Digital Investigation 8 (2012) 145–146

Contents lists available at SciVerse ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

Editorial

Editorial – Cutting the Gordian Knot: Defining Requirements for Trustworthy Tools

Digital forensics can no longer tolerate software that cannot be relied upon to perform specific functions. The root of this problem is a lack of clearly defined software requirements, which compels users and tool testers to make educated guesses and assumptions about how we expect digital forensic tools to work. This makeshift approach results in untested errors in our tools that can lead to verdicts based on incorrect information and can damage the reputation of individual practitioners and the field as a whole.

Fortunately, there is a movement toward defining clear requirements for digital forensic processes. The twist is that these requirements are not going to be defined by tool developers, but rather by the digital forensic community.

1. Where we came from

In the early days of digital forensics, tools were developed to meet immediate needs, and there was little time to follow a formal software development process. Furthermore, many of the authors of these tools were digital investigators rather than professional software developers. As a result, many of the tools that we rely on are not built on a solid foundation and are not well-documented from a software development standpoint.

Over the years, software bugs in digital forensic tools continue to emerge, misleading users into making incorrect interpretations of evidence. The release of such buggy tools effectively uses digital investigators as beta testers, and results in a bug only being fixed if it is discovered by users. This ad hoc approach to software maintenance leads to inconsistency and omissions which cannot be tolerated in a forensic discipline. In the context of a courtroom, such errors can have an adverse impact on the defendant's liberty or life, and can cause reputational harm to digital investigators.

In an effort to protect against such forensic errors and the associated reputational harm and injustice, the digital forensics community has developed mechanisms to cope with unreliable tools. Much of the training in digital forensics includes a "trust but verify" approach to using digital forensic software. In addition, organizations such as the U.S. National Institute of Standards and Technology (NIST) and the DoD Cyber Crime Center (DC3) have established formal testing programs for digital forensic tools. However, these formal test programs (and thousands of individual forensic practitioners around the world who habitually "trust but verify" the results of unreliable tools) must make educated guesses about how a given tool works, largely due to the lack of documented software requirement specifications (SRS) from tool developers.

For the future reputability of digital forensics, we cannot continue to depend on unreliable software. Recognizing this problem, the proposed ISO/IEC 27041 standard incorporates software design concepts to encourage developers to provide information about their tools that eliminate the guesswork from our test and validation efforts. However, there is much groundwork needed to support these standards and create a future of trustworthy digital forensic software.

2. ISO standards

ISO/IEC 27041 proposes a three stage process for assessing digital forensic software, using the terms Verification, Validation and Acceptance as discussed by Angus Marshall in the previous issue of this Journal (Standards, regulation & quality in digital investigations: The state we are in). This portion of the standard refers to a formal software development process. In this context, verification is a confirmation that a forensic tool conforms to its requirements specification.

A well-documented set of requirements helps tool testers know what they should be testing. Done properly, the requirements specification for a particular piece of software are written to be testable. Clearly documented, testable requirements eliminate guesswork from testing that is conducted by organizations such as NIST and DC3 and ultimately lead to the development of more robust and useful tools.

1742-2876/$ – see front matter
doi:10.1016/j.diin.2012.02.001

The Gordian knot of digital forensic software is that, in general, developers of digital forensic software are not forthcoming with their requirements specification. In fact, many digital forensic tools are not developed using a formal software development process and do not have a requirements specification document. Instead of attempting to have developers of digital forensic software provide requirement specifications in a form that is sufficient to support comprehensive testing, the emerging ISO/IEC 27041 standard puts the responsibility of setting requirements in the hands of forensic laboratories.

Once forensic laboratories establish a clear set of requirements, digital forensic tool developers would be expected to provide sufficient evidence to demonstrate that their software fulfills the requirements. As such, evidence that a tool meets the defined requirements may become a prerequisite for use of tools in accredited digital forensic laboratories.

3. Defining requirements

One goal of a software specification is to describe a tool's functionality in sufficient detail to enable testers to confirm that it satisfies all of its documented requirements. Writing a testable requirement can be challenging at the best of times, but it becomes even harder when the desired behavior is not clearly defined. This situation applies to digital forensic tools in particular because the community has not developed a standardized approach to certain routine tasks.

For example, take recovery of deleted files as an example requirement for a file system forensics tool such as EnCase, FTK, ProDiscovery, TSK or X-Ways. When dealing with FAT file systems, the requirement could be as simple as interpreting the directory entry of a deleted file to determine the starting cluster and size, and then allocating the number of contiguous unallocated clusters. However, another approach that might recover additional portions of deleted files is to skip over areas of the disk that are already allocated to files. In this way, it may be possible to piece together fragments of a deleted file that are not stored in contiguous clusters. Both approaches to recovering deleted files will be successful in some circumstances and unsuccessful in others. However, no studies have been conducted to determine which approach is more successful, and there is no consensus about which approach should be employed.

To accommodate these situations, the digital forensic community needs to define the proper behavior of our tools, at least for basic operations. When there are multiple approaches to performing a given operation, such as the recovery of deleted files discussed above, the tools need to provide users with an option to select one of the available approaches explicitly.

Overall, there is a need for a standard set of requirements for common operations in digital forensics such as acquisition of storage media, interpretation of file systems, recovery of deleted data and error handling. Only after these requirements are established by the digital forensic community, can we reasonably expect tool developers to conform to these requirements.

4. Competitive advantage

Software developers that provide evidence that their tool meets the requirements established by digital forensic laboratories will have a competitive advantage because digital investigators and digital forensic laboratories will favor those tools that can be evaluated objectively and thoroughly.

In addition to putting themselves in a strong position to meet emerging standards, software developers who provide verification evidence along with their tool will greatly enhance subsequent tool testing efforts. Comprehensively tested software helps find and fix bugs, reduces the risk of errors reaching the courtroom, and increases the trust in digital forensics as a discipline.

For the future credibility of our field, the time has come for digital forensic tools that can be tested and evaluated objectively based on rigorous documentation of software requirements. To reach this goal the digital forensic community must define requirements that set a clear expectation for tool developers.

Eoghan Casey
cmdLabs, 1101 E. 33rd Street, Suite C301, Baltimore, MD 21218, USA
E-mail address: ecasey@cmdlabs.com