Digital Investigation 11 (2014) S45–S53

Contents lists available at ScienceDirect

Digital Investigation

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

OpenLV: Empowering investigators and first-responders in the digital forensics process

Timothy Vidas [a,b,*], Brian Kaplan [a], Matthew Geiger [b]

[a] Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA
[b] Dell SecureWorks, 1 Concourse Pkwy NE 500, Atlanta 30328, Georgia

Keywords:
Triage
Incident response
First-responder
Virtualization
Preliminary forensic examination

A B S T R A C T

The continuing decline in the cost-per-megabyte of hard disk storage has inevitably led to a ballooning volume of data that needs to be reviewed in digital investigations. The result: case backlogs that commonly stretch for months at forensic labs, and per-case processing that occupies days or weeks of analytical effort. Yet speed is critical in situations where delay may render the evidence useless or endanger personal safety, such as when a suspect may flee, a victim is at risk, criminal tactics or control infrastructure may change, etc. In these and other cases, investigators need tools to enable quick triage of computer evidence in order to answer urgent questions, maintain the pace of an investigation and assess the likelihood of acquiring pertinent information from the device.

This paper details the design and application of a tool, OpenLV, that not only meets the needs for speedy initial triage, but also can facilitate the review of digital evidence at later stages of investigation. With OpenLV, an investigator can quickly and safely interact with collected evidence, much as if they had sat down at the computer at the time the evidence was collected. Since OpenLV works without modifying the evidence, its use in triage does not preclude subsequent, in-depth forensic analysis. Unlike many popular forensics tools, OpenLV requires little training and facilitates a unprecedented level of interaction with the evidence.

Introduction

In today's increasingly connected world, criminal investigations are likely to entail a digital component at some stage of the process. Even an investigation of purely physical crimes, such as murder, commonly incorporate the analysis of digital evidence, ranging from cell phone records to the victim's email messages. Unfortunately, the personnel trained to perform forensic analysis of these digital artifacts are over-taxed and the influx of cases leads to backlogs. Yet timely action may be important to hold criminals accountable for their actions or to protect others from further harm.

Various forensics process models have been proposed since DFRWS in 2001 (Reith et al., 2002; Palmer, 2001; Carrier and Spafford, 2003; Beebe and Clark, 2005), but these generally assume that the entire, lengthy process is performed. A later stage common to most models is technical analysis, a stage that necessitates trained specialists and creates the backlog of work already noted. In reaction, the application of the medical field's concept of triage has been proposed in order to quickly assign degrees of importance and urgency to items (Rogers et al., 2006; Casey et al., 2009). With respect to digital forensics, triage typically refers to rapid analysis, possibly on-scene, of digital evidence, with steps to maintain the integrity of the evidence. Since the evidence is preserved, triage does not obviate later, extended analysis using a forensic model. Digital forensic triage can provide investigative leads in a timely manner so that they can be acted upon while still applicable.

* Corresponding author. Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA.
E-mail addresses: tvidas@cmu.edu (T. Vidas), bfkaplan@alumni.cmu.edu (B. Kaplan), matthew_geiger@dell.com (M. Geiger).

http://dx.doi.org/10.1016/j.diin.2014.03.006

In practice, first-responders are often trained to recognize potential digital evidence. However, the typically prescribed action for the responder is to collect the evidence (placing a hard drive in an anti-static bag, for example) or to secure the scene until trained personnel arrive to conduct the acquisition. In either case, the next venue for the digital media is the inbound queue of a forensics lab. This "find and forward" approach places a heavy burden on the lab and its trained personnel. In addition, under this model, the investigator is at the mercy of the lab, often waiting for results in order to further the investigation. A triage model that allows the first-responder or the investigator to generate leads, can not only facilitate faster investigation but could also inform and accelerate analysis by the trained lab technician. Triage does not supplant traditional forensics processes or tools, but can augment and enhance the investigative process.

The primary contribution of this paper is a description of a tool, OpenLV, designed and deployed over the past six years under the name "LiveView." OpenLV aims to meet the demand for an easy-to-use triage tool. As such, OpenLV's target audience is digital forensics practitioners, investigators, and first-responders, though OpenLV has also been used extensively in training and educational settings. OpenLV is a free, 100% GPL-licensed tool.[1] Over the past few years, LiveView has been downloaded hundreds of times per week since originally released. A 2008 survey indicated that 30% of universities and 22% of digital forensics practitioners use the tool in some way (Tu et al., 2012). In addition to incremental updates, such as supporting the use of forensic images of current versions of Windows, the most recent version of OpenLV notably adds support for analysts using Linux to run OpenLV, support for VirtualBox virtualization software, and the ability to handle Cached Domain Credentials (discussed in Section Windows passwords). After years of development, OpenLV is a mature product that not only addresses a digital forensics need, but does so while giving users options regarding host operating system and virtualization software.

The remainder of this paper has the following structure. We first discuss background material in Section Background. In Section OpenLV, we describe the design and usage of OpenLV. Section Windows passwords describes a particular feature of OpenLV, removing the obstacle of authentication in password-protected evidence. Then, in Section Limitations, we provide limitations to the current implementation of OpenLV. We discuss related work in Section Related work and future work in Section Future work. Finally, we conclude in Section Conclusion.

Background

Forensics is often divided into classes, Live and Traditional (or "dead"). Live forensics shares many concepts with incident response (Jones et al., 2006). The user interacts with a running computer in order to identify leads and determine the next investigative steps. Since interacting with the computer necessarily changes its state, purists often shun live forensics. However, the advent of purely memory-resident malware or the need to acquire in-use encryption keys offer little alternative to conducting live forensics (Vidas, 2007; Kaplan, 2008).

Conversely, traditional digital forensics often dictates the duplication of media prior to any other interaction (Jones et al., 2006). Some evidence collection procedures demand that running computers be unplugged from power in order to prevent changes to the hard disk during the shutdown process (Best practices for seizing electronic evidence, 2002). A duplicate copy of evidence is often called an *image*. A forensics image is made by copying data to a second physical hard drive or to one of many forensic files types. A dd or (raw) image is created by simply copying data blocks from the target device to a file. Other file types improve upon this simple copy strategy by improving redundancy, storing metadata, and reducing file size with compression. In addition to the dd/raw file type, popular file types include Guidance Software's proprietary E01 format and the open Advanced Forensics Format (AFF) (Garfinkel et al., 2006). When creating forensic images, the creator may choose to duplicate the entire disk, or some subset such as a disk partition.

In additional to the general digital forensics landscape that guided the creation of OpenLV, we also provide some foundation surrounding modern virtualization platforms. VMware produced one of the earliest virtualization products for personal computers and now maintains a leading line of commercial products. VMware offers free and commercial products targeting desktop users (as opposed to data centers) in the form of its Workstation, Player, Fusion and Server range of virtualization platforms. Competing products also exist in free and commercial forms, such as Microsoft's Virtual PC, Parallels' Desktop, and Oracle's VirtualBox. For brevity, we provide background on the underlying mechanics of VMware's implementation, but general principles hold for most of these desktop virtualization products.

Fundamentally, virtualization software allows for the emulation of general computing hardware, such as the CPU, graphics card, hard disk drive, etc, on a host computer system. In this way, one physical machine (the *host*) can be used to run multiple instances of various operating systems each within a *virtual machine* (VM). The VMs each run independently from other VMs and all external interaction via network or human interface devices is mediated by the virtualization software.

Structurally, on the host, virtual machines typically consist of two core components[2]: a virtual hardware specification and a data store. VMware's desktop products store the virtual machine specification in a plain-text `.vmx` file. This file dictates what hardware settings will available to the virtual machine. For example, as shown in Fig. 1, the `.vmx` file may specify the amount of RAM, if a virtual floppy disk drive is to be present, and BIOS settings.

Similarly, the data store specifications reside in a plain-text configuration file. This `vmdk` file specifies the type of virtual drive and information about its disk geometry. VMware products support different types of virtual disks. When

[1] OpenLV may be obtained at http://www.openlv.org/.

[2] There may be other files storing the "physical" memory of the VM, additional settings, a binary BIOS, snapshots, etc.