation to any file (i.e., any existing file can be modified to hide the incriminating evidence), possessing knowledge of the scenario would not necessarily aid in the analysis process. Thus, the goal of this analysis task was to locate the incriminating evidence without the ability to rely on any associated data.

5 EVALUATION

An initial user study prototype mode was performed to evaluate the effectiveness of the environment. The basic idea behind this user study was for the test participants to identify three hidden or modified files within two sample hard disk images. During the study, we provided the participants with a basic introduction to the two techniques they would be applying, namely the visualization techniques and useful UNIX commands (i.e., the control). Typical UNIX commands would include ls, cd, grep, file, md5sum, stat, and find.

The philosophy behind the scenario presented to the user was that a computer was found at a crime scene and it is unknown as to what information on the computer might relate to criminal activity, if any. Thus, the participants were instructed to identify any number of out of place files needing further analysis that may be related to criminal activity; they were not instructed as to the exact number of files to be identified.

Each participant filled out a pre- and post-experiment survey so as provide feedback about the user experience, capability, and feelings about the tested capabilities; i.e. a qualitative assessment. In addition to capability, the questionnaire garnered insight into each subject's knowledge of computer forensics, including concepts such as how they envision files being hidden or how they would proceed with locating such hidden files. During the evaluation of each participant's results we recorded the time at which the study started and ended as well as the time at which each file was identified, if any. These temporal values provide for a quantitative analysis of the capabilities and techniques. The goal was to identify if one technique allowed for statistically more files to be identified in less time, indicating the technique is statistically more effective.

In an attempt to remove any advantage of one technique over the other we subdivided the participants into two groups. One group performed the study using the visualization techniques first while the second group performed the study using the UNIX command shell first. In addition to removing any bias, this would aid in identifying if there was an impact on the second half of the study by the technique used in the first half of the study.

Execution of each experiment was limited to 30 minutes and the number of files identified by a participant was limited to this time period. Given that forensic analysts are currently limited to simple capabilities such as those available through the UNIX command shell, though in a *very* advanced and experienced way, we wished to identify the extent to which the visualization techniques may aid investigators in locating files more efficaciously. The participants didn't have extensive experience with either the text-based or the visualization-based forensic analysis capabilities. This essentially placed both techniques on a level playing field; i.e. the participants were not guaranteed to perform better with one technique or the other due to prior experience.

6 RESULTS

At the completion of the user studies the resulting data were analyzed in terms of the effectiveness of the techniques and their impact along three dimensions. Firstly, the efficiency of the techniques as a measure of their effectiveness was compared. Secondly, we examined the impact of the order of the application of the techniques, i.e. did running the experiment with one technique first as opposed to the other have an impact on the results. Finally, we examined the different search techniques the subjects applied and the impact of each search strategy.

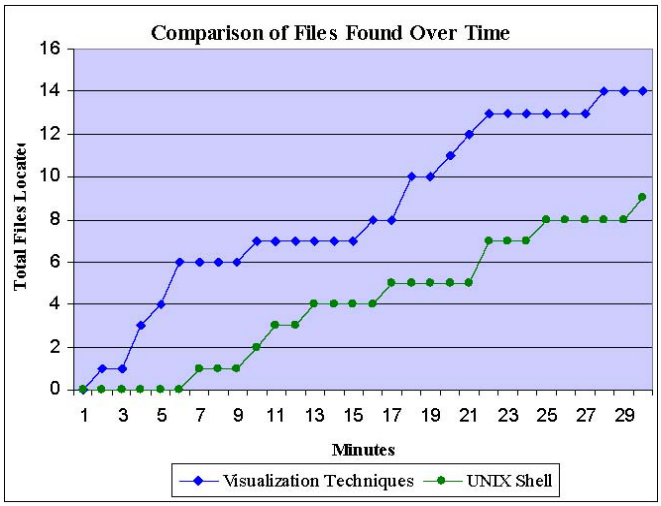

Figure 3: This graph shows the number of files located over time assuming all 6 subjects were searching simultaneously. At all points the visualization techniques show significantly more files located.

6.1 Time and Quantity

The most fundamental result desired was to determine if the participants were able to locate more hidden and modified files with the visualization techniques than with the more traditional UNIX shell commands. The results of our initial user study showed that the visualization techniques were far more effective, allowing more files to be located than the UNIX shell commands in nearly all cases. A single subject located the same number of files with each technique. All other subjects located more files with the visualization technique. More specifically, ~53% more files were located with the visualization technique than with the UNIX shell commands. This suggests that organizing information in a way that supports clustering and outlier detection increases the probability of discovering suspect files, though this finding needs to be supported by further research with a larger number of subjects.

On average the participants took ~13.7 minutes to locate a new file with the UNIX shell commands, i.e., to locate the first file or to locate a new file after one had already been identified. With the visualization techniques this time period is reduced to ~8.8 minutes, a significant improvement of 35%, or a reduction of nearly 5 minutes. Also of note, is the result that showed that locating a first file took ~57% less time with the visualization techniques than with the UNIX shell commands. This shows that the visualization techniques were easier to use and participants were able to more effectively perform the forensic analysis with minimal experience or practice. These most critical characteristics of the results are exemplified in Figure 3 which compares the two approaches by representing total number of files found, y-axis, versus time, x-axis. The deviation in the two line graphs exemplifies the improved performance achieved with the visualization techniques. At any point in time more files were found with the visualization techniques than with the UNIX shell commands and in most cases many more files.

An additional point of note is the fact that the renamed media file, namely /lib/libdth.so.420, was never identified with the UNIX shell commands. It was however, easily identified with the visualization techniques. This file was actually a renamed .jpg file sitting in a directory full of shared libraries. This deviation in file types is a clear indicator of a need for the file to be further analyzed as it is likely a file a user has attempted to hide. This anomaly could have been identified with the UNIX shell structure between techniques, regardless of their order of commands by executing 'file /lib/*'. There still would have been application. Difficulty in identifying the anomaly due to the shear volume of files but this would have been reflected in the time to identify, rather than the file simply not having been found at all.

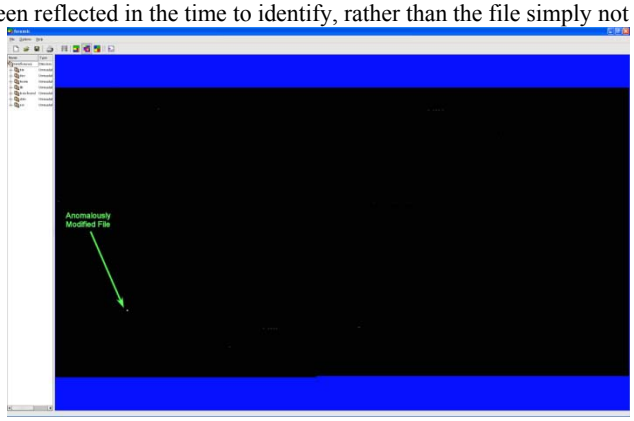

Figure 4: This example of the square block diagram shows the /lib directory filtered on modification time. The file of interest is a single white block, as identified. With this visual representation, the file stands out clearly. Attempting to locate this file using typical textual tools has proven extremely difficult.

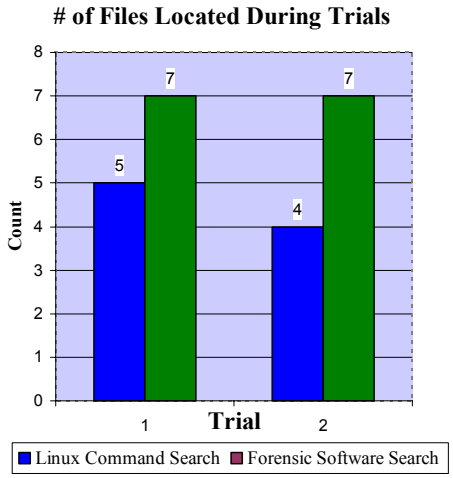

Figure 5: This graph compares the number of located files for each trial and technique. The first trial represents the performance of the UNIX command shell first while the second trial performed the GUI-based search first. Clearly, the visualization metaphor performs significantly better in either scenario.

In contrast, as exemplified in Figure 4, the anomalous file was found to be easier to identify with the visualization techniques based on the results of the experiment. In this example, we have filtered data, based on modification time, rather than file size, with the square block visualization technique. This approach results in a display filled with black squares, except for the modified file, identified in the bottom left quadrant of the display. This works because of the infrequency with which shared library and other system files are modified. Most of these types of files are installed once at system configuration time with modification dates well before the system configuration time. Alternatively, we could have displayed the file types, as represented by different colors, and identified the deviated file in this format.

Once such a suspect file is identified, the user is able to click on the identified glyph to open a detailed file view that displays the file's name, type and other characteristics. This visual feature clearly identifies the inconsistency between the file name and type.

6.2 Interaction of Methods

It was also important to identify whether each participant's performance was dependent on the order that the analysis techniques, visualization vs. UNIX command shell, were applied or whether the order in which the test datasets were investigated. In other words, did the use of one technique influence a participant's performance with the other technique? To explore this we had the test subjects alternate which technique they employed first vs. second. After the experiments we examined the number of files located and the mean time to locate a file. This will clearly indicate a preferences or lack thereof. These details are shown in Figures 5 and 6.

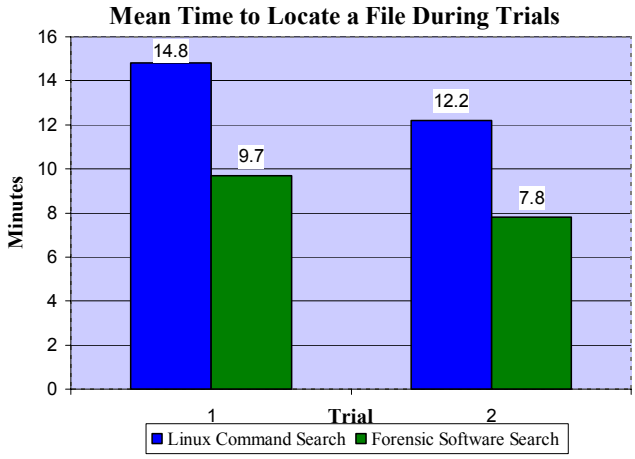

Figure 6: This graph shows the mean time to locate consecutive files for each trial and technique. The GUI-based mechanism is clearly more efficient. Additionally, user performance improved with the second technique, regardless of order.

The plot in Figure 5 does not show any evidence to indicate that the first method had a strong effect on the second, either positively or negatively. However, Figure 6 shows that the time to locate a file using the second method is reduced, regardless of the technique used. While this preliminary data does not support an overwhelming argument about the interaction of each method, it does lead to the preliminary conclusion that higher performance is associated with the second technique applied by a particular user. Since both datasets were nearly identical, it is reasonable to suppose the subjects became increasingly more familiar with the directory.

7 TECHNIQUES FOR SEARCHING

At the completion of the user studies, the researchers wanted to understand the search techniques employed by the test participants and gain greater insight into the effectiveness of the search tools. After completing both search methods, the subjects answered questions about their experience using the search tools. They were given liberty to express additional comments in addition to answering the specific questions.

The survey revealed that the subjects used two different methods when employing the UNIX shell command search and only one for the forensic visualization techniques. The two search methods using the command line were split between ls, file, and md5sum for finding recently modified/accessed files and type, and using grep (a text-based string matching tool) to identify files containing keywords related to drug trafficking. Those who used the file command and searched for modified/accessed files were more successful than those who relied on string searching. The hidden or altered files in the command search dataset did not contain any words related to drugs on which the subjects could search. One masqueraded file was a JPEG image file and the other was a delivery schedule that only had dates and names of high schools where marijuana was sold. A more experienced forensic analyst may have been able to identify the relevant data by searching for known locations of drug activity.

When using the forensic visualization techniques, the main strategy for discovering the files was to look for blocks and rectangles that stood out in size, contrast, or color. The filtered Treemaps were useful for visualizing file types quickly and helped the subjects to locate common file types such as images, office documents, text documents, compressed files, and executables. When a large contrast between the sizes of rectangles was discovered, the user would query more information from the system by clicking the surrounding files. Often discrepancies between file name and file type were enough to give the investigator reason to open the file immediately and examine its contents.

Despite the usefulness of the filtered Treemap for coloring files, it was limited by screen real estate in the number of files visible to the user; i.e. the number of pixels on the display fundamentally limits the number of representable files. For this reason, all investigators used the square block view when examining directories containing huge quantities of files. Figure 4 is a good example of where the block view triumphs over filtered Treemap. More than two thousand files are contained in the block view in such a way that they cannot all be drawn visibly in a Treemap. The filtered Treemap are more conducive to smaller numbers of files for viewing localized information in a spatial domain since it is presents more of a visual and cognitive challenge to users to search through larger domains. Tree maps cannot effectively represent partitions with large numbers of files. Enough visual space needs to exist to see the contrasts in activity between files.

8 FUTURE DIRECTIONS

Future work needs to be undertaken to further validate this tool. This proof of concept study was conducted using a limited sample size (N=6) of participant students. Future work should compare this tool with larger samples of users,

including trained forensic examiners. The tool should also be tested on a drive that has been the subject of a forensic examination. This would allow future researchers to determine how effective this tool is in assisting users to quickly discover known hidden files. In essence, we would then have a comparison using both traditional text-based techniques as well as our visualization techniques. This comparison could identify: learning curve, time to identify and analyze relevant files, the number of relevant files identified, etc.

9 LEGAL ISSUES

In most full forensic investigations, the investigator must create a forensic image of the target drive. Bit mapping of the drive captures hidden files, directories, swap data, deleted data, and information in slack space [34]. All of these may provide needed evidence to an investigator. In a full investigation, the investigator should access any available backups of the suspect's drive or system. These backups should also be copied to non-alterable media, and a chain of custody needs to be maintained over the images, backups, and copies. Ensuring that the backups are on non-alterable media should assist in showing that the backups are authentic and that the analysis of them is valid; i.e. the data is unaltered and the analysis results accurately represent the original media. This may subsequently lay the foundation for admission in a civil action and provide the necessary chain of custody for criminal prosecution. The current tool is designed so as to not alter the file contents, and should therefore preserve the integrity of the data for admission as evidence.

10 CONCLUSION

The researchers developed and studied the effects of several novel techniques for the forensic analysis of hard drives in association with the needed interactive metaphors. Specifically, we examined the impact of filtered Treemaps with positive results. These positive results are particularly valid when used in association with the detail analysis provided by the block view. This combination of capabilities is critical for the analysis of large numbers of files and directory hierarchies. We implemented a prototype environment geared towards forensic analysis. This prototype environment incorporated the two identified visualization techniques as well as a range of critical interaction techniques needed for exploration and analysis of hard drive images. Without such interactions techniques being designed directly into the visualization techniques the analysts would be limited in the specific details they would be able to derive from the visualization.

Additionally, the needs of forensic analysts were outlined and the appropriateness of the techniques were evaluated within the forensic process. The design of the techniques to fit specific needs of analysts and the forensic process in general, will help ensure the techniques will find value in deployment once they have received additional refinement. A discussion of the utility of forensic examination in law enforcement, national security and industry settings was also presented.

Finally, a set of preliminary user experiments were applied that suggested the effectiveness of the techniques. In comparison with UNIX-based shell commands, the preliminary visualization techniques have appeared to be more effective than traditional methods. This research used size of the file and modification dates as attributes for searching in order to demonstrate the general utility of visualization in performing searches of forensic drives. The development of test datasets that closely match real-world scenarios, particularly in terms of their level of challenge, should prove valuable for future research.

11 FUTURE WORK

New GUI controls should be added to allow users to more easily correlate files in separate views, (i.e. coordinated views). When a file is identified in the square block visualization, it is difficult to find the same file in the filtered Treemap view, and vice versa. New controls to sort the block visualization by file attribute are also desirable. Directory level zooming within views was also suggested by the test subjects.

Adjustment of the current visualization techniques would allow the user to see a square block diagram drawn for file type, (i.e., coloring blocks based solely on file type). Many of the investigators desired a square block diagram for situations when the investigator had only limited screen real estate for viewing the filtered Treemap. These scenarios may limit an investigator's performance due to insufficient resolution to see all available files efficiently. Adding options to reduce the number of files viewed would be desirable; therefore, it would be useful to investigate automatic and manual methods of reducing the number of displayed files.

12 REFERENCES

[1] Kruse W.G., Heiser, J.G. (2002). *Computer Forensics Incident Response Essentials*. Boston, MA: Addison Wesley.

[2] Sarbanes-Oxley Act of 2002. (2002). (Pub. L. No. 107-204, 116 Stat. 745) Washington DC: U.S. Government Printing Office.

[3] The Committee of Sponsoring Organizations of the Treadway Commission. (2004). *Enterprise risk management integrated framework*. Retrieved November 28, 2005, from http://coso.org/publications.htm

[4] Whitaker, J.A, & Howard, M. (2005). Computer Forensics. *IEEE Security & Privacy, 3*(4), 59-62.

[5] Erbacher, R., & Teerlink, S. (2006). Improving the computer forensic analysis process through visualization. *Communication of the ACM, 49*(2), 71-75.
[6] Nielson, J. (1999). *Designing Web usability: The practice of simplicity*. Thousand Oaks, CA: New Riders Publishing.
[7] Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly, 13*(3), 318-340.
[8] Neisser, U. (1967). *Cognitive Psychology.* New York: Appleton-Century-Crofts.
[9] Kelsey, C. A. (1997). Detection of Vision Information. In W. R. Hendee & P.N.T. Wells (Eds.), *The perception of visual information (Second Edition)(* p 51). New York: Springer_Verlag.
[10] Wickens, C., Sandry, D., & Vidulich, M. (1993). Compatibility and resource competition between modalities of input, central processing, and output. *Human Factors, 25*(2), 227-248.
[11] Bhaskar, R. (2006). State and local law enforcement is not ready for a cyber Katrina. *Communication of the ACM, 49*(2), 81-82.
[12] Casey, E. (2006). Investigating sophisticated security breaches. *Communication of the ACM, 49*(2), 48-55.
[13] Provos N., & Honeyman, P. (2003). Hide and seek: An introduction to steganography. *IEEE Security & Privacy Magazine, 1*(3), 32-44.
[14] OutGuess. (2003). *Steganography Detection with Stegdetect* [Online]. Available: http://www.outguess.org/ detection.php
[15] WetStone Technologies. (2004A) *Gargoyle* [Online]. Available: http://www.wetstonetech.com/page/page/ 1104418.htm
[16] WetStone Technologies. (2004B). *Stego Suite* [Online]. (May 24, 2004B). Available: http://www.wetstonetech.com/ f/Stego_Training_Datasheet.pdf
[17] AccessData. (2005). *Forensic Toolkit product page* [Online]. (November 28, 2005).Available: http://www .accessdata.com/Product04_Overview.htm
[18] Guidance Software. (2005). EnCase [Online]. Available: http://www.guidancesoftware.com
[19] Garfinkel, S.L., & Shelat, A. (2003). Remembrance of data passed: A study of disk sanitization practices. *IEEE Transaction on Security & Privacy, 11*(1), 17-27.
[20] Gutmann, P. (1996). Secure deletion of data from magnetic and solid-state memory. *Proceedings of the 6th Usenix Security Symposium.* Berkeley, CA.: Usenix Association. http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del .html
[21] Gutmann, P. (1998). Data Remanence in Semiconductor Devices. *Proceedings of the. 7th Usenix Security Symp*osium. Berkeley, CA: Usenix Association. http://www .cryptoapps.com/~peter/usenix01.pdf
[22] Scrambray J., McClure S., & Kurtz G. (2001). *Hacking exposed: Network security secrets & solutions* (2nd ed., pp.215-216). Berkeley, CA: McGraw Hill.
[23] SysAdmin, Audit, Networking, and Security (SANS) Institute. (2005). Available: http://www.sans.org
[24] Computer Emergency Response Team. (2005). Available: http://www.cert.org
[25] AtStake Corporation. (2005). Available: http://www.atstake .com
[26] Schneiderman B. (1992). Tree Visualization with Treemaps: 2-d Space-Filling Approach. *ACM Transactions on Graphics, 11*(1), 92-99.
[27] Ball, R., Fink, G. A., & North, C. (2004). Home-centric visualization of network traffic for security administration. *In Proceedings of VizSEC/DMSEC 2004*, *Fairfax, Virginia, 29 October, 2004.* (pp:55-64), New York: ACM Press.
[28] Technische Universiteit Eindhoven (2004). Available: http://www.win.tue.nl/sequoiaview/
[29] Tripwire, Inc. (2004). Available: http://www.tripwire.com/ resources/datasheets.cfm
[30] http://technet2.microsoft.com/WindowsServer/en/library/1dcfccff-ed50-4667-8136-bee1b580dae81033.mspx?mfr=true
[31] Rowlingson, Robert "A Ten Step Process for Forensic Readiness," *International Journal of Digital Evidence* (2:3), Winter 2004, pp 1-28.
[32] How al Qaeda put Internet to use, http://www.msnbc.com/avantgo/833533.htm
[33] http://en.wikipedia.org/wiki/Pattern_matching
[34] Brian Carrier, *File System Forensic Analysis*, Addison-Wesley Professional, 2005.
[35] Russell Kay, "Disk Defragmenters Demystified," Computer World, October 24, 2005, http://www.computerworld.com/printthis/2005/0,4814,105582,00.html
[36] http://www.ictlex.net/index.php/2000/09/01/forensic-computer-analysis-an-introduction/
[37] http://en.wikipedia.org/wiki/Encryption
[38] Niels Ferguson and Bruce Schneier, *Practical Cryptography*, Wiley, 2003.
[39] Stasko, John, Catrambone, Richard, Guzdial, Mark and McDonald, Kevin, "An Evaluation of Space-Filling Information Visualizations for Depicting Hierarchical Structures," *International Journal of Human-Computer Studies*, Vol. 53, No. 5, November 2000, pp. 663-694.
[40] Christopher Plaue, Todd Miller, and John Stasko, "Is a Picture Worth a Thousand Words? An Evaluation of

Information Awareness Displays", Graphics, Visualization, and Usability Center, Georgia Institute of Technology, Atlanta, GA, Technical Report GIT-GVU-04-02, February 2004.