*Avestia Publishing*
*Journal of Machine Intelligence and Data Science (JMIDS)*
*Volume 2, Year 2021*
*ISSN: 2564-3282*
*DOI: 10.11159/jmids.2021.005*

# Software Validation and Daubert Standard Compliance of an Open Digital Forensics Model

**Aparicio Carranza[1], Casimer DeCusatis[2]**
[1]New York City College of Technology - CUNY, Computer Engineering Technology Department
300 Jay Street, Brooklyn, NY, USA 11201
acarranza@citytech.cuny.edu
[2]Marist College, School of Computer Science and Mathematics
3399 North Road, Poughkeepsie, NY, USA 12601
Casimer.decusatis@marist.edu

***Abstract** - With the widespread increase in Cybersecurity incidents, there has been increased attention on the development of digital forensics tools and methodologies. We investigate a suite of Cyber-Forensic tools in the Open Source CAINE Linux Distribution, and conduct experimental software validation testing in support of open source code compliance with the well-established Daubert Standard for forensic evidence collection. We propose how tools such as Guymager, Autopsy, Fred, and PhotoRec can can be applied as part of a four tier forensic architecture, including experimental results which demonstrate the application of these tools.*

***Keywords***: CAINE, cybersecurity, Forensics, Daubert.



## 1. Introduction and Background

In recent years, both the number and severity of cybersecurity attacks have increased significantly; by some estimates, losses from such attacks are expected to exceed $2 trillion annually by 2019 [1]. The widespread increase in cybercrime has led to a corresponding interest in digital forensics investigation tools and methodologies. Several frameworks have been proposed for structured forensic analysis, but there has been comparatively little discussion around practical implementations of these frameworks. The potential use of open source forensic tools is a particularly attractive approach to this problem, since open source enables rapid development of security forensics tools and places these tools at the disposal of the security community for little or no cost. However, in order for digital evidence to be admissible in a court of law, it must comply with legal precedents such as the Daubert Standard [2]. There has been ongoing technical debate over the compliance of open source tools, and there is a need for additional documented testing in support of typical use cases. Specifically, a new suite of open source forensics tools has recently become available as part of the Computer Aided Investigative Environment (CAINE) distribution of Linux [3]. In this paper, we investigate how these forensics tools may be applied to a structured cybersecurity evidence gathering procedure, and we perform testing which may be used to assess compliance with relevant legal precedents.

While a cybersecurity incident may generate a great deal of interesting data, not all of this data qualified as forensic evidence that is admissible in a court of law. There are a number of professional organizations which have attempted to insure quality and consistency of evidence gathering within the forensic community and provide technical recommendations including how to preserve chain of custody (insuring that evidence possession is always tracked and auditable, compliance with rules of evidence, collection of volatile information first, searching disk slack space, etc.) [4]- [7]. For purposes of this paper, we will concern ourselves primarily with U.S. forensics legal precedents; these may differ from international standards, and may evolve in the future to include new techniques not available at this time. Generally speaking, most forms of digital forensic

*Date Received: 2020-05-11*
*Date Accepted: 2021-08-11*
*Date Published: 2021-09-02*

evidence fall outside the common knowledge of a jury, which must therefore rely on testimony provided by technical subject matter experts. Historically, expert testimony has been required to meet the so-called "general acceptance test" established by the Frye standard [8], which holds that scientific or technical evidence is only admissible in court if it was collected using a framework deemed generally accepted by the scientific community. Under this approach, the scientific community serves as the gatekeeper in determining whether digital forensic evidence is admissible in court.

This approach was modified in the early 1990s to a standard under which the judge, not the scientific community, determines whether forensic evidence is admissible on a case-by-case basis. The standards regarding admissibility of digital evidence and the use of expert witness testimony from computer forensics specialists were derived from the precedent setting U.S. Supreme Court case Daubert vs. Merrell Dow Pharmaceuticals, Inc., 509 U.S. 579 [2]. The court found that evidence or expert opinion derived from scientific or technical activities must come from methods that are proven to be "scientifically valid" and which meet five basic criteria [2]. In the context of digital forensics, the Daubert Standard means that tools and techniques used to collect and analyze digital evidence must be validated and proven to meet scientific standards. More specifically, digital evidence presented in a trial must have come from tools that can be proven to yield correct results through empirical testing. The tools and methodology used must pass peer review, use generally accepted theory and technique, and demonstrably meet acceptable error rates and standards. A trial judge may use the Daubert Standard to assess whether digital evidence can be properly applied to the facts at issue in a given case. Under Federal Rule of Evidence 702, digital forensic evidence in U.S. federal courts is governed by the Daubert Standard. Individual states are allowed to establish their own Rules of Evidence, which may follow the Daubert Standard or other criteria under different circumstances [9].

Establishing whether open source digital forensic tools can meet the Daubert Standard requires ongoing, periodic software validation testing. As defined by government organizations such as the U.S. FDA and CDRH [10], software validation is performed on a finished tool or piece of code, and is defined as 'confirmation by examination and provision of objective evidence that software specifications conform to user needs and intended uses, and that the particular requirements implemented through software can be consistently fulfilled.' In practice, software validation activities may occur both during as well as at the end of the software development life cycle. Such validation depends on testing, inspection, and analysis of tasks performed by the software, along with empirical evidence that software requirements have been correctly and completely implemented, and are traceable to system requirements or so-called "user stories" employed in the software design life cycle. This is not to be confused with software verification, which provides objective proof that design outputs of a given phase in the software development life cycle meet the specified requirements of that phase. Verification may include inspection of source code, documentation reviews, static and dynamic analysis, design walkthroughs, and other techniques. Software verification determines correctness, completeness, and consistency of code (often during the development process), often involving a line-by-line code review. We do not perform this level of analysis, since it has been established reproducible testing at regular intervals, with an acceptable level of granularity, is sufficient to demonstrate Daubert compliance [11], [12]. Conventional approaches to Daubert Standard compliance have favored closed-source code and tools, arguing that such code cannot be easily manipulated and citing widespread adoption and commercial product testing as proof of software validation. However, it may actually be easier to meet the Daubert Standard using open source forensic tools. This argument was originally put forth by Brian Carrier (author of the Autopsy tool discussed later in this paper) [13]. Open source forensic tools are implicitly granted community acceptance by virtue of their continued development and use, whereas closed source tools may rely on the advocacy of a single vendor. Open source tools also comply with aspects of the Daubert Standard related to publication, peer review, and periodic testing. In fact, the natural fit between open source community development efforts and software validation has led to open source forensics being called "the digital Daubert standard" [14]. In other words, open source forensic tools can easily and inherently prove that they are peer reviewed, published, falsifiable, generally accepted, and have a well established error rate.

By contrast, the Daubert Standard's stipulation are more challenging for closed source forensic tools. As noted previously, such tools often cite their large user base to prove community acceptance. However, the user base for a given closed source tool typically chooses the tool for qualities such as ease of use, intuitive interface design, and service/support/ maintenance/upgrade features [13], which are not related to procedural code development factors. If market share isn't a valid metric, then proving closed source tools to produce reliable output may be impractical. Closed source tools may be presumed reliable for incomplete reasons, such as the assumption that such tools will perform as advertised at all times [14]. Indeed, any such perceived advantages of closed source is analogous to the idea that we might achieve security through obscurity, which can lead to the Daubert Standard being entirely circumvented [14]. If it is more desirable to publish and openly evaluate forensic tools, then a framework of best practices has been recommended to insure that such tools are accepted in a legal setting.

The following steps should be taken to insure long-term acceptance of open source forensic tools [13]:

1) *Development of comprehensive tests for forensic tools*
2) *Publication of tool designs to help create more effective tests*
3) *Creation of a standard for calculating error rates for both tools and specific procedures*
4) *Publication of specific procedures that a tool uses. While open source tools already publish their source code, they should also clearly document the procedures used by their code*
5) *Public debate on the published procedural details to ensure that they are agreed upon.*

In this paper, we will apply this framework to CAINE tools in order to determine if they meet the long-term acceptance criteria listed above. Novel features of this work include empirically testing these open source tools against common use cases to validate that they produce correct and desired output within acceptable error rates. This paper will disseminate these results to the peer community, which will both add to the body of knowledge used by the digital forensics community and encourage the development of additional use cases. We document the features of the CAINE toolkit in this context, and map them against typical user requirements. Our use cases include disk imaging, file recovery, and hive management using CAINE tools including Autopsy, Guymager, Fred, and PhotoRec. We also review how these tools may be applied to a cyber-forensics architecture, and discuss relative strengths and weaknesses of this approach based on our experimental testing.

The remainder of this paper is organized as follows. After an introduction and brief review of prior art, Section 2 presents a four tier reference architecture for the various forensic tools available in the CAINE distro. Section 3 provides experimental results from using these tools, compared with other alternatives to help establish their prevailing error rates. A step-by-step description of this validation testing is provided to facilitate reproducibility of our results. Section 4 summarizes our results and conclusions of this work.

## 2. Forensic Architecture Development

The acceptability of digital evidence in a court of law is based on principles originally developed for more conventional forensic investigations. Both approaches require a chain of custody to insure that original evidence is immune to tampering. Digital evidence is more fragile than other types of physical evidence [15], thus investigative reports must be created to explain the digital evidence examination process and its limitations. There have been many efforts to develop consistent guidelines for digital forensic investigation [16], [17], including the first responders crime scene investigation model published by the U.S. Department of Justice [18]. This approach forms the basis for many different proposed data handling frameworks; while these models differ in the number and details of their process steps, they can conceptually be reduced to a basic four tier approach [19]. We have previously published an approach based on this methodology [20], so we will only provide an overview here. The first tier (*the preparation or collection phase*) involves the search, recognition, collection, and documentation of electronic evidence. The second tier is the examination phase, which helps to make digital evidence visible, explains its origins and significance, and reveals obscured information. The third tier is the analysis phase, which involves studying the product of the examination tier for its relative importance to the case under investigation. The fourth or reporting tier includes documenting the results of the examination process and limitations of the investigation. In the following sections, we describe a set of tools available in CAINE which address this four tier approach. We then demonstrate experimentally how the tools may be applied in a practical use case, and use this data as the basis for establishing compliance with the Daubert Standard.

The benefits associated with using open source tools in this framework include rapid innovation driven by a global development community and free or inexpensive access to a broad cross-section of security professionals. Such tools would also facilitate training and education efforts to address the lack of security practitioners and service industry professionals [20]. In October 2014, the forensic Linux distribution known as CAINE (*short for **C**omputer **A**ided **I**nvestigative **E**nvironment, but also named after the popular character Horatio Caine on the television series CSI Miami*) first became available from project manager Nanni Bassetti. More recent editions of CAINE are based on Ubuntu 12.04 and Linux kernel 3.2, but with the GNONE 2 fork known as MATE providing the desktop environment. CAINE differs from many other specialized distros because it also provides a suite of general purpose desktop tools which allow it to be used as a classic Ubuntu system, eliminating the need to switch back and forth between a general purpose environment and the advanced forensic tool features. Normally, a general purpose distribution would not be suitable for forensic purposes, because it automatically mounts all available drives as read/write. This poses several problems, including changing the "last mounted" times and potentially erasing data (including hidden data) when writing to the drive. To avoid these issues, CAINE never automatically mounts any device. Mounting is only possible through an applet called Mounter, accessible to the user through the system tray or command line (via the "mount" command). This applet also allows users to toggle the system policy for all future mounts from read/write to read-only and back again. In this paper, we focus on four CAINE tools which can be mapped to the four tier forensics architecture discussed previously. First, ***Guymager*** is an open source forensic disk imaging tool included with CAINE. It is a QT-based forensic imager [14]. Guymager also includes a compression engine, that can compact a disk image into one file for subsequent forensic analysis without harming the original image. This supports the principle of preserving evidence that will be admissible in a court of law, and provides the first step in our forensic framework. Second, ***Autopsy*** provides a library of tools that are designed to investigate and analyze disk images, including recovery of lost, deleted, or hidden (steganographic) data. Previously available as the GUI interface of The Sleuth Kit (TSK) project, Autopsy performs time line analysis, hash filtering, keyword search, extract web artifacts and much more. These functions are useful in the second and third tiers of our framework. Third, the ***F****orensic* ***R****egistry* ***E****ditor* (***FRED***) is a cross-platform registry hive editor including a hex viewer and data interpreter. FRED uses four different hive files: NTUSER.dat, SAM, SOFTWARE, and SYSTEM to generate a report of the interpreted data. These reports are used for the second and third tiers of the forensic analysis framework, because they show the last actions performed on the system. FRED also supports the basic forensics principle of not leaving any trace on the system being analyzed which could cause results to be altered. Normal forensic procedure would require imaging these files to avoid altering them - however hive files running on a system cannot be copied. Running FRED as a stand-alone tool on the operating system would also run the risk of affecting data and damaging files. Both of these concerns are alleviated by running FRED under the CAINE operating system. Fourth, ***PhotoRec*** is an open source data carver, used to recover deleted, lost, or damaged files or compare file checksums. PhotoRec is useful in the second through fourth tiers of our forensic architecture.

## 3. Experimental Results

We experimentally validated the use of these four tools in forensic analysis of a suspect disk drive. Tier 1 data recovery was done using Guymager to image the disk, and Tier 2-3 analysis was performed using Autopsy, Fred, and PhotoRec. Built-in features for all these tools (*particularly PhotoRec*) facilitate Tier 4 documentation throughout the forensic process. We performed disk imaging using Guymager, which provides a list of available mounted disks and details such as the disk size and serial number as shown in Figure 1. From this list, it is possible to right click on any image and select "Acquire Image". The proper image directory location has to be chosen, and information such as case number, examiner, description, and image filename may be entered in the "Acquired Image" window. Clicking Start will begin the process of creating the disk image within the specific directory that was assigned. Our testing for 25 different disk images was completely consistent with other accepted methods for creating forensically sound disk images, with no observed errors. For comparison, we used the bit-by-bit copying techniques available in practically every Linux/UNIX distribution; in particular, Kali Linux uses a version developed by the Department of Defense Digital Computer Forensics Laboratory (DCFLDD). The syntax is:

**dd if=<source> of=<destination> bs=<byte size>**

The resulting disk image is then imported to Autopsy for analysis. Autopsy was originally created by Brian Carrier [13, 15] as a graphical interface for The Sleuth Kit (TSK) and other digital forensic tools. The architecture is based on modular plug-ins, which allows the selective incorporation of file analysis routines created by third parties. It is recommended to disable JavaScript for these experiments, as it is not required to run Autopsy and may inadvertently modify files, corrupting the chain of custody. Some key features of Autopsy used in forensic analysis are summarized in Table 1.

Table 1 – Key Forensic features available in Autopsy

| | |
|---|---|
| Multi-User Cases | Enables collaboration with teams of forensic examiners on larger cases |
| Timeline Analysis | Graphical interface displays system events to help identify causal relationships |
| Web Artifacts | Extracts user activity from common web browsers |
| Registry Analysis | Uses RegRipper tool to analyze hives, including recently accessed documents |
| LNK File Analysis | Identifies shortcuts and recently accessed documents |
| Email Analysis | Parses MBOX format messages |
| EXIF Analysis | Extracts geolocation and camera information from JPEG files |

Autopsy can be accessed through CAINE in the same way as Guymager. Selecting Autopsy from the menu options automatically opens a terminal shell illustrated in Figure 1, with root authority. Users are prompted to open the GUI in a web browser; by default Autopsy installs to a local web server (localhost) accessible via port 9999.

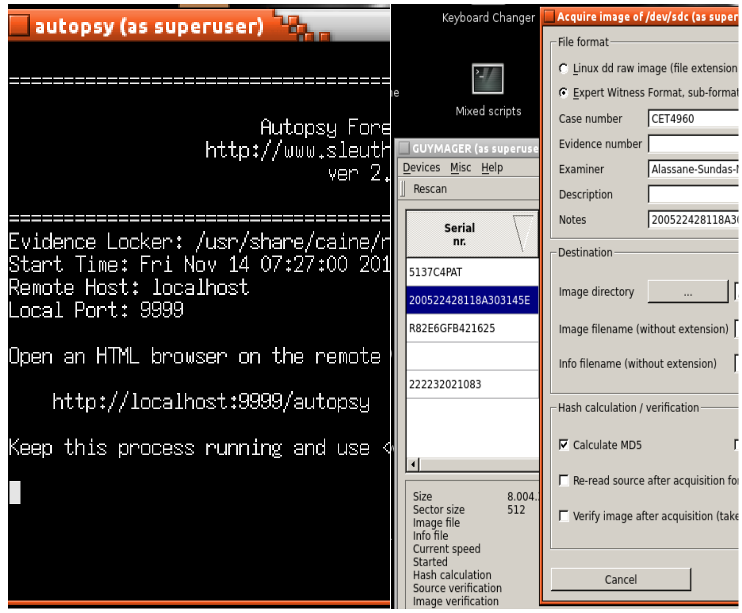


Figure 1 – (L) Autopsy terminal emulator shell (R) screen shot of Guymager disk imaging

The terminal shell will prompt the user to enter the URL provided into a browser to open up the Autopsy GUI, shown in Figure 2. From here the user can select an existing case, creates a new case, or ask for help. When the user creates a new case, they have to provide a name for the case along with up to six names of the investigators (*most practical forensic analysis is conducted by a team of experts*). There is also an option to describe the case. This prompt automatically creates a report as the investigation proceeds. As with all legal cases, it is critical to insure that the disk image has not been tampered with from the time it was captured until the time of the trial. Guymager follows the best practice of generating a hash for the captured disk image. Autopsy provides the option of importing this hash (recommended), or calculating a new hash before the start of image analysis if the hash does not already exist. This illustrates how Tier 4 requirements are enabled throughout the investigative process. In order to analyze an image, the user needs to create a new case for each image as shown in Figure 2.

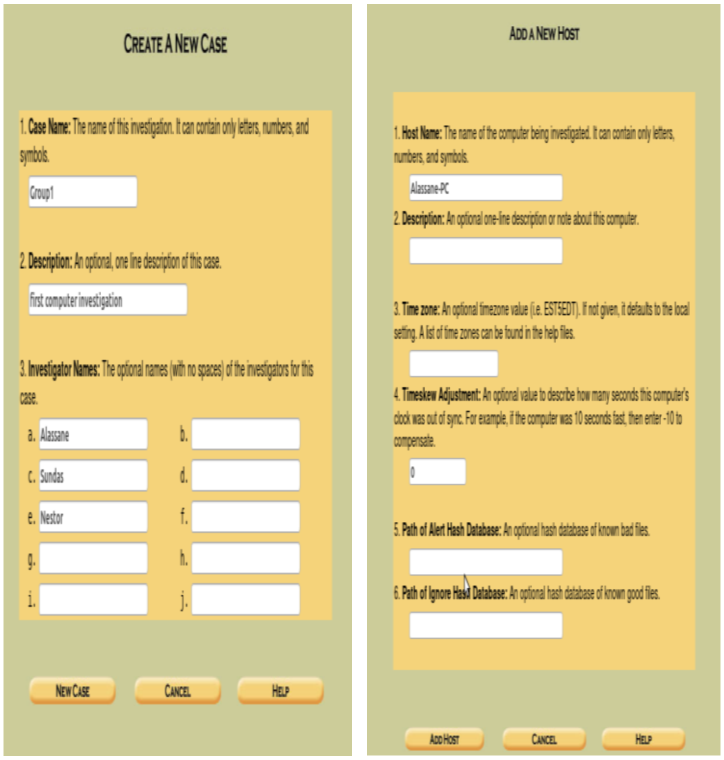


Figure 2 - (L) Creating a New Case (R) Creating a New Host

After creating the case, the user is required to specify a host. Other information are optional, however we found it helpful to include the paths for the Alert and Ignore data bases. These paths can help filter the results, and files that have been tampered with can be identified in this manner. The main part of the analysis occurs after adding the host, when the user is prompted to add an image file. The proper file type and format must be selected in order for Autopsy to analyze the image. The Hash databases option provides additional information, including options for the file activity time line, notes, and event sequencer. This allows the user to create time lines and notes for the investigation. Once the host and case are created, the user will be prompted to add the host to the case. Then an image can be selected for analysis, typically from the same location used when the file was created using Guymager. For our testing, we selected the type of image file as disk and the import method as Symlink. When the image is added, a file summary is displayed including the mount point and type of file system recognized on the image. Examples of mounted images are shown in Figure 3.


Figure 3 – (L) Mounted images within Autopsy (R) keyword search of raw data

The user can enter a keyword, string or expression that can be searched within the image. The type of data can also be selected, such as ASCII or Unicode. In this case "strings" was entered as the keyword to search. A set of predefined search options are also available. After clicking "Search", a new window shows that in our case, 92 occurrences of "strings" were found with the selected settings (see Figure 4). Subsequent views expand on this screen showing all of the occurrences within that unit. These can be viewed in both HEX and ASCII.

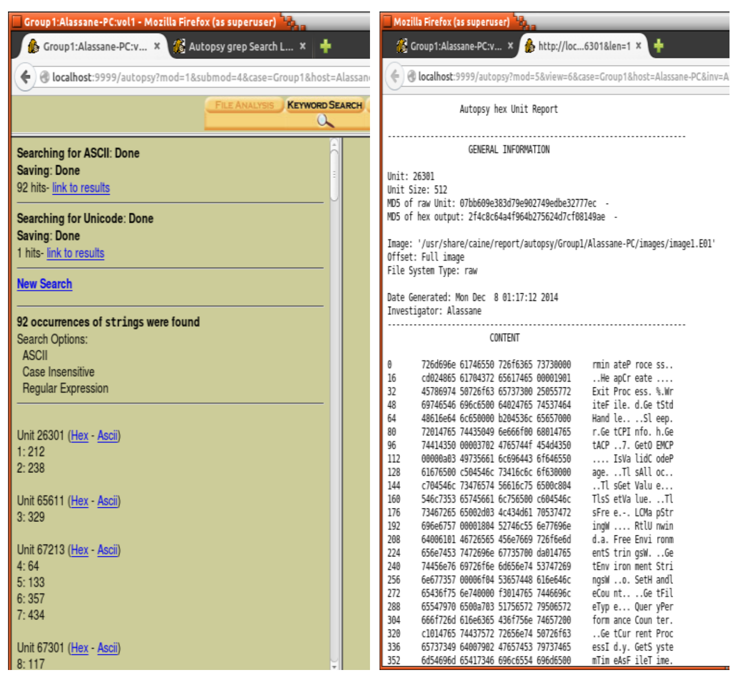


Figure 4 – (L) search data results (R) Autopsy Hex report

By selecting the "File Analysis" option, the analyzer generates a list of files within the disk image (*file browsing mode*). This allows the access of files and directory content including the date the files were created, last date accessed, size of the files, and other meta-data (Figure 5 shows the files within the C:/ directory). It is also possible to select a view of "Deleted Files" in the image, along with information on when they were created or last used (as seen in Figure 6). Figure 7 shows that files that in red are in data recovery mode. These files can be opened to view their contents (also seen in Figure 7). Although the file contents are typically encoded, Autopsy can scan and analyze the image files. The forensic analyst can explore the contents of the files using File Analysis. They can even search for files that have specific keywords or names that have been deleted. This action cannot be performed on a typical general purpose operating system. As seen in Figure 4, reports of the data can also be generated in ASCII or HEX formats. Forensic analysts can use these reports to determine if there are any dangerous/ suspicious files/directories within the disk whether they are currently on the system or have been deleted. Our testing of 25 different files revealed no detectable errors when using this approach.

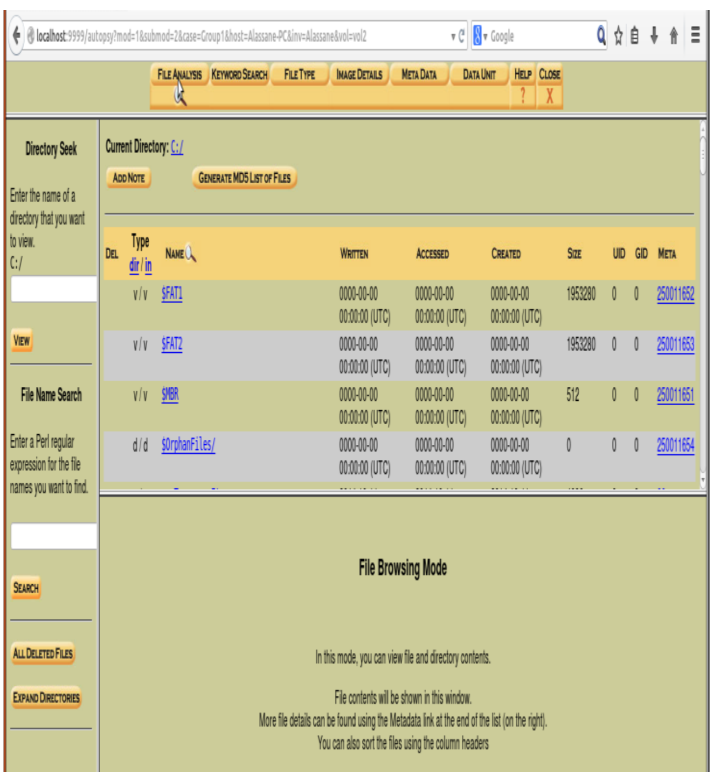

Figure 5 – Experimental scan of C: directory files

Figure 6 – View of deleted files

Figure 7 - View of files in recovery mode

To successfully analyze the hive files on this system without harming them, the user has to boot the CAINE operating system from a CD. The hard disk and any other USB storage connected to the host computer will be mounted. From there the user can double-click the mounted hard disk and explore the files of the system without any restrictions. It is not possible to tamper with the hard disk since it is read only. The first registry of hive files we scanned was NTUSER.dat. This file was found by going to the mounted hard disk and then USERS \Sundas (or hostname). After copying the file onto the desktop as an additional precaution against tampering, the CAINE GUI was used to select Menu > Forensic Tools > Fred. This enables registry and hive files to be opened, such as the NTUSER.dat file including file keys and hex views illustrated in Figure 8.

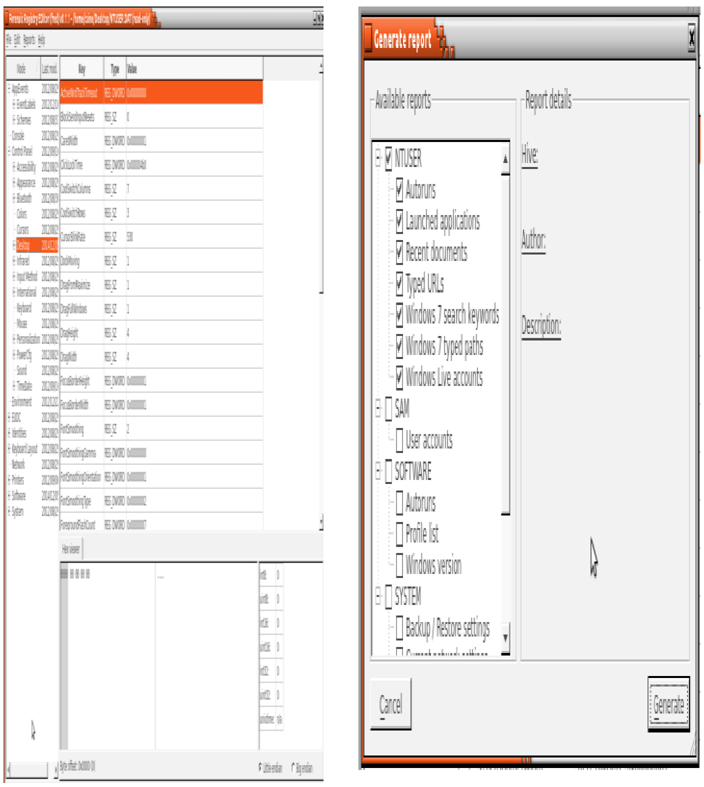


Figure 8 – (L) registry file view (R) available reports and translated data

Figure 8 shows the available reports and translated data that can be generated. For example, the NTUSER gives information on auto runs, launched applications, recent documents, typed URLS, Windows 7 searched keywords, Windows 7 typed paths and Windows live accounts. The resulting report file examples are shown in Figure 9 including the last run dates of those applications on the right. If a user even deletes their history, by running this scan, analyzers can determine which applications are being used within the system and when they were being used last. We confirmed correct operation of these functions with no errors after 25 independent trials.

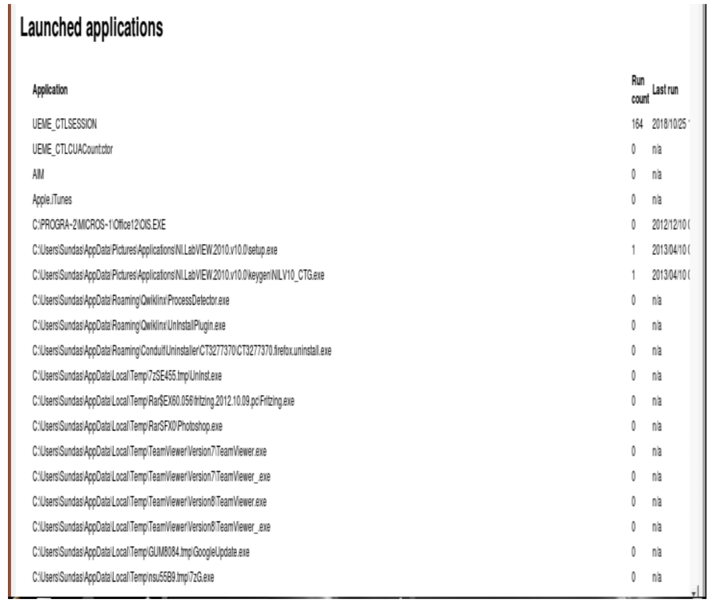


Figure 9 – file report example

Figure 10 shows all of the recent documents that have been opened along with typed URLs and typed keywords. In 25 trials, our system was able to correctly identify the complete hive history with no observable errors when compared with alternative tools such as RegEdit

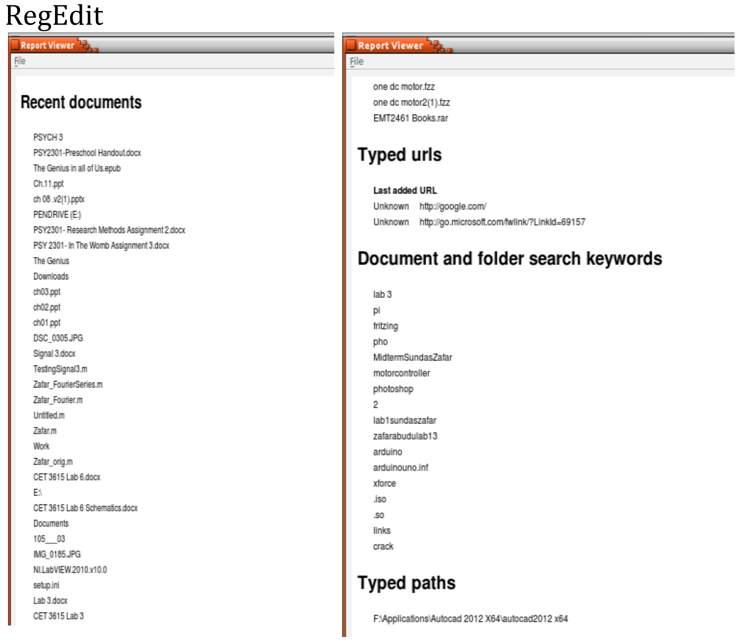


Figure 10 – recently opened documents, URLs, and keywords

We can also analyze a SAM file, as shown in Figure 11, which provides information related to user accounts on the system. Figure 12 shows which users are on the system including their name, the last time the user has been logged in, account expiry, failed logins, the last time a password was changed, and even password hints.

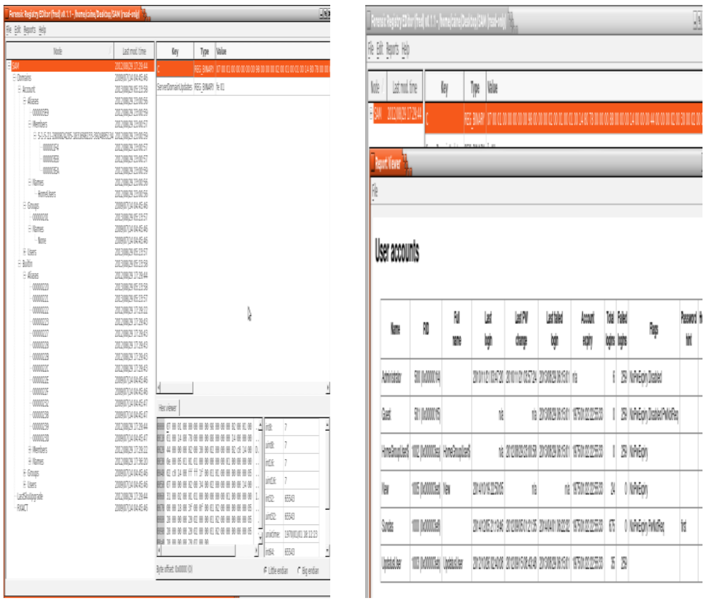

Figure 11 – (L) SAM file analysis (R) user account history

The information given by SAM can help computer forensic analyzers to determine whether someone else is trying to physically log into user accounts. If an individual does not use their computer during a certain time, they can determine from using Fred that someone is logging onto their username at odd hours. From there, further research can be done to determine which files have been accessed. A similar process can be used for the Software hive and other system archives.

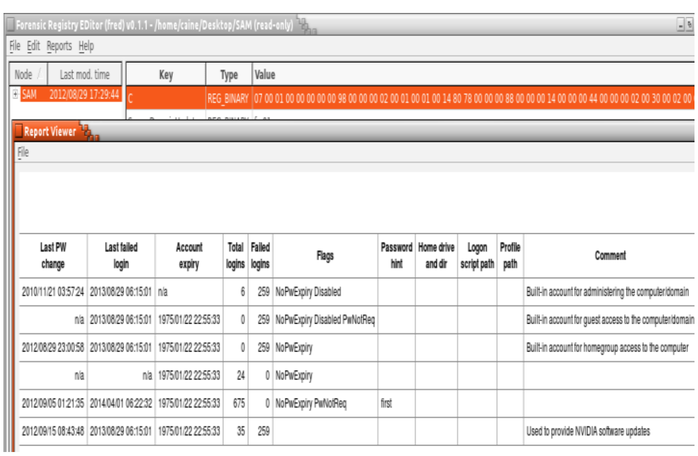


Figure 12 – example report generation screen shot

While our approach to the NTUSER file is consistent with our Tier 3 analysis objectives, we also want to get translated data by generating a report. Figure 12 shows the available reports that can be generated and the information each file provides. The NTUSER gives information on auto runs, launched applications, recent documents, typed URLS, Windows 7 searched keywords, Windows 7 typed paths and Windows live accounts. The generate button will create reports from the NTUSER hive file. If a user even deletes their history, analyzers can use this scan to determine which applications were used and when they were last used. The NTUSER report shows all of the recent documents that have been opened along with typed URLs and typed keywords. If there are unknown URLs or files that are being accessed on the system unknowingly, then the NTUSER file will provide pertinent information. Autopsy will display the current network settings, showing information on each of the five adapters connected to the device, including IP address and subnet mask. The SYSTEM file shows every USB drive ever connected to the computer. The name of the storage device, vendor name, unique ID, class and mount point are given. This type of information is important because some users may be secretly trying to steal information by connecting a USB and copying files.

To further address Tier 4 requirements, the PhotoRec application was enabled to run behind the scenes to scan and document properties of a Windows host. First the CAINE ISO must be downloaded from its open source site (caine.org) and burned into a DVD or USB media (*USB media must be made bootable*). Since CAINE is being run live from a DVD, data cannot be written to it. Therefore, an external hard drive must be connected which is where the recovered files are stored. Guymager can be used to create an image which can be read and analyzed by PhotoRec. Selection of the recovered drive is shown in Figure 13.


Figure 13 – selection of a recovered drive

Figure 14 shows the selection of the drive that will be scanned by PhotoRec, then the program is asking the user to define the space or format of the system to be scanned in order to specify its scan to the particular file system, then illustrating scan results.

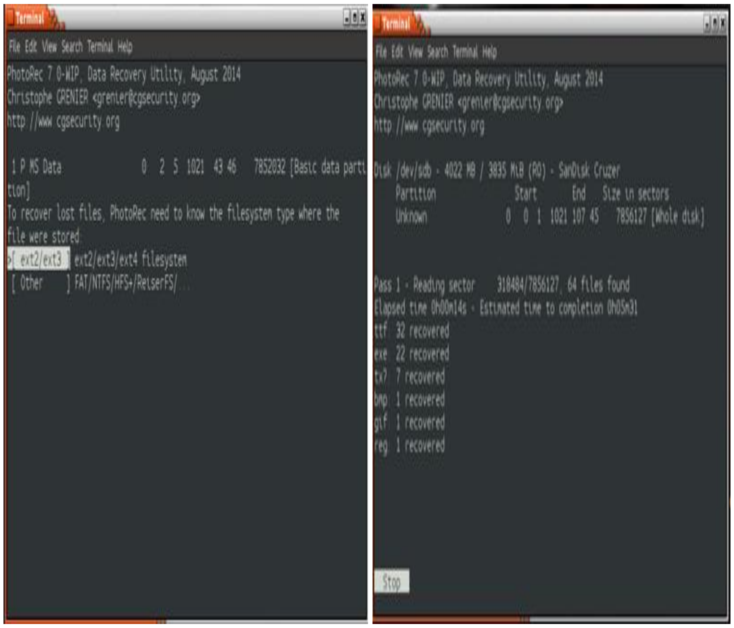


Figure 14 – (L) drive selection (R) space definition and scan results

PhotoRec works by searching data clusters, so the time required for analysis depends on the size of the drive. The screen shot of Figure 14 shows ProtoRec in the process of detecting files which had previously been deleted by the FAT file system. Once this process is completed, the recovered files are moved to a destination specified by the user. We validated that this process generated consistent and correct results for at least 50 scan attempts on various Windows systems.

## 4. Conclusions

By experimenting with tools from the open source CAINE Linux distribution, a documented process was developed which maps to the proposed theoretical four tier model of Forensic Analysis. We conducted between 25-50 independent trials using typical forensic techniques, and confirmed that these tools had an error rate equivalent to previously establish tools (i.e. no errors were observed during the course of these tests). The results of this software validation testing supports our goal of establishing that CAINE complies with the Daubert Standard of forensic evidence reporting. Another of our goals was to recommend a preferred order for applying the Guymager, Autopsy, Fred and PhotoRec tools. As a Tier 1 tool, Guymager creates images of disks for Forensic Analysis, at Tier 2, Autopsy analyzes the images and allows data recovery, at Tier 3, FRED focuses on scanning and editing registry hives, while at Tier 4 PhotoRec recovers lost or missing data. This combination affords some advantages in the forensic investigation process. For example, unless the user already has an imaged disk, Autopsy as a stand-alone tool requires the extra step of creating an image before scanning for results. Guymager performs this function as an integral part of CAINE. While Autopsy provides useful results, it requires the overhead of creating a case and adding a host before attempting to recover the contents of a disk. By contrast, PhotoRec directly scans the disk for file recovery and may detect some files missed by Autopsy. In conducting forensics research with the CAINE integrated toolkit, our main obstacle was gaining access to the right type of file for each tool to analyze, since all four tools are very particular about their input file type. Although the CAINE operating system runs within a VMware workstation, an error would occur when the image/file to scan was chosen, even if the file format was correct. Care must be taken in following the mounting directions. It is not recommended to use CAINE on the host operating system, which can result in the actual data being corrupted. Further, there were some issues with creating bootable media by writing the ISO to a USB stick. These issues did not impact our testing against the five long-term factors related to the compliance of open source forensic tools.

## References


[1] Juniper Research Report. (2018, December 6). "The future of cybercrime and security: financial and corporate threats and mitigation" [Online]. Available: https://www.juniperresearch.com/press/press-releases/cybercrime-cost-businesses-over-2trillion

[2] Daubert vs. Merrell Dow Pharmaceuticals, Inc., 509 U.S. 579 (1993)

[3] N. Bassetti. (2015, August 20). *CAINE Live USB/DVD*. [Online]. Available: https://www.caine-live.net/index.html

[4] U.S. Department of Defense. (2018). Defense Forensics Science Center standards [Online]. Available: https://www.cid.army.mil/

[5] Digital Forensics Research Workshop (DFRWS) standards. (2018). [Online]. Available: https://www.dfrws.org

[6] Scientific Working Group on Digital Evidence (SWGDE). (2018). [Online]. Available: https://www.swgde.org
[7] C. Eastton, *System Forensics, Investigation, and Response*, 3rd edition (2018) Jones and Bartlett
[8] *Frye v. United States*, 293 F. 1013 (D.C. Cir. 1923)
[9] C. Funk. (2018, December 8). "Daubert vs Frye: a national look at expert evidenciary standards" [Online]. Available: https://www.theexpertinstitute.com/daubert-versus-frye-a-national-look-at-expert-evidentiary-standards/
[10] (2018, December 6). "General Principles of Software Validation: Final Guidance for Industry and FDA Staff" [Online]. Available: https://www.fda.gov/regulatory-information/search-fda-guidance-documents/general-principles-software-validation
[11] P. Kanellis, E. Kiountouzis, N. Kolokotronis, and D. Martakos, editors, "Digital crime and forensic science in cyberspace", Idea Group Publishing, Hershey, PA (2006)
[12] C. Altheide and C. Miller, "Validating Proprietary Digital Forensic Tools: A Case for Open Source" (2011)
[13] B. Carrier. (2018, December 6). "Open source digital forensic tools" [Online]. Available: http://www.digital-evidence.org/papers/opensrc_legal.pdf
[14] Kenneally, E. E. (2001). Gatekeeping Out Of The Box: Open Source Software As A Mechanism To Assess Reliability For Digital Evidence. *Virginia Journal of Law and Technology* [Online]. Available: http://www.vjolt.net/vol6/issue3/v6i3-a13-Kenneally.html#_ednref82
[15] B. Carrier, "Defining digital forensic examination and analysis tools using abstraction layers", Int. Journal of Digital Evidence, Vol. 1, no. 4 (Winter 2003) [Online]. Available: https://www.utica.edu/academic/institutes/ecii/publications/articles/A04C3F91-AFBB-FC13-4A2E0F13203BA980.pdf
[16] B. Nikkel. (2006, May). "The role of digital forensic with a corporate organisation", Proc. IBSA Conf. [Online]. Available: http://digitalforensics.ch/nikkel06a.pdf
[17] S. Perumal "Digital forensic model based on Malaysian investigation process", Vol. 9 no. 8 (2009) [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.173.6875&rep=rep1&type=pdf
[18] J. Ashcroft. (2001). Electronic Crime Scene Investigation: A guide for first responders [Online]. Available: https://www.ncjrs.gov/pdffiles1/nij/187736.pdf
[19] I. Ademu, C. Imafidon, and D. Preston, "A new approach to digital forensic models for digital investigation", Int. Journal of Advanced Computer Science and Applications, vol 1, no 12, p. 1-4 (2011)
[20] C. DeCusatis, Aparicio Carranza, Alassane Ngaide, Sundas Zafar, and Nestor Landaez, "An open digital forensics model based on CAINE", Proc. 15th IEEE International Conference on computer and information technology (CIT 2015), October 26-28, Liverpool, U.K.