Open Source Digital Forensics Tools

The Legal Argument[1]

Brian Carrier
carrier@cerias.purdue.edu

Abstract

This paper addresses digital forensic analysis tools and their use in a legal setting. To enter scientific evidence into a United States court, a tool must be reliable and relevant. The reliability of evidence is tested by applying "Daubert" guidelines. To date, there have been few legal challenges to digital evidence, but as the field matures this will likely change. This paper examines the Daubert guidelines and shows that open source tools may more clearly and comprehensively meet the guidelines than closed source tools.

1 INTRODUCTION

The debate between open source and closed source software has historically been waged on issues such as philosophy [5], security [18], reliability [11], and support. Each side has arguments that resonate with different user populations and there seems to be no clear winner.

This paper addresses software that is used for digital forensic analysis and examines the role of open source. These tools are used to analyze digital data and often find evidence that someone did or did not commit a crime. As the tool output may be evidence introduced in a court trial, it must meet certain legal requirements. This paper examines the legal requirements of digital forensic tools and addresses how open source tools satisfy them.

Digital forensics has existed for as long as computers have stored data that could be used as evidence. For many years, digital forensics was performed primarily by government agencies, but has become common in the commercial sector over the past several years. Originally, much of the analysis software was custom and proprietary and eventually specialized analysis software was made available for both the private and public sectors. Recently, open source alternatives have been developed that provide comparable features.

---

[1] I originally published this paper in October 2002 as an @stake Research Report. The original report can be found at www.atstake.com/research/reports/acrobat/atstake_opensource_forensics.pdf. This version was published in September 2003 and has updated URLs and a different format.

The first part of this paper provides a brief overview of how digital forensic tools are used, followed by the legal guidelines for proving the reliability of scientific evidence. Those guidelines are then addressed with respect to open source software. Finally, a balanced solution is proposed that allows commercial software companies to remain competitive by keeping interface-related code closed source while having the extraction code open source and available for publication and peer review. The solution allows users to have commercial support and the freedom to choose a tool based on interface and ease of use.

As a disclaimer, I am a researcher and software developer and not a lawyer. Therefore, this should not be taken as legal advice. In addition, I develop open source digital forensic analysis tools [3].

2 DIGITAL FORENSIC ANALYSIS

In general, the goal of digital forensic analysis is to identify digital evidence for an investigation. An investigation typically uses both physical and digital evidence with the *scientific method* to draw conclusions. Examples of investigations that use digital forensics include computer intrusion, unauthorized use of corporate computers, child pornography, and any physical crime whose suspect had a computer. At the most basic level, digital forensics has three major phases:

- Acquisition
- Analysis
- Presentation

The *Acquisition Phase* saves the state of a digital system so that it can be later analyzed. This is analogous to taking photographs, fingerprints, blood samples, or tire patterns from a crime scene. As in the physical world, it is unknown which data will be used as digital evidence so the goal of this phase is to save all digital values. At a minimum, the allocated and unallocated areas of a hard disk are copied, which is commonly called an *image*.

Tools are used in the acquisition phase to copy data from the suspect storage device to a trusted device. These tools must modify the suspect device as little as possible and copy all data.

The *Analysis Phase* takes the acquired data and examines it to identify pieces of evidence. There are three major categories of evidence we are looking for:

- Inculpatory Evidence: That which supports a given theory
- Exculpatory Evidence: That which contradicts a given theory
- Evidence of tampering: That which can not be related to any theory, but shows that the system was tampered with to avoid identification

This phase includes examining file and directory contents and recovering deleted content. The scientific method is used in this phase to draw conclusions based on the evidence that was found.

Tools in this phase will analyze a file system to list directory contents and names of deleted files, perform deleted file recovery, and present data in a format that is most useful. This phase should use an exact copy of the original, which can be verified by calculating an MD5 checksum. It is important that these tools show all data that exists in an image.

Regardless of the investigation setting (corporate, federal, or military), the steps performed in the acquisition and analysis phases are similar because they are dominated by technical issues, rather than legal. The *Presentation Phase* though is based entirely on policy and law, which are different for each setting. This phase presents the conclusions and corresponding evidence from the investigation. In a corporate investigation, the audience typically includes the general counsel, human resources, and executives. Privacy laws and corporate policies dictate what is presented. In a legal setting, the audience is typically a judge and jury, but lawyers must first *evaluate* the evidence before it is entered. In order to be admissible in a United States legal proceeding, scientific evidence must pass the so-called "Daubert Test", which stems from the U.S. Supreme Court's ruling in Daubert vs. Merrell Dow Pharmaceuticals (1993) [17]. This paper will address the requirements of the Daubert Test.

3 ADMISSIBILITY OF DIGITAL FORENSIC EVIDENCE

3.1 Overview

To be admissible in a United States court, evidence must be both relevant and reliable. The reliability of scientific evidence, such as the output from a digital forensics tool, is determined by the judge (as opposed to a jury) in a pre-trial "Daubert Hearing". The judge's responsibility in the Daubert Hearing is to determine whether the underlying methodology and technique used to identify the evidence was sound, and whether as a result, the evidence is reliable. The Daubert process identifies four general categories that are used as guidelines when assessing a procedure:

- **Testing:** Can and has the procedure been tested?
- **Error Rate:** Is there a known error rate of the procedure?
- **Publication:** Has the procedure been published and subject to peer review?
- **Acceptance:** Is the procedure generally accepted in the relevant scientific community?

The Daubert Test is an expansion of the Court's prior approach to the admissibility of scientific evidence. Previously, under the "Frye Test", courts placed responsibility of identifying acceptable procedures on the scientific community using peer-reviewed journals. However, as not every field has peer-reviewed journals, the Daubert Test offered additional methods of testing the quality of evidence.

Each guideline will now be addressed in more detail with respect to digital forensics. The guidelines will be examined for both data acquisition tools and analysis tools. Currently, the majority of digital forensics involves the acquisition of hard disks and analysis of file systems. Therefore, special attention will be paid to these tools and the

procedures for copying data from one storage device to another and extracting files and other data from a file system image.

3.2 Testing

The testing guideline identifies if a procedure can be tested to ensure it provides accurate results and if so, has it. This is a complex problem with digital forensics because computers themselves are complex. Two major categories of tests must be performed on the tool output:

- False Negatives
- False Positives

The false negative tests will ensure that the tool provides all available data from the input. For example, when a tool lists the contents of a directory then all files should be displayed. Similarly, if the tool is capable of listing deleted file names, all deleted names should be displayed. An acquisition tool must copy all data to the destination. With digital forensic tools, this category is the easiest to test and most formalized testing is of this type. Known data is planted on a system, it is acquired, analyzed, and it is verified that the data can be found.

The false positive tests will ensure that the tool does not introduce new data to the output. For example, when a tool lists the contents of a directory then it does not add new file names. This category is more difficult to test. A common technique used to verify that a tool is not introducing data is to validate the results with a second tool. While this is a good practice, it does not replace the need for a formalized test methodology.

The National Institute of Standards and Technology (NIST) has a dedicated group working on Computer Forensic Tool Testing (CFTT) [14]. They develop test methodologies for a category of tools and conduct tests using specific input cases. The specification for disk imaging tools was published [15] and the tests were conducted on several different tools. Currently, the results have not been fully disclosed. In addition, they have not created a test methodology for analysis tools yet.

Prior to the NIST CFTT results, one of the only public comparisons of forensic tools was published in SC Magazine [8]. The test planted data in different locations of different disk types, acquired the disk, and tried to find the planted data. While these evaluations are useful when considering which tool to purchase (and they even found several bugs), they should not be the only ones available for a legal process.

The proper way to test forensic tools is by using an open method. Requirements must be created for each tool type and corresponding tests must be designed that enforce the requirements. Using specific test conditions for all tools can only go so far at catching bugs because of the large number of possible tests. For example, designing a comprehensive set of test requirements for all NTFS file system analysis tools is a massive task, especially because the file system structures are not public. Based on time requirements, it is unlikely that a test suite can be developed that can validate every possible file system configuration. In fact, the testing requirements are likely to be stricter for digital forensic analysis tools then the original application or operating system tests. The analysis tool must handle every possible condition; otherwise a suspect could

potentially create a condition that would hide data from the investigator. In general, the original application only has to test that it can handle every condition that it can create.

Even though a standard testing methodology has not been created, bugs in today's closed and open source applications are being identified by investigators in the field and reported to the vendor. A common argument against open source applications is that people with malicious intent can find flaws in the source code and exploit them without publishing the details. While this scenario is possible, it is not unique to open source applications. Flaws and bugs are found in both closed and open source applications, so it is just as probable that a malicious person could exploit a closed source application. The long-term solution is to have a comprehensive test methodology to decrease the total number of flaws so that the chances of a malicious person exploiting them are decreased.

Having access to a tool's source code will improve the quality of the testing process because bugs can be identified through a code review and by designing tests based on the design and flow of the software. Experienced and unbiased experts should conduct these tests and all details should be published. At a minimum, closed source tools should publish design specifications so that third parties, such as NIST CFTT, can more effectively test the tool's procedures.

3.3 Error Rates

The error rate guideline identifies if there is a known error rate of the procedure. Digital forensic tools typically process data through a series of rules. The developers of the original application being analyzed designed these rules. For example, a file system analysis tool uses the specifications of a file system. If the specification is public, then there should be no errors in the tool except programming mistakes. If the specification is not public, NTFS for example, then there could be errors because the specification is not fully understood. This is similar to the testing techniques associated with natural systems such as DNA tests or fingerprints, which have an error rate that is based on how the test was conducted.

As discussed in [1][2], two categories of errors can exist in digital forensic tools, Tool Implementation Error and Abstraction Error. *Tool Implementation Error* is from bugs in the code or from using the wrong specification. An *Abstraction Error* is from the tool making decisions that do not have a 100% certainty. This typically occurs from data reduction techniques or by processing data in a way that it was not originally designed for.

It is relatively easy to give each procedure an Abstraction Error value and, as in other areas of science, this value will improve with research. It is more difficult to assign a value for Tool Implementation Error. As was recommended in [1][2], an error rate could be calculated for each tool based on the number and severity of bugs. To maintain such a value, would require access to the bug history of a tool. This is relatively easy for open source tools because even if the bug is not documented, the latest source release can be compared with the previous one to find out which code changed. The error rate would be very difficult to maintain with closed source applications because if the bug was never made public, it could be quietly fixed and not added to the error rate. In addition, the error rate would also be difficult to start calculating because commercial tools are driven

by revenue and volume of sales. Publishing error rates would be a guarded topic for those who fear sales loss because of it.

Since a formula for calculating an error rate has not been proposed, market share has been used as a metric since a tool with a high error rate would not be purchased and used [7]. At a minimum this maybe true, but a more scientific approach should be taken as the field matures. Sales figures do not show how often a tool is used or the complexity of the data it is processing. An error rate must account for both simple analysis scenarios and complex ones where the suspect has tried to hide data from the tool.

To calculate an error rate we must first develop the testing methodology required by the first guideline. Open source (or closed source/documented design) tools allow a testing methodology to be created more easily. Furthermore, it is much more difficult for an open source application to hide its history of bugs and errors.

3.4 Publication

The publication guideline shows that the procedure has been documented in a public place and has undergone a peer review. Under the earlier Frye test, this was the main condition for evidence admission. In the area of digital forensics, it is only recently that a peer-reviewed journal has existed [9] and thus far it has not covered tool procedures.

Before the International Journal of Digital Evidence, technology magazine articles were used to show publication [7]. One article [6] states that the tool has widespread usage and it lists some of the tool's features. This type of publication may address the high-level procedures of disk acquisition and analysis topics, but it does not address the technical procedures used to extract the evidence.

For a file system analysis, the procedures that need publication are those that are used to break the one large file system image into the, sometimes, thousands of files that users create in folders and directories. Some file systems have a published detailed specification, such as FAT [13], yet others, such as NTFS, do not. It is only from efforts in the Linux community that detailed NTFS structures are publicly known [12]. It is crucial that a tool publishes the procedures used to process a file system type, especially undocumented ones.

Furthermore, most digital forensic file system analysis tools show the files and directories that were recently deleted and, sometimes, can recover them. These tasks were not part of the original file system specification and therefore there is no standard method of performing them. Deleted file names are found by processing unused space and finding data that meets certain sanity check requirements. If the sanity checks are too strict, some deleted names will not be shown and evidence cannot be found. If the requirements are too weak, then erroneous data will be shown. The details of this process must be published so that an investigator can identify how the procedures are being performed.

The FBI's forensic journal published a document on the use of digital photography in 1999 [16]. In the *Software Guidelines* section, a legal note is made:

> Manufacturers of software used for image processing may be required to make the software source code available to litigants, subject to an appropriate protective order designed to protect the manufacturer's proprietary interests. Failure on the part of the manufacturer to provide this information to litigants could result in the exclusion of imaging evidence in court proceedings. This should be considered when selecting software.

This statement shows that software developers must be willing to release their source code if it is used to generate evidence. If a developer is unwilling to do so, then it should be known ahead of time so that it can be a factor when purchasing an analysis tool. If the courts allow the source code to be reviewed by an expert witness but not disclosed, then this guideline can be satisfied if there is a generally accepted technique for data processing. The expert witness can compare the source code with the accepted procedures and verify that they are properly implemented.

The publication guideline is very important and the one that is most lacking in digital forensic analysis. Little has been published on deleted file recovery and file system analysis. At a minimum, closed source tools should publish a design specification that documents the procedures and details of their analysis. Open source tools disclose all of their procedures through source code and allow one to verify that the tool is indeed following the published process and not publishing only the minimum required. In addition, open source tools should publish the procedural details in a language other than just source code.

3.5 Acceptance

The acceptance guideline is a framework for the associated scientific community to evaluate published procedures. For this guideline to be assessed, published procedures are required. Closed source tools have previously responded to this guideline by citing the large number of users they have [7]. Acceptance of a tool is different than acceptance of a procedure. If there are few tool options that perform a procedure and none of them have published procedure details or major flaws, then the selection choice will likely be based on non-procedural factors such as interface and support.

Until the procedural details are published and become a factor when purchasing digital forensic analysis tools, the size of the user community is not a valid measure of procedural acceptance. Open source tools document the procedures they use by providing the source code, thus allowing the community to accept or reject them.

3.6 Federal Rules of Evidence

As documented in [7], there is debate about whether digital evidence falls under the Daubert guidelines as scientific evidence or the Federal Rules of Evidence as non-scientific technical testimony. Rule 901(b)(9) [4] illustrates that a process can be authenticated with "evidence describing a process or system used to produce a result and showing that the process or system produces an accurate result." This rule directly addresses the same topics as the publication and error rate guidelines of Daubert. Indirectly, the rule addresses the testing guideline because a testing methodology must be developed before an error rate can be calculated. Furthermore, when the courts evaluate the "evidence describing a process", they will likely be assessing if the process is

considered valid by using its acceptance in the field. Therefore, the underlying concepts of the Daubert guidelines will be applied to digital evidence regardless if it is considered non-scientific technical testimony or scientific evidence.

4 A BALANCED SOLUTION

As many of the common digital forensic analysis tools are developed with commercial interests, it is unlikely that vendors would be willing to publish all of their source code. Using the tool definitions from [1][2] though, may provide a more practical solution than 100% open source.

The papers describe two tool categories: extraction and presentation. *Extraction tools* are those that process data to extract a subset of it. For example, an extraction tool would process a file system image and output the file content and descriptive data, such as the last access time. *Presentation tools* are those that arrange the data from an extraction tool into a useful format. One tool can take on both roles or they can be separate. For example, an extraction tool can analyze a file system image and output the name and times for every file. One presentation tool could display that data sorted by directories, which is how most people view a file system. Another presentation tool could display the same data, but sorted by the Modified, Access, and Change (MAC) times to create a timeline of file activity. The same data is shown, but in a different order.

If the extraction tools are open source and the investigator has access to the output of this layer, then s/he can verify the output of the presentation tool. Therefore, the presentation tools could remain closed source, but with a published design. In addition, many new features in file system digital forensic analysis tools are based on presentation. For example, looking up a hash in a database, enterprise solutions over a network, comparing the file type with extension, and keyword searches are all actions that occur after the data is extracted from the file system image. Therefore, creating standard techniques of data extraction would not limit a software company's ability to remain competitive. User interface, features, and support will be the differentiators between vendors. In fact, this would allow vendors to focus on innovative presentation techniques for file system analysis and improve the extraction and presentation tools of less mature areas, such as networks and log reduction.

By publishing source code through open source extraction tools, the digital forensic community can examine and validate the procedures used to produce digital evidence. This model allows an accurate error rate to be calculated because all data extraction-related bug fixes would be made public. If multiple tools used the same code base for data extraction, one could develop a stable code base and test methodology fairly quickly. Furthermore, the tools would have a similar error rate because the only difference would be because of bugs in their interface and presentation of data. Therefore, vendors may be more willing to participate in an effort to calculate error rates.

This open source model is different than the one that most are familiar with. The goal of many open source projects is to have a large number of developers who can access and update the code. The goal of this project would be for easy access to the code for review, but limited access for updating. The developers would be a limited group of people and

a panel of auditors would validate all code updates. When the software and code is released, a cryptographic signature from the group would accompany it.

5 CONCLUSION

Using the guidelines of the Daubert tests, we have shown that open source tools may more clearly and comprehensively meet the guideline requirements than would closed source tools. To further the acceptance of analysis tools in a legal setting, the following steps must be taken in the future:

- Development of comprehensive tests for file system (and other) analysis tools in addition to the ones that NIST has already developed for disk imaging tools
- Publication of tool design to help create more effective tests
- Creation of a standard for calculating error rates for both tools and specific procedures
- Publication of specific procedures that a tool uses. While open source tools already publish their source code, they should also describe the procedure in words.
- Public debate on the published procedural details to ensure that they are agreed upon

Digital forensic tools are used to fire employees, convict criminals, and demonstrate innocence. All are serious issues and the digital forensic application market should not be approached in the same way that other software markets are. The goal of a digital forensic tool should not be market domination by keeping procedural techniques secret.

Digital forensics is a maturing science that needs to be continuously held to higher standards. The procedures used should be clearly published, reviewed, and debated. The availability of analysis tools to the general public has likely increased their quality and usability. The next step is to increase confidence in the tools through publication, review, and formal testing.

6 ACKNOWLEDGEMENTS

Samir Kapuria (@stake), Eoghan Casey (Knowledge Solutions), Emily Sebert (@stake), and John Tan (@stake) provided many helpful comments and feedback on this paper.

7 ADDENDUM

After initial publication of this paper, I was referred to a similar open source paper published in the Virginia Journal of Law and Technology [10]. The paper examines the reliability of software and digital evidence, but expands its scope beyond specialized forensic analysis software. This is important because the data produced by normal business applications can be introduced as evidence, server logs for example. Having

access to the source code of the software can help show the requirements needed to cause the server application to create a log entry and the procedures used to create the log.

REFERENCES

[1] Brian Carrier. Defining Digital Forensics Examination and Analysis Tools. In *Digital Research Workshop II*, 2002. Available at: http://www.dfrws.org/dfrws2002/papers/Papers/Brian_carrier.pdf

[2] Brian Carrier. *Defining Digital Forensics Examination and Analysis Tools Using Abstraction Layers*. International Journal of Digital Evidence. Vol 1, Issue 4, Winder 2003. Available at: http://www.ijde.org/docs/02_winter_art2.pdf .

[3] Brian Carrier. The Sleuth Kit and The Autopsy Forensic Browser. Available at http://www.sleuthkit.org/.

[4] *Federal Rules of Evidence*. Article IX. Authentication and Identification. Available at: http://www2.law.cornell.edu/cgi-bin/foliocgi.exe/fre/query=[JUMP:'Rule901']/doc/{@1}?firsthit

[5] Free Software Foundation. *Philosophy of the GNU Project*. Available at http://www.gnu.org/philosophy/philosophy.html

[6] Lee Garber. *EnCase: A Case Study in Computer-Forensic Technology*. IEEE Computer Magazine January 2001. Available at: http://www.computer.org/computer/homepage/January/TechNews/technews2.htm

[7] Guidance Software. *EnCase Legal Journal, Second Edition*. March 2002. Available at: http://www.encase.com/support/downloads/LegalJournal.pdf

[8] James Holley. *Computer Forensics Market Survey*. SC Magazine September 2000. Available at: http://www.scmagazine.com/scmagazine/2000_09/survey/survey.html

[9] *International Journal of Digital Evidence*. Available at: http://www.ijde.org/

[10] Erin Kenneally. *Gatekeeping Out Of The Box: Open Source Software As A Mechanism To Assess Reliability For Digital Evidence*. Virginia Journal of Law and Technology. Vol 6, Issue 3, Fall 2001. Available at: http://www.vjolt.net/vol6/issue3/v6i3-a13-Kenneally.html

[11] open source. *Open Source Case for Business*. Available at: http://www.opensource.org/advocacy/case_for_business.php

[12] Linux-NTFS Project. *NTFS Documentation*. Available at: http://linux-ntfs.sourceforge.net/ntfs/

[13] Microsoft Organization. FAT: General Overview of On-File Format, 1.08 Edition, December 2000.

[14] NIST. Computer Forensics Tool Testing. Available at http://www.cftt.nist.gov/

[15] NIST CFTT. *Disk Imaging Tool Specification, 3.1.6 Edition*, October 2001. Available at: http://www.cftt.nist.gov/DI-spec-3-1-6.doc

[16] Scientific Working Group on Imaging Technologies. Definitions and Guidelines for the Use of Imaging Technologies in the Criminal Justice System. *Forensic Science Communications*, 1(3), October 1999. Available at: http://www.fbi.gov/hq/lab/fsc/backissu/oct1999/swgit1.htm

[17] Supreme Court of the United States. *Daubert v. Merrell Dow Pharmaceuticals Syllabus*. June 28, 1993. Available at: http://supct.law.cornell.edu/supct/html/92-102.ZS.html

[18] David Wheeler. *Is Open Source Good for Security?*, Secure Programming for Linux and Unix HOWTO 13 July 2002. Available at: http://www.dwheeler.com/secure-programs/Secure-Programs-HOWTO/open-source-security.html.