Memory Acquisition: A 2-Take Approach

Frank Y.W. Law, Pierre K.Y. Lai, K.P. Chow, Ricci S.C. Ieong,
Michael Y.K. Kwan, Kenneth W.H. Tse, Hayson K.S. Tse
Department of Computer Science
The University of Hong Kong
{ywlaw, kylai, chow, scrieong, ykkwan, whktse, hkstse}@cs.hku.hk

***Abstract*— When more and more people recognize the value of volatile data, live forensics gains more weight in digital forensics. It is often used in parallel with traditional pull-the-plug forensics to provide a more reliable result in forensic examination. One of the core components in live forensics is the collection and analysis of memory volatile data, during which the memory content is acquired for searching of relevant evidential data or investigating various computer processes to unveil the activities being performed by a user. However, this conventional method may have weaknesses because of the volatile nature of memory data and the absence of original data for validation. This may cause implication to the admissibility of memory data at the court of law which requires strict authenticity and reliability of evidence. In this paper, we discuss the impact of various memory acquisition methods and suggest a 2-Take approach which aims to enhance the confidence level of the acquired memory data for legal proceedings.**

***Keywords*— memory forensics, memory acquisition, live forensics**

I. INTRODUCTION

In the conventional approach, we analyze computer memory by preserving its content via dumping of data from the physical memory [2, 4, 6]. Data searching or carving will then be performed on the acquired memory snapshot for locating any meaningful information therein [9, 10]. There also exist various kinds of methodology to the reconstruction of memory process data [11, 12] to assist computer forensic examiners in better understanding the activities of a computer user. Very often, the examination of memory data involved the collection of a single memory snapshot at specific time. The integrity of preserved memory data could not be verified due to the ever-changing internal statuses and volatile nature of the data [14]. Unlike traditional computer forensics which allows verification of the acquired image against the original data, memory forensics could be considered as a one-off examination whereas the integrity of data relies on the techniques and tools that are used at the memory acquisition process. With technology advancement and price-down of Random Access Memory (RAM), computer memory nowadays becomes another data storage location which is highly likely to contain evidential data that does not exist on the computer hard drive. The ruling of the Columbia v Bunnell, 2007 WL 2080419 [1] case from the US District court further suggested that memory content were "electronically stored information" that could be used in court proceedings. However, the traditional concept of "forensically sound" duplicate [16] is no longer applicable to memory image and this may require a new legal definition for memory data before it could be fully admissible at the court of laws. Prior to the creation of such new concept for the current legislation, which is often a lengthy process, there is a need to address the admissibility issue of memory data.

The rest of the paper is organized as follows. Section 2 gives the literature review of the current technologies and practices on memory forensics. Experimental results for various memory acquisition methods and the analysis of acquired memory data are presented in Section 3. Then, we propose a new approach for memory acquisition in Section 4. Section 5 concludes the paper.

II. LITERATURE REVIEW

In these years, the concept of memory forensics is evolving in the field of computer forensics. A lot of research has been conducted in this area with a view to search for viable ways to enhance the effectiveness in the collection and analysis of memory data. Farmer and Venema pointed out the major problem faced in memory analysis is the great variation in the internals of how memory is implemented on various operating systems [17].

Based on different user requirements, a number of live incident response tools have been developed to address the needs [2, 3]. Existing toolkits are often automated programs to be run on the target live system to collect volatile data at the physical memory, whilst the collected data are often recommended to be stored at external storage device to minimize the change to the original system [18]. The credibility of the acquired data relies solely on the reliability of the tool and the expertise of the user. If the tool is run on a compromised system, the integrity of the collected data could hardly be secured [4]. Some tools may even substantially alter the digital environment of the original system and cause an adverse impact to the preserved memory data. As a result, it is often required to study the effect of running such tools and determine if the alterations would affect the acquired data [5, 19].

Carrier and Grand pointed out the potential flaws in acquiring volatile data via the original system and proposed a hardware-based approach for making an accurate and reliable copy of volatile memory contents [4]. Its purpose is to avoid the volatile data being compromised by untrusted code from the operating system or other applications. Antonio further

978-1-4244-4946-0/09/$25.00 ©2009 IEEE

discussed the problems when acquiring live data via a network-based model and suggested using Firewire devices to acquire memory data through the Direct Memory Access (DMA) controller [6]. J. Alex et al. even experimented to retrieve data from physical DRAM via cold boot attack and successfully recovered encryption keys from memory content [20]. There are also suggestions to obtain the image of memory via system hibernation [22] or system crash [23]. Basically, the operating environment and hardware configuration determine which tools and methods to be used and how effectively the methods are in computer forensic analysis of memory [24].

There are many memory researches focus on the proper methods or techniques that should be used in data acquisition and analysis of memory data, few of them discuss the way to verify the integrity and reliability of retrieved evidential data that may be used for legal proceedings. It is generally accepted that the key in validating the authenticity and integrity of data rests with the acquisition process. Provided the acquisition process preserves a complete and accurate representation of the original data [21], together with a good documentation, the data extracted from such acquired image should be forensically sound. The ruling in Lorraine v Market American Insurance Company, 2007 WL 1300739 [25] laid out the requirements on the admissibility of digital evidence and emphasize the importance of authentication to the data. The United States Department of Justice also pointed out the main challenge for the admission of digital evidence is authenticity [26]. It is therefore important to have a methodology to aid computer forensic examiners in memory acquisition process to ensure memory data collected could be verified.

III. EXPERIMENTS

The focus of this paper is to explore the way to enhance the confidence level of memory data acquired from live computer memory. In the following experiments, we are going to evaluate the data change in memory snapshots collected continuously to study the effect of running different memory acquisition tools on the computer memory content.

The testing platform consists of a Windows XP Service Pack 2 machine running on an Intel Pentium 4 3.0 GHz CPU with 512mb of RAM. The virtual memory is set to the default system managed size setting and a firewire interface card is installed on the testing platform to allow direct memory access to the memory. We set up a Windows 2003 Service Pack 2 server on a network to provide accessible web resources to the testing platform.

A. Physical Memory Acquisition

The first experiment relates to the testing of different kinds of memory acquisition tools that could be used for collecting memory snapshots. The purpose of the experiment is to assess to what extent a tool would tamper the memory content during the acquisition process. To simulate a dynamic environment, we collected multiple memory snapshots from the testing platform in the context of downloading a large text file from a Windows server via the File Transfer Protocol (FTP) program came along with the Windows System. Apart from the FTP process, none of the tasks was run by the user. There were a total of twenty-one memory processes running on the system. In the controlled environment, the memory content was acquired by the tools listed in Table 1. A total of ten physical memory snapshots were collected one by one immediately after the previous dumping process. An external USB hard drive was used to save the dumped memory images. The time gap between two consecutive dumps was around 20 seconds Every two dumps were analyzed in the way that each byte in one dump was compared with the corresponding byte in the other dump. After the testing of an individual tool, the system was rebooted. The FTP program would be initiated again to download the text file, another acquisition tool would then run on the testing platform and perform the same procedure. Table 1 shows the results of the experiment.

TABLE I. TESTING RESULT OF MEMORY ACQUISITION TOOLS (MEMORY USAGE IS MEASURED IN KB)

Tools	Avg. Memory Usage	Peak Memory Usage	Avg. Virtual Memory Usage	Peak Memory Usage	Avg. % change between memory images
Nigilant32[26]	3300	6836	920	2272	50.8%
Memdump [27]	2836	3425	1156	1204	45.8%
MemDD [28]	2238	2250	49	52	39.0%
FauDD [29]	1544	1544	470	472	50.9%
KntDD [30]	5104	5152	1952	2452	3.6%
Win32DD[31]	1168	1168	292	294	47.4%
X-Ways Capture [32]	1624	1636	444	444	49.4%
F-Response [33]	3720	4508	1956	3260	6.1%
Memoryze [34]	7912	10052	4764	4792	55.6%
1394memimage* [35]	NA	NA	NA	NA	2.3%

* The 1394memimage tool access computer memory through DMA and does not create any memory process at the testing platform.

It is observed that some memory tools cause less impact to the collected memory content. Some memory acquisition tools are more invasive than the others and their general alterations to the memory image content collected at various time intervals are over 35% on average. After comparing the individual imaged memory content, we discovered that some of the acquisition tools had shuffled the memory data in the context of memory dumping, causing movement of data to the content of consequent acquired snapshots.

It is worthy to note that software acquisition tools cause more changes to the memory images than the hardware approach. Should the software acquisition tools fail to acquire the memory image at the first attempt, it should be aware that the original content of memory data may be changed. Though the hardware acquisition method has very little intrusiveness, it would easily crash the target system.

It is interested to note that there exists persistent data at the memory images which could be identified at consecutive dumps. These kinds of persistent data could be validated between consecutive memory images and the integrity of data found within these areas could be assured.

B. Logical Memory Process Acquisition

Similar experiments in Section 3.1 were conducted on a number of common dynamic/static memory processes to observe any implications by the memory acquisition tools. Given that there are not many tools in the market specifically designed to dump logical process memory, we have tested pmdump [13], and X-Ways Capture in the experiment. Table 2 shows the percentages of data change in logical memory process dumps collected by various tools.

TABLE II. TESTING RESULT OF LOGICAL MEMORY ACQUISITION

Process	Size of memory process dump (mb)	Average % change between images	
		pmdump	*X-Ways capture*
System	0.8	0.0001%	0.0001%
Smss.exe	1.0	0.021%	0.0019%
Services.exe	34.2	0.0063%	0.0078%
Winlogon.exe	33.0	0.0068%	0.0072%
ftp.exe	25.5	4.95%	5.1%
Iexplore.exe*	68.3	8.3%	7.9%
Msnmsgr.exe*	46.6	3.6%	3.9%

* The percentage of change is calculated with respect to the size of the latest memory dump

It is observed that the percentage of difference is relatively small, suggesting that only a little portion of data was changing in the context of logical memory process acquisition. It seems that logical memory acquisition caused fewer changes to the memory content when compared with the physical memory acquisition, suggesting logical memory acquisition tools are less invasive. With the existence of large amount of persistent data, the chance of extracting reliable memory data should be much higher than physical memory acquisition.

By analyzing consecutive memory process snapshots, the major data changing area was identified. Fig. 1 demonstrates the location of that portion in the process memory dump.

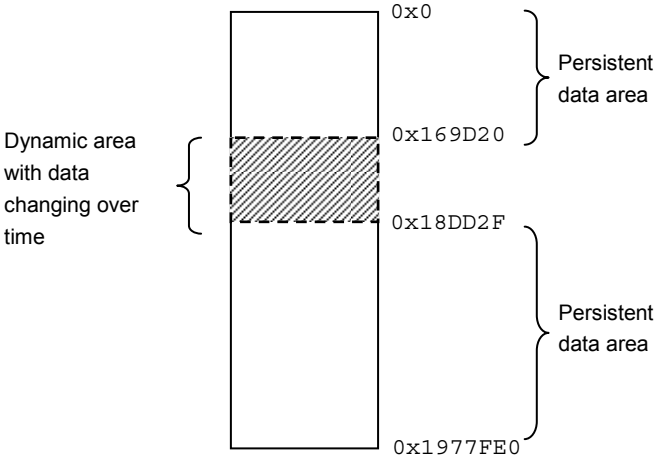

Figure 1. Memory buffer map of a process memory dump against a FTP program

It is believed that the identified portion of data is the memory buffer of the FTP program that is used for handling dynamic data received from the server. The file being downloaded from the server side is a text file with content extracted from a story book. We have examined the content at the memory buffer location of each memory process snapshot and successfully identified the corresponding content of the downloaded text file. Similar experiments were conducted on Internet Explorer version 7 as well as Windows Live Messenger when the subject text file was transferred between the testing platform and another computer host. Very similar results have been obtained, suggesting that we should be able to differentiate both the persistent and dynamic data area within a memory process snapshot.

IV. FORENSICALLY SOUND MEMORY ACQUISITION

There are many definitions on "forensically sound" duplicate when performing the traditional disk-to-disk imaging from "dead" computer system. However, from the above experiments, the acquisition of memory data is very likely to change the memory content and the traditional definition seems not to be applicable nowadays.

From the experimental results, it is suggested that there exists persistent memory content at consecutive memory snapshots. The data recovered from this persistent area may be verified and could be more trustworthy than the remaining dynamic and inconsistent memory content in the unmatched areas. When evidential digital data is retrieved in these consistent areas, its reliability could be higher than the remaining data at the same acquired image. However, one should note that memory dumps should be collected in short time intervals or the consistent data area would be insignificant due to the dynamic nature of the memory [5].

A. The 2-Take Method

Though obtaining multiple memory snapshots could achieve the purpose of data authentication on memory data, it can be tedious and time-consuming in practice. To compass a similar effect, a 2-Take method for memory acquisition is suggested. In a physical crime scene, a forensic specialist will first take a picture of the evidence to be collected, followed by the actual evidence collection. We believe such two-step approach could be adopted in digital evidence collection at computer memory:

1. Take a first snapshot of the current memory content to be used for future verification.
2. Take a second snapshot of the memory content to be used for computer forensics examination.

To achieve this, the time interval between the two memory snapshots to be obtained should be kept as small as possible. The snapshots are then compared to identify the persistent data set in which the data recovered in these areas could be validated on its accuracy and completeness. Since the evidential data recovered from these areas could be authenticated, it is considered to be forensically sound and could be presented to the court of law as evidence. For data recovered from the remaining dynamic data area, it relies on the conventional memory forensic methodology, i.e. the assurance of the forensic examiner who acquired the memory

image, to prove its reliability. In any case, to decide the most appropriate approach in collecting memory data from a running computer, the examiner should take into account the effect of various memory acquisition tools and circumstances at live investigation scene.

V. CONCLUSIONS

Memory forensics is obviously one of the core parts in live forensics investigation and it is foreseeable that more and more evidential data will be resided in memory in future. During conventional live memory investigation, one can only rely on documentation, the tools used by computer forensic examiners as well as the examiners' expertise to ensure the authenticity and reliability of the acquired memory data. However, the current practice in obtaining a single memory snapshot may be subject to challenge by the defense since the integrity of data could not be verified. In order to utilize memory data in legal proceedings, we must ensure the memory data collected are reliable and could be authenticated. In this paper, we observe the possibility of using two memory snapshots to enhance the reliability of acquired memory data via the identification of persistent data within consecutive memory dumps. It provides sources of authentication so that the confidence level of using the acquired memory data at legal proceedings can be improved.

Furthermore, should multiple memory snapshots be obtained from live computer memory, it may be able to assist computer forensics examiner in doing a thorough analysis on dynamic memory processes, especially in the analysis of buffering data which is transient in nature.

The proposed 2-Take method is easy to perform in practice. It is hoped that such method could supplement the conventional memory forensic analysis by introducing another way to present memory data at legal proceedings. Further research may be conducted to investigate if it is feasible to create a tool for capturing memory content in a continuous manner to assist computer forensic examiners in studying persistent memory data and better understanding the information flow at the memory.

REFERENCES

[1] Judgment from Honorable Jacqueline Chooljian, Verdict of United States District Court, http://i.i.com.com/cnwk.1d/pdf/ne/2007/Torrentspy.pdf

[2] Harlan Carvey, "Windows Forensics and Incident Recovery", Addison Wesley, 2005.

[3] Kevin Mandia, Chris Prosise, and Matt Pepe, "Incident Response and Computer Forensics", McGraw-Hill Osborne Media, 2 edition, 2003.

[4] Brian D. Carrier and Joe Grand, "A Hardware-Based Memory Acquisition Procedure for Digital Investigations", Journal of Digital Investigations, 1(1), 2004.

[5] A. Walters and N. Petroni, Jr, "Volatools: Integrating Volatile Memory Forensics into the Digital Investigation Process", Komoku, Inc., College Park, MD, USA, Jan 2006, http://www.komoku.com/forensics/basic/bh-fed-07-walters-paper.pdf

[6] A. Martin, "FireWire Memory Dump of Windows XP: A Forensic Approach", Boston University, April 2007, http://www.friendsglobal.com/papers/FireWire%20Memory%20Dump%20of%20Windows%20XP.pdf.

[7] III Golden G. Richard and V. Roussev, "Next-generation digital forensics", Communications of the ACM, 49(2):76–80, February 2006.

[8] G. L. Garcia, "Forensic physical memory analysis: an overview of tools and techniques", Helsinki University of Technology, October 2007, http://www.tml.tkk.fi/Publications/C/25/papers/Limongarcia_final.pdf

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

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

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

[12] H. Carvey, "Windows Forensics Analysis", Syngress, 2007.

[13] Pmdump v1.2, http://www.ntsecurity.nu/toolbox/pmdump/

[14] Frank Y.W. Law, K.P. Chow, Michael Y.K. Kwan and Pierre K.Y. Lai, Consistency Issue on Live Systems Forensics, The 2007 International Workshop on Forensics for Future Generation Communication environments (F2GC-07), Jeju Island, Korea, 6 – 8 December 2007. pp. 136-140.

[15] M. Michael, "Forensically Sound Duplicate", http://forensiccomputing.blogspot.com/2006/08/forensically-sound-duplicate.html, August 2006

[16] D. Farmer and W. Venema, Forensic Discovery, Addison-Wesley, p.161, 2005

[17] Scientific Working Group on Digital Evidence, "Best Practice for Computer Forensics", http://www.swgde.org/documents/swgde2007/SWGDELiveCaptureFinal.pdf

[18] S. Iain, E. Jon, T. Theodore and B. Andrew, "Acquiring Volatile Operating System Data Tools and Techniques", http://carloshnrb.googlepages.com/artigoso15.pdf

[19] J. Alex Halderman et al., "Lest We Remember: Cold Boot Attacks on Encryption Keys", http://citp.princeton.edu/pub/coldboot.pdf

[20] Eoghan Casey, "What does "forensically sound" really mean?", Journal of Digital Investigation, Vol.4 (2007), pp. 49-50

[21] S. Matthieu, "Sandman Project", http://sandman.msuiche.net/docs/SandMan_Project.pdf, Feb 2008.

[22] K. Michael, "Red Hat, Inc's network console and crash dump facility", http://www.redhat.com/support/wpapers/redhat/netdump/

[23] H. Ewa, B. Derek, H. Francs and W. Mark, "Persistent Systems techniques in forensic acquisition of memory", Journal of Digital Investigation, Vol. 4 (2007), pp.129-137

[24] Lord Bissell Brook LPP, "From E-Discovery to E-Admissibility?", http://www.lordbissell.com/Newsstand/2007-06_EDiscovery_Neiditz_Hatfield_Safer.pdf

[25] Computer Crime and Intellectual Property Section, Criminal Division, United States Department of Justice, "Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations", http://www.usdoj.gov/criminal/cybercrime/s&smanual2002.htm#_VB1_, July 2002

[26] Nigilant32 Beta v0.1, http://www.agilerm.net/publications_4.html

[27] Memdump v2.0, http://www.tssc.de/index.htm#

[28] Memory DD v1.3, http://www.mantech.com/msma/MDD.asp

[29] Forensics Acquisition Utilities v. 1.3.0.2378(beta2), http://gmgsystemsinc.com/fau/

[30] KntDD v1.16, http://gmgsystemsinc.com/knttools/

[31] win32dd v1.2.1.20090106, http://win32dd.msuiche.net/

[32] X-Ways Capture v1.18, http://www.x-ways.net/capture/index-m.html

[33] F-Response Field Kit Edition v1.18, http://www.f-response.com/

[34] Mandiant Memoryze v.1.2.18.0, http://www.mandiant.com/software/memoryze.htm.

[35] 1394memimage, http://www.storm.net.nz