available at www.sciencedirect.com

ScienceDirect

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

Digital Investigation

Distributed forensics and incident response in the enterprise

*M.I. Cohen**, *D. Bilby*, *G. Caronni*

Google Inc, Switzerland

Keywords:
Remote forensics
Live forensics
Digital forensics
Incident response
Information security
Malware
Memory forensics
Distributed computing

ABSTRACT

Remote live forensics has recently been increasingly used in order to facilitate rapid remote access to enterprise machines. We present the GRR Rapid Response Framework (GRR), a new multi-platform, open source tool for enterprise forensic investigations enabling remote raw disk and memory access. GRR is designed to be scalable, opening the door for continuous enterprise wide forensic analysis. This paper describes the architecture used by GRR and illustrates how it is used routinely to expedite enterprise forensic investigations.

1. Introduction

Digital forensics is an established field in civil and criminal investigations. In the enterprise, digital investigations are often associated with incident response – the detection and investigation of system compromises and targeted attacks. Within the corporate sphere, investigations typically focus on timely response and damage assessment, in addition to maintaining evidentiary standards. The typical enterprise owns and deploys many machines serving a multitude of roles – for example, workstations, laptops and servers. All these machines can be used as launch points for internal attacks and may become involved in forensic investigations. Digital readiness has been previously defined as "the ability of an organization to maximize its potential to use digital evidence whilst minimizing the costs of an investigation" (Rowlingson, 2004). A distributed forensic investigation platform therefore serves to increase digital readiness by lowering the investigative cost and increasing the quality of digital evidence obtainable.

Traditional forensic acquisition consists of shutting the target system down, removing its disk and acquiring a bit for bit copy of the drive, followed by a manual analysis of the drive image. Forensically sound write blocker hardware is often employed during the acquisition step – often requiring the physical removal of the system's hard disk (Carrier and Spafford, 2003). Advantages of this technique include the preservation of digital evidence, and court validated procedures potentially enhancing the admissibility of evidence (Scientific Working Group on Digital Evidence, 2006).

This process is time consuming, requiring a trained forensic investigator to be physically present for the acquisition. This increases the cost of imaging and response time and reduces the number of machines that can be imaged in an investigation. Shutting down a running system may also lead to the loss of important volatile evidence (Walters et al., 2007). Additionally, without knowledge of all assets involved in an incident, even the best evidence preservation techniques fail to give an accurate picture. Timeliness of response becomes more significant to preservation of evidence, than the potential admissibility of the evidence.

Live response has been used as a critical part of the investigative process in order to capture and document volatile system state prior to hard disk imaging (Walters et al., 2007). A number of toolkits are available for automating live evidence capture prior to the seizing of computers (e.g., Microsoft COFFEE – a law enforcement only tool (Microsoft Corporation, 2011)). Volatile evidence can be used after

* *Corresponding author.*
E-mail address: scudette@google.com (M.I. Cohen).
1742-2876/$ – see front matter
doi:10.1016/j.diin.2011.05.012

acquisition to support the investigation in addition to a disk image. Arguably, the complete acquisition of physical memory is sufficient to enable most volatile system state to be subsequently deduced (Suiche, 2011; MANDIANT, 2011).

Triaging has been proposed as way of reducing the time required to image many systems (Garfinkel, 2010). The goal is to both identify relevant evidence quickly and guide the investigative process by directing human resources to reduce the overall response time (Rogers et al., 2006). Triaging often employs less thorough forensic techniques. For example, rather than analyzing the disk by keyword searching across all files, undeleting files, or unpacking container files, an investigator might choose to only search for keywords in documents located in a certain directory. This might be considered a quick way of determining the relevance of this specific disk to the case. Similarly rather than a bit for bit image of the target media, an investigator may selectively image only the data which is likely to be evidentiarily relevant (Termed selective imaging (Turner, 2006)).

Remote live forensic analysis has been proposed as a solution to reducing the time required for response. Some simple tasks can be performed using inbuilt operating system facilities, e.g., examining files via domain file sharing or standard remote administration tools. Typically however, remote system level access to deployed systems is provided via a pre-installed agent servlet. The remote agent allows an investigator to directly connect to the system, and perform rapid analysis or triage the system for subsequent acquisition (Guidance Software Inc., 2011; Various, 2011a). Care must be taken in order to protect the communications between the agent and the management console, typically this communication is encrypted and secured using sophisticated protocols (Mccreight et al., September 2004; Various, 2011a).

Since the system under investigation itself is used for running the remote agent, data obtained from the live system might be subject to subversion by locally active malware, or an attacker. Although case law is rare in this area, legal analysis suggests that authenticity challenges could be successfully defended by thorough tool testing (Kenneally, 2005), although some additional legal risk is afforded by remote live forensics (Kenneally and Brown, 2005).

1.1. *Privacy*

Having a remotely accessible and often silent enterprise management agent installed can pose serious concerns for privacy of the users. Many enterprise management tools give the investigator complete access to the underlying filesystem and memory. This power can be abused to seriously compromise a user's privacy since personally identifiable information (such as credit card numbers, credentials or encryption keys) can be gleaned from memory images and browser cache objects (Kornblum, 2009).

With tighter regulations regarding privacy it is becoming imperative to protect the user's information (Lasprogata et al., 2004; Kim, 2006). This requirement is in direct competition with the need for expedited and thorough analysis for locating compromises (Kim, 2006). Typically in a large organization, routine investigative tasks (e.g., malware infections) must be delegated quickly to a large number of operators, while sensitive investigations which might require access to private information must be legally managed within a small number of operators (e.g., those granted access under a legal warrant).

Unfortunately, current products do not provide fine grained access controls or the auditing flexibility to tailor access to user data (Casey and Stanley, 2004). Due to the interactive nature of most tools, any user granted access to its management console, automatically provides that user with complete access to all files on the remote system. Furthermore, auditing the files that the user accessed is not usually implemented. This may give the investigator complete, unfettered and unaudited access to all files.

On the other hand, by automating pre-devised analysis tasks, our system allows only specific, audited tasks to be initiated on the remote system. For example, searching for files matching a particular keyword can be initiated automatically without the investigator seeing what files actually exist in the user's home directory. These files may then be quarantined for subsequent review by an authorized person.

1.2. *Correlation*

Deployed systems have a variety of software installed (often by users) presenting different kinds of risks for the security of the system and the network. When investigations are conducted on an enterprise scale, the ability to query the entire fleet of deployed systems can in itself provide useful signals for compromise. For example, if a new vulnerability is discovered in a particular version of a product, up to date statistics of all systems in inventory running the vulnerable version can be used to rapidly determine risk exposure and formulate a response plan. This can occur even when the vulnerable software is not part of the standard build environment, but was installed by a user.

As malware is developed to appear more innocuous, it attempts to blend in with other system artifacts (Peron and Legary, 2005). For example, malware commonly uses the same process name as common system processes or use similar registry keys or files as legitimately installed software. Detection of the malware is thus more difficult when analyzing the system in isolation. However, when considering the correlation of artifacts across a large fleet of machines, many running the same install base, indicators of compromise are easily seen. For example, if a commonly installed executable has a particular hash value on most systems, but a different hash value on a few systems in the fleet – this might indicate that this binary was maliciously replaced.

In addition, correlating information across time may also provide valuable indication of compromise – as in the case of new and unusual artifacts suddenly appearing. For example, assume a series of snapshots in time of autorun processes (processes which start when the system is booted) is available for each system. If subsequently a malware is detected on a particular system, inspecting these snapshots allows us to establish a timeline of compromise more reliably (e.g., without relying on potentially manipulated timestamps on the compromised system).