Column: Every Last Byte

Simson Garfinkel
Naval Postgraduate School
California, USA
slgarfin@nps.edu

Inheritance powder is the name that was given to poisons, especially arsenic, that were commonly used in the 17th and early 18th centuries to hasten the death of the elderly. For most of the 17th century, arsenic was deadly but undetectable, making it nearly impossible to prove that someone had been poisoned. The first arsenic test produced a gas—hardly something that a scientist could show to a judge. Faced with a growing epidemic of poisonings, doctors and chemists spent decades searching for something better.

James Marsh's test for arsenic, published in 1836, ushered in a revolution in forensic toxicology. Unlike previous tests, the Marsh test detected not just the presence of arsenic, but could be used to determine the amount as well. The test was exceedingly sensitive. Most importantly, the test produced a stable black powder that could be both observed with the eye and preserved so that it could be shown to a jury at a later time. Poisonings continued, of course, but the chance of being caught increased dramatically.

Today's digital forensics has more than a passing resemblance to forensic toxicology. Instead of poisons, we have malware and hacker tools. Instead of the Marsh test and its modern replacement, gas chromatography, we have anti-virus scanners and lists of hash values. But in many ways the goal is still the same. The forensic examiner scans for that which does not belong. Such materials, when found, can be evidence of foul play.

There are, however, fundamental differences between human bodies and computer systems. Humans are live, natural objects, interacting moment-by-moment with a complex environment. Bodies are filled with chemicals and viruses that are unknown and for which no test exists. There is no way to enumerate the state of a human being—let alone preserve or restore it! A forensic pathologist would never attempt to explain every cell, virus, and chemical that's found in corpse. The task would be foolhardy for the simple reason that even healthy bodies are filled with stuff that has yet to be observed or identified, let alone explained.

Computer systems, in contrast, are knowable. We can precisely capture the contents of a computer's hard drive—even restore it on another system. (Try that with a human!) We can isolate a computer from its environment, or alternatively we can capture every piece of information that moves between a computer and the Internet. We can know the contents of every last byte.

Indeed, using record and reply technology (now standard on VMWare Workstation), it is possible to watch the same piece of malware executing again and again over the same piece of data.

Instead of looking for the bad stuff, an alternative way to conduct a forensic examination might be to explain every last byte—or at least to try.

Today's digital forensic examiners lack both the tools and the training to explain every last byte of what's on a hard drive. More importantly, they lack the science. Computer system may be enumerable, but even simple computers have an astounding number of enumerable states. This means that simplistic approaches to classifying the content of a computer system are bound to fail, for the simple reason that they are not sufficient to handle the computer's inherent complexity.

For example, many examiners today use the National Institute of Standards and Technology (NIST) National Software Reference Library (NSRL) Reference Data Set (RDS) to scan for malware and hacker tools. This approach won't find a computer that was hacked due to running an out-of-date copy of Adobe Flash for the simple reason that the vulnerable copy of Flash is in the RDS. The RDS won't identify it as "malware" because the program was previously published by Adobe, vulnerabilities and all. Likewise, hash-based approaches will never be able to find a computer hacked because of an unknown zero-day attack, or one that's been compromised with custom-written malware.

An alternative approach is to take the observable information and assemble it into a set of models that explain the computer's history—and presumably, the user's actions. Distinguish data that fits the model from those that don't. Bring inconsistencies to the attention of the forensic examiner.

With a few years of research, this approach might be able to detect the presence of unknown steganography techniques in complex document types. The detector would have a model that describes how the bytes within a Microsoft Word, JPEG, or ZIP file are structured. The detector could then examine a corpus of documents and look for those that don't follow the model—or for those that show unexpected variability in regions where other documents show consistency.

Turing teaches us that we can't *really* understand every last byte on a computer system, for the simple reason that many of those bytes are both programs and data. Still, much may be gained from automated tools designed to build an affirmative model of what is known, rather than tools that search out evidence of malfeasance. After all, criminals have shown a remarkable ability to come up with new poisons.