Study on Forensic Analysis of Physical Memory

Liming Cai, Jing Sha ,Wei Qian
Key Laboratory of Information Network Security, Ministry of Public Security,
People's Republic of China The 3rd Research Institute of Ministry of Public Security. Shanghai, China
{cailiming, shajing, qianwei}@stars.org.cn

***Abstract*—In this paper we firstly describe the importance of the study on forensics analysis of physical memory. Further we introduce some tools and techniques commonly used in forensics analysis of physical memory. Lastly we present an example of forensic analysis to illustrate how to do physical memory forensics and analysis in a windows system by using existing tools.**

Keywords- computer forensics; physical memory forensic; physical memory analysis

I. INTRODUCTION

As the development of computer technology, computer crime has become an increasing problem. In such circumstance, more and more attention has been paid on the computer forensic. Analyzing system memory for artifacts is a technique of computer forensic used by forensic analysts, security specialists and those that analyze malware.

Since a considerable part of the computer crime evidence can not be extraction from permanent storage medium such as system's hard disk. We must access the computer system's physical memory to find more important information, such as the intruder's IP address, information about the running malicious program's processes, worms, trojans and so on. We also could extract some sensitive information from e-mail and instant messaging tools through the analysis of the dumped physical memory. But once computer system is turned off, all these information will be lost. Therefore, the research on forensics and analysis of physical memory has practical significance on the promotion of the development of computer forensics technology.

Foreign research on the computer's physical memory forensic began in 2005. However, the study in this field in our country is still very weak. And so far very few domestic research papers about forensics analysis of physical memory were published.

In this paper we will introduce the key technology of physical memory forensics and some common software and hardware tools used in physical memory forensics. Then we will describes how to obtain a complete copy of system memory from a live computer using program "Dumplt", and how to analyze this memory dump with the popular memory analysis tool Volatility. Finally, we will present some problems and further work about the current research of physical memory forensics.

II. TOOLS AND TECHNIQUES ABOUT FORENSICS ANALYSIS OF PHYSICAL MEMORY

A lot of information on computer such as the system process, network information, login information, registry information and so on must be accessed by obtaining and analyzing the target computer's physical memory dump. In this section we will introduce some tools and techniques commonly used in forensics analysis of physical memory to find the important information.

The key technology of forensics analysis of physical memory mainly includes two aspects: acquisition of physical memory and analysis of collected physical memory. That is to say, firstly how to obtain physical memory and generate physical memory image file; secondly how to find out important evidence through the analysis of physical memory image file.

A. Acquisition of physical memory

There are two approaches to acquire physical memory images: hardware based tools and software based tools.

1) Hardware based tools: The hardware based approach 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.

One is a dedicated PCI card named Tribble, which 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. The advantage of this approach is the ease of use and the null impact on the system. The biggest disadvantage of this approach is that the hardware must be pre-installed. So the device can't yet be widely used.

Another is FireWire bus, also known as IEEE 1394 bus. Investigators can obtain the system's physical memory mirroring by utilizing the special properties of FireWire device. FireWire devices with the technology of direct memory access (DMA) can directly access the system memory without CPU. The advantage of this approach is the FireWire port is a popular port present in many systems. And the data transfer speed is very fast through FireWire port. The disadvantage of this approach is the generation of physical memory mirroring may cause system crash or lose some information in memory.

2) Software based tools: There are several software tools used in obtaining physical memory image. Here we will introduce two commonly used software tools.

Data Dumper (DD) is the most commonly used software tools in UNIX system. It can be used in copying files or creating images. GMG System has released a free download forensic acquisition tool bag so that it can run on Windows system and generate the system's physical memory dump.

DumpIt is a compact portable tool which makes it easy to save the contents of PC's physical memory. It's a console utility, but there's no need to open a command line, or master a host of cryptic command line switches. Instead, all you do is double-click the program's executable, press "Y" to confirm that you're "sure you want to continue". And that's it, DumpIt will save the contents of RAM to a file in DumpIt's current folder. We will use this tool in the following example.

B. *Analysis of physical memory*

The next work is to analyze the obtained physical memory dump. We should extract useful evidence in the memory dump. Commonly, we could extract the following information in the memory dump.

- processes running in memory
- loaded module and DLL, including implanted malicious programs
- system's registry information

The information listed above is most commonly concerned by forensic investigators. Of course, we also could analyze other data in the memory dump to get other information. In next section we will present an example of forensic analysis of physical memory to illustrate how to acquire the information listed above by using existing tools.

III. EXAMPLE OF FORENSIC ANALYSIS

A. *Obtaining a memory dump with "DumpIt"*

Simply download DumpIt [6], put it onto a USB drive or save it on your hard drive, double click it, select yes. And then we obtain a complete copy of machine's memory (See Figure 1). The file extension of the image file generated by DumpIt is *raw.

The only thing we need to notice is that it is large enough to hold the file that is created if using a USB drive. The memory dump will be a little larger than the size of your installed RAM.

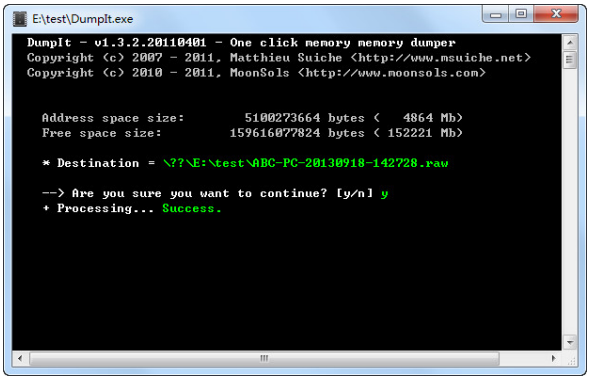


Figure 1. Creating a memory dump file with DumpIt

B. *Analyzing a memory image with "Volatility"*

Once we have the memory dump saved, we can now analyze it with Volatility[7]. First, we should view the summary information of the memory dump obtained by using imaginfo command. Specifically, this command can display the host operating system version, service packs and hardware architecture (32-bit or 64-bit) and the time of the generation of the memory dump and other basic information(See Figure 2). For now, we just need to know the profile type of the memory dump, in this case Win7SP1x64. We will use this in the next few steps.

```
E:\test>volatility imageinfo -f ABC-PC.raw
Volatile Systems Volatility Framework 2.2
Determining profile based on KDBG search...

          Suggested Profile(s) : Win2008R2SP0x64, Win7SP1x64, Win7SP0x64,
8R2SP1x64
                     AS Layer1 : AMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (E:\test\ABC-PC.raw)
                      PAE type : PAE
                           DTB : 0x187000L
                          KDBG : 0xf800047f40a0L
          Number of Processors : 4
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0xf800047f5d00L
                KPCR for CPU 1 : 0xf880009e9000L
                KPCR for CPU 2 : 0xf88004764000L
                KPCR for CPU 3 : 0xf880047d5000L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2013-09-16 12:48:37 UTC+0000
     Image local date and time : 2013-09-16 20:48:37 +0800

E:\test>
```

Figure 2. Execution of imaginfo command

1) Analyzing process list. Now let's take a look at recovering a list of the running processes and active network connections from the captured memory file. Using Volatility's "pslist" command can be used to view the processes that were running on the Windows system

```
E:\test>volatility pslist -f ABC-PC.raw --profile=Win7SP1x64
Volatile Systems Volatility Framework 2.2
Offset(V)          Name                    PID   PPID   Thds     Hnds   Sess  Wo
w64 Start                 Exit
------------------ -------------------- ------ ------ ------ -------- ------ ---
--- ------------------- -------------------

0x0000fa80039e1040 System                    4      0    137      929 ------
  0 2013-09-16 11:07:39
0x0000fa8004dd59d0 smss.exe                336      4      2       32 ------
  0 2013-09-16 11:07:39
0x0000fa8004cba060 csrss.exe               484    428     11      822      0
  0 2013-09-16 11:07:43
0x0000fa800581e330 wininit.exe             552    428      3       83      0
  0 2013-09-16 11:07:44
0x0000fa8005abe060 csrss.exe               584    564     15      689      1
  0 2013-09-16 11:07:44
0x0000fa8005b02b30 services.exe            608    552      7      257      0
  0 2013-09-16 11:07:44
0x0000fa8005b4bb30 lsass.exe               632    552     11      797      0
  0 2013-09-16 11:07:44
0x0000fa8005b44b30 lsm.exe                 640    552     11      171      0
  0 2013-09-16 11:07:44
0x0000fa8005b92b30 winlogon.exe            696    564      5      122      1
  0 2013-09-16 11:07:44
0x0000fa8005c33b30 svchost.exe             788    608     12      382      0
  0 2013-09-16 11:07:45
```

Figure 3. Execution of pslist command

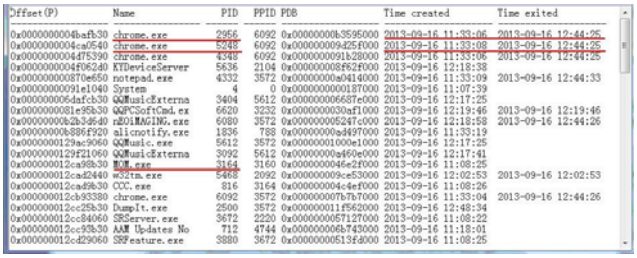
From the output of the command (See Figure 3), we see the physical memory location, process name and the PID number of all process that were running.

But pslist command is not able to detect hidden processes in memory and process information disappeared in process list as the system is attacked. Psscan command can solve this problem. Psscan command can list all processes in system through the technology of memory pool tag finding. This command is not only able to display the current active process information in memory, but also is able to display

terminated process and hidden process. And it is the hidden process in memory forensics has important reference value.

```
E:\test>volatility psscan -f ABC-PC.raw --profile=Win7SP1x64 >psscan.txt
Volatile Systems Volatility Framework 2.2
```

Figure 4. Execution of psscan command



Offset(P)	Name	PID	PPID	PDB	Time created	Time exited
0x0000000004bafb30	chrome.exe	2956	6092	0x000000000b3595000	2013-09-16 11:33:06	2013-09-16 12:44:25
0x0000000004ca0540	chrome.exe	5248	6092	0x000000009d25f000	2013-09-16 11:33:08	2013-09-16 12:44:25
0x0000000004d75390	chrome.exe	4348	6092	0x000000009fb28000	2013-09-16 11:33:06	2013-09-16 12:44:25
0x0000000004f062d0	KTDeviceServer	5636	2104	0x000000008f62f000	2013-09-16 12:18:38	
0x0000000003f70ef650	notepad.exe	4332	3572	0x000000000a0414000	2013-09-16 11:33:09	2013-09-16 12:44:33
0x000000009fe1040	System	4	0	0x0000000000187000	2013-09-16 11:07:39	
0x0000000006dafcb30	QQMusicExterna	3404	5612	0x000000006687e000	2013-09-16 12:17:25	
0x0000000081e95b30	QQPCSoftCmd.ex	6620	3232	0x000000003af1000	2013-09-16 12:19:46	2013-09-16 12:19:46
0x0000000002b3d640	qDUMAGING.exe	6080	3572	0x000000005247c000	2013-09-16 12:18:58	2013-09-16 12:44:26
0x0000000006886f920	alicnotify.exe	1836	788	0x000000009d497000	2013-09-16 11:33:19	
0x0000000129ac9060	QQMusic.exe	5612	3572	0x000000001000e1000	2013-09-16 12:17:25	
0x0000000129f21060	QQMusicExterna	3092	5612	0x000000009a460e000	2013-09-16 12:17:41	
0x0000000012ca95b30	WMI.exe	3164	3160	0x000000004e2f000	2013-09-16 11:08:25	
0x0000000012cad2440	w32tm.exe	5468	2092	0x000000009ce53000	2013-09-16 12:02:53	2013-09-16 12:02:53
0x0000000012cad9b30	CCC.exe	816	3164	0x000000004c4ef000	2013-09-16 11:08:26	
0x0000000012cb93380	chrome.exe	6092	3572	0x000000007b7b7000	2013-09-16 11:33:04	2013-09-16 12:44:26
0x0000000012cc25b30	DumpIt.exe	2500	3572	0x000000011f562000	2013-09-16 12:48:34	
0x0000000012cc84060	SRServer.exe	3672	2220	0x000000005712f000	2013-09-16 11:08:22	
0x0000000012cc95b30	AAU Updates No	712	4744	0x000000006b743000	2013-09-16 11:18:01	
0x0000000012cd29060	SRFeature.exe	3880	3672	0x000000005134d000	2013-09-16 11:08:25	

Figure 5. Output of psscan command

Part of the results after running psscan command showed in Figure 5. We can discover what actions users have done by analyzing the condition of three running process which are underlined in Figure 5. User opened the chrome browser at 2013-9-16 11:33:06. After 2 seconds the user opened the chrome browser again at 2013-9-16 11:33:08. The user closed the chrome browser at 2013-9-16 12:44:25. User ran the process of MOM.exe at 2013-9-16 11:08:25. The process has been running since the moment of obtaining the memory mirroring. And the process ID is 3164.

2) Analyzing information of DLL. What is MOM.exe? Is it a virus program? Now we may view DLL files loaded in the specified process and then learn more about what the procedure MOM.exe is.

We could view information of DLL in memory by dllllist command. If we only need to look at loaded DLL files of a specific process instead of all processes, we can use –p PID option to filter the output.

```
E:\test>volatility psscan -f ABC-PC.raw --profile=Win7SP1x64 >psscan.txt
Volatile Systems Volatility Framework 2.2

E:\test>volatility dlllist -p 3164 -f ABC-PC.raw --profile=Win7SP1x64
Volatile Systems Volatility Framework 2.2
************************************************************************
MOM.exe pid:   3164
Command line : "E:\Program Files\ATI Technologies\ATI.ACE\Core-Static\MOM"


Base                             Size Path
------------------ ------------------ ----
0x00000000009b0000            0x12000 E:\Program Files\ATI Technologies\ATI.ACE\Core-Static
0x00000000076ff0000            0x1a9000 C:\Windows\SYSTEM32\ntdll.dll
0x000007fefa330000             0x6f000 C:\Windows\SYSTEM32\MSCOREE.DLL
0x00000000076ed0000            0x11f000 C:\Windows\system32\KERNEL32.dll
0x000007fefd0f0000             0x6c000 C:\Windows\system32\KERNELBASE.dll
0x000007fefd6c0000             0xdb000 C:\Windows\system32\ADVAPI32.dll
0x000007fefdfb0000             0x9f000 C:\Windows\system32\msvcrt.dll
0x000007fefd4a0000             0x1f000 C:\Windows\SYSTEM32\sechost.dll
0x000007fefdaf0000            0x12d000 C:\Windows\system32\RPCRT4.dll
0x000007fefa230000             0x70000 C:\Windows\Microsoft.NET\Framework64\v4.0.30319\mscor
0x000007fefd4c0000             0x7f000 C:\Windows\system32\SHLWAPI.dll
0x000007fefd650000             0x67000 C:\Windows\system32\GDI32.dll
0x00000000076dd0000             0xfa000 C:\Windows\system32\USER32.dll
```

Figure 6. Execution of dllllist command

From the output of the command (See Figure 6), we can see the path of the program MOM.exe and all the loaded DLL files with its base address when the process is running. Now we could determine the program MOM.exe is not a virus program. It is normally just a part of a program called the ATI Catalyst Control Center. We could know more about the running process or do more another analysis by executing dllllist command.

3) Analyzing registry keys. Registry plays a very important role in configuration and control of a computer's operating system. It is primarily used to store the configuration information of the operating system and user's application. Registry is not a simple large file, but a collection. Registry file mainly include two categories, the first category stores static data on the disk. Such registry files are generally stored in the operating system system32\config folder. Second category stores volatile registry data. Such registry file is the data in RAM maintained by the Windows kernel which will be created at every system startup time .

```
E:\test>volatility hivelist -f ABC-PC.raw --profile=Win7SP1x64
Volatile Systems Volatility Framework 2.2
Virtual            Physical           Name
------------------ ------------------ ----
0x0000f8a00f1a0010 0x00000000c6ce5010 \??\C:\System Volume Information\Syscache.
hve
0x0000f80004b47b10 0x0000000004b47b10 [no name]
0x0000f8a00000d010 0x000000000a802b010 [no name]
0x0000f8a000024010 0x000000000a80ac010 \REGISTRY\MACHINE\SYSTEM
0x0000f8a000065010 0x000000000a7fed010 \REGISTRY\MACHINE\HARDWARE
0x0000f8a00011e010 0x00000000975b2010 \SystemRoot\System32\Config\DEFAULT
0x0000f8a000b87410 0x000000000a429d410 \Device\HarddiskVolume1\Boot\BCD
0x0000f8a000d22010 0x000000000a3981010 \SystemRoot\System32\Config\SOFTWARE
0x0000f8a001297010 0x000000008ee4d010 \??\C:\Windows\ServiceProfiles\LocalServic
e\NTUSER.DAT
0x0000f8a0012d2010 0x000000008581f010 \SystemRoot\System32\Config\SECURITY
0x0000f8a001379410 0x000000091e5a410 \SystemRoot\System32\Config\SAM
0x0000f8a0015b3010 0x000000091ee5010 \??\C:\Windows\ServiceProfiles\NetworkServ
ice\NTUSER.DAT
0x0000f8a00205f410 0x000000005d334410 \??\C:\Users\abc\ntuser.dat
0x0000f8a002604010 0x000000005d535010 \??\C:\Users\abc\AppData\Local\Microsoft\W
indows\UsrClass.dat

E:\test>
```

Figure 7. Execution of hivelist command

The results after running hivelist command showed in Figure 7. We could locate the virtual addresses and physical addresses in memory corresponding to the registry files on hard drive. If we need to do further analysis of a registry key, we must execute this command previously in order to locate the virtual addresses of the registry key.

In general, forensic investigators can retrieve data from registry related to information about hardware and software configuration, user's preference, information about system's initialization configuration, user login and password information.

IV. CONCLUSION

Forensics analysis of physical memory is relatively a new field of research. The study started late. Although some progress has been obtained, there are still many problems.

Firstly, it is a lack of reliable and practical hardware device to access physical memory. Hardware based method to get the system's physical memory is an ideal solution as it almost doesn't affect system's physical memory. We could obtain a very complete physical memory without irruption by using hardware method. But the present hardware device such as Tribble, FireWire and so on still need further improvement

Secondly, although there are a lot of software tools for acquisition of system's physical memory, software tools will inevitably damage or even override the contents of physical memory. How to improve software tools to make minimize impact on physical memory is our next issue to research.

Thirdly, it is a lack of relatively mature tools for analysis of physical memory. More powerful analysis tools need to be researched and developed.

ACKNOWLEDGMENT

This work was financially supported by the basic science project of Ministry of public security, project number: 2012GABJC035 and National Development and Reform commission, project number: [2012]1424.

REFERENCES

[1] H. Carvey. Windows Forensics Analysis. Syngress,2007.

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

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

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

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

[6] MoonSols DumpIt goes Main stream. http://www.moonsols.com

[7] The Volatility Framework. http://code.google.com/p/volatility/

[8] http://code.google.com/p/volatility/wiki/FAQ

[9] Stefan Vomel,Felix Freiling. A survey of main memory acquisition and analysis techniques for the windows operating system.Digital Investigation 8 (2011)3-22

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

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

[12] Lodovico Marziale Golden G. Richard III Vassil Roussev Andrew Case, Andrew Cristina. Face: Automated digital evidence discovery and correlation. In proceedibgs of the annual DFRWS conference, 5, 2008.

[13] Ali Reza Arasteh and Mourad Debbabi. Forensic memory analysis: From stack and code toexecution history. In proceedings of the annual DFRWS, 2007.

[14] Mariusz Burdach. Digital forensics of the physical memory. March 2005.

[15] Carrier. Joe Grand carrier. A hardware-based memory acquisition procedure for digital investigations. 2004.