On the Identification of Information Extracted from Windows Physical Memory

Funminiyi Olajide, Nick Savage
Department of Electronic and Computer Engineering, University of Portsmouth, UK

Abstract

Forensic investigation of the physical memory of computer systems is gaining the attention of experts in the digital forensics community. Forensic investigators find it helpful to seize and capture data from the physical memory and perform post-incident analysis when identifying potential evidence. However, there have been few investigations which have identified the quantity and quality of information that can be recovered from only the computer system memory (RAM) while the application is still running. In this paper, we present the results of investigations carried out to identify relevant information that has been extracted from the physical memory of computer systems running Windows XP. We found fragments of partial evidence from allocated memory segments. This evidence was dispersed in the physical memory that had been allocated to the application. The identification of this information is useful to forensic investigators as this approach can uncover what a user is doing on the application which can be used as evidence

1. Introduction

Physical memory contains information that may not be found using traditional hard disk forensic investigation tools and techniques. The physical memory of a computer system running Windows XP contains address spaces for both user and kernel processes, free pages that are not currently allocated to any process and cached file system blocks[1]. The acquisition and analysis of the physical memory can reveal facts about the current and past usage of the computer system. In addition to this, memory that has been allocated to applications can be extracted and used to infer what the user has been using the application for; information which may not be visible when using traditional hard drive forensic investigation tools and techniques. A recent workshop on digital forensics highlighted the need in the community for the development of tools and techniques for capturing and analysing the physical memory of computer systems [2]. This is also echoed by Carrier and Grand who stated that analysing the extracted memory dump requires new approaches to be developed [3].

In order to progress the development of physical memory forensic investigation, experiments have been designed to investigate the quantity and quality of information that can be recovered from the physical memory allocated to applications in Windows XP. In this research, the most commonly used Windows XP applications have been identified and 100 measurements of the information retrievable from those applications has been conducted. The results of this investigation will facilitate future investigations and the analysis of data found in the physical memory allocated to applications. The investigation of our experiments is restricted to only the memory allocated to applications. This may become an essential tool for information assurance as well as solving crime and tracing fraud.

2. Related Work

Previous work has included the development of tools to extract or dump the physical memory of computer and to extract relevant partitions from that memory. Although Burdach describes memory extraction tools as not fully developed, [4], some tools do currently exist. Some of these tools require specialist hardware to be added to the computer system in order to extract the memory image. An example of this type of tool has been developed by Garcia [5]. Garcia’s tool is among the few hardware-based memory acquisition methods that use a PCI extension card to dump the memory content to an external device. Hardware based acquisition tools have advantages such as not requiring any additional software to be executed on the computer system when acquiring evidence. However they do need to have the tool installed on every machine which a laborious and expensive process. For this reason there has been much focus on software-based tools.

Msuiche has developed a command line tool that can be used on Windows to extract the physical memory of the computer system [6]. It also reconstructs the virtual address space of the system process and other processes. In an attempt to find the most applicable tool to use, Memory dd developed by ManTech International Corporation was tested [7]. This tool is capable of revealing hidden and terminated processes and threads. Win32dd [8], developed by Solomon and Russinovich, was also tested during our investigation. The final tool tested was Nigilant32 [9]. This tool allows an investigator to preview a memory image and take a snapshot of it. It has a small footprint, using less than 1 MB in memory when loaded and with a minimal impact during acquisition. This tool was used to acquire the physical memory of the computer system in this investigation.

There are only a few tools that can perform memory analysis. Some examples are MemParser [10] and the Volatility Framework [11]. Of these two, the Volatility Framework is more extensive. This tool is capable of performing the analysis on a variety of memory image formats such as DD format, crash dump and Hibernate Dumps. Volatility is able to list OS kernel modules, drivers, open network socket, loaded DLL modules, heaps stacks and open files. Recently, a seminar addressed the need for more sophisticated tools on physical memory acquisition and analysis [12]. This is because external and internal intrusions will continue even in the robust security infrastructures of the best government and industry systems. But the key to successfully preventing and responding to any digital fraud investigations is the sound identification, collection, preservation and analysis of computer evidence.

The workshop of [2], issued a memory analysis challenge to encourage research and tool development in this direction. Therefore, a method of [13] laid emphasis on the importance of forensic live response and event reconstruction methods. The extension of this work relies on the research of application level evidence from physical memory [14]. This research identified the important aspects of memory analysis and proposed an approach for application level evidence.

A recently published paper [15], identified the seven most commonly used application and the aspects of memory analysis on the basis of how much information can be recovered from the memory content of the application. This approach provides prospective evidence regarding the application of memory analysis.

3. Methodology

The first stage of this research was the identification of the most commonly used applications on Windows XP systems. To ensure that the results from these investigations are as applicable as possible, the applications that are commonly used by businesses were identified. Different organisations, including banks, commercial retailers, telecommunication companies and public sector organisations were asked which applications were most commonly used on their systems.

The most commonly used applications identified by this enquiry were Word 2007, Excel 2007, PowerPoint 2007, Outlook 2007, MS Access 2007, Internet Explorer 7.0 and Adobe Reader 8.0. In order to make the results from this investigation as applicable as possible a normal working environment was replicated. At the start of the day the computer under investigation would be turned on and at the end of the business day it would be turned off. When the computer is first turned on, the seven applications that are being investigated were opened. During the day the user will interact with the applications, recording their actions. The physical memory of the computer system would be extracted every 30 minutes using Nigilant32. The user will not use the computer for any other purpose during the investigation period.

The typical activities of the user are shown in Table 1. As can be seen, user actions on each application vary during each time period. In some cases, no user input was made. A series of tests were completed over 6 days until 100 images were captured. As the physical memory in the computer was 2 Gigabytes (GB) this resulted in 200 GB of images being captured. The Volatility framework was used as a basis for processing the physical memory that was extracted. Modules were added and some modules were adapted in the framework for this purpose. The memory that was allocated to the applications was extracted and saved in different files, named according to their process ID.

This results in a file that contains fragments of data that may be considered as evidence. The next stage in processing the data is to identify those fragments extracted from memory that contain evidence which may be used to recreate what the user was using the application for. In order to do this, we developed a program to assert that a certain piece of original user input data is equal to a certain pattern of the application processes that was extracted from the memory dump. In this, we match a string of user input to the extracted memory. An example of the result of this process is shown in Figure 1.

Table 1. User actions during the investigation

Applications	User Action every 30minutes
Word 2007	Write a paragraph of text (including alphanumeric and punctuation characters) containing long and short sentences or do nothing on the document. Save document or do not save document.
Excel 2007	List a set of numbers, draw a graph of the numbers or do nothing, save document or do not save. Input may contain alphanumeric and punctuation characters.
Outlook 2007	Write an email containing a paragraph of text (including alphanumeric and punctuation characters) with long and short sentences. Send and receive email data to and fro or do nothing. Save email or do not save.
PowerPoint 2007	Write slides of text (including alphanumeric and punctuation characters) or do nothing. Save document or do not save.
MS Access 2007	Write text (including alphanumeric and punctuation characters) and numbers on database or do nothing. Save document or do not save.
IE 7.0	Open or click on news. Click backwards or forwards. Save or do not save. Highlight text, search for text or do nothing.
Adobe Reader 8.0	Highlight text, search for text or do nothing. Save document or do not save.

4. Results

The aim of this investigation was to identify the quantity and quality of information that could be recovered from only the computer system memory (RAM) while the application was still running. In order to present the results we generated the following statistics which are indicative of the quantity of information that can be recovered; mean evidence repetition, mean percentage of evidence found and mean length of evidence found in continuous blocks.

Mean evidence repetition is mean number of repeated pieces of evidence extracted. This statistic is important as it states how often evidence is found in memory, indicating the likelihood of finding that evidence. The calculation of this statistic involves counting the number of times evidence appears in the memory allocation of that application.

The approach used to determine the mean percentage of evidence found was to count the characters of user input information within the extracted evidence from the memory. The statistic is calculated by dividing the user input characters by the total count of those characters found in the evidence extracted from the memory.

In calculating the mean evidence in continuous block, we counted the actual length of the original user input within the extracted memory dump of the application. Table 2 describes the quantitative memory analysis result of the commonly used application tested in this paper.

The quality of information can be inferred from the ability to reconstruct the user's input based upon the fragments of evidence identified in memory.

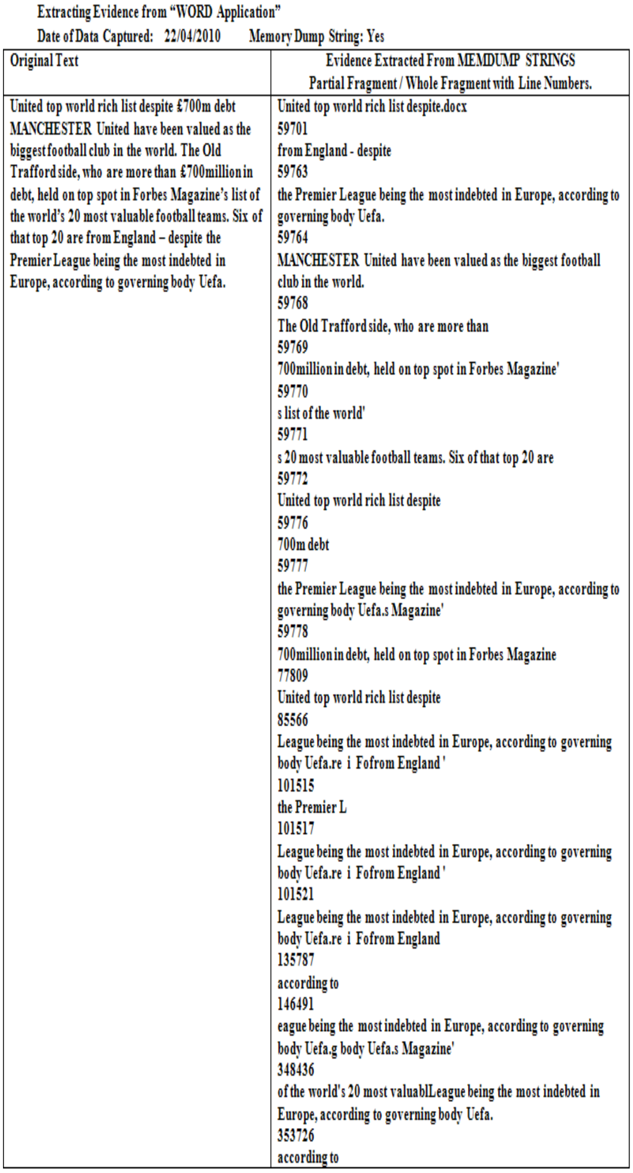
An example of fragments of evidence recovered from memory is shown in Figure 1.0. Based on the way that these sentences overlap, it is possible to reconstruct the evidence found in the memory.

Table 2. Result analysis applications

Seven Commonly used Applications	Mean evidence repetition	Mean % of evidence found	Mean length of evidence found in continuous block
Word 2007	194.70	96	48.65
Excel 2007	62.30	44	21.33
Outlook 2007	110.90	94	51.89
PowerPoint 2007	291.00	95	24.59
MS Access 2007	453.92	39	17.22
IE 7.0	152.00	99	37.40
Adobe Reader 8.0	95.00	35	34.48

The left column of Figure 1, shows the original user information that has been input using the Word application. This is, in turn, used to identify evidence from the extracted application memory. The right-hand column of Figure 1, illustrates the sample evidence of user information that was extracted from the physical memory. This includes the partial and whole fragment of evidence. It illustrated how partial evidence is dispersed in the memory allocated to the Word application. The line number is shown as the location where the evidence resides. The information placed below the line number is the partial fragment of evidence. Some of this evidence was partial while other are located in continuous blocks. This approach was repeated for other applications by reconstructing the evidence extracted from the application memory and match with the original user input.

Figure 1. Pattern matching of known information with memdump evidence



Extracting Evidence from "WORD Application"
Date of Data Captured: 22/04/2010 Memory Dump String: Yes

Original Text	Evidence Extracted From MEMDUMP STRINGS Partial Fragment / Whole Fragment with Line Numbers.
United top world rich list despite £700m debt MANCHESTER United have been valued as the biggest football club in the world. The Old Trafford side, who are more than £700million in debt, held on top spot in Forbes Magazine's list of the world's 20 most valuable football teams. Six of that top 20 are from England – despite the Premier League being the most indebted in Europe, according to governing body Uefa.	United top world rich list despite.docx 59701 from England - despite 59763 the Premier League being the most indebted in Europe, according to governing body Uefa. 59764 MANCHESTER United have been valued as the biggest football club in the world. 59768 The Old Trafford side, who are more than 59769 700million in debt, held on top spot in Forbes Magazine' 59770 s list of the world' 59771 s 20 most valuable football teams. Six of that top 20 are 59772 United top world rich list despite 59776 700m debt 59777 the Premier League being the most indebted in Europe, according to governing body Uefa.s Magazine' 59778 700million in debt, held on top spot in Forbes Magazine 77809 United top world rich list despite 85566 League being the most indebted in Europe, according to governing body Uefa.re i Fofrom England ' 101515 the Premier L 101517 League being the most indebted in Europe, according to governing body Uefa.re i Fofrom England ' 101521 League being the most indebted in Europe, according to governing body Uefa.re i Fofrom England 135787 according to 146491 eague being the most indebted in Europe, according to governing body Uefa.g body Uefa.s Magazine' 348436 of the world's 20 most valuablLeague being the most indebted in Europe, according to governing body Uefa. 353726 according to

5. Analysis

In Table 2.0 there is a high percentage of evidence found for the applications Word, PowerPoint, Email and IE70. However the applications Excel, MS Access, and Adobe Reader show a low percentage of evidence found in the memory. The reason for this is that in Excel and MS Access more user input was numeric data.

This numeric data was difficult to identify in the application memory dump. As seen in Figure 1 partial evidence was extracted from the application's memory dump. The extracted user input on the application was found as allocated and dispersed in the physical memory. For example in Figure 1, user input on the Word application *"United top world rich list despite £700m debt"*. This information was found repeated in the extracted evidence from the physical memory. This evidence is partially dispersed and appears repeatedly as indicated by the line numbers. This evidence information was found repeated in the memory with line numbers shown as *59701, 59776, 59776 and 85566*.

The original user input was reconstructed to form a block chain of evidence in the memory. This evidence was reconstructed in association with the allocated line numbers of where the evidence resides in the physical memory. For example, the first line number *59701* was reconstructed with line number *59777* to form the actual sentence that the original user input.

In this experiment, it is obvious that fragments of information related to what user is doing on the application can be recovered and that these fragments can be coherently combined to form larger examples of evidence.

6. Conclusions

In this research work, the process of securing data from the memory content of windows systems has been described. This includes when images were captured while the application is active and currently running. Both quantitative results and qualitative results related to the partial fragments of evidence found in the memory have been identified. The statistical results were calculated and presented. This includes the mean evidence repetition, mean percentage of evidence found and mean length of evidence found in continuous block.

By reconstructing the evidence found in the physical memory of applications with the allocated line numbers, the original user input information was identified. The amount of relevant evidence obtained from each application was calculated. The approach taken in this research has become part of forensic analysis in digital investigation.

7. References

[1] Schuster A, "Searching for processes and threads in microsoft windows memory dumps. ," Digital Forensic Research Workshop (DFRWS), 2006.

[2] Digital Forensic Research Workshop (DFRWS). (2007, July) http://www.dfrws.org/2007/challenge

[3] Carrier BD, Grand J., "A hardware-based memory acquisition procedure for digital investigations.," The International Journal of Digital Forensics & Incident Response, vol. (1), no. 2, pp. 50-61, February 2005.

[4] M. Burdach, "Windows memory forensic toolkit;," Journal of Information and Computing Systems, vol. (5), no. 2, pp. 45-75, March 2007.

[5] Gabriela Limon Garcia, "Forensic Physical Memory Analysis: an overview of tools and techniques," in TKK T-110.5290 Seminar on Network Security, Helsinki, Finland, 2007.

[6] Msuiche. (Accessed 2008, March) Msuiche.net at:, Capture memory under win2k3 or vista with win32dd. http://www.msuiche.net/2008/06/14/capture-memory-under-win2k3-orvista-with-win32dd.

[7] ManTech Memory. (Aceesed 2010, March) at:, ManTech International Corporation. Memory dd. http://www.mantech.com/msma/MDD.asp

[8] Solomon DA. Russinovich ME, Microsoft Windows internal Covering Windows Server 2008 and Windows Vista, 5th ed. Washington, USA: Microsoft Press, 2009.

[9] Agile Risk Management. (Accessed 2009, October) Nigilat32 small footprint, Agile. [Online]. http://www.agilerm.net/nigilant32

[10] Betz C., "Mempaser analysis tool.," in DFRWS 2005 Forensic Challenge Can be accessed at: http://www.dfrws.org/2005/challenge/memparser.shtml, MA, 2005, pp. 100-115.

[11] Volatile Systems. (2009, April) The Volatility framework: volatlile mwmory artifact extraction utility framework.
http://www.volatilesystems.com/default/volatility

[12] Kleiman D. Carvey H, "Windows Forensic Analysis Incident Response and Cybercrime Investigation Secrets," 1st ed. Syngress Publishing; , July 2007.

[13] Olajide F. Savage N, "Forensic Live Response and Events Reconstruction Methods in Linux Systems," in PGNET The Convergence of Telecommunications Networking and Broadcasting, Liverpool, December 2009, pp. 141-147.

[14] Olajide F. Savage N, "Application Level Evidence from volatile memory," Journal of Computing in Systems and Engineering, December 2009.

[15] Olajide F. Savage N, "On the extraction of forensically relevant information from physical memory," in World Congress on Internet Security (WORLDCIS-2011), Technically Co-Sponsored by IEEE UK/RI Computer Chapter, London, 2011.