Acquisition and Analysis of Physical Memory on Windows Systems

Shijin Chandran R.S, Haseena S
Cyber Forensics & Information Technology
ER&DCI Institute of Technology
C-DAC, Thiruvananthapuram, India

***Abstract*—As with the areas of digital forensics especially in memory forensics, the validity and in some cases the sheer possibility of data analysis depends upon successfully acquisition of data from the memory. The analysis of acquired Random Access Memory has been an active area of recent research. It is an important area of digital forensics especially in incident response, malware analysis and behavior analysis of application and system software in physical memory. The network related information is one of the crucial artifacts that can be collected from the volatile memory, which can't be collected anywhere from the hard disk. Currently available methods use kernel data structure for analyzing the Random Access Memory. Since the kernel variables in Windows 7 differ from other versions of Windows, analyzing Windows 7 memory for collecting crucial evidence is very tedious task. This paper demonstrates a method of memory acquisition invoked via a kernel module-device driver, which provides access to the memory and a method which is suitable for any windows version to collect network packets from memory dump and retrieving important network related artifacts.**

Keywords— memory forensics, sensitive information, livev System, network packets,ethernet frame

I. INTRODUCTION

The Primary Memory or Random Access Memory (RAM) also referred to as main memory is typically considered to be a volatile resource. This variety of information is lost due to volatile nature of physical memory or when the system is turnoff. Therefore analyzing windows physical memory is useful for both criminal investigation and incident response due to importance of available evidences in volatile memory. The Physical memory is accessed through \\device\PhysicalMemory section object. Memory is divided in two areas/modes: user area and kernel area. Operating system (OS) resides in kernel area whereas the user programs reside in user area. According to OS structure, a user program cannot directly access kernel memory. Therefore, memory forensics tools need special privileges to capture/dump kernel memory. A device driver is a particular form of software application which is designed to enable interaction with hardware devices. It has kernel privilege so that ant program accessing physical memory through device driver will get access. This method of memory acquisition fully relies on the driver program created for kernel privilege. So the main research work is on development of such a driver program to support acquisition program. Writing a device driver requires an in-depth understanding of how the hardware and the software works for a given platform function. Because drivers require low-level access to hardware functions in order to operate, drivers typically operate in a highly privileged environment and can cause system operational issues if something goes wrong. In contrast, most user-level software on modern operating systems can be stopped without greatly affecting the rest of the system. The live forensics approach involves gathering possible evidences from running system. It contains details of volatile data such as running processes, logged-in users, current network connections, users' sessions, cryptographic keys, open files, Ethernet frames, residual IP packets etc.

Network related crimes are one of the most critical issues faced by the cybercrime investigators now a day. Network forensics analyses the traffic data captured and logged through intrusion detection systems or firewalls or at other network devices such as routers. In this approach the packet headers analysed which gives an idea about sender and recipient of particular payload, sequence number and acknowledge number of that payload. Also the Ethernet header gives information about MAC addresses of the communicating machines. The same approach can be applied to retrieving the network packets which can be found in the RAM dump by using signature based analysis.

II. EXISTING SYSTEM

There are many known techniques and tools available for capturing a volatile memory image. They can be broadly categorized into Hardware and Software based techniques. This project concentrates mainly on software based acquisition. There are software tools existing for physical memory acquisition like DumpIt, Magnet Live Ram Capture, mdd, Belkasoft RAM Capture etc. The existing tools have got so many limitations in acquiring the physical memory in a forensically sound manner. One cannot say mapping of

a particular section of memory will be a success. Since the acquisition process is as important as memory analysis, these limitations must be taken care of. The development of both applications is still growing and needs support from community in order to create more 'user friendly' options for both applications.There is no tools currently available for analyzing network packets from physical memory dump.

III. SYSTEM DESIGN

A. *Memory Acquisition*

A common technique for acquiring a (raw) forensic copy of a computer's RAM relies on leveraging the \\.\Device\ PhysicalMemory section object that, as the name suggests, provides access to sections of physical memory. For security reasons, however, permissions to open the resource in user space were revoked with the introduction of Microsoft Windows Server 2003 (Service Pack 1, see Microsoft Corporation, 2013a). For this reason, all of the imaging applications we considered for our evaluation did not only consist of a user-mode administration program, but also of a kernel-level driver. The latter typically calls the ZwOpenSection function in the first step to retrieve a handle to the \\.\Device\PhysicalMemory object[1]. After determining the actual size of memory, portions of RAM may then be read out page wise, for instance, with the help of the ZwMapViewOfSection function. In the last step, a mapped section can either be directly written to the image file in kernel space or transferred to user space via a buffer for further processing.

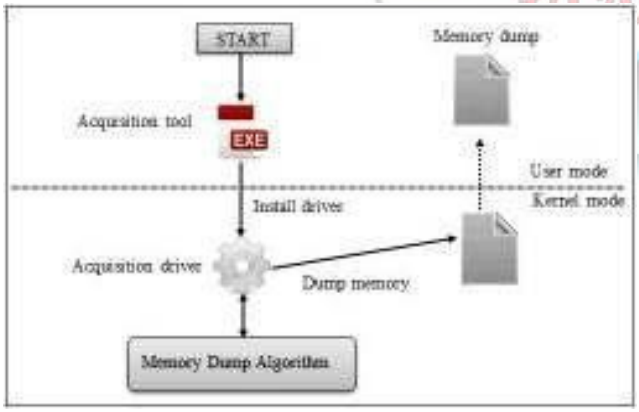


Fig.1 Redline results extracted from

The focus of this research is on extracting data from the volatile random access memory (RAM) on a personal computer running Microsoft's Windows Vista operating system, while minimally affecting the existing data. The research work includes the development of a kernel-mode device driver with the capabilities on one or more versions of Microsoft Windows Vista, a user-mode application that interacts with the driver, and a thesis that documents the development process and the research findings. The device driver is capable of exposing the volatile memory as a read only media (similar to a CD-ROM)[6]. The MemReader tool will also include a user mode application that makes use of this device driver and writes the content of the memory unto a specified storage media.

B. *Memory Dump Analysis*

The dump file is generated in raw format and signature based is performed for carving network information. Analyzing Ethernet header reveals network information like source and destination IP addresses MAC address of communicated machines, sequence and acknowledgement number of packets, type of application protocol used, source and destination port addresses etc.

The Ethernet frame contains Ethernet header and a payload. The Ethernet payload contains IP header and IP payload. The IP payload is the wrapping up of TCP header and data [7]. Almost all devices including Windows systems use the Ethernet version 2 protocols to transmit IP packets by default.

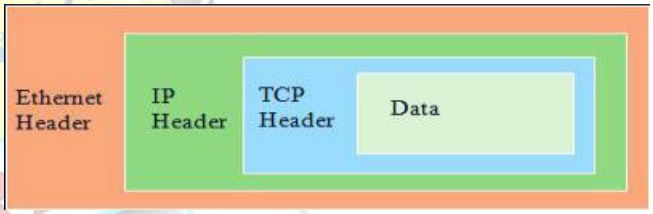


Fig.2 Ethernet Frame Structure

1) Ethernet Frame Header

The original Ethernet IEEE 802.3 standard defined the minimum Ethernet frame size as 64 bytes and the maximum as 1518 bytes. Fig. 3 represents the structure of Ethernet frame header.

Fig.3 Ethernet Frame Header

The protocol type used in the Ethernet frame header gives idea about next level protocol that is contained within the Ethernet packet. 0x 0800 is the signature of IP header, 0x 0806 is the signature of ARP header and 0x86DD denotes it's an IPV6 packet.

2) IP Header Structure

The standard IPv4 header is 20-byte in size.Fig.4 shows structure of IP header.

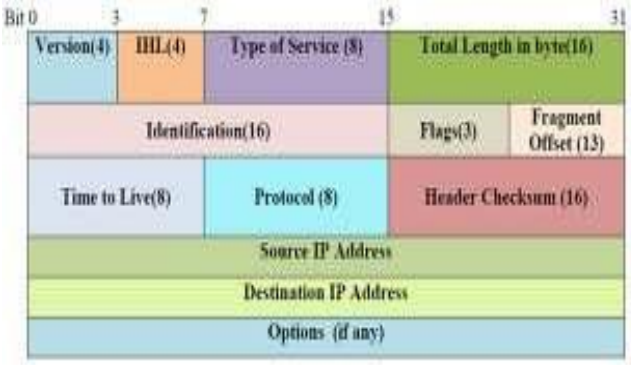


Fig.4 IP Header Structure

Version number specifies the format of the IP packet header. In IPv4, this value is 0x4 and in IPv6, this value is 0x6. The second nibble in the first byte of IPv4 packet header refers the number of 32-bit (4-byte) words in the IP Header and is termed as Internet Header Length. Since the standard IPv4 Header is 20 bytes in length, so this value is 0x5.The type of service field is 8 bit in size. It carries information to provide quality of service features, such as prioritized delivery, reliability, minimum cost and throughput for IP datagrams. The total length specifies the total length of the IP datagram including the header, in bytes. The identification field is says each of the fragments belonging to which message. This field is used by the recipient to reassemble messages without accidentally mixing fragments from different messages. The flag is referred by 3 bits. Two of which are used to manage fragmentation and other is reserved.

The protocol field helps to identify the higher layer protocol carried in the datagram (Either Higher layer protocol or encapsulated network layer protocol).The Figure 4 represents the signatures of protocols. The Hex value 01 means packet follows ICMP protocol, 06 corresponds to TCP and 11 corresponds to UDP.

TABLE-I: Protocol Signature

Value (Hexadecimal)	Value (Decimal)	Protocol
00	0	Reserved
01	1	ICMP
06	6	TCP
04	4	IP- in - IP Encapsulation
11	17	UDP

3) Protocol Header

The Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are core protocols of the Internet protocol suite. Both TCP and UDP work at transport layer of TCP/IP model and both have very different usage.

a) TCP

TCP is the most commonly used protocol on the Internet. The TCP offers error correction and "guaranteed delivery." TCP uses flow control mechanism and it ensures that a fast sender does not overrun a slow receiver. The congestion control mechanism prevents too much data from being injected into the network.

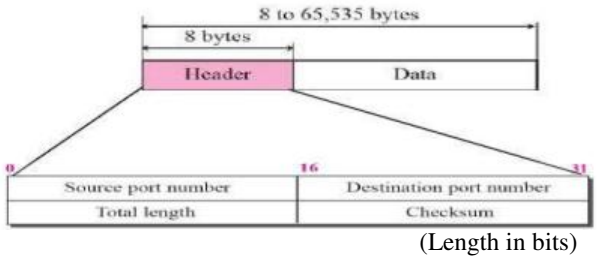
TCP header is 20 byte in size. The Table-II below represents different fields in the TCP header. The sequence number in the packet represent the ID number of the packet and ACK number represents ID number of the next packet to be expected. The urgent pointer refers the packet offset contain urgent data.

TABLE-II: Fields in TCP Header

Fields	Size(in Bits/Bytes)
Source Port	2 Byte
Destination Port	2 Byte
SEQ Number	4 Byte
ACK Number	2 Byte
Header Length	4 Bits
Reserved Bits	4 Bits
Flags	6 Flags
Window Size	2 Bytes
TCP Checksum	2 Bytes
Urgent Pointer	2 Bytes

b) UDP

UDP is another commonly used protocol on the Internet. UDP is commonly used for streaming audio and video. UDP is faster than TCP is because there is no form of flow control or error correction. Any collisions and errors that occurs on the transmission is not a problem on UDP since those will be properly handled by the IP stack. Fig.5 shows the structure of UDP datagram.



(Length in bits)

Fig.5 UDP Datagram

The total length represents the length of the entire UDP datagram, including both header and data fields. An optional 16-bit checksum computed over the entire UDP datagram plus a special “pseudo header” of fields. It is the 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the UDP header, and the

data, padded with zero octets at the end (if necessary) to make a multiple of two octets.

c) ICMP (Internet Control Messaging Protocol)

The Internet Control Message Protocol (ICMP) is one of the main protocols of the Internet Protocol Suite. ICMP differs from transport protocols such as TCP and UDP in that it is not typically used to exchange data between systems, it only used for checking the availability of a particular host or router. It is assigned protocol number 1. ICMP is a protocol useful in Internet Protocol (IP) network management and administration. The ping utility uses ICMP to probe remote hosts for responsiveness and overall round-trip time of the probe messages. When ping a remote computer, there is actually happening sending of a message called echo request, the remote computer then replies to the message, which is called echo reply.

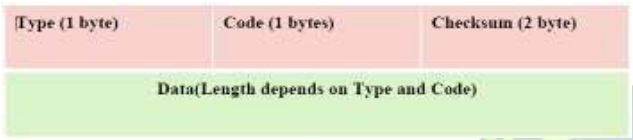


Fig.6 ICMP Header structure

d) Address Resolution Protocol (ARP)

Address Resolution Protocol, a network layer protocol used to convert an IP address into a MAC address. ARP is a non-routable protocol, and can therefore only be used between systems on the same Ethernet network. In the case of ARP packet within the Ethernet frame, the MAC header is followed by the ARP packet data. The length of the ARP packet data is 28 bytes (224 bits). The minimum Ethernet data length is 46 bytes. Since the ARP packet length is less than minimum Ethernet data length, the ARP data will always be followed by at least 18 bytes of padding.

Bit	0-7	8-15	16-31
0	Hardware Type		Protocol type
32	Hardware Len	Protocol Len	Operation
64	Sender HW Addr		
96	Sender HW Addr		Sender Protocol Addr
128	Sender Protocol Addr		Target HW Addr
160	Target HW Addr		
192	Target Protocol Addr		

Fig.7 ARP packet structure

The Opcode field 0001 for ARP Request or 0002 for ARP Reply The field ARP Hardware Type identifies the specific data-link protocol being used. For Ethernet, the value of this field is by default 0001 (Ethernet). ARP Protocol Type by default is IPv4. Link Layer Hardware Address Length is the length of the MAC address and the length of the protocol address is specified in the Network Protocol Address Length field. The Sender Hardware Address represents the MAC address of the machine sending the request. Christo Ananth et al. [3] proposed a secure hash message authentication code. A secure hash message authentication code to avoid certificate revocation list checking is proposed for vehicular ad hoc networks (VANETs). The group signature scheme is widely used in VANETs for secure communication, the existing systems based on group signature scheme provides verification delay in certificate revocation list checking. In order to overcome this delay this paper uses a Hash message authentication code (HMAC). It is used to avoid time consuming CRL checking and it also ensures the integrity of messages. The Hash message authentication code and digital signature algorithm are used to make it more secure . In this scheme the group private keys are distributed by the roadside units (RSUs) and it also manages the vehicles in a localized manner. Finally, cooperative message authentication is used among entities, in which each vehicle only needs to verify a small number of messages, thus greatly alleviating the authentication burden. The protocol address of the machine sending the ARP request is specified in the field Sender Network Protocol Address. Target Hardware Address is the MAC address being sought. Target Network Protocol Address is the protocol address of the destination. For ARP requests, this is the IP address that we wish to find.

08 00 45 xx xx xx xx xx xx xx 06 - IPv4 + TCP 08 00 45 xx xx xx xx xx xx xx 11 - IPv4 + UDP 08 00 45 xx xx xx xx xx xx xx 01 - IPv4 + ICMP

During analysis search for the above signature to retrieve the TCP, UDP& ICMP packet and search 0x 0806 for ARP packet.[7]

IV. IMPLEMENTATION OF ACQUISITION TOOL

A. Background Of Windows Device Drivers

Digitally signed kernel-mode software is a process that attempts to ensure security on computer systems. Microsoft's Windows Vista relies on digital signatures for kernel-mode code to increase the safety and stability of the Windows platform. All 64-bit kernel-mode software intended for x64-basedcomputers running Windows Vista must be digitally signed. This applies to boot start drivers for x86 and x64 versions of Windows. Microsoft however encourages publishers to digitally sign all software, including device drivers for both 32-bit and 64-bit platforms [4].

B. Driver Installation

A driver using the Kernel Mode Driver Framework (KMDF) provided by the WDK is a kernel module that lives permanently inside the system. In many ways, a driver is treated by Windows as a regular service that can be started and stopped just like any other system service. The two steps involved in installing a driver are:

1. Registering the driver as a system service
   The driver file that created is installed into the system and is started as a service.
2. Enabling/ Starting the driver/ service

The Service Control Manager exposes APIs that can be used to install and start a device driver/ service from any win32 application. It is therefore possible to register and start a driver programmatically. The registry values of the name, path and start-type of the driver need to be supplied. TABLE-III shows the registry key values and types created for the new service.

TABLE-III: Registry key values

Registry Value	Type	Data
DisplayName	REG_SZ	<display name>
ImagePath	REG_EXPAND_SZ	<directory path>
Start	REG_DWORD	3
Type	REG_DWORD	1
ErrorControl	REG_DWORD	1

C. Accessing Memory

The Windows memory manager provides a set of routines that kernel-mode drivers use to allocate and manage memory. Kernel-mode drivers allocate memory for purposes such as storing internal data, buffering data during I/O operations, and sharing memory with other kernel-mode and user-mode components. Christo Ananth et al. [3] proposed a secure hash message authentication code. A secure hash message authentication code to avoid certificate revocation list checking is proposed for vehicular ad hoc networks (VANETs). The group signature scheme is widely used in VANETs for secure communication, the existing systems based on group signature scheme provides verification delay in certificate revocation list checking. In order to overcome this delay this paper uses a Hash message authentication code (HMAC). It is used to avoid time consuming CRL checking and it also ensures the integrity of messages. The Hash message authentication code and digital signature algorithm are used to make it more secure . In this scheme the group private keys are distributed by the roadside units (RSUs) and it also manages the vehicles in a localized manner. Finally, cooperative message authentication is used among entities, in which each vehicle only needs to verify a small number of messages, thus greatly alleviating the authentication burden. Memory management is the functionality of an operating system which handles or manages primary memory. Memory management keeps track of each and every memory location either it is allocated to some process or it is free. It checks how much memory is to be allocated to processes. Some of the memory management routines which will help in memory acquisition are listed below. All these routines are reserved for system use and run with kernel privilege which is so helpful for this work.

For the purpose of this research, a kernel mode device driver was developed and loaded into the kernel. The driver opens a handle to the physical memory segment and when the user mode application interacts with the device driver, the driver maps sections of the physical memory into the process's virtual address space. Since a driver is being installed, administrative privileges are required for the user running the tool. The driver then copies the information from the physical memory into the buffer of the application. The application then saves the data on the buffer into an image file in a file system present on the computer system.

The acquisition starts with calling the ZwOpenSection memory manager routine for retrieving a handle to \\device\PhysicalMemory section object. It is followed by MmGetPhysicalMemoryRanges to get number of pages and address range in physical memory. Once we have the number of pages and address rang, we take section by section by calling ZwMapViewOfSection memory routine. It simply maps a view of a section. If the mapping is success then the section is written to output using NtWriteFile routine. If the mapping is failed then that particular area in output file is filled with zeros. Then the page is unmapped and checks whether it is last section in memory. If not execution flow moves back to mapping next section. It is written to output file also. When the process reaches last section, the condition becomes true and loop is exited. After that the handle is closed and resource released.

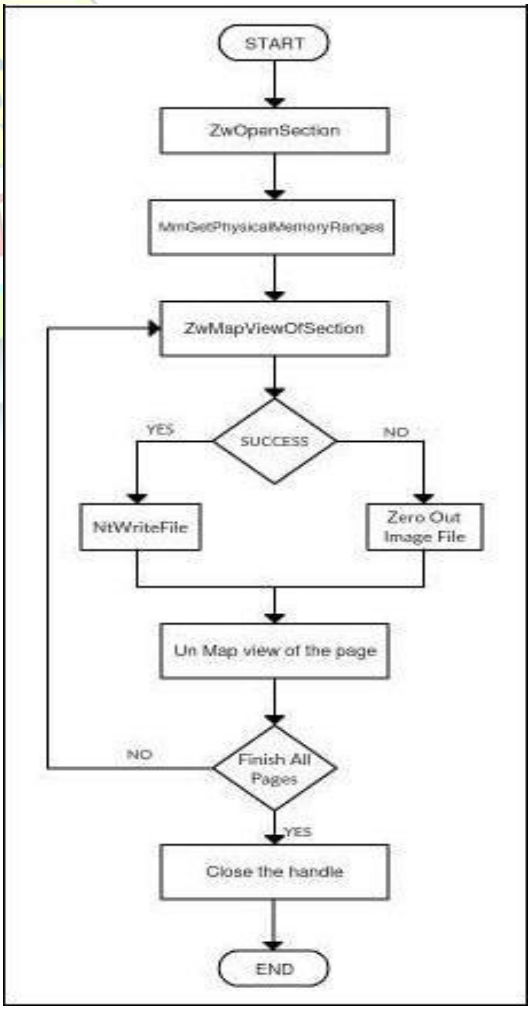


Fig.8 Acquisition flow

All these operations are carried out in escalated privilege mode because of the driver service running. By this acquisition is completed. All routines are kept on disk in a re-locatable load format. The main program is loaded into memory and is executed. The memory management routines will help in memory acquisition process by passing all information from and to the memory. All these routines run with kernel privilege which is so helpful for this work. The acquisition flow diagram as shown in Fig. 8 is a representation of the execution of the acquisition system and depicts the specifications of acquisition.

V. CONCLUSION

The physical memory of a computer is one of the most secure places of data storage. Acquiring contents of physical memory in a forensically sound manner and with improved acquisition rate is a challenging task. Latest versions of windows claims to be the most secure version of windows yet, thereby causing the forensic investigations a tedious one. Resources like physical memory are accessed in a highly secure manner. A device driver program is the main part of the proposed system. So with a suitable device driver, it is possible for a user mode process to create an image of RAM. The research work includes the development of a kernel-mode device driver to get access to RAM. By this technique, memory acquisition process will become more efficient which will make analysis part effective.

References

[1] Stefan Vömel, Johannes Stüttgen, 2013, "An evaluation platform for forensic memory acquisition software" - Digital Investigation Journel 10 S30–S40.

[2] A. Martin. FireWire memory dump of a Windows XP computer: a forensic approach. http://www.friendsglobal.com/papers/FireWire%20Memory%20Dump%20of%20Windows%20XP.pdf

[3] Christo Ananth, M.Danya Priyadharshini, "A Secure Hash Message Authentication Code to avoid Certificate Revocation list Checking in Vehicular Adhoc networks", International Journal of Applied Engineering Research (IJAER), Volume 10, Special Issue 2, 2015,(1250-1254)

[4] B. D. Carrier and J. Grand. A hardware-basedmemory acquisition procedure for digitalinvestigations.Digital Investigation, 1(1):50–60, 2004.

[5] J. Rutkowska. Beyond the CPU: Defeating hardware based RAM acquisition tools, 2007. http://invisiblethings.org/papers/cheating hardware-memory-acquisition-updated.ppt

[6] Luka Milković-"Defeating windows memory forensics", 2012, 29c3

[7] Robert Beverly, Simson Garfinkel*, Greg Cardwell, "Forensic carving of network packets and associated data structures", Elsevier Journal - Digital investigation 8-S78 e S89, 2011