C31 Testing Computer Forensic Tools at NIST

James R. Lyle, PhD, National Institute of Standards and Technology, 100 Bureau Drive, Stop 8970, Gaithersburg, MD 20899-8970*

After attending this presentation, attendees will have an awareness of the issues in validation of forensic software used in the examination of digital data and the role of the National Institute of Standards and Technology (NIST) Computer Forensics Tool Testing (CFTT) project. This presentation provide the information necessary for toolmakers to improve tools, for users to make informed choices about acquiring and using computer forensic tools, and for interested parties to understand the tools capabilities.

Regardless as to whether it is a criminal investigation, the discovery process of civil litigation or the response to an unauthorized computer system intrusion, whenever digital evidence must be examined, the inves- tigator needs to know that the forensic software tools used in the process produce reliable, accurate, and objective results. The goal of the CFTT project at the NIST is to establish a methodology for testing computer forensic software. A methodology using the *conformance testing* model consisting of tool requirements specifications, test procedures, test criteria, test sets, and test hardware has been developed.

The test process begins with the selection by a steering committee of a specific forensic tool function for development of requirements that must be met. After specification, the requirements are published to the internet for public comment. In this way the entire computer forensic community participates in the process. After the requirements are in final form, a test plan is produced, also for public comment. At this point the steering committee selects a list of tools to test. The test plan is applied to the selected tools and test reports are published.

There are significant challenges for testing forensics tools. First, there are no standards or specifications for the expected behavior of forensic tools. Second, very arcane and often undocumented knowledge is required to understand the critical testing issues. Third, the behavior of the tools when executed in the presence of hardware errors is often relevant.

Currently specifications have been developed for acquiring digital data for examination, protecting original digital data during acquisition and recovery of deleted digital data. Test reports have been produced for the most widely used tools for acquiring digital data and for protecting original digital data during acquisition. CFTT test reports have been cited in some high profile court cases, *e.g*., Zacarias Moussaoui.

In general, the software tools used to examine digital evidence produce reliable consistent results. However, the tested tools often exhibit operational quirks that an examiner should be aware of. For example, acquiring all the digital data on a hard drive can be an issue. Depending on how the acquisition software asks a hard drive to report its size, different answers can be obtained. In addition, a hard drive may be configured to only report part of the actual drive. In other words, it is quite easy to establish hidden areas on a hard drive that can be missed in an examination if the tools used for the acquisition do not check for the hidden areas.

Testing reveals that there are sometimes significant tradeoffs in the selection of a tool. For example, software and devices for protecting original digital data usually follow one of two possible designs. Access to a digital storage device, e.g., hard drive, is provided by a set of commands through an interface. There is usually a set of possible read commands (to get data from the device), a set of write commands (to put data on the device), a set of control or configuration commands and a number of unas- signed command codes. One design is to allow only the read commands and block all other commands; the other design is to block all write commands and allow any other command. This becomes an issue when an access protocol is revised and new commands are assigned to the unused command codes.

Several lessons learned during the testing of widely used tools are dis- cussed. For example, the behavior of an acquisition tool used on an unre- liable (i.e., has bad sectors) disk is of interest. However, an unreliable disk is just that, unreliable. For testing, a *reliable bad disk* is needed. This was accomplished by using software to simulate a disk with bad sectors on a normally functioning hard disk.

Digital Evidence, Software, Validation

** Presenting Author*