Forensic physical memory analysis: an overview of tools and techniques

Gabriela Limon Garcia
Helsinki University of Technology
glimonga@cc.hut.fi

Abstract

Forensic physical memory analysis has gradually evolved from basic techniques such as string searching to more complex methods. As computer malware becomes more sophisticated, tools and techniques for memory analysis suffer inadequacies. Given this, it is essential to examine tools and techniques for physical memory analysis. By understanding their behaviour, limitations and advantages it is possible then to select the most suitable solution.

KEYWORDS: forensics, physical memory analysis

1 Introduction

In recent years, terms such as computer viruses, worms and trojans have become common words to identify offensive attacks. The following statistics provide a deeper understanding of the current situation; a recent analysis by Kapersky Labs shows a raise in the number of malicious programs detected in a rate of 89% per month only for the second half of 2006 [18]. Moreover, the 2007 Computer Crime and Security Survey indicates an increase in the average of the annual losses due to cyber crime [27]. This number duplicates last year's amount. In addition, 184,944 companies surveyed, suffered a malware attack.

Evidently, there is an urgent need to react to such threats. Digital forensic science is useful for this purpose since it employs techniques which help to determine the origin of incidents such as cyber crimes. Digital Forensics comprehends the collection, validation, analysis, interpretation, documentation and presentation of the digital evidence [24].The latter, is defined as information of value stored or transmitted in a binary form [23].

In order to find digital evidence, traditional forensics procedures examine storage devices such as hard disks by acquiring an exact sector-by-sector copy (image) for later analysis [6]. Even though disk forensics provides file system data such as stored files and installed programs, several reasons prevent from solely applying this method. First, the required time to image a hard disk increases proportionally to hard disk storage capabilities; as a result, imaging large scale hard disks is not convenient when time is a crucial factor. Secondly, since disk imaging requires unplugging the system, availability of the system is interrupted which might be expensive and inconvenient for companies. Moreover, all the volatile information such as network connections, command history or information related to executing applications is lost [16]. This information resides in random access memory (RAM) and it is not possible to access it by means of disk forensics. As a result, memory forensics becomes the cornerstone to find significant evidence,

A differentiation in memory definitions should be noted. In Windows operating system, the Windows manager maps pages (block of data) in virtual memory to pages in physical memory [32]. This means that the data can temporarily reside in virtual memory and afterwards be allocated at a specified address in memory. That is, physical memory provides the main storage while virtual memory optimizes the use of physical memory [8]. This concepts are relevant, since some kind of malware is capable to hide itself without leaving any trace of activity. The mechanisms they apply directly affect the operating system. In consequence, since the operating system is compromised, the mechanism for memory management might be compromised as well. Because of this it is not longer possible to rely on the information it provides.

Early attempts to establish memory scanning techniques clearly declare the need for memory scanning [26]. The essential idea is to obtain reliable information of the system state. Running processes rely on applications that could be manipulated to produce false data. Therefore the only solution available is to take a snapshot of the memory state along with the static memory. Given this, this paper focuses in the analysis of physical memory rather than virtual memory. Specifically it concentrates in Windows physical memory.

Like any other science, computer forensics makes use of specialized tools to retrieve significant information from the object of study. The use of correct techniques and tools is decisive to accomplish its objective. This paper presents a state-of-art research of some tools for forensic memory analysis. It focuses in two stages of digital forensics: data collection and data analysis.

2 Motivation

The simplest definition of malware specifies it as any software with malicious intentions [3]. Advanced malware techniques not only comply with this definition, but also add new features to remain undetected by modifying the behaviour of the operating system kernel, or some sensitive application without user consent. This is the case of malware in the form of rootkits. They behave under two principles [20]: 1. rootkits need to execute and 2. rootkits need to remain hidden.

Traditional rootkit classification comprehends user mode

rootkits and kernel mode rootkits [30]. User mode rootkits generally install other programs to access the system secretly and hide its presence by modifying system libraries. Kernel mode rootkits take control over kernel level functions which difficulties their detection. This classification emphasizes the privileges for execution within the operating system complying with the first principle of rootkit behaviour.

Another categorization of rookits [14] emphasize persistence characteristics. This classification is persistent rootkits, which modify static components in the operating system, and memory based rootkits. In the latter case, rootkits implement "hooking techniques " [13]; Hooking refers to the alteration of kernel internals. For example, execution paths which are generated by invoking a Windows API function. The request is usually placed by a user-mode application

Both rootkit types, persistence and user mode, are easily detected within a disk image since they perform modifications to static data stored in the disk. Basically, the techniques to achieve this include file integrity check and memory scanning.

However, the ease of detection does not apply for the kernel and memory based rootkits. A method for preventing detection from memory scanning is commonly known as the Shadow walker memory subversion technique [12] Rootkits with this technology, hide in memory by means of redirecting processes execution to a replaced handler; they modify the dynamic objects loaded in memory used by the kernel to manage system resources. This technique is called Direct Kernel Object Manipulation (DKOM) introduced by James Butler [15]. By means of DKOM, rootkits control procedures such as virtual memory mappings described earlier, this technique prevents rootkits from detection. Traditional forensic approach, which is, relying in either virtual memory scanning or persistent data, is not sufficient. Subversion techniques such as shadow walker clearly illustrate the need for memory scanners tools aware of malware hiding techniques. Understanding the effects of forensic tools will help to differentiate them from the effects of stealth malware.

3 Tools and techniques

This analysis focuses in two phases of memory analysis: acquisition of the data and analysis of the collected data. The next section describes briefly some of the tools commonly employed for each purpose.

3.1 Evidence collection

Collection of evidence focuses in obtaining digital evidence in an acceptable form. There are two approaches to acquire physical memory images: software and hardware oriented. This section presents some tools for both approaches.

3.1.1 Hardware based acquisition tools

The main idea is to bypass the operating system by means of a physical device. The dedicated hardware will open a dedicated communication port to copy the contents of the physical memory. Two main technologies are in the limelight:

- **Tribble**. This solution uses a dedicated PCI card. (Peripheral Component Interconnect) The PCI card requires installation before incident occurrence. The card can easily be detached after the incident. In this way the state of the system is preserved to search for digital evidence [10].
  Advantages. The ease of use and the null impact on the system.
  Disadvantages. The pre-installation requirement is the major drawback. Unauthorized access to physical memory can easily be obtained through PCI cards by means of libraries [25]. Moreover, it is possible to perform Denial of Service attacks (DoS), covering attacks, full replacing attacks by hiding system memory on the PCI bus [19].

- **FireWire bus**. Also known as IEEE 1394 bus, supports among other functionalities such as high speed communication and data-transfer, physical access to the system memory.
  Advantages. The FireWire port is a popular port present in many systems.
  Disadvantages. For some system configurations, Fireware bus presents problems with a region of the memory called Upper Memory Area (UMA) citeUMA.

In general, the main advantage of hardware based acquisition tool is the absence of interaction with the operating system avoiding the risk of writing data to the target machine. However, since hardware based technologies use Direct Memory Access (DMA) to read physical memory, systems are vulnerable to attacks using this same feature.

3.1.2 Software based acquisition tools

- **Data Dumper**. A common used tool for acquiring an image of physical memory is data dumper (DD). DD is a Linux utility program whose objective is to access physical memory through the }Device}Physical Memory object. The windows version for DD is available within the windows forensic tools images developed by Garner [1]. DD relies in a core dump function which creates a special image structure in a *key-word = value* format; therefore, the output file is readable in many tools and applications like Windows kernel debugger.
  Advantages. Executing the program does not require rebooting the system or any disruption of the service.
  Disadvantages.The dumped memory has the same size as the RAM. [21]. The physical memory object is not accessible in Windows Server 2003 SP1 [17].

- **Windows crash dump utility**. A crash dump can be the result of instability on the system. Internally, when a crash dump occurs, the system state is frozen and the contents of the physical memory are swapped or copied to the disk. The result is a crash dump file with the contents of the RAM and extra debugging information. Windows provides a keyboard configurable feature to generate dump files [7]. The output file is .DMP format, compatible only with Microsoft tools. This format adds information of CPU state [29].
  Advantages. The file obtained from a crash dump is a

faultless copy of the RAM [16].
Disadvantages. The format is not specifically for forensics but for debugging purposes. Currently, minidump versions are available, complete dumps are not available. There are two main issues when configuring the feature: it requires to reload the system for the changes to take into account and its functionality is limited to a specific keyboard driver [28].

The major disadvantage of software based solutions is the executing conditions. As any other program, they require the use of kernel memory and processing, meaning the

3.2 Data Analysis

The basic task of this phase is to translate the obtained stream of bytes into structured information. The data structures within the dumped data are the cornerstone for data analysis. A data structure is a table of data including structural relationships. Therefore, relevant information for its analysis consists of the data structure fields and the correlation to other data structures.

First , general and isolated techniques are presented . Secondly, we present tools that comprehend a graphical user interface.

3.2.1 Techniques for memory analysis

String Searching

One of the most traditional techniques in memory forensics is the search for valuable strings such as passwords or network addresses that are relevant for the investigation.

- **String.exe**. Linux man pages define strings as a program that returns the strings from initialized and loaded sections of object file. This utility was ported to Microsoft environment by Mark Russinovich [4]. The main advantage is the simplicity of the tool. It is not required to have knowledge of operating systems internals. However, it is undesirable to apply this technique without a set of predefined keywords to pursue.

- **Grep.exe**. The grep program is designed to search a sequence of characters in a file. The main advantage of grep is its capabilities to specify regular expressions for searching.

- **Clustering algorithms**. This new approach was proposed at the Digital forensic Research workshop 2007 DFRWS [22]. The idea is to group the results of string searches using a neural network approach. This approach provide context information to the search results.

Advantages. Ease of use.
Disadvantages.These techniques, except for the clustering algorithms, provide information without an overall context. Mostly it is inferred by the investigator. Clustering algorithms are still on testing phase.

Finding Process objects

A Windows process has an associated EPROCESS structure. In the same way, a Windows thread has associated an ETHREAD structure. One ore more threads belong to one process [31]. The following techniques search for EPROCESS structures.

- **PTFinder**. Process and Thread Finder is a Perl script created by Andreas Schuster [9] to detect and list all the processes and threads in a memory dump. This script searches for EPROCESS structures and perform a series of comparisons against rules to ensure its authenticity. The layout output displays a hierarchical view of the processes that provides legibility to the results. A very interesting characteristic, is that PTFinder recognizes objects using the DKOM (Data Kernel Object Manipulation) technique
Advantages. Detection of processes implementing DKOM techniques.
Disadvantages. Definition of the structures depends upon the service pack and windows version, since the structures change for every verion and for every service pack.

- **Lsproc**. Harlan Carvey proposes a set of Perl scripts to list processes and its properties. [16]. Lsproc is similar to PTFinder except that Lsproc searches for processes only. Lsproc is part of a set of scripts that jointly offer a comprehensive view of the processes properties and relationships. The downside is the independence of the scripts; they are command line oriented programs which give a basic idea of extraction of information from memory dump.

Techniques based on the detection of EPROCESSEs are a reliable source for finding evidence of malware since a thread which does not belong to any process is more likely to be a suspicious object. The main drawback relies on the fact that sizes and the values of the structures do change between Windows operating systems versions and service packs.

Finding Objects signatures

Object specific signatures can be used to identify them in memory. The main approach is to scan the pool of signatures to identify hidden objects

- **GrepEXEC** . This tool is the result of the Digital Forensic Research Workshop challenge. The main purpose is to verify objects such as driver object, device object, EPROCESS and ETHREAD objects. It searches through the acquired image for recognizable objects signatures [11]. The disadvantage for this tool is the lack of source code for implementation.

Tools for memory anaysis

- **Windbg Tool** Windbg is a debugger part of Windows debugging tools. It provides an interface to the user

Tool/technique	Advantages	Disadvantages
Tribble	Null system impact	Pre-installation requirement/Security hole
FireWire bus	Hardware (port) availability	Memory incompatibilities
	Hardware based	
DD	No service disruption	Physical Memory object not available in Windows 2003
CrashDump utility	Minidump format	Incompatible with some keyboards
	Software based	

Table 1: Data Acquisition

to perform source-level debugging [5] Windbg applies debug symbols format to map strings to identifiable objects. The powerful property of Windbg is its scripting capabilities. It defines commands to display memory, structures, executable objects and other objects from a memory dump.
Advantages. Defines an API for access to memory objects.
Disadvantages. There is not a standard definition for the scripts. The results are isolated from each other.

- **KnTTools with KnTList** This tool performs two tasks: first, the acquisition of physical memory (KnTTools) based on DD and secondly the analysis (KnTList) interprets the structures in memory. The latter is based on mechanism used in PTFinder [2]
Advantages. Two main procedures of digital forensics are available in one tool.
Disadvantages. It is a commercial software.

4 Summary

The information retrieved for each tool is sinthehtized in table 1, 2 and 3. Table 1 offers a summary for the datda acquisition tools. Table 2 and Table 3 cover data anaysis.

5 Conclusion

It should be noticed that, despite physical memory offers a comprehensive collection of reliable information, it only provides a snapshot of the state of the memory in a given time. Data is not static, it changes at every time. We can deduce then that, physical memory analysis is still in an early deployment stage. The next generation of digital forensic tools for physical memory should employ more sophisticated data analysis techniques to adequate to the sophistication techniques of malware. Moreover, new developed tools should integrate different approaches. The result

Tool/technique	Advantages	Disadvantages
Strings	Simplicity	Does not offer context information
Grep	Search by means of regular expressions	No context information
Clustering agorithms	No service disruption	Physical Memory object not available in Windows 2003
PTFinder	Detection of DKOM techniques	Different structures for each Windows version and SP
LsProc	ease of execution	isolated results
GREPexec	covers different types of objects	It is not implemented

Table 2: Data analysis - Techniques

Tool/technique	Advantages	Disadvantages
Windbg	Comprehensive API/ Free software	Isolated results
KnTTools	Integration of acquisition and analysis	Commercial software

Table 3: Data analysis - Techniques

should translate in tools ease of use, generation of results in an efficient and accurate manner and adaptable to new challenges new threats pose. This should be reinforced by standardization of basic concepts to define a common language.

References

[1] Garner george m. forensic acquisition utilities. http://www.gmgsystemsinc.com/fau/.

[2] Knttools with kntlist. gmg systems, inc. http://www.gmgsystemsinc.com/knttools/.

[3] The linux information project. http://www.linfo.org/malware.html.

[4] Mark russinovich. strings v2.40. http://www.microsoft.com/technet/sysinternals/Miscellaneous/Strings.mspx.

[5] Microsoft windows debugging tools. http://www.microsoft.com/whdc/devtools/debugging/debugstart.mspx.

[6] Steps for recovering from a unix or nt system compromise. cert coordination center. http://www.cert.org/tech_tips/win-UNIX-system_compromise.html.

[7] Windows feature lets you generate a memory dump file by using the keyboard. http://support.microsoft.com/kb/244139.

[8] M. Abd-El-Barr. *Fundamentals of Computer Organization and Architecture*. John Wiley Sons, Incorporated, 2005.

[9] Andrea Schuster. Searching for processes and threads in Microsoft Windows memory dumps. Digital forensic research workgroup, 2007.

[10] J. G. Brian D. Carrier. A hardware-based memory acquisition procedure for digital investigations. Technical report, Journal of Digital Investigations, March 2004. http://www.digital-evidence.org/papers/tribble-preprint.pdf.

[11] C. Bugcheck. Grepexec: Grepping executive objects from pool memory. *Report from the Digital Forensic Research Workshop (DFRWS)*, May 2006.

[12] J. Butler and S. Sparks. Shadow walker. raising the bar for rootkit detection. *Black Hat Briefings and trainning Japan*, 2005.

[13] J. Butler and S. Sparks. Windows rootkits of 2005, part one. *Security Focus*, November 2005. http://www.securityfocus.com/infocus/1850.

[14] J. Butler and S. Sparks. Windows rootkits of 2005, part two. *Security Focus*, November 2005. http://www.securityfocus.com/infocus/1851.

[15] P. J. Butler J., Undercoffer J. Hidden processes: The implication for intrusion detection. In *IAW*, pages 116–121. IEEE, 2003.

[16] H. Carvey. *Windows Forensics Analysis*. Syngress, 2007.

[17] M. W. S. T. Center. Deviceobject.

[18] A. Gostev. Malware evolution: January to july 2007. Technical report, Kaspersky Lab, October 2007. http://www.viruslist.com/en/analysis?pubid=204791966.

[19] Joanna Rutkowska. Beyond The CPU: Defeating Hardware Based RAM Acquisition Tools (Part I: AMD case. Black Hat Japan, February 2007.

[20] J. D. Kornblum. Exploiting the rootkit paradox with windows memory analysis. Technical report.

[21] B. M. An introduction to windows memory forensic. Technical report, July 2005. http://forensic.seccure.net/pdf/.

[22] Nicole Lang Beebe, Jan Guynes Clark. Digital forensic text string searching. Digital forensic research workgroup, 2007.

[23] F. B. of Investigation. Digital evidence: Standards and principles. forensic science communications. *Forensic Science Communications. Scientific Working Group on Digital Evidence (SWGDE)*, 2, April 2000. http://www.fbi.gov/hq/lab/fsc/backissu/april2000/swgde.htm.

[24] G. Palmer. A road map for digital forensic research. Technical report, Report from the Digital Forensic Research Workshop (DFRWS), November 2001.

[25] P. D. R. L. Pimenidis. Targeting physically addressable memory. Technical report, September 2007. http://www.springerlink.com/content/84px763856x48vv2.

[26] Péter Ször. Memory scanning under windows NT. Virus Bulletin Conference, September 1999.

[27] R. Richardson. 2007 csi computer crime and security survey. Technical report, Computer Security Institute, 2007. http://gocsi.com/forms/csi_survey.jhtml.

[28] N. Ruff. Windows memory forensics. *Journal in Computer Virology*, November 2007.

[29] A. Schuster. Notes on computer forensics - international edition. http://computer.forensikblog.de/en/2006/03/dmp_file_structure.html.

[30] E. Skoudis and L. Zeltser. *Malware: Fighting Malicious Code*. Prentice Hall, 2003.

[31] D. A. Solomon and M. E. Russinovich. *Inside Microsoft Windows 2000, Third Edition*. Microsoft Press, 2000.

[32] Windows Hardware and Driver Central.). *Memory Management*, 2005.