available at www.sciencedirect.com

ScienceDirect

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

Digital Investigation

A survey of main memory acquisition and analysis techniques for the windows operating system

*Stefan Vömel**, *Felix C. Freiling*

Department of Computer Science, Friedrich-Alexander University of Erlangen-Nuremberg, Am Wolfsmantel 46, 91058 Erlangen-Tennenlohe, Germany

ARTICLE INFO

Article history:
Received 9 May 2011
Received in revised form
8 June 2011
Accepted 11 June 2011

Keywords:
Memory forensics
Memory acquisition
Memory analysis
Live forensics
Microsoft windows

ABSTRACT

Traditional, persistent data-oriented approaches in computer forensics face some limitations regarding a number of technological developments, e.g., rapidly increasing storage capabilities of hard drives, memory-resident malicious software applications, or the growing use of encryption routines, that make an in-time investigation more and more difficult. In order to cope with these issues, security professionals have started to examine alternative data sources and emphasize the value of volatile system information in RAM more recently. In this paper, we give an overview of the prevailing techniques and methods to collect and analyze a computer's memory. We describe the characteristics, benefits, and drawbacks of the individual solutions and outline opportunities for future research in this evolving field of IT security.

1. Introduction

With the widespread use of computer systems and network architectures, digital cyber crime has, unfortunately, aggravated as well. According to a recent publication by the Internet Crime Complaint Center (2010), a partnership between the National White Collar Crime Center (NW3C) and the Federal Bureau of Investigation (FBI), the number of complaints filed to the institution has almost gone up by the factor 20 within less than a decade. In 2009, more than 336,000 reports about different types of illicit activity such as online fraud, identity theft, and economic espionage were registered. The yearly monetary loss of complaints referred to law enforcement was estimated to be nearly $560 million. As a survey by the Computer Security Institute (2009) shows, companies may lose up to several hundred thousand dollars in the course of an incident. In such cases, a forensic investigation of the affected machines may prove helpful for reconstructing the actions that led to the security breach, finding relevant pieces of evidence, and possibly taking legal actions against the adversary.

Traditional approaches in computer forensics mostly described the acquisition and analysis of *persistent* system data. Respected procedures usually involved powering off the suspect machine, creating an exact bit-by-bit image of the corresponding hard disks and other storage media, and performing a post-mortem examination of the collected information (U.S. Secret Service, 2006; U.S. Department of Justice, 2008). Obtaining a copy of physical Random Access Memory (RAM) was, on the other hand, frequently neglected by first responders (Shipley and Reeve, 2006; Hoglund, 2008), even though guidelines stressed the necessity of securing digital evidence with regard to the *order of volatility*, i.e., from the volatile to the less volatile, as early as in 2002 (Brezinski and Killalea, 2002; Casey, 2004; Farmer and Venema, 2005). In the face of ever-growing hard drive storage capabilities (Oswald,

* *Corresponding author.*
E-mail addresses: stefan.voemel@informatik.uni-erlangen.de (S. Vömel), felix.freiling@informatik.uni-erlangen.de (F.C. Freiling).
1742-2876/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.diin.2011.06.002

2010), and correspondingly, tremendous efforts to analyze media in time (Mrdovic et al., 2009; Walters and Petroni, 2007; Shipley and Reeve, 2006) as well as a rising number of memory-resident malicious software applications (Moore et al., 2003; Rapid7 LLC, 2004; Sparks and Butler, 2005; Bilby, 2006), the restoration of transient and system state-specific information has, however, also moved more gradually into the focus of current research, beginning with the *Digital Forensic Research Workgroup* (DFRWS) challenge in 2005 (DFRWS, 2005).

This shift in practices has been driven and inspired by several other developments, too: First, "pulling the plug" on a company server may negatively affect productivity in certain cases and cause substantial losses due to unexpected down times. Furthermore, depending on the configuration of the system, file system journals may be damaged during the shutdown process, or the machine may be difficult to restart. Thus, there is a demand to minimize interferences with existing business and enterprise processes. Second, some programs are explicitly designed to make no or preferably as little persistent changes as possible on the hard disk of the user. Contemporary examples for this type of software are the *Mozilla Firefox* web browser with its private browsing capability (Mozilla Foundation, 2008; Aggarwal et al., 2010) or utilities included in the *PortableApps.com* project (Rare Ideas, 2010). For this reason, forensic analysts must adapt their strategies and also search in volatile system storages for traces and data remnants, including usernames, passwords, and text fragments. Moreover, many modern operating systems include support for file or even full disk encryption (Microsoft Corporation, 2009; Apple Inc., 2010; Saout, 2006). Similar functions are provided by freely-available open source tools, e.g., *TrueCrypt* (TrueCrypt Foundation, 2010), or commercial products such as *SafeGuard Easy* (Sophos Plc, 2010) or *PGP* (PGP Corporation, 2010). Because of the transparent design and ease of use of these software products, security professionals are likely to face an increasing number of encrypted drives that make traditional investigations infeasible (Getgen, 2009). Restoring a cryptographic key from memory might be the only possibility to get access to the protected data area in this case. The same holds true for packed malicious binaries. Malware writers typically employ compression, armoring, and obfuscation techniques to make reverse engineering and static analysis of their code more difficult (Sharif et al., 2009; Rolles, 2009; Brosch and Morgenstern, 2006; Young and Yung, 2004). Memory inspection is a viable solution to cope with these issues and extract the unpacked and decrypted executable directly from RAM.

As can be seen, a myriad of valuable information is stored in volatile memory that is usually lost when the target computer is powered off. Failing to preserve its contents may thus destroy a significant amount of evidence.

1.1. Motivation for this paper

Over the last 5 1/2 years, considerable research has been conducted in the field of memory forensics, and various methods have been published for capturing and examining the volatile storage of a target machine. However, many techniques solely apply to specific versions of operating systems and architectures or only work under certain conditions. Moreover, depending on the technology used, the reliability and trustworthiness of generated results may vary. For these reasons, security professionals must have a thorough understanding of the capabilities and limitations of the respective solutions in order to successfully retrieve pieces of evidence and complete a case. A complete description of the current state of the art appears to be missing at the time of this writing though, restricting (research) activities in this area to a number of renowned experts.

In this paper, we give a comprehensive and structural overview of proven approaches for obtaining and inspecting a computer's memory. We explain the technical foundation of existing tools and methodologies and outline their individual strengths and weaknesses. Based on these illustrations, security analysts and first responders may choose an adequate acquisition and analysis strategy. In addition, we give an extensive summary of the relevant literature. This review serves as a good starting point for own future studies.

Please note that our explanations refer to the product family of *Microsoft Windows* operating systems. We assume that due to their high popularity and dominant market position (Net Applications, 2010), investigators are particularly likely to face Windows-based machines in practice. In addition, as we will see, a deep knowledge of internal system structures is required to collect digital evidence from a volatile storage. Covering other platforms such as Linux or Mac OS is therefore out of the scope of this paper. Interested readers are referred to Movall et al. (2005) and Suiche (2010) for more information on these topics.

1.2. Outline of the paper

This paper is outlined as follows: In Section 2, we briefly describe the memory management process and give an overview of the most important data structures that are required for this task. Current techniques and methods for creating a memory image from the target system are presented in Section 3, followed by a detailed illustration of the different investigative procedures in Section 4. A special framework for memory analysis activities, *Volatility*, is subject of Section 5. We conclude with a summary of our work and indicate opportunities for future research in this area in Section 6.

2. Technical background

Modern multi-tasking operating systems typically do not access physical memory directly, but rather operate on an abstraction called *virtual memory*. This abstraction of physical RAM needs specific hardware support (the so-called *Memory Manager* or *Memory Management Unit*) and offers several inherent advantages, e.g., the possibility of providing each process with its own protected view on system memory as well as monitoring and restricting read and write activities with the help of privilege rules (Intel Corporation, 2011). The layout between the virtual and physical address space may differ though, and blocks of virtual memory do not necessarily map to contiguous physical addresses as illustrated in Fig. 1.