Contents lists available at ScienceDirect

Digital Investigation

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

DFRWS 2015 Europe

Acquisition and analysis of compromised firmware using memory forensics

Johannes Stüttgen[*], Stefan Vömel, Michael Denzel

Department of Computer Science, Friedrich-Alexander University of Erlangen-Nuremberg, Martensstraße 3, 91058 Erlangen, Germany

Keywords:
Memory forensics
Memory acquisition
Live forensics
Firmware rootkits
Incident response

A B S T R A C T

To a great degree, research in memory forensics concentrates on the acquisition and analysis of kernel- and user-space software from physical memory to date. With the system *firmware*, a much more privileged software layer exists in modern computer systems though that has recently become the target in sophisticated computer attacks more often. Compromise strategies used by high profile rootkits are almost completely invisible to standard forensic procedures and can only be detected with special soft- or hardware mechanisms. In this paper, we illustrate a variety of firmware manipulation techniques and propose methods for identifying firmware-level threats in the course of memory forensic investigations. We have implemented our insights into well-known open-source memory forensic tools and have evaluated our approach within both physical and virtual environments.

© 2015 The Authors. Published by Elsevier Ltd on behalf of DFRWS. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Introduction

Over the last decade, forensic practitioners have increasingly started to perceive the value of volatile information in system RAM for the outcome of a case. Particularly when suspecting potentially-installed malicious applications on a compromised host, so-called *memory forensic* investigation techniques can frequently confirm assumptions and yield results more quickly than traditional, persistent data-oriented approaches (Walters and Petroni, 2007). The majority of research has thereby concentrated on *memory analysis*, i.e., identifying relevant data structures and subsequently extracting pieces of evidence (for an overview of current practices, please refer to Ligh et al. (2010) and Vömel and Freiling (2011)). The process of securing said data in a forensically-sound manner on the other hand, i.e., *memory acquisition*, has been described to a significantly lesser degree in the literature. This fact is quite surprising, considering that recent works have indicated quality deficiencies in popular imaging solutions under certain conditions that may severely impede or even prevent thorough artifact examination at a later time (Vömel and Stüttgen, 2013; Stüttgen and Cohen, 2013).

In order to mitigate this research gap, we will discuss several scenarios in this paper that exemplify various hurdles and pitfalls modern memory acquisition technologies must overcome. With respect to this, we will focus on specific system manipulation strategies on the *firmware* level, in contrast to other studies that illustrate issues in kernel or user space (see Ligh et al., 2010; White et al., 2013). Although only a comparatively small amount of malicious programs distributed in the wild are known of explicitly attacking base components and features of a system, the lower machine layers have received broader attention by malware authors in the last years, especially with regard to the development of *Advanced Persistent Threats* (APTs) and special *rootkits* that are remarkably difficult to detect and remove (Shields, 2008; Davis et al., 2010). As we will see, however, the vast number of

[*] Corresponding author.
E-mail addresses: johannes.stuettgen@cs.fau.de (J. Stüttgen), stefan.voemel@cs.fau.de (S. Vömel), m.denzel@cs.bham.ac.uk (M. Denzel).

http://dx.doi.org/10.1016/j.diin.2015.01.010
1742-2876/© 2015 The Authors. Published by Elsevier Ltd on behalf of DFRWS. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

forensic programs and toolkits available on the market to date are unfortunately ill-prepared for dealing with such threats and fail at properly duplicating all sources of a computer's physical address space in which malicious code may reside.

Contributions of the paper

In the scope of the paper, we will see that most firmware code and data can be accessed over the memory bus. We present a broad survey of firmware rootkit techniques and the traces such threats leave in a system. To facilitate acquisition and analysis of potential infections, we have developed methods for enumerating, imaging, and examining firmware components and implemented them into free, open-source utilities. The reliability of these utilities is evaluated with the help of a proof-of-concept ACPI rootkit as well as manipulated BIOS ROMs.

Related work

As we have already indicated, the development of malicious programs that are directly stored and executed on the hardware or firmware layer has increased over the last years. Stewin and Bystrov (2012), for instance, outline techniques for silently capturing keystrokes or extracting sensitive information such as cryptographic keys from a running machine with the help of *Direct Memory Access* (DMA). Embleton et al. (2008) illustrate how the processor's *System Management Mode* (SMM) may be misused for storing malevolent code, while Tereshkin and Wojtczuk (2009) manipulate the chipset of the *Memory Controller Hub* (MCH) for these purposes.

With respect to forensic investigations, other authors have attempted to leverage specific features or components of the hardware. Most notably, the IEEE 1394 (*FireWire*) interface or the more recent *Thunderbolt* port permit duplicating the contents of memory from a target system quite easily (see Zhang et al., 2011; Maartmann-Moe, 2012). In contrast, Wang et al. (2011) as well as Reina et al. (2012) describe a prototype implementation for creating a consistent copy of the physical address space by entering the System Management Mode. Memory acquisition based on DMA operations over a network card is subject of the work of Balogh and Mydlo (2013). Last but not least, obtaining the contents of firmware ROMs is the specific goal of the *Copernicus* project (Butterworth et al., 2013). Copernicus permits extracting firmware data directly over the *Serial Peripheral Interface* (SPI) bus. Because this approach is vulnerable to malicious software running in System Management Mode, the latest implementation utilizes the *Intel TXT* extensions (Intel Corporation, 2014c) to isolate the acquisition module from other parts of the system (Kovah et al., 2014).

Most of the previously described technologies have not or only insufficiently been evaluated. First, more extensive studies concerning the quality of imaging applications were presented by Vömel and Stüttgen (2013) and Stüttgen and Cohen (2013). We will further elaborate upon these works and assess the performance of common imaging products, specifically when sophisticated malware species are present.

Outline of the paper

The remainder of this paper is outlined as follows: In Section Background, we briefly define the type of malicious program we will concentrate on throughout our discussion. We also describe standard approaches for acquiring a copy of main memory from different operating systems and give an overview of several firmware features and technologies that help the reader better understand later parts of our work. A survey of current firmware rootkit techniques and their implications for memory forensic investigations is presented in Section Rootkit Strategies for Compromising Firmware. Methods for enumerating and acquiring firmware code and data are subject of Section Firmware Acquisition Using Memory Forensics. In Section Firmware Analysis, we discuss the analysis of the acquired data, followed by an evaluation of how well these approaches are already incorporated in common forensic suites and applications in Section Evaluation. Aspects and limitations that need to be considered when applying the respective concepts in real-world investigations are discussed in Section Discussion. We conclude with a short summary of our work and indicate various opportunities for future research in Section Conclusion.

Background

In the scope of this paper, we discuss challenges and pitfalls for memory acquisition solutions that operate within a hostile environment, i.e., the target system is likely to have been affected by malicious programs, and the integrity of the machine cannot be trusted. In Section Rootkits, we specify the conditions the respective acquisition technologies must cope with in more detail and give a more thorough definition of the *rootkit* term. In Section Memory Acquisition Process, we outline basic approaches for duplicating volatile information on different operating systems. The characteristics of major firmware technologies are subject of Sections System Firmware–ACPI. The information presented in these sections is mainly based on the work of Salihun (2006), Dice (2013) as well as the respective vendor specifications (Intel Corporation, 2014b; PCI-SIG, 2010) and is required to better understand the technical complexities illustrated in later parts of this paper. Readers who are already familiar with rootkits and have a solid knowledge of the BIOS, PCI, and ACPI environment may safely skip these explanations and directly proceed to Section Rootkit Strategies for Compromising Firmware.

Rootkits

A particularly sophisticated type of malicious programs are so-called *rootkits*. Rootkits are defined to consist of a set of programs and code that allows a permanent or consistent, undetectable presence on a computer (Hoglund and Butler, 2005, p. 4). They are intentionally designed to project a manipulated view to the system user in order to achieve the primary objectives *concealment*, *surveillance* as well as

Download English Version:

https://daneshyari.com/en/article/457884

Download Persian Version:

https://daneshyari.com/article/457884

Daneshyari.com