International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:6, No:10, 2012

Hybrid Approach for Memory Analysis in Windows System

Khairul Akram Zainol Ariffin, Ahmad Kamil Mahmood, Jafreezal Jaafar, and Solahuddin Shamsuddin

***Abstract*—**Random Access Memory (RAM) is an important device in computer system. It can represent the snapshot on how the computer has been used by the user. With the growth of its importance, the computer memory has been an issue that has been discussed in digital forensics. A number of tools have been developed to retrieve the information from the memory. However, most of the tools have their limitation in the ability of retrieving the important information from the computer memory. Hence, this paper is aimed to discuss the limitation and the setback for two main techniques such as process signature search and process enumeration. Then, a new hybrid approach will be presented to minimize the setback in both individual techniques. This new approach combines both techniques with the purpose to retrieve the information from the process block and other objects in the computer memory. Nevertheless, the basic theory in address translation for x86 platforms will be demonstrated in this paper.

***Keywords*—**Algorithms, Digital Forensics, Memory Analysis, Signature Search.

I. Introduction

FOR over the years, computer has become common in our daily life. Many activities in our daily life, starting from writing the document to observing the stock market has involved the use of computer system. Therefore, it can be concluded that the introduction of computer and the technology within it has ease our daily routine. However, it also brings a negative effect such as cybercrime to our world.

The Council of Europe's Cybercrime Treaty defines the cybercrimes as a range of crime that is committed using the computer, network and hardware devices [1]. Further, Symantec divides the cybercrimes into two main perspectives such as the single event that is facilitated by the crimeware and the range of activity, starting from cyber stalking to stock market manipulation [2]. In RSA 2012 Cybercrime Trends Report, it is reported that the cybercrime has shown no signs of slowing down. The report also concluded that in every minute, 232 computers have been infected with the malware [3].

Due to the increase in cybercrimes, a field known as Digital Forensic has been established. This field is involved with collecting, preserving, analyzing, documenting and presenting the contents of computer as evidence of cybercrime [4]. For over a decade, the investigation in Digital Forensics has been focused on the analysis of the non-volatile devices. Until recently, and with the merge of online storage, the volatile device such as computer memory has become critical in the investigation of Digital Forensic.

The research on the volatile memory has started in 2005 where Digital Forensic Research Workshop (DFRWS) has organized a Windows Memory Challenge. During the event, two analysis tools have been developed which are known as Memparser [5] and KntList [6]. Apart from that, the important of volatile memory analysis has also been listed out in Jesee, K [7]. Nevertheless, the sensitive information such as password, encryption keys and username are only available in volatile memory where they are used to store temporary processes and data before being transmitted to the Central Processing Unit (CPU).

K.A Zainol Ariffin is with the University Teknology Petronas,Perak, Malaysia (e-mail: akram.zainolariffin@ gmail.com).

S A.K Mahmood., is with the Computer Information System Department, University Teknology Petronas,Perak, Malaysia (e-mail: kamilmh@petronas.com.my).

J Jaafar with the is with the Computer Information System Department, University Teknology Petronas,Perak, Malaysia (e-mail: jafreez@petronas.com.my)

S Shamsuddin with the Research Department, CyberSecurity Malaysia, Mines, Selangor, Malaysia (e-mail: solahuddin@cybersecurity.my).

II. Literature reviews

A. Internal Structures and Address Translation in Computer Memory

An explanation on theories of internal structures and address translation has been outlined in Dhamdhere, M [8] and Russinovich and Solomon [9]. Both explain fully on the function of the internal structure and address translation in the computer memory with Russinovich and Solomon focussing directly towards Windows Operating System. In addition, the books also explain on the procedure of executive object creation and information storage with regard to the object.

Nevertheless, the procedure of storing data by kernel is also demonstrated by Amari, K [10]. In the demonstration, it illustrated that the kernel has set on pool to store the objects.

Most of the data in the computer memory are stored in the paged pool. Carrier, B [11] deduct that most of the data is stored in the paged pool as it allows the data to be transferred into hard disk if the computer memory is running low in space. Meanwhile, the important objects such as process and thread blocks are stored in the non-paged pools as the kernel need to access them frequently. Since the running process still remains intact all the time if the system still on power, therefore they will be available during the acquisition of the physical memory [12].

Apart from that, Virtual Address Description of a process block has a purpose of tracking the status of the process's address space. This internal structure is maintained by the memory manager and it stores the information on the attributes of the object such as range of the address, inheritance of child and object's security. Due to the information that is stored in this structure, a tool known as VADtool has been developed with the purpose to track the memory mapped files from the memory dump [13].

The demonstration address translation algorithm for x86 Windows system has been presented in [14].

B. Persistence of Data in the Computer Memory

A study has been carried out by Garfinkel, Chows and Rosenblum in 2004 that concluded 86% of the data contents still remain in the computer memory. The results from the test that reflect on the duration and system reboot are displayed in Table I and II respectively [15].

TABLE I
EFFECT ON THE DURATION

Duration	Description
14 days	23Kb of data still remains in the memory
28 days	7Kb of data remains in the memory. (Password and cryptographic keys)

TABLE II
EFFECT ON REBOOT

Duration	Description
Soft reboot	Data still remains in memory
Hard reboot	Data only remains up to 30 seconds

C. Analysis Technique for Memory Dump

XORSearch [16] is a tool that is designed based on string search technique. It takes a keyword as an input and then performs the search throughout the memory dump. Further, the tool can also help to find the keyword that has obfuscated by using either Exclusive OR (XOR) or Rotate Left (ROL) function that comes with it.

Windows Operating System represents each process in volatile memory as process block. It contains pointers to both next and previous process blocks. Further, the block stores the information on attributes of the process together with pointer to other data structures that is related to it. Process Environment Block (PEB) which is one of the internal structure of process block, is responsible to store the location of executable files and the DLL's path. Due to this fact, AccessData [17] group has designed a tool that is known as Forensic Toolkit (FTK). This tool will parse the process block to enumerate all the contents within memory. It also applies Directory Table Base (DTB) information in the address translation algorithms in order to identify the process in memory. Further, Windows Memory Forensic Toolkit (WMFT) has applied this technique by tracking the PsActiveProcessHead link to capture all the active processes in the memory dump [18]. Apart from that, Ruichao Zhang, Lianhai Wang, Shuhui Zhang [19] had demonstrated the data extraction from memory dump by using Kernel Processor Control Region (KPCR). In the demonstration, the information such as running processes, current network connection, file content and other data can be extracted from the image.

S. M. Hejazi, C. Talhi, M. Debbabi [20] had outlined the use of aplication or protocol fingerprint to trace the active application in the memory. The test was conducted to track online application such as email and messenger where from the result, it showed that each of the application had a use fingerprint representation.

PTFinder [21] is a tool that applies the file carving where the technique is done linearly to recover only the contiguous file. This technique is applicable because most operating systems will convert the file to be contiguous file instead of fragment files. [4]

III. METHODOLOGY

The algorithm for the hybrid approach is based on combination of process signature search and process enumeration techniques. In K.A.Z Ariffin, A.K Mahmood, J. Jafreezal, S Shamsuddin [22], it has outlined that all the process block in computer memory can be retrieved by using process signature search. Due to this information, this new approach will make use of the process signature to track the process block that is available in the computer memory. Then, once all the process blocks have been retrieved, the algorithm will continue to capture the information from them. The approach in tracking the information from the process block is divided into two types as listed below:

a) The information is retrieved directly from the process block.
b) An address translation algorithm is used to track the internal objects where information within them will be retrieved.

Apart from that, a process block tree will be constructed to link the processes within the computer memory due to their status (parent or child process). The overall working of the new hybrid approach is based on the rules that are discussed below:

RULE 1: Searching for process signature and detect the true process block

In theory, all process blocks in the computer memory except idle process are defined by a unique signature value. This signature is known as proã which has a constant hexadecimal value of 50726fe3 (H). This signature is located outside and before the location of the process block. The location of the signature is different between the Windows operating system. Table III lists out the location of the process signature for all Windows version until Windows Vista.

TABLE III
OFFSET FOR PROCESS SIGNATURE

Windows Operating System	Offset of the signature from starting Process Block (in Hex)
2000	0x01c
2003	0x0c0
XP	0x01c
Vista	0x024

After all possible process blocks have been allocated, they are then captured, dumped and stored in the database. The size of the possible process that is to be dumped and stored is based on the equation below:

$$\text{size} = (\text{start process offset} - \text{proã offset}) + \text{size of process block} \quad (1)$$

Once all the possible process blocks have been dumped, a process of selecting the true entity is run to remove the false process blocks. In general, it is not guaranteed that all the captured possible process blocks represent the true process block. This is due to the nature of computer memory which is random and volatile, the processes or data from different period of time will still remain in it until they are overwritten by the new data. Due to this scenario, additional information is needed to remove the data or entity from different period of time. The easier way to accomplish this is by using the information that is stored in the Image Filename. This information is stored in "uchar" format and only the process block with recognizable character (in word with .exe) is chosen as a true entity. When all the true entities have been detected, the false process will be removed from the database. The offset of Image Filename in the process block for all Windows version is listed in Table IV.

TABLE IV
OFFSET FOR IMAGE FILENAME

Windows Operating System	Offset of the signature outside the Process Block (in Hex)
2000	0X1fc
2003	0x164
XP	0x174
Vista	0x14c

RULE 2: Retrieve information and data from the process block

Once the true process blocks have been detected, the information within them can be retrieved. This information can be critical in digital forensics investigation as it can give a snapshot on how the process has been active in the computer system. However, only few information is available directly from the process block itself while others is stored in internal object (Process Environment Block (PEB), Thread, Exception, and Handle). In general, the process block stores the pointer of the other objects within it which is also known as virtual address. In order to proceed to other internal objects, a process enumeration technique is applied.

In this technique, an address translation algorithm will be used to translate or convert the virtual address (pointer) of internal object into physical address. The address translation algorithm for 32 bit system has been discussed in detail in [14]. The most important internal structure that is involved in this algorithm is Directory Base Table. This structure stores the value that represents the CR3 register for Central Processing Unit (CPU) and it also points to the start of Page Directory. This value together with the virtual address of the internal objects is applied in the algorithm to allocate the location of the object. Once the objects have been allocated, the information within them can be retrieved.

RULE 3: Construct the Process Block Tree

In theory, the process block in the computer system has a unique process ID (pid). This value together with Parent process ID (ppid) can be used to determine the child and parent for the process. The Parent process ID is stored in InheritProcess ID offset in the process block. There are two cases in determining the parent and child process in the memory dump:

- If the value in the ppid of the process is the same as the value of the pid for other process (still remains in the memory dump), then this process is a child of other.
- If the ppid value of the process block does not reflect on any process, then it is a standalone process. This normally happens if the parent process is no longer in the computer system

Table V shows the offset of pid and ppid in the process block for Windows Operating System.

TABLE V
OFFSET FOR PID AND PPID

Windows OS	ppid	pid
2000	0x1c8	0x09c
2003	0x138	0x094
XP	0x14c	0x084
Vista	0x124	0x09c

Hence, by applying the above rules, the Process Block Tree for Windows System can be constructed.

IV. EXPERIMENT

A. Comparison Test between Two Methods

The comparison study between Process Signature Search and Process Block Enumeration techniques is conducted. The test is aimed to identify the advantages and disadvantages for both techniques. The factors for this comparison test are the required knowledge, involvement of address translation algorithm and the ability to retrieve information about the process. Once the advantages and disadvantages have been identified, a new approach will be developed to minimize the setback from each of the techniques.

B. Retrieve Information from Memory Dump by Using the Hybrid Approach

In this test, the author will run test to the hybrid technique ability to track the information about the processes that are stored in the computer memory. The algorithm for the hybrid approach is listed in the Appendix A.

This test is conducted on the existing memory dumps that are available in Digital Forensics Conference and CFReDs Project website [23], [24]. It allows the author to directly compare the result with the work that has been done in the past as many researchers have made use of these memory dumps.

V. RESULT AND DISCUSSION

From the comparison test, the advantages and disadvantages of process signature search and process block enumeration techniques have been outlined. The result for the test has been summarized in Table VI.

TABLE VI
COMPARISON TEST BETWEEN METHOD 1 AND METHOD 2

Factor	Method 1	Method 2
Technique	Process Signature Search	Process Block Enumeration
Requirement knowledge	Operating System	Operating System and System Architecture
Involvement address translation algorithm	No	Yes
Ability to retrieve hidden and exile process	Yes	No
Ability to track other information from other internal object (i.e Threads, PEB, etc)	No	Yes
Advantages/Disadvantages	+ less knowledge needed + easy for new officer + fast + able to trace hidden/exile process _ not able to retrieve other internal objects _ information only at surface level	+ can retrieve many information with regard to the process + able to trace other internal object _ require more knowledge and complicated _ not able to retrieve hidden and exile process

From the table, the process signature search technique only requires the knowledge of Operating System. This knowledge covers the computer memory management and the internal structure within the process block. On the other hand, the process block enumeration requires both operating system and computer system architecture knowledge. It is because in this technique, an address translation algorithm is applied together with the technique. Address translation algorithm is a technique that converts the virtual address of the internal structure into the physical address. In the process block enumeration technique, the address translation is applied to track the active process blocks that reside in the memory. It is done by tracking the Active Process Link offset where the pointer to the next active process block is stored in the forward link (flink). Hence, by tracking the flink, the entire active process block can be retrieved. This technique is based on tracking the doubly link between process block. However, the exile and hidden processes is no longer in the doubly link between the process blocks. Due to this scenario, these processes cannot be retrieved by using process block enumeration technique since the empty pointers are stored in the structure. Table VIII shows the information with regard to rundll32.exe process which has already ended.

TABLE VII
INFORMATION ON RUNDLL32.EXE PROCESS BLOCK

Information	Quantity
Name	Rundll32.exe
Parent PID	228
PID	296
Start Time	Sun Jun 5 2005 14:07:32
End Time	Sun Jun 5 2005 14:08:43
Forward pointer	00000000
Backward pointer	00000000
Starting Process Block	6601020(H)

In process signature search technique, the algorithm is based on tracking the signature for process block. This signature will still remain in the computer memory, although the process has already ended. Hence, the exile and hidden process can still be retrieved although it is no longer retained in the Doubly Link between process block.

In term of information, the method 1 is able to retrieve the information that is stored in the process block. However, it cannot trace any information from other object (linked with the process) due to not applying address translation algorithm. Hence, the information that can be retrieved by using this technique is limited which included the start time, end time, and memory usage. As for method 2, the information that can be retrieved is broader as it allows searching on the linked objects and reading the information within them. Thus, with method 2, the information on threads, ports, dynamic link libraries path (dllpath), current directory, command line and image path name can be retrieved.

From the comparison test, it shows that method 1 is able to retrieve all the process blocks while with method 2, further information about the process can be identified. Due to this reason, an algorithm that combines method 1 and 2 is designed and tested to retrieve all the information from the computer memory. The process block tree for the test is shown in Appendix B. Once the process block tree (for hybrid approach) has been outlined, further information with regard to specific process can be retrieved by tracking the Internal Object that is linked with process block. As an example, the dll path, current directory, command line and image path name of Explorer.exe have been captured by tracking the location of Parameter Process Block. Figure 1 shows the step of retrieving the information from Parameter Process Block.

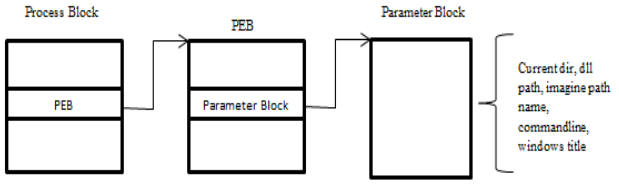


Fig. 1 Step of retrieving information from Parameter Process Block

Figure 2 shows the location of Parameter Process Block that stored the information on Explorer.exe process block. Table IX shows the information on dll path, current directory,

command link and image path name for Explorer.exe process block.

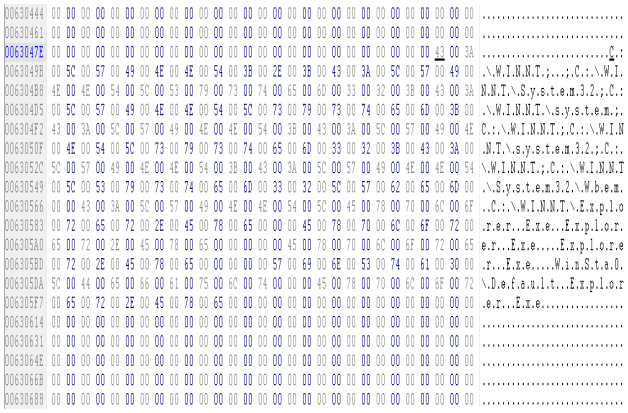


Fig. 2 Location of Process Parameter Block in computer memory

TABLE VIII
INFORMATION ON EXPLORER.EXE

Information	Value
Current Directory	C:\
Dllpath	C:\WINNT\; C:\WINNT\system32; C:\WINNT\System; C:\WINNT; C:\WINNT\System32; C:\WINNT;
ImagePathName	C:\WINNT\Explorer.exe
Command line	C:\WINNT\Explorer.exe
Windows Title	C:\WINNT\Explorer.exe

VI. CONCLUSION

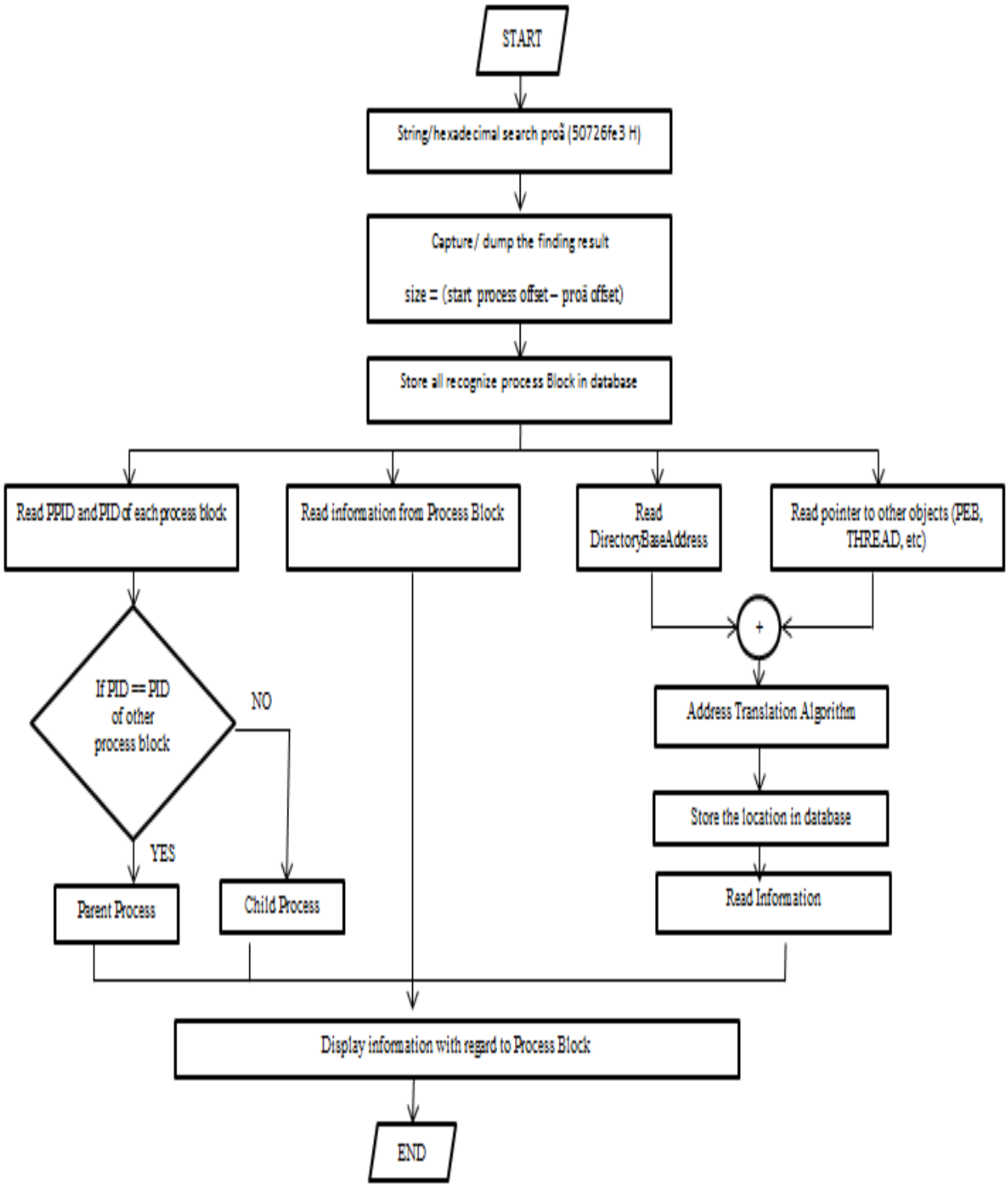
From the result of the experiment, it shows that the process signature search is an excellent technique to retrieve the entire process blocks in the memory dump of Windows Operating System. However, this technique can only retrieve the information at surface level as it cannot capture information that is stored on other internal object. Apart from that, the process block enumeration technique is unable to trace the hidden and exile process in memory dump, but excellent in retrieving information from other internal objects. Due to this scenario, a new approach which combines both techniques is designed and it is successfully able to retrieve all the information with regard to the process blocks that are stored in the memory dump. This new hybrid algorithm works on two step mechanism:

- First : detect all the process block in memory dump by using the process signature search
- Second : capture information from other internal objects by using process block enumeration

For future work, a combination of VADRoot and the hybrid algorithm can be applied/tested to improve the chance of retrieving other important internal structure.

APPENDIX

Appendix A: Hybrid approach algorithm



Appendix B: Process Block Tree from Hybrid Approach

PROCESS	PPID	PID	Starting Time	End Time
System	0	8		
Smss.exe	8	156	Sun Jun 5 2005 00:32:40	
Csrss.exe	156	180	Sun Jun 5 2005 00:32:43	
Winlogon.exe	156	176	Sat Jun 4 2005 23:36:31	
Winlogon.exe	*156*	*176*	*Sun Jun 5 2005 00:32:44*	
Isass.exe	176	240	Sun Jun 5 2005 00:32:46	
Metasploit.exe	240	788	Sun Jun 5 2005 00:38:38	Sun Jun 5 2005 00:38:38
Metasploit.exe	240	600	Sun Jun 5 2005 00:55:09	Sun Jun 5 2005 00:55:09
UMGR32.exe	600	668	Sun Jun 5 2005 00:55:09	
Dfrws2005.exe	668	784	Sun Jun 5 2005 01:00:54	Sun Jun 5 2005 01:00:59
Dfrws2005.exe	668	784	Sun Jun 5 2005 01:00:54	Sun Jun 5 2005 01:00:59
Services.exe	176	228	Sun Jun 5 2005 00:32:46	
Svchost.exe	228	408	Sun Jun 5 2005 00:32:49	
JogServ2.exe	408	1064	Sun Jun 5 2005 00:34:02	
Svchost.exe	228	480	Sun Jun 5 2005 00:32:49	
Spoolsv.exe	228	436	Sun Jun 5 2005 00:32:49	
Avsynmgr.exe	228	464	Sun Jun 5 2005 00:32:49	
VsStat.exe	464	612	Sun Jun 5 2005 00:32:54	
Avconsol.exe	464	628	Sun Jun 5 2005 00:32:55	
WinMgmt.exe	228	672	Sun Jun 5 2005 00:32:55	
WinMgmt.exe	672	760	Sun Jun 5 2005 00:33:10	Sun Jun 5 2005 00:33:24
Rundll.exe	228	296	Sun Jun 5 2005 14:07:32	Sun Jun 5 2005 14:08:43
Regsvc.exe	228	540	Sun Jun 5 2005 00:32:52	
MsTask.exe	228	552	Sun Jun 5 2005 00:32:52	
Dfrws2005.exe	228	592	Sun Jun 5 2005 01:00:54	
Userinit.exe	176	800	Sun Jun 5 2005 00:33:53	
Explorer.exe	800	820	Sun Jun 5 2005 00:33:53	
Helix.exe	820	324	Sun Jun 5 2005 14:09:27	
Cmd2k.exe	324	1112	Sun Jun 5 2005 14:14:38	
dd.exe	1112	1152	Sun Jun 5 2005 14:14:38	Sun Jun 5 2005 14:18:48
dd.exe	1112	284	Sun Jun 5 2005 14:53:43	
Cmd2k.exe	324	1132	Sun Jun 5 2005 14:10:53	
cmd.exe	820	1076	Sun Jun 5 2005 00:35:18	
HKServ.exe	820	972	Sun Jun 5 2005 00:33:57	
Apoint.exe	820	964	Sun Jun 5 2005 00:33:57	
DragDrop.exe	820	988	Sun Jun 5 2005 00:33:58	
Alogserv.exe	820	1008	Sun Jun 5 2005 00:33:58	
HKServ.exe	820	972	Sun Jun 5 2005 00:33:57	
Tgcmd.exe	820	1012	Sun Jun 5 2005 00:33:59	
PcfMgr.exe	820	1048	Sun Jun 5 2005 00:34:01	
Winlogon.exe	164	144	Fri Jun 3 2005 01:25:54	
Csrss.exe	144	168	Fri Jun 3 2005 01:25:54	
Apntex.exe	864	1072	Sun Jun 5 2005 00:34:03	
Nc.exe	592	1096	Sun Jun 5 2005 01:00:54	

Appendix C: Process Block Retrieval by Process Enumeration Technique

PROCESS	DTB	PT	Starting Time
Idle			
System	030000	1032000	
Smss.exe	03104000	1032000	Sun Jun 5 2005 00:32:40
Csrss.exe	0429f000	1031000	Sun Jun 5 2005 00:32:43
Winlogon.exe	04fe4000	103b000	*Sun Jun 5 2005 00:32:44*
Services.exe	04fe4000	103b000	Sun Jun 5 2005 00:32:46
Isass.exe	052e5000	103b000	Sun Jun 5 2005 00:32:46
Spoolsv.exe	05e2e000	103b000	Sun Jun 5 2005 00:32:49
Avsynmgr.exe	06173000	103b000	Sun Jun 5 2005 00:32:49
Svchost.exe	05f78000	103b000	Sun Jun 5 2005 00:32:49
WinMgmt.exe	01a24000	103b000	Sun Jun 5 2005 00:32:55
Explorer.exe	03ca1000	103b000	Sun Jun 5 2005 00:33:53
Apoint.exe	02bf1000	103b000	Sun Jun 5 2005 00:33:57
HKServ.exe	02ce7000	103b000	Sun Jun 5 2005 00:33:57
DragDrop.exe	02dbc000	103b000	Sun Jun 5 2005 00:33:58
Alogserv.exe	02e4c000	103b000	Sun Jun 5 2005 00:33:58
Tgcmd.exe	02ce5000	103b000	Sun Jun 5 2005 00:33:59
PcfMgr.exe	043de000	103b000	Sun Jun 5 2005 00:34:01
JogServ2.exe	05a23000	103b000	Sun Jun 5 2005 00:34:02
Apntex.exe	05d0c000	103b000	Sun Jun 5 2005 00:34:03
cmd.exe	0575e000	103b000	Sun Jun 5 2005 00:35:18
UMGR32.exe	075a5000	103b000	Sun Jun 5 2005 00:55:09
Dfrws2005.exe	03e02000	103b000	Sun Jun 5 2005 01:00:54
Nc.exe	0600d000	103b000	Sun Jun 5 2005 01:00:54
Helix.exe	06f53000	103b000	Sun Jun 5 2005 14:09:27
Cmd2k.exe	058dd00	103b000	Sun Jun 5 2005 14:14:38
Cmd2k.exe	039a2000	103b000	Sun Jun 5 2005 14:10:53
dd.exe	01d9e000	0000000	Sun Jun 5 2005 14:53:43

ACKNOWLEDGMENT

The author would like to present my gratitude towards CyberSecurity Malaysia and Universiti Teknologi Petronas for the help and support that had been accorded during the research period.

REFERENCES

[1] Krone, T., 2005. "*High Tech Crime Brief*". (Book style). Australian Institute of Criminology. Canberra, Australia. ISSN 1832-3413. 2005.
[2] US-Cert, government organization, "*Computer Forensic*", (published report). USA, 2008
[3] RSA, "The current state of cybercrime and what to expect in 2012", (published report) USA, 2012.
[4] Hill, C.E., "*What is the Definition of Digital Forensics? "*, in *eHow, How to do just about everything*(Unpublished work sytle). Unpublished
[5] DFRWS." *Memparser Analysis Tool by Chris Betz"*.(Unpublished work sytle). Unpublished
[6] DFRWS. "*Kntlist Analysis Tool by George M. Garner Jr."* . (Unpublished work sytle). unpublished
[7] Jesee, K., "*Using every part of the buffalo in Windows memory analysis*(Published Journal style)". Journal Digital Investigation, vol **4**: pp. 24-29, 2007.
[8] Dhamdhere, D.M., "*Operating Systems: A Concept based Approach*."(Book style) 1 ed., McGrawHill, 2009.
[9] Russinovich, M.E., D.A. Solomon, and A. Ionescu, "*Windows®Internals Covering Windows Server® 2008 and Windows Vista®*" (Book style). J. Pierce, Editor., Microsoft Press, 2009.
[10] Amari, K., "*Techniques and Tools for Recovering and Analyzing Data from Volatile Memory*"(unpublished work style), SANS Institute, 2009.
[11] Carrier, B.G., J.," *A Hardware Based Memory Acquisition Procedure for Digital Investigations"*. (Published Journals style). Journal of Digital Investigation, 2004, March.
[12] Schuster, A., "*Searching for processes and threads in Microsoft Windows memory dump"*(Published Journal style).Journal Digital Investigation, vol **3,** pp. 10-16, 2006.
[13] Dolan-Gavitt, B.," *The VAD tree: A process eye view of physical memory"*(Published Journal style).Journal Digital Investigation, pp. s62-s64, 2007
[14] Khairul A.Z, Ahmad K.M, Jafreezal J.," *Investigating the PROCESS block for memory analysis(* Published Conference Proceedings style)," ", in ACS'11 proc, WSEAS Conf, pp 21-29, 2011
[15] Garfinkel, T., Pfaff, B., Chow, J., & Rosenblum, M. "*lifetime is a systems problem (*Published Conference Proceedings style)," In *Proc of the ACM SIGOPS European Workshop*, ACM, 2004
[16] Stevens, D."*XORSearch*", (unpublished work style).2007, January 30.
[17] AccessData Corporation, " *Importance of memory Search and Analysis*" (Published White Paper) Lindon, UT, 2006.
[18] Burdach, M. *"An Introduction to Windows memory forensic"*(unpublished work style). Unpublished.
[19] Ruichao Zhang, L. W., Shuhui Zhang. "*Windows Memory Analysis Based on KPC"*(Published Conference Proceedings sytle) . In *Proc of the 2009 Fifth International Conference on Information Assurance and Security*, IEEE, Xi'An China.
[20] S. M. Hejazi, C. T., M. Debbabi "*Extraction of forensically sensitive information from windows physical memory*".(Published Journal Style) Journal d i g i t a l i n v e s t i g a t i o n vol 6, pp.S 1 2 1 – S 1 3 1, 2009
[21] Schuster, A." *PTFinder"*. 2006.(Unpnlished work style).unpublished.
[22] Khairul A.Z, Ahmad K.M, Jafreezal J, S Shamsuddin, "Process *Block Tree (PBT) for Windows Operating System* (Published Conference Proceedings style),"", in ICCEMS 2012 proc, ICCEMS Conf, pp121-128.
[23] DFWRS, "*Windows Memory Challenge*", (Online Workshop/Conference page) 2005.
[24] Jesse K, "*Computer Forensics Reference Datasets"*, (Online Research data sets) CFReDs Project,ManTech, 2011.