The Contribution of Tool Testing to the Challenge of Responding to an IT Adversary

James R. Lyle

National Institute of Standards and Technology
Gaithersburg, Maryland, USA
jlyle@nist.gov

Abstract: The investigator is being presented with more data and more types of data to analyze. The investigator cannot work without tools. Tools are needed to acquire and analyze the data and solve the case. If the accuracy of any tools is successfully challenged in a court of law, then any results based on the tools can be suppressed and not presented. Even if an investigation is not going to any formal proceeding, the investigator wants to know the limitations of any tools used in an investigation. This can best be accomplished by an independent assessment of the tools. This paper describes the Computer Forensics Tool Testing (CFTT) project at the National Institute of Standards and Technology (NIST) in the United States. Currently, the CFTT project is developing tool specifications, test plans, test procedures, and test sets. The results provide the information necessary for toolmakers to improve tools, for users to make informed choices about acquiring and using computer forensics tools, and for interested parties to understand the tools capabilities. Our approach for testing computer forensic tools is based on well-recognized international methodologies for conformance testing and quality testing

1 Introduction

Someone notices that something is not quite right. An investigation is started. This may be something like a system security officer looking at strange behavior on a corporate system or a police officer following information developed in an ongoing criminal investigation. At some point an investigator starts looking for answers to questions like:

- What is going on?
- Is the activity deliberate?
- Who is doing it?
- How is it being done?
- Is criminal activity involved?

The answers might be found in the recovery of a single deleted file. At other times, the answers may require analysis of data from many different sources such as system log files, e-mail messages, network packets, files scattered over a large multi-national IT network and multiple live memory acquisitions. No matter how the answers are obtained, sound IT-incident management or skilled IT-forensics is a central component.

There are many challenges for the IT investigator. Some of the recent trends challenging the investigator can be summarized with one word: *more*. The investigator has more cases to solve, more data to analyze in each case, more types of data to analyze, more wide spread use of encryption and more willingness by the defense to challenge the analysis of digital data.

These challenges are difficult to address. For example, the volume of cases can be addressed by either more investigators or devoting less time per case. Availability of additional resources for more investigators is not keeping pace with the increased caseload and the amount of time devoted to each case is unlikely to decrease.

The investigator is being presented with more data and more types of data to analyze. Just a few years ago a 2GB hard drive on a single computer seemed like ample space for a single user. Currently a single user may have many computers, often with several hundred gigabytes of data directly attached to each system. In addition, there may be data stored on numerous removable devices. The problems compound and expand rapidly for investigations that move from a single user to a large corporation. The investigator cannot work without tools. Tools are needed to acquire and analyze the data and solve the case. If the accuracy of any tool is successfully challenged in a court of law, then any results based on the tool can be suppressed and not presented. Even if an investigation is not going to any formal proceeding, the investigator wants to know the limitations of any tools used in an investigation. This can best be accomplished by an independent assessment of the tools

2 Origins of CFTT

In 1999 some members of the American law enforcement community saw a need for independent assessment of software tools used in the forensic acquisition and analysis of digital data. They approached the National Institute of Standards and Technology[1] (NIST) and the result was the creation of the Computer Forensics Tool Testing (CFTT) project at NIST. The mission of the project is to establish a methodology for testing computer forensic software tools by development of general tool specifications, test procedures, test criteria, test sets, and test hardware. The results provide the information necessary for toolmakers to improve tools, for users to make informed choices about acquiring and using computer forensics tools, and for interested parties to understand the tools capabilities. The approach for testing computer forensic tools is based on well-recognized international methodologies for conformance testing and quality testing. The National Institute of Justice (NIJ), the NIST Office of Law Enforcement Standards (OLES), the Federal Bureau of Investigation, the U.S. Department of Defense Cyber Crime Center, Internal Revenue Service Criminal Investigation's Electronic Crimes Program, and the U.S. Department of Homeland Security's Bureau of Immigration and Customs Enforcement and U.S. Secret Service support CFTT.

The first issue to arise was how to organize the testing program. A common ad hoc approach is simply to "put together some test data and see what happens." This is a very appealing idea; it allows testing to get started quickly, the product user manual indicates the expected results and it is intuitively very satisfying. However, there are major short comings to this approach. It does not allow easy comparison between different tools, it is not likely to produce repeatable results for different test organizations, it fails to identify what a sound forensic tool should do and it lacks sufficient formality. On the other hand, this situation is exactly what conformance testing is designed for.

3 CFTT Conformance Testing

Conformance testing is testing to see if a software tool meets a set of requirements defined in specification for the tool. There are several discrete components required to do conformance testing in CFTT:

---

[1] **Certain trade names and company products are mentioned in the text or identified. In no case does**

such identification imply recommendation or endorsement by the National Institute of Standards

and Technology, nor does it imply that the products are necessarily the best available for the

purpose.

A **specification** giving an unambiguous list of requirements,
A **test plan** describing test cases to run, criteria for selecting test cases and criteria for conformity assessment,
A set of **test tools** for creating test data and extracting test results,
A set of **test procedures** to follow during test execution.

The CFTT testing methodology developed by NIST is functionality driven. The activities of forensic investigations are separated into discrete functions or categories, such as hard disk write protection, disk imaging, string searching, etc. A specification is then developed for each forensic function. Currently we have published specifications for digital data acquisition, software write blocking, hardware write blocking, and deleted file recovery. String searching tools will be the next category for development.

The CFTT testing process is directed by a steering committee composed of representatives of the law enforcement community. Included are the FBI, DoD, NIJ (representing state and local agencies), IRS, NIST/OLES and other agencies. Currently the steering selects tool functions for investigation and selects tools implementing a function for actual testing by CFTT staff. Specification development proceeds according to the following steps:

1. NIST, with input from forensic practitioners, develops a specification (requirements) for the selected forensic function.
2. The specification is posted to the web for peer review by members of the computer forensics community and for public comment by other interested parties.
3. Relevant comments and feedback are incorporated into the specification.
4. A test plan, including test assertions and test cases, is developed.
5. The test plan is posted to the web for peer review by members of the computer forensics community and for public comment by other interested parties.
6. Relevant comments and feedback are incorporated into the test plan.
7. Final versions of the specification and test plan are posted to the web.
8. The test tools and test procedures are developed and posted to the web.

After the specification is complete, tool testing can begin. The specification is designed for easy use. Tools can be tested against the specification by professional test labs, by tool vendors or tool users. NIST also does testing limited to tools selected by the steering committee with the results published by NIJ. The test process used by NIST is as follows:

1. Steering Committee selects tool to test
2. NIST acquires the tool to be tested.
3. NIST reviews the tool documentation.
4. NIST selects relevant test cases depending on features supported by the tool.
5. NIST executes tests
6. NIST produces test report.
7. Steering Committee reviews test report.

8. Vendor reviews test report.
9. NIST posts support software to web.
10. NIJ posts test report to web.

4 Creating Specifications and Test Plans

Currently, CFTT has published specifications for disk imaging (also called digital data acquisition), software write block tools, hardware write block devices, and deleted file recovery. The process of creating a specification is challenging. Obscure and poorly documented aspects of hardware and operating system implementations must be understood, if not for creating the specification then for development of test tools. This section discusses some examples of the challenges in each area.

4.1 Disk imaging

In order to verify the accuracy and completeness of digital data acquisition an understanding of hard drives, other storage media and operating system access methods is required. For example, to completely acquire a hard drive, the total number of data sectors must be determined. Because there are several different ways to determine the size of a drive, it is important to know what method is used by a forensic tool.

- BIOS interrupt 13, function 08h. This method returns the disk geometry in cylinders, heads per cylinder and sectors per head. The number of sectors is the product of the three values. Early versions of the DOS operating system used this method to determine disk size. Some BIOS versions would not report the real geometry, but would adjust the head and cylinder values to keep the number of cylinders less than 1024. In this case, there might be a few extra sectors left over. Disk imaging tools that use this method to determine disk size fail to acquire the left over sectors. Other disk imaging tools tried to allow for the underreporting could sometimes acquire more sectors by might still miss a few sectors.
- Direct access via drive interface. This method directly accesses the drive to determine the drive size. The details vary depending on the type of drive: ATA, SCSI, etc. For an ATA drive, this usually means using the *max user sectors* as returned by the *identify device* ATA command.
- BIOS interrupt 13, function 48h. This was an extension to the BIOS to allow access to larger drives. The value returned is usually the same as for the direct access method.

However, there are additional complications. For ATA drives, the *max user sectors* value may be changed. This can be done in two ways; either creating a *host protected area* (HPA) or a *device configuration overlay* (DCO). Also, while an HPA can be removed temporally, removing a DCO requires a permanent change to a hard drive.

The above is an example of an apparently simple requirement, determine the number of sectors on a hard drive, can become rather complicated in a specification.

4.2 Write Blocking

Write blocking, protecting a hard drive from modification during access, can be accomplished either by software tools or hardware devices. Both methods raise an important issue for tool testing. Access to a storage device is by commands sent via some protocol between the host computer and the storage device. Usually there are several commands to get data from the device (read commands), several commands to store data on the device (write commands), other commands (control and configuration), and some possible command codes that are not used. Read commands should be allowed, and write commands should be blocked. Some vendors allow the other commands, but some vendors block the other commands. It is not always clear what should be in the specification.

Another issue with write blocking was usability of the specification. As far as possible it should be possible for tools to be tested by as diverse a population of organizations as possible. In particular, forensic labs should be able to use the specifications to test the tools used by the lab in actual investigations. The most complete method for observing hardware write block device behavior is a protocol analyzer. However, it is not the most effective use of scarce resources for a forensic lab to obtain such a device. The specification for write block devices was developed such that if a protocol analyzer is available for testing it can be used to obtain detailed observations of blocker behavior, but the blocker can still be tested, although with less detailed results, if a protocol analyzer is not available.

5 Test Results

The most basic requirement for disk imaging is to produce a complete and accurate copy of the original. We have found several types of problems with imaging tools. First, some imaging tools operating in the DOS environment do not make a complete copy of the original drive. This could usually be traced to the method used by the tool to access the hard drive. If the tool accessed the drive via the BIOS rather than via a direct interface (ATA), part of the drive was sometimes omitted. This would occur if the BIOS underreported the size of the drive. In another case, using direct access to a SCSI hard drive, one tool silently omitted a significant fraction of the original from the copy while BIOS access to the same SCSI drive yielded a complete copy. This turned out to be triggered only under certain combinations of hardware. A third problem was unique to the Linux environment. In Linux, since I/O to a hard drive used a block size of 1,024 bytes, the last sector (of 512 bytes) is omitted from the copy if the drive has an odd number of sectors. The same program, **dd**, used to image a hard drive in the FreeBSD environment acquired all sectors of the drive. Another problem occurs in some Windows environments (NT, 2K and XP). An attempt to restore a copy from a drive image file to a second hard drive may be incomplete if the second drive is exactly the same size as the original. This led the vendor to change the wording of the tool documentation to emphasize that a drive slightly larger than the original should be used for the restore operation and not one of the same size as the original.

The write blocker tests have provided some interesting results.

- Some blockers cached the results of the IDENTIFY DEVICE command so that after the first time the command was issued from the blocker to the drive, the command was never issued again to the drive. Whenever the host afterwards issued the IDENTIFY DEVICE command the cached result was returned. This had the side effect that if a SET MAX ADDRESS command was used to change the total number of sectors on the device, the value for number of sectors on the drive returned for the IDENTIFY DEVICE command was not updated to reflect the new value established by the SET MAX ADDRESS command.
- Some blockers substituted a different read command for the command issued by the host.
- Some blockers allowed an obsolete low level formatting command. This command cannot modify drive contents with meaningful data but can erase the drive.

Some of the newer blocker designs and firmware will require small revisions to the specification. For example, for some blockers using a bus other than ATA to connect to the host computer, recent firmware updates automatically do a temporary removal of an HPA if present on an ATA drive. This should be easy to accommodate with small revisions to the specification.

6 Summary and Conclusions

A tool testing program for forensic tools enhances the ability of the investigator to respond.

- The test reports allow the tool user to make informed choices about acquiring and using forensic tools.
- The test reports provide the tool vendors with feedback for tool improvement.
- Results from independently tested tools are less likely to be successfully challenged in court.
- The specification process highlights technical issues where the forensic community needs to develop consensus.
- Having a common, public tool specification helps diverse organizations test forensic tools in a comparable way.
- Having a common, public tool specification provides a first step toward development of international standards for forensic tools.
- Programs similar to CFTT should be established; for example, so that more types of forensic tools could be tested.

James R. Lyle wrote his first FORTRAN program in 1968 and has been programming ever since. He received the B.S. (72) and M.S. (75) degrees in mathematics from East Tennessee State University and the M.S. (80) and Ph.D. (84) degrees in computer science from the University of Maryland College Park.

He is the project leader for the Computer Forensics Tool Testing (CFTT) project at the National Institute of Standards and Technology (NIST). Before joining NIST full time in 1993, he was an Assistant Professor at the University of Maryland Baltimore County and a Faculty Associate at NIST.

Dr. Lyle is a member of the IEEE Computer Society and the Association for Computing Machinery.