An Analyzing of different Techniques and Tools to Recover Data from Volatile Memory

[1]Mr. Arpit Patel [2] Prof. Nilay Mistry
[1]GTU, PG School
[2] Gujarat Forensic Science University, Ahmedabad, Gujarat, India

Abstract— Computer forensics has recently gained significant popularity with many local law enforcement agencies. It is currently employed in fraud, theft, drug enforcement and almost every other enforcement activity. There are many relatively new tools available that have been developed in order to recover and dissect the information that can be gleaned from data storage area like hard-disk, pen drive, etc. it's all like a volatile memory, but because this is a relatively new and fast-growing field many forensic analysts do not know or take advantage of these assets. Memory like Volatile memory may contain many pieces of information relevant to a forensic investigation, such as passwords, cryptographic keys, and other data. Having the knowledge which type of method use and tools needed to recover that data is essential, and this capability is becoming increasingly more relevant as hard drive encryption and other security mechanisms make traditional hard disk forensics more challenging. This research will cover the theory behind volatile memory analysis, including why it is important, what kinds of data can be recovered, and the potential pitfalls of this type of analysis, as well as techniques for recovering and analyzing volatile data and currently available toolkits that have been developed for this purpose.

I. INTRODUCTION

Forensic analysis of physical memory is gaining good attention from experts in the community especially after recent development of valuable tools and techniques. Investigators find it very helpful to seize physical memory contents and perform post-incident analysis of this potential evidence [1]. Standard approach to forensic analysis of a computer is called static analysis. Hard disks and other nonvolatile memory devices, of powered off computer, are cloned bit by bit. Such forensically correct copies are analyzed, usually in read only mode to ensure data integrity, even on, cloned memory devices. Static analysis enables investigator to thoroughly search stored data and find relevant evidence. There are number of documents that describe procedure and number of tools to support and automate search and analysis. [2] Traditional or "dead" forensics involves the recovery of evidence from computer systems that have been powered down. Unfortunately, shutting down a system results in the loss of important volatile data. Also, it may be impossible to shut down vital enterprise systems to conduct forensic investigations [3]. One relatively new capability available to examiners is memory forensics. As attackers learned that they could leverage volatile memory to store data and execute code instead of or in addition to the hard disk, it became necessary for analysts to take that into consideration and develop their own methodologies for recovering this important information in their investigations. It also aims to provide guidance as to why memory forensics is valuable, and argues that it is in fact essential to the future of forensic analysis.

The goal of digital forensics is the extraction and analysis of electronic evidence. Traditionally static media has been the principal source of digital evidence; however, research into the use of physical memory as a data source has spawned the digital forensics sub-stream of physical memory forensics [4].

II. DATA ANALYSIS

When analyzing digital data, we are looking at an object that has been designed by people. Further, the storage systems of most digital devices have been designed to be scalable and flexible, and they have a layered design. I will use this layered design to define the different analysis types [5].

If we start at the bottom of the design layers, there are two independent analysis areas. One is based on storage devices and the other is based on communication devices. I am going to focus on the analysis of storage devices, specifically non-volatile devices, such as hard disks. The analysis of communication systems, such as IP networks, is not covered, but is elsewhere [6] [7] [8].

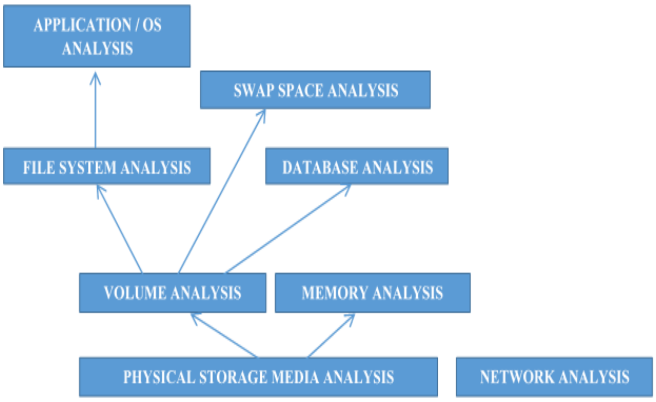

Fig 1. Layers of analysis based on the design of digital data

Fig 1 shows the different analysis areas. The bottom layer is Physical Storage Media Analysis and involves the analysis of the physical storage medium. Examples of physical store mediums include hard disks, memory chips, and CD-ROMs. Analysis of this area might involve reading magnetic data from in between tracks or other techniques that require a clean room. We are going to assume that we have a reliable method of reading data from

the physical storage medium and so we have a stream 1s and 0s that were previously written to the storage device.
The analysis process in Fig 2. This shows a disk that is analyzed to produce a stream of bytes, which are analyzed at the volume layer to produce volumes. The volumes are analyzed at the file system layer to produce a file. The file is then analyzed at the application layer.

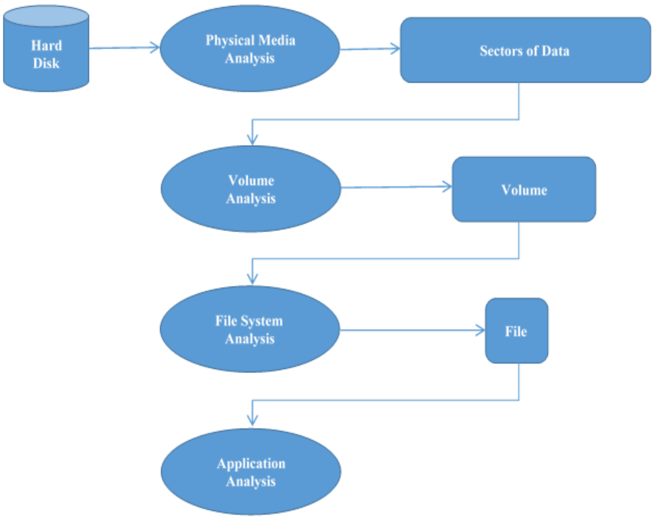

Fig 2. Process of analyzing data at the physical level to the application level.

III. FAT & NTFS FILE SYSTEM

Detailed analysis and study on the Memory Principles of the NTFS file system, this puts forward the object-oriented idea to design NTFS file parsing system, parse the most derived binary data that was saved to disk, achieving the totally analysis of the normal files and the deleted files. Then a friendly interface is used to display all these data of tree structure to the users.

AFF, Advanced Forensic Format, is one of well-known proposals in dealing with how to store digital images of hard disk or any stored digital data for potential investigations.[9]

A. Information hiding methods [10]

1. Hidden Files and Folders
2. Deleted Files
3. Hidden/Deleted Partitions
4. Alternate Data Streams
5. Slack Space
6. File Slack Space Hiding
7. Bad Clusters
8. Steganography

B. Analysis of NTFS file system [11]

1) Framework of NTFS

In NTFS, system obtains the storage location of data by Master File Table (MFT).MFT is a database relative to the file, consisting of a series of File Record.

2) The Volume Access Process By MFT In NTFS

- Small File(SF) And Small Directory(SD) Access
- Mass File Access

3) Directory Access

Boot	MFT	System file	File area

Fig 3. Structure of NTFS partition

C. The Recovery Techniques Of Deleted Files[11]

In Windows OS, when the NTFS file is deleted, file data area is not cleared at once. The system only alter file status byte value from 01(remain using) to 00(deleted) in file record, and all other important information is remained.

1) Totally recoverable model:

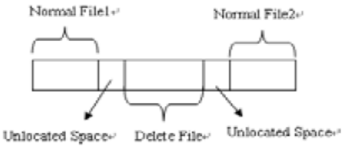

Fig 4. Totally recoverable model

2) Partial recoverable model:

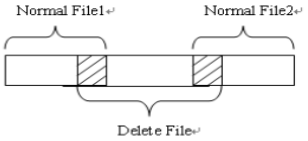

Fig 5. Partial recoverable model

3) Totally non-recoverable model:

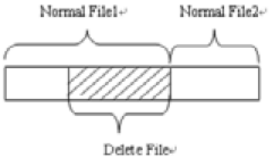

Fig 6. Totally non-recoverable model

One technique investigators use to improve temporal reconstruction, as summarized in Table 3, is the use of unallocated portions of file systems as evidence. When a file system stops using a block of data, the data remains within the block itself, but the metadata describing it indicates that the block is unallocated [9].

Category	Pros	Cons	Exemplary Systems
Concurrent versioning file systems	Consistent tracking of changes to the same file	Burdensome to users No continuous data retention	CVFS
Journaling File systems	Transparent to users	Relatively easy for a sophisticated user to overwrite files upon deletion Ad hoc retention of file data	XFS, ZFS, EXT3
Backups	Useful on virtually any file system Potentially long-term retention of forensically useful data	No continuous data retention	EnCase, Forensic ToolKit, Sleuth Kit
File System Snapshots	Low overhead means of retaining file data	No continuous data retention Must be initiated by an administrator	A feature of ZFS

Table. 1 Sources of temporal forensic data

IV. FILES MAPPED IN MEMORY

Physical memory of a computer was mainly captured to retrieve strings, e.g. passwords, IP addresses or e-mail addresses. Going through these results manually is a tedious job; it is preferable to have a tool that can automatically identify relevant structures.

File carving is the process of reassembling files from disk fragments based on the file content in the absence of file system metadata. [12]

A method often used for reconstructing files from an image is carving. When carving for files, characteristic signatures are used to identify the start of a file. A popular carving program is Scalpel (http://www.digitalforensicssolutions.com/Scalpel/). Scalpel uses a linear carving technique, which is effective only for contiguous files. When a file is fragmented, linear carving algorithms fail to reconstruct the file and the file will be incomplete after the first fragment. Smart carving algorithms may be able to recover fragmented files. [13]

File mappings are administrated using a number of different data structures. An overview of these structures is shown in Fig. 7. These structures are allocated from memory pools. A memory pool is a dynamic memory area allocated by the kernel where it stores administrative structures. The type of a pool structure can be determined through pool tag, a four byte magic number (e.g. Proc, Obtb, and MmCa) stored in the header of the structure.

The structure is identified via its pool tag Proc and contains pointers towards the Virtual Address Descriptor (VAD) Root and the Object Table.

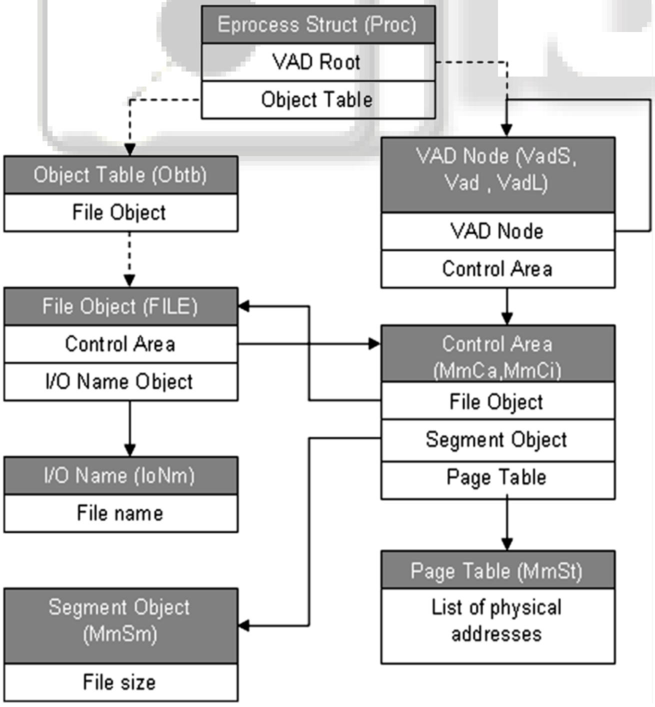

Fig 7. Overview of links between different memory structures related to mapped files

A. Mapped-file recovery

- Allocated file-mapping structures
- Unallocated file-mapping structures
- Unidentified file pages
- Prototype implementation

V. MEMORY FORENSIC TECHNIQUES

Memory forensics is the area of computer forensics that relates specifically to volatile memory. More specifically, it is the acquisition and analysis of physical memory [14]. Memory forensics is more challenging than disk-based forensics for several reasons: it is volatile in nature and therefore difficult to collect. It is also difficult to analyze, as memory does not use a set structure. Acquisition and analysis of the data represent separate distinct research areas and are the focus of much research within the discipline. [15]

A. The need for memory forensics

There are a number of factors that have driven the development of memory forensics. The increased technological ability of criminals who are able to use technologies to conceal data or activity is a major factor [16]. Specifically, the increased use of hard drive encryption has lead to a need to recover encryption keys. Encryption keys stored in physical memory may be retrieved and used to decrypt data on static-media sources. A number of researchers have published very promising work in this area. Another reason for the increased attention to memory forensics is the growth in disk usage. Conventional methodologies may no longer be sufficient in dealing with the increasing storage ability.

Most of the current tools have a low level focus, working only with a small subset of operating system memory structures. Memory is structured differently between operating systems and often varies between different versions of the same operating system.

Tools/project	**Created by/Author**	**OS Support**	**Description**
Memparser [17]	Open Source Chris Betz (Winner of 2005 DFRWS challenge)4	Win OS	***1)*** Memparser is a memory forensics analysis tool for microsoft windows systems. ***2)*** Parses a string into a number. The number stored at ptr is potentially suffixed with K (for kilobytes, or 1024 bytes), M (for megabytes, or 1048576 bytes), or G (for gigabytes, or 1073741824). If the number is suffixed with K, M, or G, then the return value is the number multiplied by one kilobyte, one megabyte, or one gigabyte, respectively. ***3)*** Performs enumeration of the process list and dumps other information such as process environment information, DLL's in use and can extract memory space of individual processes. ***4)*** License – GNU General Public License

			version 2.0 (GPLv2) http://sourceforge.net/projects/memparser/
KnTTools KNTDD - Acquires memory [18]	George M. Garner Jr. and Robert-Jan Mora	Win OS	Acquisition to removable drive or network. Cryptographic integrity checks, auditing. Conversion to Microsoft crash dump format. Remote deployment as a service. http://forensic.seccure.ne/ http://gmgsystemsinc.com/knttools/
KnTTools KnT List - Lists Kernel Structures [18]	George M. Garner Jr. and Robert-Jan Mora	Win OS	Reconstructs virtual address space. Drives, Device Objects, System Tables. Threads, access tokens, handle table, objects, etc. Outputs as text and XML. http://forensic.seccure.ne/ http://gmgsystemsinc.com/knttools/
Windows Memory Forensic Toolkit (WMFT)	Mariusz Burdach	Win OS	It is a collection of utilities intended for forensic use. It can be used to perform forensic analysis of physical memory images acquired from Windows 2000/2003/XP machines. There are two versions of toolkit. The version for Windows is written in C# for .NET technology and can analyze memory images. This version has additional functionally such as detecting hidden objects. Limitation :- can't find Hidden (via DKOM) and terminated processes.
Technique by Schuster [19]	Andreas Schuster	Win OS	Created a novel method for finding process objects, thread objects and many other object types.
Lspi.pl			It is a Perl script that takes the same arguments as lspd.pl and lspm.pl and locates the beginning of the executable image for that process.
ssdeep [20]	Jesse Kornblum		It implements *"context triggered piecewise hashing"*. This can be used to compare files where some of the bits differ - such as executable files recovered from memory and the original file that is stored on the hard drive. The output of the tool is a percentage of how similar the files are. A higher value indicates a greater chance that two files were derived from the same location.
Procloc.pl [21]	Tim Vidas, Nebraska University	Win OS	It (derived from process locator) is another tool that scans a Microsoft Windows physical memory image in order to find process structures; it is available under a GNU licence. http://nucia.unomaha.edu/tvidas
pmodump.pl [22]	Joe Stewart, Truman Project	Win OS	It extracts the virtual memory of a process by using a Microsoft Windows memory space characteristic. This characteristic is leveraged to find all of the memory pages that make up the target process's memory.
Forensic Toolkit (FTK) [23]	AccessData	Win OS	The FATKit software contains support for page file integration during analysis. http://www.accessdata.com/support/product-downloads#FTKImager
Using every part of the buffalo [24]	John C. Kornblum	Win OS	Recovering additional data from memory images by retrieving data based on the type of invalid virtual address. In additional information was recovered by translating just three types of invalid virtual addresses. Invalid virtual addresses that referenced page file entries were not used in the experimentation as a page file was not available
Volatility [25]	AAron Walters & Nick L. Petroni	Win OS/ linux	It is both a framework and a set of tools for physical memory analysis. It is available to the public under a GNU licence. The philosophy behind Volatility is to be open source and free. The creators "*encourage people to modify, extend, and make derivative works, as permitted by the GPL*". Extracts: Image date & time Memory map for each running process Network sockets DLLs loaded for each process https://www.volatilesystems.com/VolatileWeb/volatility.gsp http://volatility.tumblr.com/
Mdd (Memory DD) (ManTech)	ManTech Internati-onal Corpora-tion	Win OS	MDD is a physical memory acquisition tool for imaging Windows based computers. http://sourceforge.net/projects/mdd
Memoryze	Mandiant	Win OS	Mandiant's Memoryze is free memory forensic software that helps incident responders find evil in live memory. Memoryze can acquire and/or analyze memory images, and on live systems, can include the paging file in its analysis. https://www.mandiant.com/resources/download/memoryze
DumpIt	MoonSols	Win OS	This utility is used to generate a physical memory dump of Windows machines. It works with both x86 (32-bits) and x64 (64-

				bits) machines. The raw memory dump is generated in the current directory, only a confirmation question is prompted before starting. Perfect to deploy the executable on USB keys, for quick incident responses needs. http://www.moonsols.com/wp-content/plugins/download-monitor/download.php?id=7
	winen.exe (Guidance Software - included with Encase 6.11 and higher)	Guidance Software	Win OS	It is a standalone ram acquisition tool that ships with the forensic software encase. The Winen Executable can run as a command line tool, user prompt or from a configuration file. We can run Winen.exe from a USB drive that you plug into the target machine. The tool collects RAM and places the collected information into an .E01 file. There is a 32-bit version as well as a 64-bit version. http://forensiczone.blogspot.com/2008/06/winenexe-ram-imaging-tool-included-in.html http://www.guidancesoftware.com/
	OSForensics	PassMark Software	Win OS	OSForensics can acquire live memory on 32bit and 64bit systems. A dump of an individual process's memory space or physical memory dump can be done. Output can be a straight dump or a Microsoft crash dump file, for use with Micrsoft's WinDbg debugger http://www.osforensics.com/osforensics.html
	WinPmem	Michael Cohen	Win OS	WinPmem is a free, actively developed, opensource forensic memory acquisition tool for Windows. It supports Windows XP to Windows 8, both 32 and 64 bit architectures. It can produce raw dumps as well as dumps in crashdump format (for analysis with Volatility or windbg). It supports output to STDOUT for piping the dump through tools like netcat or ssh. WinPmem can be used together with the Volatility Technology Preview to analyse a live windows system for live response and triaging. https://volatility.googlecode.com/svn/branches/scudette/docs/pmem.html
	Fastdump and Fastdump Pro	HBGary	Win OS	Fastdump and Fastdump Pro Fastdump (free with registration) Can acquire physical memory on Windows 2000 through Windows XP 32 bit but not Windows 2003 or Vista. Fastdump Pro Can acquire physical memory on Windows 2000 through Windows 2008, all service packs. Additionally, Fastdump Pro supports: – 32 bit and 64 bit architectures. – Acquisitions of greater than 4GB. – Fast acquisitions through the use of larger page sizes (1024KB) but also supports a strict mode that enforces 4KB page sizes. – Process probing which allows for a more complete memory image of a process of interest. – Acquisition of the system page file during physical memory acquisition. This allows for a more complete memory analysis.
	Windows Memory Reader	Mem Marshal project, cybermarshal	Win OS	It is a simple command-line utility to capture the contents of physical RAM. Results are stored in a Windows crash dump or raw binary file. Researchers can also use Windows Memory Reader to capture memory-mapped device data, such as shared video memory. Windows Memory Reader supports Windows XP through Windows 8, both 32-bit and 64-bit versions, and is available free of charge. http://cybermarshal.com/index.php/cyber-marshal-utilities/windows-memory-reader
x86 Hardware	WindowsSCOPE CaptureGUARD PCIe card	Commercial; desktops, servers	Win OS	Publicly available, supports all Windows OS; windd and other formats. CaptureGUARD Gateway performs DRAM acquisition even on locked computers http://www.windowsscope.com.
	WindowsSCOPE CaptureGUARD ExpressCard	Commercial; Laptop	Win OS	Publicly available, supports all Windows OS; windd and other formats. CaptureGUARD Gateway performs DRAM acquisition even on locked computers http://www.windowsscope.com
	Tribble PCI Card	Brian D. Carrier, Joe Grand		A Hardware-Based Memory Acquisition Procedure for Digital Investigations http://www.digital-evidence.org/papers/tribble-preprint.pdf
	CoPilot	Komoku		Komoku was acquired by Microsoft and the card was not made publicly available.

Forensic RAM Extraction Device (FRED)	BBN Technologies		Not publicly available. http://www.ir.bbn.com/~vkawadia/
LiME	Joe Sylve	Linux	It is a Loadable Kernel Module (LKM), which allows the acquisition of volatile memory from Linux and Linux-based devices, such as those powered by Android. The tool supports dumping memory either to the file system of the device or over the network. http://code.google.com/p/lime-forensics/
Fmem		Linux	It is kernel module that creates device /dev/fmem, similar to /dev/mem but without limitations. This device (physical RAM) can be copied using dd or other tool. Works on 2.6 Linux kernels. Under GNU GPL. http://hysteria.sk/~niekt0/foriana/fmem_current.tgz
Second Look®: The Linux Memory Forensic Acquisition	Raytheon Pikewerks http://pikewerks.com/	Linux	This commercial memory forensics product ships with a modified version of the crash driver and a script for safely dumping memory using the original or modified driver on any given Linux system. http://secondlookforensics.com/
Goldfish	Pavel Gladyshev, Afrah Almansoori ; Cybercrime Technologies	MAC OS	MAC OS X automated memory acquisition and analysis tool. It is a MAC OS X live forensic tool for use by law enforcement. Its main purpose is to provide an easy to use interface to dump system RAM of a target OS X machine via a firewire connection. It then automatically extracts the current user login password and any open AIM conversation fragments that may be available. License: GNU GPLv3
Mac Memory Reader	Cyber Marshal	MAC OS	It is a simple command-line utility to capture the contents of physical RAM on a suspect computer, letting an investigator gather volatile state information prior to shutting the machine down. Results are stored in either a Mach-O binary file or a raw-format file. It can also capture memory-mapped device data, such as shared video memory.
OSXPmem	Johannes Stuettgen	MAC OS	The OSX Memory Imager is an open source tool to acquire physical memory on an Intel based Mac. It consists of 2 components: The usermode acquisition tool 'osxpmem', which parses the accessible sections of physical memory and writes them to disk in a specific format. A generic kernel extension 'pmem.kext', that provides read only access to physical memory. After loading it into the kernel it provides a device file ('/dev/pmem/'), from which physical memory can be read.

B. Sensitive information in memory [26]

Memory is like a game table for all running applications and processes. To be part of the game, data should be brought to this table. This data includes, but is not limited to, executable code of the processes, data files accessed by processes, URLs accessed via a web browser, usernames, and passwords. The data resident in memory can be classified into the following categories:

1) Metadata
2) Files
3) Sensitive data
4) Case irrelevant data

VI. CONCLUSION

This paper highlights the need for forensics in different ways and the potential for future research. The need has been demonstrated by emphasizing the weaknesses in conventional forensic computing methodologies, tools and techniques. The future research potential has been identified by a review of the relevant literature in the area. Memory forensic techniques have the potential to recover digital evidence where conventional static-media based techniques cannot. As the digital evolution progresses, the challenges to forensic investigators are likely to expand. These challenges - the number and capacity of electronic devices, the network connectivity and bandwidth, and the potential for the use of anti-forensic techniques - are. There are also other challenges in the collection and analysis of devices technical challenges, such as the moving of data storage to off-site systems. Forensic computing research needs to meet these challenges with tools, techniques and processes to understand the potential of digital evidence and also to provide means to detect and analyses systems where these are utilized. Current tools and techniques in memory forensics need to be expanded to encompass more than low-level analysis. The research need is for tools and techniques that can extract high level data about applications and technologies that are problematic for conventional forensic computing methodologies.

ACKNOWLEDGEMENTS

At last but not least, we would like to appreciate all the people who help us directly or indirectly in preparing this survey paper.

REFERENCES

[1] S. M. Hejazi, C. Talhi, M. Debbabi (Computer Security Laboratory, Concordia Institute for Information Systems Engineering, Concordia University, Montreal,

Quebec, Canada); *"Extraction of forensically sensitive information from windows physical memory"* digital investigation 6 (2009) S121- S131, 1742-2876 © 2009 Digital Forensic Research Workshop.

[2] Sasa Mrdovic, Alvin Huseinovic, Faculty of Electrical Engineering Sarajevo; "*Forensic Analysis of Encrypted Volumes Using Hibernation File*"; 2011 IEEE.

[3] Zhen Su, LianHai Wang; "*Evaluating the Effect of Loading Forensic Tools on the Volatile Memory for Digital Evidences*", 2011 Seventh International Conference on Computational Intelligence and Security, 978-0-7695-4584-4/11, IEEE 2011.

[4] Matthew Phillip Simon, Jill Slay; "*Recovery of Pidgin Chat Communication Artefacts from Physical Memory – A Pilot Test to Determine Feasibility*"; 2011 Sixth International Conference on Availability, Reliability and Security; 978-0-7695-4485-4/11, IEEE 2011.

[5] Carrier, Brian. "Defining Digital Forensic Examination and Analysis Tools Using Abstraction Layers." *International Journal of Digital Evidence*.

[6] Bejtlich, Richard. *The Tao of Network Security Monitoring: Beyond Intrusion Detection*. Boston: Addison Wesley, 2005.

[7] Casey, Eoghan. *Digital Evidence and Computer Crime*. 2nd ed. London: Academic Press.

[8] Mandia, Kevin, Chris Prosise, and Matt Pepe. *Incident Response and Computer Forensics*. 2nd ed. Emeryville: McGraw Hill/Osborne.

[9] Ryan Q. Hankins and Jigang Liu, Member, IEEE; "*Towards a Forensic-aware File System*"; 978-1-4244-2030-8/08 2008 IEEE.

[10] Jeremy Davis, Joe MacLean, David Dampier; "*Methods of Information Hiding and Detection in File Systems*"; 2010 Fifth International Workshop on Systematic Approaches to Digital Forensic Engineering, 978-0-7695-4052-8/10 2010 IEEE.

[11] Zhang Kai,Cheng En, Gao Qinquan; "*Analysis and Implementation of NTFS File System Based on Computer Forensics*"; 2010 Second International Workshop on Education Technology and Computer Science, 978-0-7695-3987-4/10, 2010 IEEE.

[12] Scott Hand, Zhiqiang Lin, Guofei Gu, Bhavani Thuraisingham; "*Bin-Carver: Automatic recovery of binary executable files*" Digital Investigation 9 (2012) S108–S117.

[13] R.B.van Baar, W.Alink, A.R.van Ballegooij; *Netherland Forensic Institute, 2497 GB, The Hague, Netherlands*; "*Forensic memory analysis: Files mapped in memory*", digital investigation 5 (2008) S52 – S57.

[14] Matthew Simon, Jill Slay; "Enhancement of Forensic Computing Investigations through Memory Forensic Techniques"; 2009 International Conference on Availability, Reliability and Security, 978-0-7695-3564-7/09 IEEE 2009.

[15] E. Huebner, D. Bem, F. Henskens, and M. Wallis, "Persistent systems techniques in forensic acquisition of memory," Digital Investigation, vol. 4, pp. 129-137, 2007.

[16] Matthew Simon, Jill Slay; "Enhancement of Forensic Computing Investigations through Memory Forensic Techniques"; 2009 International Conference on Availability, Reliability and Security, 978-0-7695-3564-7/09 IEEE 2009.

[17] DFRWS. (2005). Memparser Analysis Tool by Chris Betz. [Online].
Available:
http://www.dfrws.org/2005/challenge/memparser.shtml.

[18] GMG Systems Inc. (2007). KnTTools with KnTList. [Online].
Available: http://gmgsystemsinc.com/knttools/.

[19] A. Schuster, "Searching for processes and threads in Microsoft Windows memory dumps," Digital Investigation, vol. 3 (S), pp. 10-16, 2006.

[20] J. Kornblum, "Identifying almost identical files using context triggered piecewise hashing," Digital Investigation, vol. 3(S), pp. 91-97, 2006.

[21] T. Vidas, "The Acquisition and Analysis of Random Access Memory," Journal of Digital Practice, vol. 1, pp. 315-323, 2006.

[22] N. Ruff, "Windows Memory Forensics," Journal in Computer Virology, vol. 4, pp. 83-100, 2008.

[23] N. L. Petroni Jr., A. Walters, T. Fraser, and W. A. Arbaugh, "FATKit: A framework for the extraction and analysis of digital forensic data from volatile system memory," Digital Investigation, vol. 3, pp. 197-210, 2006.

[24] J. D. Kornblum, "Using every part of the buffalo in Windows memory analysis," Digital Investigation, vol. 4, pp. 24-29, 2007.

[25] Volatile Systems. (2008). The Volatility Framework: Volatile memory artifact extraction utility framework. [Online].
Available:
https://www.volatilesystems.com/default/volatility#overview.

[26] S. M. Hejazi, C. Talhi, M. Debbabi; "*Extraction of forensically sensitive information from windows physical memory"*; digital investigation 6 (2009) S121-S131, 1742-2876 © 2009 Digital Forensic Research Workshop.