Column: File Cabinet Forensics

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

Researchers can spend their time reverse engineering, performing reverse analysis, or making substantive contributions to digital forensics science. Although work in all of these areas is important, it is the scientific breakthroughs that are the most critical for addressing the challenges that we face.

Reverse Engineering is the traditional bread-and-butter of digital forensics research. Companies like Microsoft and Apple deliver computational artifacts (operating systems, applications and phones) to the commercial market. These artifacts are bought and used by billions. Some have evil intent, and (if society is lucky), the computers end up in the hands of law enforcement. Unfortunately the original vendors rarely provide digital forensics tools that make their systems amenable to analysis by law enforcement. Hence the need for reverse engineering.

There is no legal requirement for the cell phone makers to support data extraction or to help us to understand the extracted data. As a result, the developers of forensic tools must painstakingly reverse engineer cable pinouts, master the hardware, software and data layouts of phones, figure out ways to extract data from devices, decipher the meaning of each binary field, and track arbitrary (and frequently undocumented) vendor changes. The same problems are played out in reverse engineering application programs, file formats, over-the-wire protocols, proprietary file systems, and indeed the vast majority of information that is processed by modern computer forensics tools.

Much of this reverse engineering effort is best described as “file cabinet forensics.” That is, the reverse engineers are trying to figure out information that’s locked away in vendor file cabinets (or file servers). It would certainly be a lot easier to have the vendors provide their design documents, specifications and source code that’s in their file cabinets, rather than trying to decipher those bits without technical assistance. But there is no way to compel the vendors to yield their secrets.

The situation faced by police investigators (and thus by forensic developers) today is similar to the situation that law enforcement faced in the early 1990s when attempting to execute wiretaps on cell phone networks. Although the wireless companies of the time were legally required to provide law enforcement with help, they were not required to design or deploy systems that were technically capable of meeting law enforcements' requirement.

Faced on the one hand with equipment that was not "wiretap-ready" and on the other hand with wiretap orders, providers were forced to come up with creative back doors to let law enforcement conduct voice intercepts on their deployed networks. In New York City, for example, AT&T gave the FBI access to cellular networks through the use of "technical ports" that had been created for servicing the switches. But there weren't enough ports to satisfy the demand, and a significant backlog built up.[1]

In the 1994 Congress addressed the wiretap issue with the Communications Assistance to Law Enforcement Act. Passed over the objection of civil libertarians, CALEA created a $10,000-per-day fine for companies that sell voice communications equipment into the US market that cannot be wiretapped. Today, as a result, the US has a telephone infrastructure that offers no privacy against court-ordered intercepts, and the technology is available world-wide, to democracies and totalitarian regimes alike.

(Clearly, allowing for lawful access can be a double-edged sword. We would ideally like a system that provides law enforcement access in a manner that is fully audited and not subject to abuse. Instead, the technology has clearly been misused, as it was in Greece in 2005[2]. But that, alas, would best be discussed in another article.)

Unfortunately, reverse engineering is ultimately a no-win game for digital forensics. That's because there are more people building new digital artifacts than reversing the artifacts currently in use. Forensic researchers simply can't keep up. For this reason, I believe that we will need to address this problem legislatively, as we did with CALEA, and reserve our reverse-engineering capabilities for developers who operate outside the law—for example, malware authors.

Beyond mere reverse engineering, there is a world of research that needs to be done. My concern is that researchers are now spending so much effort on reverse engineering that other important research is being delayed or deferred, as that other work requires the benefits of file cabinet forensics to be practical.

Consider the case of the location information stored within smart phones and GPS devices. Such information can be of great use in a criminal investigation, and it is widely acknowledged that there is a wealth of location information

[1] See "Snoops are vexed by digital era," S. Garfinkel, *The Boston Globe,* 1991, for a discussion of the problems faced by law enforcement at the time. http://simson.net/clips/1991/1991.Globe.Digital_Telephany.pdf

[2] Prevelakis, V. & Spinellis, D. (2007, July). The Athens Affair. *IEEE Spectrum*, *44*(7), 26-33.

that is not immediately visible through the user interfaces that these machines provide. Although vendors know about the information that their machines explicitly record, there is other information that is inadvertently captured or improperly erased. That information can only found with work beyond basic reverse engineering—finding that information requires *reverse analysis*.

A good example of reverse analysis was the discovery of the iPhone tracking database. This data, which was used by computer forensics examiners for more than a year, was widely publicized in April 2011[3] and soon corrected by Apple. At this incident illustrates, reverse analysis can produce discoveries are useful, but their use can be fleeting, as many correspond to bugs and privacy violations in consumer products which vendors will be highly motivated to correct. Zero-day exploits are and good example of the fruits of reverse analysis.

It's my belief that the biggest multiplier for digital forensics research comes not from reverse engineering or reverse analysis, but from the development of new techniques that transcend the specifics of the systems being analyzed. Such techniques are powerful because they can be applied to a wide range of digital artifacts, rather than to the specific system being analyzed.

A good example here is the development of a technique for finding AES keys in memory through the analysis of the key schedule.[4] Another example is my program, bulk_extractor, which uses opportunistic decompression to search for features in compressed data. Both of ideas that can be generally applied to a range of different situations, creating powerful capabilities that can be applied to many different devices.

What's both exciting and frustrating about digital forensics is that we need constant work in reverse engineering, reverse analysis, and underlying science in order to make progress against a problem that keeps getting harder. Like a person on a treadmill, we need to run just to stay in place. That's because our adversary is not just the bad guys—it's also the multitude of developers. For this reason, I hope that in the future we will turn more to legislation to solve these "file cabinet forensics" problems. After all, it's more efficient to get the data out of file cabinets, rather than resorting to reverse engineering.

[3] Allan, A. & Warden, P. (2011, April). iPhone Tracker. http://petewarden.github.com/iPhoneTracker/

[4] Halderman, J.A., Schoen, S.D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J.A., Feldman, A.J., Appelbaum, J., & Felten, E.W. (2008, July). Lest We Remember: Cold Boot Attacks on Encryption Keys. In: *Proc. 17th USENIX Security Symposium (Sec '08)*, San Jose, CA.