Forensic Analysis of Physical Memory and Page File

Hameed Iqbal

Master's Thesis
Master of Science in Information Security
30 ECTS
Department of Computer Science and Media Technology
Gjøvik University College, 2009

Avdeling for
informatikk og medieteknikk
Høgskolen i Gjøvik
Postboks 191
2802 Gjøvik

Department of Computer Science
and Media Technology
Gjøvik University College
Box 191
N-2802 Gjøvik
Norway

Forensic Analysis of Physical Memory and Page File

Hameed Iqbal

2nd November 2009

Abstract

With the passage of time, the field of computer forensics is maturing and the traditional methodology of disk forensics has now become a standard. In the same manner volatile data forensics is also getting serious attention from forensic investigators and researchers. Physical memory is an integral part of volatile data forensics. It can provide a forensic examiner with wealth of information like passwords, encrypted keys, typed commands, web addresses, shared and executable files, currently running processes and terminated processes, open ports and active connections. This thesis explores the forensic analysis of physical memory and page file in search of sensitive data using the currently available tools. Experiments are carried out in virtual environment on Windows XP operating system. The immediate purpose of this thesis is to study the impact of increased memory size, operating system and applications on the retention of sensitive data in today's computers. We will also explore the capabilities and limitations of the currently available tools for the acquisition and analysis of memory and page file.

Acknowledgements

I wish to extend my deepest gratitude to some people who helped me in the completion of this thesis work.

First of all I am thankful to Almighty Allah for giving me the ability and strength to contribute to the service of humanity in the shape of this research work. Then I would like to say many thanks to my supervisor Andre Årnes for his continuous support and encouragement during this whole research work. I am also thankful to my friends for their encouragement. Last but not the least I am thankful to my family and those special people who are away from me but their endless prayers and support enabled me to undertake this thesis work.

I would like to dedicate this research work to the Higher Education Commission (HEC) of Pakistan for financing my research work in Norway.

Hameed Iqbal November 02, 2009

Contents

List of Figures

1 Introduction

1.1 Topic Covered by the Project

The technology is developing rapidly in the field of computer industry. The data processing ability has increased, capacity to store data in digital format has expanded dramatically and the access to information has been made easy and most of all these facilities are becoming cheaper for the common people. This trend has also raised a concern in security professionals to keep sensitive and private data like passwords and credit card numbers secure as long as it is being processed in physical memory and to discard it securely when no longer needed to keep it away from adversaries and criminals.

In this research work we will do the forensics of physical memory and page file in search of sensitive data. Data retention in physical memory is influenced by operating system (system software), applications processing the data and the underlying hardware of the computer [55], [56] and [57]. Experiments will be conducted in virtual environment using VMware incorporating both physical memory and page file. Different scenarios will be established in windows operating system environment that will closely resemble real world situations. The whole analysis will be conducted using the freely available tools on Internet.

1.2 Keywords

Computer forensics, Digital forensics, Digital evidence, Digital investigations, Incident response, Physical memory forensic and analysis, Volatile data forensics, Memory analysis tools, Page file forensics, Swap space analysis, Sensitive data forensics

1.3 Problem Description

When a digital crime occurs, the main focus of the forensic examiners is to acquire and analyze non volatile data from the suspect machines. But this approach is no longer applicable due to the following reasons.

- In today's end user computers the hard disk data storing capacities has increased dramatically. 500 GB is a normal capacity in today's computers and there is a growing concern how to increase the forensic abilities of the currently available tools and techniques to process such huge amount of data [23], [84], [51] and [64]. The same is the case with physical memory in today's computers. 2 to 4 GB of physical memory is a normal standard in current end user computers. And when this amount of data in physical memory is incorporated with the swapped data in page file on hard disk, the prospects of finding the required results are even higher. Therefore it is not a wise idea to ignore physical memory of such huge capacities in digital crime investigations.
- All transient data and volatile information such as network connections, chat logs, command histories, process information and open files that reside in physical memory will be lost if the "pull the plug" approach is followed. [108] and [48].

- Some times evidence can be resident only in memory. For example there are many Malware programs that rum directly from physical memory [75], [30], [52], [81], [41] without being installed on the hard disk thus leaving no trace on the hard disk.
- In case of traditional disk forensics if the acquired evidence is encrypted then physical memory is the next immediate place where we can find keys for the encrypted data.

The importance of physical memory forensics is obvious from these points but it is still in the stage of infancy and require more serious attention.

The currently available tools and methods for the analysis of memory evidence are limited and require a lot of attention compared to the importance of sensitive data in memory. The available free tools on the Internet from different authors are limited in functionality and scope especially when physical memory is incorporated with page file in search of finding the evidence. The currently available tools should be practically tested by practical examples. This will show us not only their efficiency and ability but also their trust worthiness in a court of law.

1.4 Research Questions

Physical memory forensics is an evolving field of research. Since the operating systems from Microsoft are closed source therefore it requires even more attention from the researchers and forensics practitioners. This master thesis will try to answer the following questions in the field of physical memory forensics:

- What is the current state of physical memory forensics?
- What are the currently available physical memory forensics tools and techniques?
- Have the currently available tools the ability and scope to analyze the physical memory and page file in currently available end user computers?
- Can we extract sensitive data from Physical Memory and Page File when the system is in a particular state?
- Are the current hardware devices, system softwares (operating systems) and application softwares taking care of the sensitive data with the increase in memory size?

1.5 Planned Contributions

This thesis work will contribute to the knowledge and understanding of physical memory and page file forensics both theoretically and practically by experiments. A state of the art in the field of physical memory and page file analysis on windows systems will be provided. This will help both forensic professionals and researchers to get more understanding of this emerging area.

A Summary of the currently available tools for the analysis of physical memory and page file will be provided with their features. To see the strength of the available tools, experiments will be conducted on selected tools on different scenarios in virtual environments. This will not only give an idea to the researchers working on windows systems but also forensic practitioners will benefit from this.

Finally and importantly from this research work we will get an idea of how long

physical memory and page file can maintain sensitive data and the influence of operating system and applications on data retention in different states of computer.

1.6 Research Method

The research methodology for this thesis will cover both theoretical and practical approaches. In theoretical part an introduction to the field of digital forensics will be provided covering digital forensics process and basic definitions of terms and then discussing physical memory and page file forensics which is the main focus of this work. A summary of the currently available tools and techniques for the acquisition of physical memory and page file will be discussed in detail. Then a detail description of the analysis tools will be discussed.

Based on the discussion of the analysis tools in theoritical part, some freely available tools will be selected for the practical part of the thesis work. A set of scenarios will be selected and then implemented in virtual environment. Then the analysis of the physical memory and page file will be performed for the extraction of sensitive data using the selected tools from the theoretical part of the thesis.

1.7 Research Limitations

This master thesis will be limited to the forensics of physical memory and page file. All the experiments will be conducted using virtual environments. VMware will be used as the test bed for the experiments and the operating systems will be windows XP professional SP3.

1.8 Outline of the rest of the Report

The rest of this thesis report is organized as follows:

- Chapter 2 will provide background information to the field of computer forensics. The basic terms will be discussed together with the forensic process in a digital investigation.
- Chapter 3 will discuss physical memory and page file forensics. State of the art in this area will be discussed. The chapter will conclude with the acquisition and analysis methods and techniques for the physical memory and page file.
- Chapter 4 will discuss the methodology for the experiments.
- Chapter 5 will discuss the experiments.
- Chapter 6 will discuss the results of the experiments.
- Chapter 7 will discuss conclusions and future directions.

2 Science of Digital Forensics

This chapter provides an introduction to the field of digital forensics highlighting the sub disciplines of this field and finally depicting the position of physical memory. Definitions of the basic terms and principles are explained. The chapter also explains the process and methodology while investigating digital crime incidents. The same methodology will be used later for the experiments part in this thesis.

2.1 Digital Forensics

Digital forensics is a sub branch of forensic science. Forensic science is defined as " The application of a broad spectrum of sciences to answer questions of interest to a legal system" [47] and [113]. Forensic science encompases many fields. These include but are not limited to:

- Physiological sciences
  Examples include Pathology, Dentistry, Finger print and DNA analysis.
- Social sciences
  Examples include psychology and toxicology.
- Cybertechnology
  Emaples include Information forensics and Computer forensics.

Digital forensics is placed in the last discipline. Many definitions have been proposed by various people. Here I will mention the most commonly used definitions.

The Digital forensics research work group 2001 has defined digital forensics as "The use of scientifically derived and proven methods towards the preservation, collection validation, identification, analysis, interpretation, documentation, and presentation of digital evidence derived from digital sources for the purpose of facilitation or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations" [42].

Another simple definition has been proposed by Schweitzer [99] and Carvey [37] . "Digital forensics is the science of acquiring, retrieving, preserving, and presenting data that has been processed electronically and stored on computer media".

A closely related definition to the one stated by Schweitzer is that of McKemmish [70] and Michael et al [72]. "Digital forensics is the process of identifying, preserving, analyzing and presenting digital evidence in a manner that is legally acceptable".

The history of digital forensics can be traced back to 1984 when FBI established the Computer Analysis and Response team (CART) in response to the demands for services from the law enforcement community. By the year 1996 the postal inspection service with the help of FBI had established a Computer Forensic Unit to help each other in the development of computer forensic capabilities. By the end of 1998 they had formed the Scientific Working Group Digital Evidence (SWGDE). The investigation done in that

time was mainly on digital photography, digital video and audio evidence [31] and [80] . Since then the field of computer forensics is progressing and has expanded to many other sources that contain digital data in the shape of evidence. Rapid advancement in technology coupled with the volume and sophistication of digital crimes has made the field of digital forensics even more diverse. The number of incidents are continuously rising each year. To quote an example of our dependence on computers and the way digital crimes are affecting all sectors of society, Figure 1 details the CSI Computer Crime and Security Survey results of computer security practitioners in U.S. corporations, government agencies, financial institutions, medical institutions and universities from 2004 to 2008 [85].

Table 1	2004	2005	2006	2007	2008
Denial of service	39%	32%	25%	25%	21%
Laptop theft	49%	48%	47%	50%	42%
Telecom fraud	10%	10%	8%	5%	5%
Unauthorized access	37%	32%	32%	25%	29%
Virus	78%	74%	65%	52%	50%
Financial fraud	8%	7%	9%	12%	12%
Insider abuse	59%	48%	42%	59%	44%
System penetration	17%	14%	15%	13%	13%
Sabotage	5%	2%	3%	4%	2%
Theft/loss of proprietary info	10%	9%	9%	8%	9%
from mobile devices					4%
from all other sources					5%
Abuse of wireless network	15%	16%	14%	17%	14%
Web site defacement	7%	5%	6%	10%	6%
Misuse of Web application	10%	5%	6%	9%	11%
Bots				21%	20%
DNS attacks				6%	8%
Instant messaging abuse				25%	21%
Password sniffing				10%	9%
Theft/loss of customer data				17%	17%
from mobile devices					8%
from all other sources					8%

Figure 1: CSI Computer Crime and Security Survey

To cope with the types of incidents and to facilitate detection and correction of crimes the field of digital forensics can be broadly divided into the following areas:

2.1.1 Internet Forensics

Internet Forensics covers the investigation of criminal activity that has occurred on the Internet. It deals with the analysis of the origins, contents, patterns and transmission paths of e-mail and Web pages as well as browser history and Web server scripts and header messages. Internet forensics covers a vast area. Some of the purposes of Internet forensics include tracing anonymous emails, IP addresses and DNS, probing unauthorized access, tracing cyber crimes, protecting online reputation and gathering evidence from multiple sources [46].

2.1.2 Network Forensics

Network forensics deals with the capture and inspection of packets passing through a selected node in the network In order to identify attacks. Networks contain digital evidence that can be used to establish that a crime has been committed, determine how a crime was committed, provide investigative leads, reveal links between an offender and victim and disprove or support witness statements. Packets can be inspected on the fly or stored on disk for later analysis. Within network, evidence can be gathered from various sources depending on the unique requirements and nature of the investigation. It can be gathered at the server level, proxy level or from several other sources such as routers, switches, IDSs and firewalls [76] and [65].

2.1.3 Computer Forensics

Computer forensics can be referred to as the forensic examination of computer components and their contents [65]. In general, computer forensics is used to identify evidence when personal computers are used in the commission of crimes. This evidence can be located in the components of computer where data can reside or store. These locations include but are not limited to the following locations:

Hard drives,
Compact disks,
Flash Drive
Zip drives
Printers,
Random Access Memory (RAM),
Cache Memory
Registers

When a digital crime occurs, depending on the nature of the incident the role of computer can be one of the following [99]:

- Computer as Object of the Crime:
  When a computer is affected by the criminal act, it is the object of the crime. For example, when a computer is stolen or destroyed.
- Computer as Subject of the Crime:
  When a computer is the environment in which the crime is committed, it is the subject of the crime. For example, when a computer is infected by a virus or a malicious piece

of code.

- Computer as Assistant in the Crime:
  In this case computer can be used as the tool for conducting or planning a crime. For example, when a computer is used to forge documents or break into other computers, it is the helper in the crime.

When a digital incident occurs, there are usually two approaches followed by investigators to diagnose the incident.

1. Traditional Forensics
   In traditional forensics approach also called "snatch and grab" method, an investigator pulls the plug on the machine, and then images (copies) the hard disk, either on site or (after confiscating the machine) in a lab. An analyst examines the image in a controlled environment by repeatable steps in search of evidence.
   In this case the digital incident has occurred but the computer is without power or not in running state. In such kind of digital incidents the investigators will consider mostly those sources that store the data for longer duration. One main source is hard drive. Thus the investigator will make images of the sources containing permananent data and then continue with the analysis. Traditional digital forensics attempts to preserve all (disk) evidence in an unchanging state.
2. Live Forensics
   In this case the digital incident has occurred and the computer is still running.

   Live digital forensics can be further subdivided into two sections.

- Live Response Forensics Scenario:
  In live response scenario an investigator surveys the crime scene, collects the evidence, and at the same time probes for suspicious activity. This was the traditional approach to investigating volatile volatile data sources in a live forensics. During this process the investigator will query the suspect system using API style tools before imaging the hard drive. The purpose of the first responder was to trace rouge connections or suspicious running processes and commands [99] and [61].
  During a live response scenario, the state of the running computer is not static. This could lead to the same query producing different results based on when it is run. Therefore an investigator could compute a cryptographic checksum of the tool outputs and make a note of this hash value. This would help throw away any doubts that the results had been altered later.
- Volatile Memory Forensics Scenario:
  Physical memory (volatile memory) forensics is a relatively new field of computer forensics where an investigator collects the memory dump and performs analysis in an isolated environment.
  In this scenario along with running some commands on live system depending on the requirements of the investigation and type of incident, the first responder would acquire a physical memory dump and page file of the compromised system and transmit it to the data collection system along with the image of hard drives. Unlike traditional hard drive forensics, no hash is calculated for memory before acquisition. Since

physical memory is volatile by it nature, the imaging process is taking a snapshot of a "moving target." The main difference between this approach and Live Response is that no additional evidence is needed on the compromised system. Therefore, the memory evidence can be analyzed on the collection system [61] and [37]. A comparison of physical memory forensics with live response scenario and traditional forensics will be discussed in chapter 3.

2.2 Digital Investigation Process Models

As mentioned in the beginning of this chapter, there is an increase in the number of digital crimes worldwide and techniques and tools to counteract these crimes from laws enforcement people must keep the same progress to ensure that the playing field remains level. Another part of this race, and perhaps more crucial, is the development of a methodology in digital forensics that encompasses the forensic analysis of all types of digital crime scene investigations.

According to "A digital forensic investigation is a process that uses science and technology to examine digital objects and that develops and tests theories, which can be entered into a court of law, to answer questions about events that occurred." A process model helps us to organize and implement the procedures conducted during a typical digital crime scene investifgations. Several conceptual process models have been proposed by many people some focusing on tools and technology while others on the types of incidents. In this section a brief description of the currently proposed models will be outlined and finally a general detail of the stages in models will be defined where physical memory can be included in the process.

Mandia and Prosise [61] have proposed an incident response methodology. This methodology is composed of such steps as "pre-incident preparation, detection of incidents, initial response, response strategy formulation, duplication, investigation, security measure implementation, network monitoring, recovery, reporting, and follow-up." The model is oriented towards those who respond to the physical crime scene. This model serves the intended purpose of investigating computer crimes and can be applied to general crime scenes of this nature.

The U.S. Department of Justice (DOJ) [78] presented an abstract process model consisting of Şcollection, examination, analysis, and reportingŤ as the main steps. This model tried to apply the traditional physical forensic knowledge to electronic evidence. The DOJ model also lists the types of evidence, potential locations of that evidence as well as the types of crimes that may be associated with the evidence.

The Digital Forensics Research Workshop (DFRW) is another important participant in developing forensic process models. The unique feature of DFRW model is that it is one of the first large-scale consortium governed by academic professionals and researchers rather than law enforcement agencies. This focus is helping to define the direction of the scientific community towards the challenges of digital forensics. Because the practitioners and forensic researchers are not using standardized procedures and protocols [42]. The DFRW model consists of "Identification, Preservation, Collection, Examination, Analysis, Presentation, and Decision" steps.

Reith and Gunsch at the U.S. Air Force have also proposed an abstract process model incorporating different features of several models. The steps of this model are almost the similar to the ones proposed by DOJ [42] and DFRWS [78]. This model introduces

a more general model, making it possible to use for any digital investigation, regardless of technology or crime. The main steps of this model are "Identification, Preparation, Approach strategy, Preservation, Collection, Examination, Analysis, Presentation and Returning evidence" [69].

Baryamureeba and Tushabe [24] have proposed an enhanced digital investigation model. The model is similar to IDIP model [33] in phases but the approach to each phase is iterative rather than linear. The components of this model are "Readiness phase, Deployment phase, Trace back phase, Dynamic phase, and Review phase."

At the end of this section we will discuss The Integrated Digital Investigation Process Model(IDIP) proposed by Carrier and Spafford [33] in 2003. This digital investigation process model is build upon the experience gained from physical investigations. Using the concept that a computer is itself a crime scene, the investigation theory for a physical crime scene is applied to a digital investigation. The digital crime scene investigation is integrated with the physical crime scene so that physical evidence can be collected that ties the digital activity to a person [33].T his model called The Integrated Digital Investigation Model(IDIP) consists of 17 different phases which has been organized into 5 groups. We will define each phase of this model and elaborate where physical memory can be included in this model. The five main components of this model are shown in figure 2.

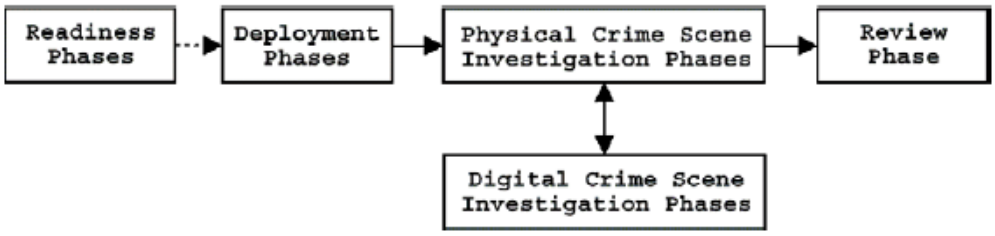

Figure 2: The five main phases of IDIP

1. Readiness Phases

In this phase operational and infrastructure preparations are done. The purpose is to ensure that the means and resources to conduct the crime investigation. operational readiness means that the personnel have have the required training and equipment for the investigation. While infrastructure readiness means that the target of investigation exists. If there is no target to be investigated then the purpose bears no results [33].

2. Deployment Phases

This phase consists of detection, notification, Confirmation and Authorization. In detection and notification the incident is detected and the appropriate people are notified. While confirmation and authorization means receiving authorization to fully investigate the incident and the crime scene. This can be in the form of a search warrant or some other legal document or permission [33].

3. Physical Crime Scene Investigation Phases

In this phase physical evidence regarding the crime is collected. This evidence is used later to reconstruct the crime scene of the incident. The phases of physical crime scene are presented in figure 3 showing its interaction with the digital crime scene. During physical crime scene investigations if some digital evidence in encountered the the digital crime

scene is triggered and the investigation for this digital evidence will switch to digital investigations. We will focus only on the details of the digital crime scene [33].

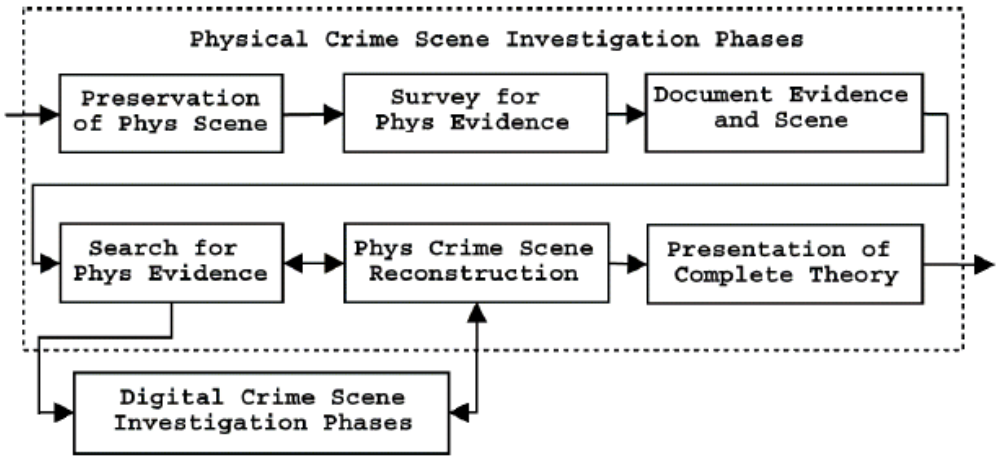

Figure 3: The six phases in the physical crime scene investigation and the interaction with the digital crime scene investigation.

4. Digital Crime Scene Investigation Phases

Physical crime scene is initiated when some device is traced as physical evidence from the physical crime scene. After the completion of this phase its results are sent back to the physical phase to continue with the rest of the investigations. The six phases of the digital crime scene are depicted in the figure 4 [33].

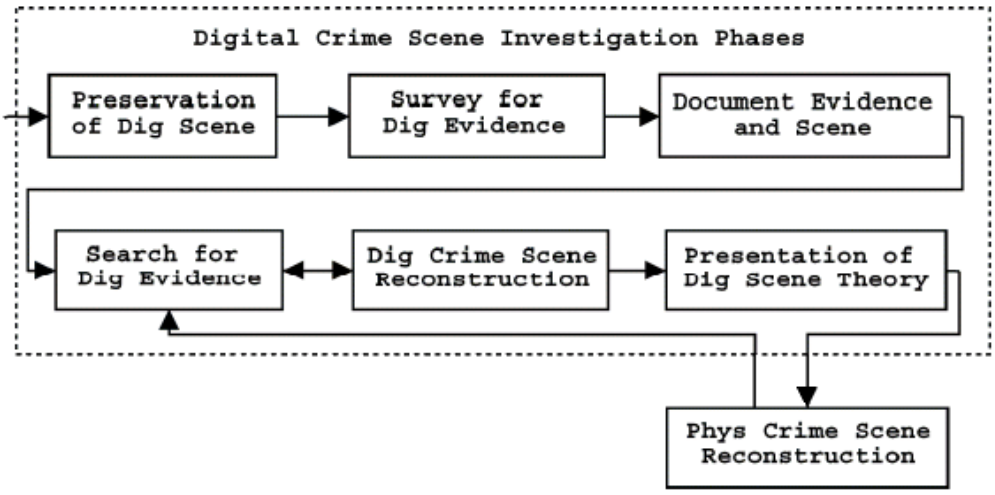

Figure 4: The six phases in the physical crime scene

4.1 Preservation Phase

The goal of this phase is to keep the digital crime scene as intact as possible. Upon arriving at a crime scene, investigators should be careful not to interfere with the digital evidence found, as the state of the digital data may be extremely volatile [33]. Some of the steps included during this phase are isolating the system from the network, collecting

volatile data that would be lost when the system is turned off, and identifying any suspicious processes that are running on the system. The collection of volatile memory is introduced during this phase of investigation [18]. In normal investigation during this phase the disk drives will be imaged. Along this the the physical memory will also be copied for later analysis. Physical memory can be captured using a number of possible mechanisms. These acquisitions mechanisms are discussed later in chapter two. During this phase if the system is live then the first responder must be very careful about two principles when dealing with volatile memory (physical memory) These are Order of Volatility (OoV) and Locard exchange principle. Order of Volatility principle outlined in RFC 3227 [27] states that one should proceed with collection from volatile to non-volatile mediums; that is physical memory should be collected before hard disks. The Locard exchange principle says that "Every contact leaves a trace" [116]. In terms of memory forensics it means we can't image physical memory without disturbing it . The other important point is that the status of memory is not static, it is constantly changing and the image of such memory will represent the status of a dynamic system that can't be reproduced at a later stage [19]. The challenge is to minimize the changes to the evidence, understand the effect of the changes, and minimize the trust that is placed on the operating system of the live investigations. While preserving the digital evidence write blocker should be utilized when imaging hard drive, physical memory or page file to prevent contamination of the original data, and to ensure the integrity of the copied data. Hashing algorithms like SHA-1, SHA-256 or MD5 should be used to ensure the integrity of the original data [33].

4.2 Survey Phase

After the digital crime scene is preserved, the next stage is to look for the required evidence. This is conducted in the survey phase of the investigation on the images of the digital evidence acquired during preservation phase. The Survey Phase looks for the obvious pieces of digital evidence of the crime under investigation. Depending on the type of incident the survey phase can be conducted live or in controlled isolated laboratory. Just like the disk image forensics, the images of physical memory and page files can be searched for the required evidence [18]. When physical memory is integrated in this phase, the preserved data can be verifiably copied and analyzed in a more trusted and well defined environment. Since at this stage we have the the images of memory and page file so the procedures and techniques will remain the same as applied to hard disk images. One thing should be noted here that if the results are generated from live response by using the 1st responder tool kit that do not include image of physical memory, then the results cannot be repeated at this phase. If the Survey Phase is performed on a live system, then a forensic image of the system should still be taken so that any digital evidence can also be collected in a controlled lab environment. The main purpose of the survey Phase is to show the investigator the skill level of the suspect and what analysis techniques the investigation will require [33].

4.3 Documentation Phase

During this phase each and every step taken during analysis should be properly written, saved in a proper format and documented. Chain of Custody forms should be created in this phase if the purpose of the evidence is to be presented in a court of law. In practice the documentation phase is a continuous phase during the whole investigation process. It is initiated from from the movement the first the incident is detected or reported to the responders and is ended when the final report is handed over to the authorities and case

is resolved [33].

4.4 Search and Collection Phase

During this phase a thorough analysis of the acquired digital evidence is performed to look for the required information depending on the type of investigation. The analysis is done in a controlled environment preferably on a computer equipped with the required tools and programs [33]. Some of the initial analysis is already done in the survey phase of the investitigation. Here it is complemented with detailed search for the required evidence. Here different analysis. techniques are applied on the acquired images. For the memory and page file images, their own analysis methods are used. There are many tools and techniques available for this purpose. They are discussed in chapter 3 in detail.

4.5 Reconstruction Phase

In reconstructing phase the pieces of evidence identified in the search and collection are combined to get a complete picture of the cause of incident. Based on the initial assumptions set for the analysts try to reconstruct the crime to prove or disprove their findings. On volatile memory side the the pieces of evidence recovered from physical memory and page file will be combined to get a full understanding of a suspicious activity, processes running and pages visited. Questions like what,when and how are answered in this phase [33].

4.6 Presentation Phase

This is the last phase of the digital crime scene investigation during which the results of the investigations are forwarded to the physical crime scene investigation. The physical crime scene investigators then integrate the results of both phases to reconstruct the crime and arrive at the final conclusions. During digital crime scene investigation phase the investigaorts can share data and information with physical crime inspectors. So both phases go side by side [33].

5. Review Phases

This phase is mainly done to see the performance of both digital and and physical investigations. The roles played by both types of investigations are analyzed and areas of improvement are identified for future enhancements [33].

2.3 Terminology

In this section the basic fundamental concepts, rules and guidelines used in the digital forensic investigation particularly the ones related to physical memory forensics, are described. During the experiment part of this thesis we will follow these rules and guidelines.

- Digital Data
  "Digital data is information and data of value to an investigation that is stored on, received, or transmitted by an electronic device." [33] and [34]. This evidence is acquired when data or electronic devices are seized and secured for examination.
- Digital Evidence
  "A digital evidence of an incident is any digital data that contain reliable information that supports or refutes a hypothesis about the incident". [34] and [35].
- Digital Investigation
  A digital investigation is a process that uses science and technology to examine digital

evidence and develops and tests hypotheses to answer questions about events that occurred during a digital incident. Example questions include What caused the incident to occur, When did the incident occur, and Where did the incident occur [34].

- Chain of Custody
  Chain of custody refers to the process of maintaining and documenting the chronological history of the events and steps done during the course of a digital forensic investigation. The chain of custody documents should include name or initials of the individual collecting the evidence, each person or entity subsequently having custody of it, dates the items were collected or transferred, agency and case number, victimŠs or suspectŠs name, and a brief description of the item [78]. The main purpose of this process is to maintain the record of all activities taken on the evidence. This not only shows consistency of the investigation process but also this process is very important if the evidence is to be presented in a court of law to prove some one guilty or innocent.

- Evidence Reliability and Integrity
  This is another important factor for a sound and complete forensics investigations. Reliability refers to the forensic trustworthiness of the digital evidence. This can be ensured by using the standard practice and rules during the whole investigation process. The people conducting the investigation should be experienced and expert and the tools used should be acceptable to the court of law.
  Integrity in simple terms means the original digital evidence should not be altered during the whole course of investigation. Chain of custody is one such process for keeping the integrity of the digital evidence. At any stage during chain of custody is interrupted, the integrity of the evidence is compromised. The best way to keep the integrity of evidence is to use cryptographic hash functions. Hashing algorithms like SHA-1, SHA-256 or MD5 should be used to ensure the integrity of the original data [33]. Hash functions make it difficult though not impossible, to break the integrity of the evidence.

- Order of Volatility (OoV)
  If the collection and preservation of digital evidence is done correctly, it is much more useful in finding the required results, and the chance of admissibility in the court are increased. Order of volatility plays an important role particularly in the case of physical memory forensics. Accoring to this rule during the preservation phase of the digital investigation we should start with the collection of the most volatile data first. According to RFC 3227 [27], the order of volatility is as follows:

  1. registers, Cache
  2. routing table, arp cache, process table, kernel statistics, memory
  3. temporary file systems
  4. disk
  5. remote logging and monitoring data that is relevant to the system in question
  6. physical configuration, network topology
  7. archival media

From this list we can see that collection of physical memory should be done before hard disk. Since physical memory is volatile by nature and the data residing inside this source could disappear quickly.

- Locard Exchange Principle
  This is another principle that plays a vital role during the forensic investigation of physical memory. The Locard exchange principle says that "With contact between two items, there will be an exchange". Essentially this principle is applied to physical crime scenes in which the perpetrator (s) of a crime comes into contact with the scene, so he/she will both bring something into the scene and leave with something from the scene [116]. The same principle is applied to digital crimes. When one computer connects to another, there will be an exchange of information between them. This holds true whether the intention is to use one system to gain unauthorized access or to perform incident response information collection from one system to another. Data will be exchanged during communications between the two systems [37]. This becomes even more important when dealing with volatile data. The physical memory in most cases cannot be imaged without changing it. Some piece of code must be introduced into memory to acquire its contents which by itself will change the original evidence and violating the integrity of the evidence.Therefore the purpose of the 1st responder should be to minimize the changes to memory evidence while acquiring its image. Since the status of the memory is constantly changing and the image of such memory will represent the state of a dymanic system that cant be regenerated later. In other words, the investigator should be aware of the overall context of the investigation in order to make informed decisions during memory acquisition [116].

3 Physical Memory Forensics

This chapter provides an introduction to the forensics of physical memory and page file in windows operating systems. A basic understanding of the internal working of the physical memory wil be introduced followed by the page file. We will also highlight the available memory and page file acquisition methods. The chapter will conclude with the analysis methods of the memory and page file.

3.1 Background

We will continue the discussion from chapter 3 about the investigation options available to investigators. Here more detail will be added comparing the different scenarios and then coming to the current state of physical memory forensics. In the field of Digital forensics, attention has been paid mostly to the forensics of non-volatile media such as hard drives or storage peripherals. This type of forensics is called classic or traditional forensics. A typicsl scenario of such kind of forensics could be like this. On the detection a digital incident the incident response team will arrive at the scene. They will find the systems dead (in shut down state) or live (up and running) [114] and [37]. While following the traditional approach of digital forensics, in both of the above cases the first responder will seize the machine, attach the hard drive to a forensic system, make a complete image of the hard drive and then continue with rest of the investigation process. This kind of investigation is depicted in figure 5 [63]. This kind of investigations is no longer a wise action because of the several reasons highlighted in chapter one.

The other kind of investigation is called live forensics. Here as mentioned briefly in chapter 2, the first responder will conduct the acquisition of volatile data before making images of the hard drive. Such information is obtained in most cases by the first responder tool kit mostly consisting of free ware utilities, tools native to the systems, and some Perl scripts. The purpose of this information is mostly for trouble shooting and to get an idea about what was happening on the system while it was in running state. [112] and [61]. The volatile information obtained though the first responder tool kit generally consists of process information, network connections,status of the network,command histories and logged on users [37] and [61]. Depending on the type of incident and organization the investigation and analysis is done either done locally or over a network by using covert or overt method. Figure 6 shows the flow of events in graphical manner [63].

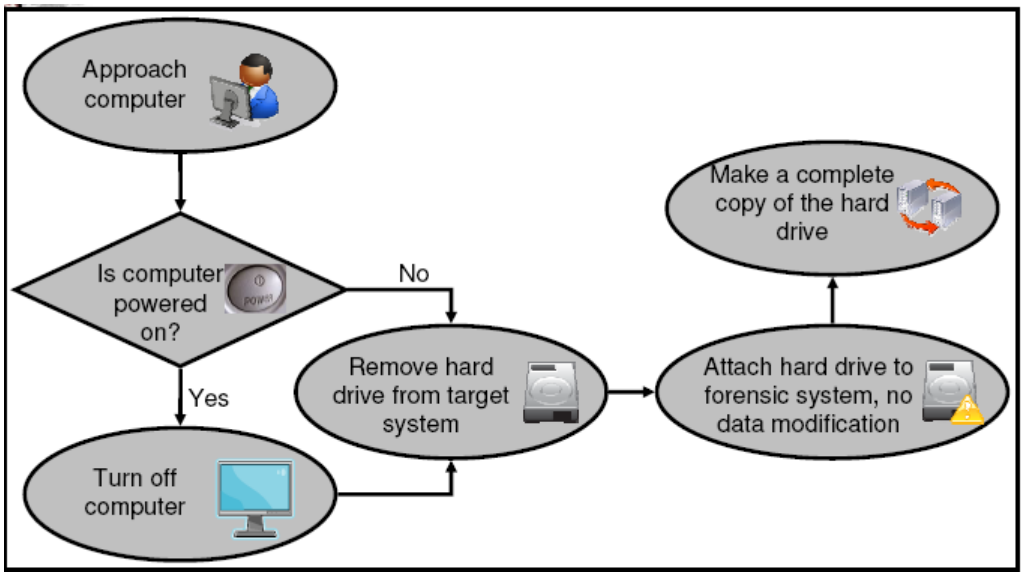

Figure 5: Traditional Forensics [63]

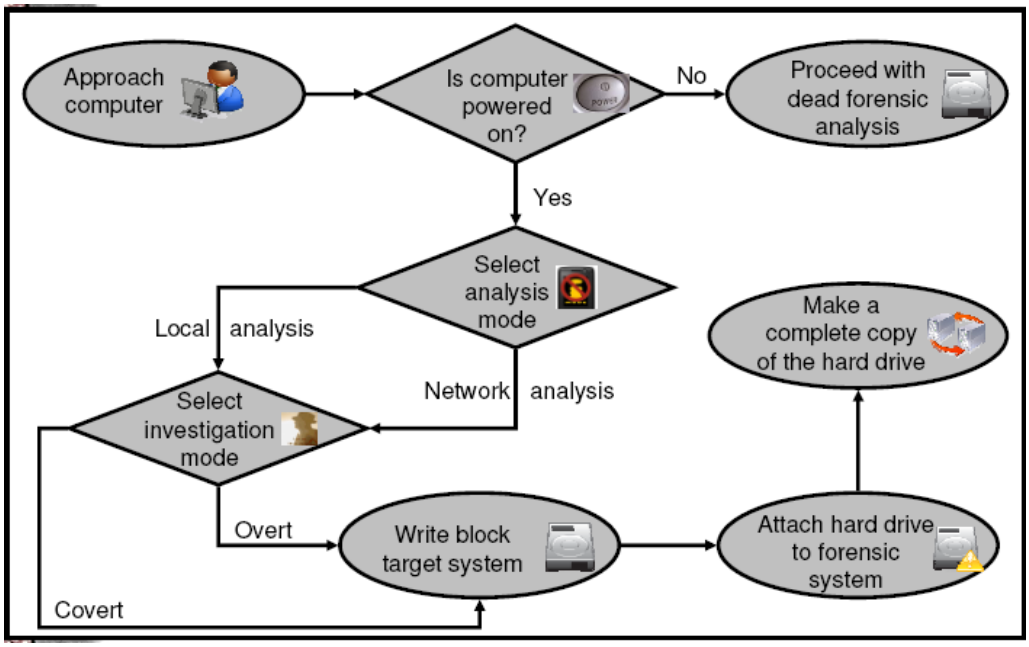

Figure 6: Live Forensics [63]

Traditionally this is the only useful approach to investigating memory. traditional But as the amount of permanent storage and memory is increasing, the importance of volatile information has become even more important. But this kind of volatile information has several draw backs. There are four main issues with such kind of live response forensics [18], [19] and [29].

3.1.1 Impact on the Target System

During a forensics investigation one rule of best practice is to minimize the impact on the target system. But In case of live forensics it is an unavoidable action. Volatile data cannot be acquired without disturbing it and there is no measure how much the impact of the live response tools will be on the target system. Ultimately this will raise questions on the reliability and integrity of the whole process [18] and [29].

3.1.2 Repeatability of the Results

The 2nd issue with live response investigation is that the results are not repeatable. Live response tools are run on an unknown system in an unknown state. The tools used can be trustworthy and they can be verified by third parties but the results produced by them cannot be repeated or reproduced in the case of live response scenario. Because they are run on a system whose state is constantly changing. And this is this is essential when the results need to be proved in a court [18] and [29]. Thus it is quite hard to to prove the authenticity of the results produced.

3.1.3 Asking New Questions

This issue is a continuation of the previous issue.It is had not to reproduce the same results by using the same tools but at nay stage later in the investigation if the examiner needs more details about a process or about the output of a command, it cannot be answered. The reason is that the evidence in hand is not the same. It will give different results [18], [29].

3.1.4 Trust

The first responder tool kit is run on a system whose status is not trustworthy. The system might be compromised. In this case the tools have to rely on the underlying operating system and hardware. It means the tool set by itself can be trustworthy but we cannot be sure about the trustworthiness of the results from from such a system [18] and [29].

Considering the issues with classical disk forensics and live response forensics it becomes even more important that we should include physical memory forensics to the traditional forensics process. As described in chapter one it can be included during the preservation phase of the crime investigation. But it was until 2005 when much attention was given to the analysis of physical memory. It all started from the memory challenge of DFRWS 2005 [43]. Since then much attention has been paid to the extraction of meaningful information from both physical memory and page file. The Standards for acceptance and inclusion of physical memory in the forensic cycle are evolving, and legal precedents are still being established. The big challenge to the acceptance of physical memory is that, it represents a state of a system which is dynamic. But as the capacity of memory is on the increase and new tools and techniques are being developed and soon the paradigm of physical memory forensics will become the accepted norm.

3.2 Physical Memory Management

This section provides an introduction to the organization of memory in windows environment. The purpose is to get an idea about the internal working of the memory, how data is handled once it is reserved by processes and how physical memory and virtual memory correlates with each other. Our discussion will be mainly on 32 bit systems but some concepts are also applicable to 64 bit systems. For the whole discussion [45], [88],

[105], [91] and [60] are used as main references.

3.2.1 Basic Organization of Physical Memory

Computer systems organize memory into fixed size pages just as they organize file systems into fixed-size disk blocks. The size of this size is generally 4 KB or 4 MB depending on the hardware architecture. The usage of this memory is centrally controlled by the memory manager. The memory manager provides a set of core services e.g allocating memory to pages, sending data to page file when needed and bring them back and sharing the limited memory among files [88] and [105]. Since the amount of available memory is small therefore the most opeating systems including windows use a concept called virtual memory. According to this concept any process running on 32 bit windows systems is allocated a set of virtual memory addressees of 4 GB in size. Half of this size is reserved for the for the private use of the process and the remaining half is reserved for the operating system use [60].

3.2.2 Virtual to Physical Address Translation

The memory manager maintains page tables for every process that needs to keep their addresses and the CPU will translate these virtual addresses into physical addresses. For each virtual address is given its own space called page table entry (PTE). In this PTE the virtual and physical addresses are mapped with each other [88] and [105]. Windows uses a two level page table structure to translate virtual to physical addresses. In this case the virtual address of a process is divided into three parts; page directory index, page table index, and byte index (page frames). The purpose of each of these structures is as follows:

- The page directory index is used to locate the page table in which the virtual address's PTE is located.
- The page table index is used to locate the PTE.
- The byte index is used to find the address of the physical page.

Figure 7 shows the schematic diagram between all these tables in the conversion process.

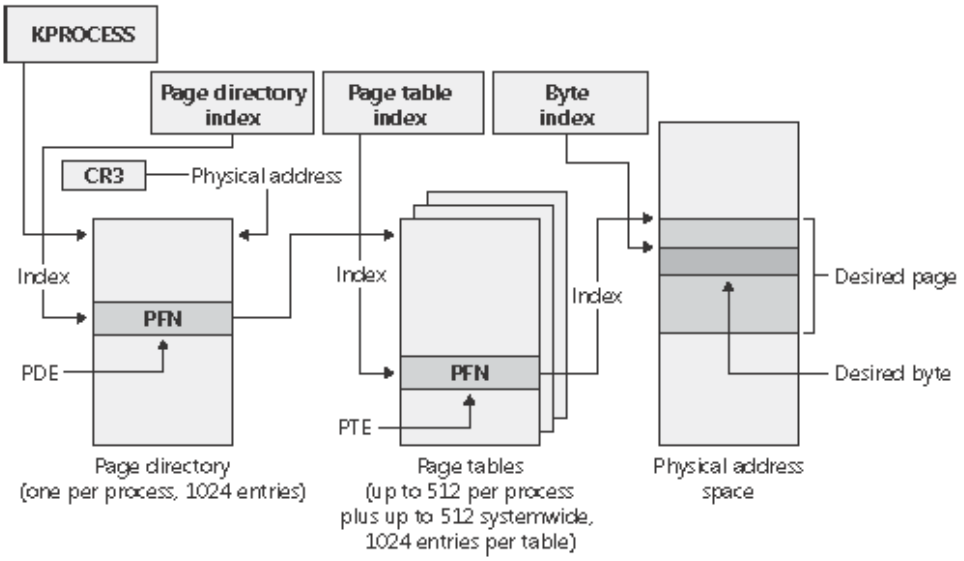

Figure 7: Virtual to physical address translation process [88]

The following basic steps are involved in translating a virtual address: [88] and [60]

1. The MMU locates the page directory for the current process by setting a special register called the CR3 register with the address of the page directory table.
2. The page directory index is used to locate a specific page directory entry (PDE) which describes the position of the page table needed to map the virtual address. The physical address within the PDE is referred to as page frame number (PFN).
3. The page table index is used to locate a particular page table entry (PTE) that describes the physical location of the virtual address being translated.
4. The PTE is necessary to locate physical memory page. If the page is valid (valid bit set to one), the PTE contains the page frame number of the page in physical memory. Conversely, if the page is invalid (valid bit set to zero) the MMU fault handler locates the page and tries to make it valid.
5. When the PTE is pointed to a valid page, the byte index is used to locate the address of the desired data within the physical page.

We Will not go into the details of memory management since this is beyond the scope of this thesis work. For more details the reference books and articles should be consulted.

3.2.3 Page File Management

As we discusses in the previous section physical memory is limited in capacity compared to the big space reserved by each process by the memory management unit. Therefore, not all processes can be accommodated in the memory at the same time. To cope with this problem the memory management unit uses a technique called paging. In this technique pages of 4 KB in size are pages out to the hard disk thus freeing space for more processes in memory. Which pages should be paged out to page file entirely depends on the operating system and application developers [101]. Different algorithms are used for this purpose like first in first out (FIFO), last in first out (LIFO), priority level of the data

and size of the process. When a page is swapped out to page file it is also marked in the corresponding page table in the memory. The same procedure is used to bring the page back to memory when referenced. There are some pages that cannot be paged to page file these consists of ranges of system virtual addresses that are guaranteed to be resident in physical memory at all times. The paged out pages are stored in one or more files with the name "pagefile.sys". By default the page file is located in "C:/pagefile.sys" or in "C:/windows/win386.swp" in windows 9x systems [101]. On windows systems up to 16 different page files may be present. In most cases the size of page file is 1.5 to 3 times the size of the installed physical memory on the computer. It is not mandatory that page file should be utilized when the physical memory is exhausted; a small portion of space in page file will still be used even when there is no paging. In this case the page file will be allocated to virtual memory pages for which no corresponding pages are available in physical memory [91]. Page file is not erased by default in windows xp and vista. The settings can be found here in registry in windows operating systems: "HKEY LOCAL MACHINE/System /CurrentControlSet/Control/Session Manager/Memory Managemen".

3.3 State of the Art in Physical Memory Forensics

Physical memory analysis, like all other forensic endeavors, is concerned with the retrieval of information that can serve as evidence in criminal investigations. Plenty of work has been done in the field physical memory and page file forensics. In this section we will look at the work of different authors who have worked on extracting meaning from either physical memory or page file or including both in their analysis.

Several people have worked on the life span of data in physical memory. Jason Solomon et al [54] in their paper User data persistence in physical memory 2007 determined the age of unallocated, deallocated and terminated pages of memory and cache. The observation were done in the user space of the memory on windows xp service pack 2 and suse Linux 10. Three experiments were conducted with the load of 100, 1000 and 1500 processes by using a C language code. The results were surprising both in terms of load on the systems and the rate of decay of both types of pages. Under light load of 100 processes, ages of the pages is less than two minutes for both xp and Linux. While under heavy load, where the systems were forced to do paging, the age of pages is less than 5 minutes in windows xp while less than 2 minutes in Linux. Page file was not considered in this work. Same sort of work was done by Walters and Petroni [18] by determining the rate of decay of data in in memory. They conducted their experiments in VMware on windows XP using 256 MB and 512 MB respectively. They concluded that the rate of decay in memory is a function of time and the more the memory, the less is the rate of decay.

Chow and others wrote a series of papers [55], [56], and [57] highlighting the importance of the life time of sensitive data in volatile memory. They analyzed the life time of sensitive data in a simulated environment called TaintBochs. By using this environment they tainted sensitive data processed by Mozilla, Apache, GNU Emacs and Linux. Then the tainted data was tracked as it propagated in hardware, system software and applications. In the analysis of their experiments they found sensitive data in plain text scattered for an indefinite period of time in many different places like circular queues, I/O buffers and strings. They also conducted experiments on windows 2000, by tainting sensitive data at the windows log on and when processed by Internet explorer 5. They

found scan code and Unicode representation of the passwords in memory. But they didn't do more on this platform due to lack of source code availability.

The authors of the famous cold boot attack paper [53] recovered the encryption keys by executing a number of experiments. This attack is based on the data remanence properties of DRAM and SRAM chips that allows the data to be retrieved from physical memory even when the power is removed from computer.The authors executed this attack by 1st resetting the machine and then booting it from a light weight operating system and the memory contents dumped to a file. 2ndly they cooled the memory modules,removed them from the original machine and immediately placed in another machine under attacker control.They were able to recover popular encryption keys. In this work the authors have exploited the hardware problem which keeps sensitive data for hours when cooled. Andre Arnes and others [20] have also worked on the extraction of cryptographic keys from windows memory. Using their own developed tool they recovered the keys for popular crypto systems. They established different system states for their experiments. Our thesis work would be extending this work by incorporating page file in search of sensitive data. Carsten Maartmann Moe in his master thesis [66] established some of the scenarios to recover chat logs and passwords and cryptographic keys from memory. Our work will be adding some more scenarios and including page file as well.

Skorobogatov [104] measured the data retention ability in different models of SRAM chips by applying two factors - power consumption and temperature. His 1st observation was that the smaller the power consumption of the chip, the longer is its data retention time at room temperature. His 2nd observation was that data retention time increases with the decrease in temperature for all the tested chips.

Peter Guttmann in his technical paper [49] has described the technical details of the semi conductor devices and the different ways that can affect the existence of data on these types of devices. These include electro migration, hot carriers, and ionic contamination. These patters can help in the recovery of over written data or data when no power is being provided to memory. He has also suggested several ways how to deal with such problems.

Many people have worked on the extraction of sensitive data from from both sources. Zhao and Cao [82] have extracted sensitive data like user names and their passwords from physical memory of computers running under windows systems. The authors have considered page file, hibernation file, crash dump files and RAM directly for their experiments. Despite that they extracted sensitive data from these sources; they have not explained the details of their experiments like version of the operating system and the duration of the sensitive data in their experiment, since time plays a vital role in keeping the data in physical memory.

Lee and others have written a series of papers on the collection and analysis of physical memory and page file.In their 1st paper [100] paper have acquired RAM and page file by using the Linux distribution from windows systems and then extracted user names and passwords from these sources by keyword search. They suggest that with the increase in RAM, the chances of finding sensitive data in page file also increases. In their other works [102] and [101] they have used their own page file collection called PCT for the extraction of page file from running windows systems. By applying their own filtering algorithm they were able to recover sensitive data including passwords from publicly accessible computers. They also claim that probability of password recovery is about 66

percent if page file size is over 768 MB. But their PCT tool is not publicly available. Andreas Schuster [95] and Walters and Petroni [18] have mentioned some techniques to find some useful information like TCP connections and open sockets in memory.

Some authors have also worked on new techniques for the extraction of sensitive data from memory. Hejazi and others [90] have proposed the methods of application finger print analysis and process stacks to extract sensitive data like user names, passwords and URL's from windows memory.Their second method needs more experiments on closed source applications.

Several authors have worked on locating processes and threads and their reconstruction in physical memory. Andreas Schuster [96] has worked on the extraction of hidden and terminated processes and threads in memory. Based on his theory he developed the PTFinder tool for visualizing processes and threads in memory. Harlan Carvey [37] has also worked in the same area by trying to trace link and unlinked objects in memory. Burdach [28] also worked on process information by following the unbroken links in memory threads. Dolan Gavitt [95] has used virtual address descriptor (VAD) tree to explain the pattern of memory allocations used by a process and files mapped into its virtual address space. But this technique does not work for the exited processes.

Baar and others [83] have worked on file mapping utilizing the concepts of Andreas schuster [96] and Dollan gavit [95]. Petroni and others [77] has also worked on address translation. They also included visualization in their frame work called FATKit.

Work has also been done on the reconstruction of processes and threads in both memory and page file. Kornblum [62] in his paper has mentioned a method of robust address translation in which he has mentioned six types of invalid pages can be located in virtual memory. This invalid pages of course also includes the pages swapped to page file. Savoldi and Gubian [92] have also worked on the same area like Andreas Schuster [96]. They used the modified version of PTFinder tool and tried to link the processes in page file with their corresponding pages in memory by considering only invalid PTE's. They have recommended to study page file and memory in different periods of time to get a more clean picture which we are going to study in this thesis.

In response to digital forensic challenge in physical memory in 2005 [43] several solutions were proposed and many tools were developed. Most of the proposed solutions worked on enumerating processes and lists in memory. Knttools [59], [109] from Garner and memparser from Betz [25] were the joint winners. Both tools are capable of extracting information about threads,access tokens and other data structures from windows memory. The Volatility Framework [110] was another tool from Volatile Systems which is capable of extracting information from windows memory including loaded DLLs, running processes and open network connections. Aratesh [22] has presented a method for partially reconstructing process execution history from a windows memory image.

Some people have also worked on the integration of physical memory with other sources like file systems data and network capture One such work is ramparser by case and others [71] and PyFlag [21].

3.4 Acquisition Methods

In this section I will discuss the tools and techniques for the acquisition of memory images. While then I will mention the tools and procedures developed for the acquisition of page file. Mostly the discussion will be about Microsoft windows systems when

it comes to operating system while I will not keep my discussion to a specific hardware unless mentioned by a tool or technique explicitly.

There are two approaches to the acquisition of memory contents; Hardware based and software based. Both approaches have their pros and cons. Following are the most commonly used and suggested hardware and software based memory acquisition methods and tools.

3.4.1 Hardware Based Memory Acquisition Methods

The main concept of hardware based approach is to access physical memory directly without disturbing the host operating system and without writing any thing to physical memory while acquiring its image. The main disadvantage of such approach is that the hardware must be installed on the machine prior to incident. In this case it can also be used by an attacker for illegal purposes. In this section I will present a summary of the mostly used hardware based approaches. The pros and cons of each approach will be presented according to the set criteria like impact on the system, post or prior installation, required privileges, acquired image format, safety features and flexibility according to changing hardware and operating systems [97].

1. Fire wire
   This interface was first introduced by Apple in early 90s. It is also known as i.link and IEEE 1394. It is a peripheral bus standard available in many desk top computers and laptops. The fire wire port is used to access the physical memory of computers directly without disturbing the host operating system. This kind of physical access is called direct memory access (DMA). The Fire wire port was first of all used by Quinn in 2002. He used his own code to manipulate an MAC system in 2002 [12] and [98]. Fire wire can be used to read and write to physical memory at much faster rates. All that is needed is a controller device and some code to get access to memory and perform the read and write operations. Adam Boileau extracted the contents of physical memory from both windows and Linux systems using his python library codes [26]. It means fire wire is quite handy. But every approach has some draw backs also. First of all fire wire port is not universally used in all computers. There have been some technical problems while acquiring image through fire wire too. In some windows systems it cause blue screen of death (BSoD). It also causes hanging of the systems. It also causes problems with a region of the memory called Upper memory Area (UMA) [48] and [115]. Direct memory access is not enabled by default in Microsoft windows by default [86]. The fire wire option can be disabled from device manager deny access through 1394 port [67].

2. Crash Dumps
   This feature is available in windows systems. It is also called blue screen of death (BSoD). Normally when there is some driver conflict or hardware problem then the system is hanged and goes into blue screen and had to be restarted after resolving the issue. During this process the windows writes the contents of physical memory into page file location. The default location is C:/pagefile.sys. The output file is written in Microsoft proprietary format which is readable only in Microsoft debugging tools [2]. The file format for the output is .DMP. Though, the crash dump file is no longer limited to Microsoft debugging tools. Andreas Schuster in his blogs [93] has shown how to read the .dmp file for analysis. The crash dump can be generated by pressing

a sequence of keyboard keys. By default this feature is disabled. In order to get this feature the value of the registry key for CrashOnCtrlScroll must also be set to 1, depending on the keyboard whether it is USB or PS/2. When all these settings are done then we can generate a memory dump file by holding down the right CTRL key and pressing the SCROLL LOCK key two times [74]. Except these changes we also need to make changes for the type of crash dump that we want to generate. There are three types of crash dumps: small, Kernel and complete crash dump. The series of Microsoft operating systems have different default options for generating dump of physical memory. According to MS Knowledge article Q254649 [11], following are the default dump file options:
Windows 2000 Professional: Small memory dump (64 KB)
Windows 2000 Server: Complete memory dump
Windows 2000 Advanced Server: Complete memory dump
Windows XP (Professional and Home Edition): Small memory dump (64 KB)
Windows Server 2003 (All Editions): Complete memory dump
Windows Vista (All Editions): Kernel memory dump
Windows Server 2008 (All Editions): Complete memory dump
This means we also need to make changes to this option because we would like complete memory dump for forensic purposes. The article Q254649 [11] also states that complete crash dumps are not available on systems with more than 2GB of memory. But according to Nicolas paper [86] and the knowledge base article Q237740 [8], this option can be enabled. But it requires additional settings in Boot.ini file. While, if we have 4 or more than 4GB of physical memory, then we need to set the PAE value also. Windows crash dump is a nice feature but it has several draw backs too.
The first one is that this option is not enabled by default and by setting this option, it requires reboot.
The complete crash dump option is not the default choice in any Microsoft operating systems except Microsoft windows server family as mentioned above. It means it has to be set prior to incident.
The original contents of page file will be lost since by generating crash dump its contents are replaced.

3. Virtualization
   Virtualization is the process of running a guest operating system inside a virtual environment. While the host operating system provides control to the program, that runs the guest operating system. Currently the popular virtual products in the market are VMware [13] and Virtual PC [10]. The virtualization products give us a snap shot of memory and page file which is an identical replica of the original data. When we suspend the virtual machine, the whole activities on the guest machine are stopped and the contents of the physical memory as ".vmem" file. This format is supported by many analysis tools While the contents of page file can be recovered any time by mounting the system drive in read only mode. The virtual products provide an excellent environment for forensic and other testing purposes. We can install any kind of program or Mal ware and see the behavior and then we can revert to previous state of the system by using the snap shot facilities. Virtualization provides a nice solution to acquire a clean copy of physical memory but they are not so commonly used by the people especially when some incidents occur and forensic investigation is required.

But still there has been a growing interest in these products in the last few years and they will become more common as the computers systems are getting more and more processing ability [15].

4. Hibernation File
   Hibernation is a feature in many computes especially it is available in laptop systems. In hibernation mode the contents of the physical memory are written to disk in compressed Microsoft proprietary format, and then the system is shut down. When the system is restarted, the boot loader checks for the hibernation file if there is one and then the contents of RAM are loaded and we see all our open programs left while hibernating the system. Hibernation provides two advantages: The start process after hibernate is faster than a fresh start. On laptops it can be useful if the battery power goes low. In order to get the hibernation file this option must be enabled in advance. The hibernation file is stored in C/: hiberfil.sys. This file is hidden by the operating system. This is not recommended to enable hibernate option after the incident because it will consume extra space on hard disk and might over Write existing data. The hibernation is very useful for forensic investigation but the compressed hibernated file format is undocumented and no details have been released by Microsoft yet. But a security researcher Matthieu Suiche [107] has converted this file into readable format. The other demerit of this approach is that hibernate option is not available in all computers. According to Microsoft, computer that has more than 4 GB of memory can not be put into hibernation in Windows XP and Windows Server 2003 [17]. Another weakness is that the hibernated file will not contain the unused space of the physical memory.

5. Tribble
   In 2004 Carrier and Grand [36] presented their idea of dedicated PCI expansion card. They also presented their proof of concept device called tribble by which memory can be captured bypassing the operating system. The main advantage of this approach is that the contents of physical memory can be captured without introducing any new data in memory. But the main problem is that the device must be installed prior to incident. In that case it can also be used by the attackers for unauthorized access to memory. The other draw back is that the device is not widely available for use and testing.

3.4.2 Software Based Memory Acquisition Methods

The main purpose of software based memory acquisition is to install a piece of code in memory that will give us control to acquire the image of the physical memory. The main requirement from such techniques is that the impact of the toll on the memory should be nominal and the format of acquired image should be supported by the analysis tools to get meaningful results.

1. Data Dumper (DD)
   Data Dumper is a Linux utility program whose purpose is to get access to physical memory through "Device/ physical memory/" object [3]. DD is considered a standard for acquiring forensic memory images as well as hard disk. The image acquired by DD is supported by most forensic analysis utilities. Garner from GMG systems [58] produced the modified version of DD for windows based systems. This version is part

of the forensic acquisition utilities which is a freely available tool for download. This tool has the ability to access physical memory from user mode. This DD provides a very flexible solution to the acquisition of memory. It has minimal impact on the system and does not require rebooting the computer. The memory image can be copied locally to USB device or sent over a network. And it also provides compression and hashing facility. But this approach has some draw backs as well. Microsoft is no longer providing access to "Device/ physical memory/" memory object from user land since windows 2003 SP1. Thus only kernel drivers are allowed to get access to that object. To cope with this problem GMG system developed a new utility called KntDD which is part of the Knt Tools discussed later. The acquisition process can last for hours and memory is constantly changing which can have a significant impact of the acquired image. The attacker can hook several places to tamper with the image [48].

2. KnTDD
   This tool was developed by GMG systems [109]. Knt DD is part of the Knt Tools kit for the extraction of physical memory. Currently this tool is available only for commercial purposes. This tool kit is able to run on Microsoft operating systems from windows 2000 to vista. It also works on 64 bit versions of the windows. The acquired image of the memory by using this tool can be saved locally, on removable USB device or through network. The raw format can also be converted to Microsoft crash dump format to be analyzed and read by Microsoft debugging tools. The main issue with this tool is that it needs to be loaded into memory while acquiring its image. And while it is reading the image the status of the memory is not frozen. Thus, this tool not only consumes space from memory but as it is traversing the memory, pages that have already been read can change [38].

3. Memory DD
   ManTech Memory DD [40] is a freely available tool for capturing memory on Windows Vista and 2003 Server and other Microsoft operating systems. According to the documentation of this tool, it acquires a forensic image of physical memory and stores it as a raw binary file from a running system. MD5 is used for data integrity and the resulting acquired image can be opened in external tools for analysis. MDD software can copy up to 4 GB of memory. MDD is also available in Helix. MDD is free tool and supports almost the full range of Microsoft windows family. It is a nice feature but I am not sure about its impact on the memory while acquiring the image.
   There are many other software based memory acquisition tools available both free and commercial. Here I am briefly naming those tools with their links for download and more information.

4. Win32 DD
   Win32dd [106] is a free kernel land tool to acquire physical memory. This tool claims to have the capability of acquiring memory images of windows family from XP to 2003. While it doesn't say any thing about supporting 64 bit operating systems.

5. Encase This is a commercial tool providing comprehensive solution to forensic acquisition and analysis of memory and hard drive [106].

6. FastDump FastDump [6] is a freely available tool for the acquisition of physical memory on windows systems. The resulting acquired image in binary format can be

easily saved to a USB drive.

7. F-Response
   F-Response [5] is a commercial software utility that enables an investigator to conduct live forensics, Data Recovery, over an IP network using the tool(s) of choice.

3.4.3 Page File Acquisition Methods

Paging is a memory management scheme used by the operating system to store and retrieve data from secondary storage for use in main memory. Paging is an important part of virtual memory implementation in most currently available operating systems, allowing them to use disk storage for data that does not fit into physical memory. When the physical memory is full and has no more space for the incoming processes then the memory management swaps out pages to an area on the hard disk and retrieve them when needed. On today's computers the swap space ranges between 1.5 to 3 times the physical memory size. The page file by default is located in C:/pagefile.sys or in C:/windows/win386.swp in windows 9x systems [101]. If the capacity of the physical memory is small then there wold be more swapping thus giving the investigators opportunity to find evidence there. But in this case the data in swap file will be replaced quickly. On the other hand, as the capacity of the physical memory increases with time the process of swapping decreases. But on the other hand this gives opportunity to reside there for longer time once it is swapped out. Following are the possible tools and techniques to acquire page file on windows systems.

1. Injecting Unsigned Drivers
   In live memory acquisition it is not possible to acquire page file data at the same time because the page file is being used by the operating system. But it is still possible to "access" this file using a specially crafted driver. Incidentally this technique has been used by Joanna to inject an unsigned driver in windows vista64 memory kernel [89].

2. PCT Tool
   The page collection tool [101] and [102] was developed by Seokhee and his co researchers to acquire page file from a system running Microsoft windows. According to their experiments they were able to copy the contents of 1 GB page file from a running system to an external USB storage in 3 to 4 minutes. But their tool is not available for public use until now.

3. Using Virtual Environment
   As we talked about the virtual products in the hardware acquisition methods this approach can be used to acquire the contents of page file by freezing the virtual machine at any instance. Thus page file can be collected at the same time with the image of physical memory. This gives a more reliable method because there would be almost no changes to the contents of the page file thus maintaining the integrity of the acquired images.

There are also some commercial tools available for acquiring the page file of running system. All these tools must be installed on the system unless they are already available on the system. These are: Disk Explorer [4] Forensic Tool kit [7], X-Ways Forensics [16] and iLook [9]

3.5 Tools Used for the Analysis of Physical Memory and Page File

Volatile data forensics is an emerging field and alot of tools are being developed recently. The focus of this chapter is to develop a better understanding of some of the more commonly used tools for the analysis of memory and page file. The first section provides an overview of the tools. The most important features are identified and described. The purpose of this section is to provide information necessary for developers to improve tools, users to make informed choices, and the legal community and others to understand the capabilities of the available tools and softwares. All the mentioned tools in this section will be discussed by covering the background information about the tool and its functional overview. We have categorized the tools into commercially and freely available tools.

There are several organization who are working on the developments of standards and practices for computer forensic tools. One such organization is National Institute of Standards and Technology (NIST). This organizations have mainly focused on developing standards for disk imaging, hard disk write blocking and deleted file recovery. Recently mobile phone forensics has been added to this list [79]. But the standards for volatile data forensics are still in its infancy stage.

3.5.1 Commercial Tools

1. KnTList

   - Background

   KnTList tool was developed by GMG systems [109]. The KnTList tool are currently available to the military, civilian law enforcement and other civilian governmental agencies, and higher educational institutions. This tool is part of the Knt tools package. The author of this tool was the winner of memory challeng in 2005 [59]. This tool kit came into existence when Microsoft prohibited access to memory from user land through "Device/ physical memory " object in windows 2003 SP1. KnTList are command line tools for the and analysis of physical memory.

   - Capabilities

   The knTlist has both the acquisition and examination capability of physical memory from a live Windows system. KnTList can reconstruct meta data, virtual address space and other processes. The output of the tool is both in text and XML format.

2. Responder Professional

   - Background

   Responder professional is a commercial tool developed by HBGARRY [50]. This is a GUI tool supporting memory images from Windows opearting systems.

- Capabilities

According to the specification of the tool it has the following functionalities:
Operating System Information, Application information, Binary Analysis, Malware Detection and Network connections and listening ports.

3. WinHex

- Background

WinHex is in its core a universal hexadecimal editor from X-Ways Software Technology [111]. It is a commercial tool but the evaluation versions has enough features to examine memory and disk images from most Windows systems. Though it has limitations under windows Vista.

- Capabilities

WinHex can be used for key word search on memory and disk images. It can also be used to inspect and edit all kinds of files, recover deleted files or lost data from hard drives.

4. Forensic ToolKit

- Background

Forensic ToolKit (FTK) was developed by AccessData Corporation [39]. This is a well known commercial tool for disk and memory analysis on Windows computers.

- Capabilities

FTK is composed of several components. Each component can be installed independently. The components include FTK Imager, the Registry, Viewer and the Known File Filter (KFF). The FTK Imager is used for imaging and acquisition. It has also the capability of previewing and searching the evidence. This component is freely available in HELIX distribution [44]. The Registery Viewer component is used for the analysis of registry files. This includes searching for information in a registry file and accessing areas containing passwords and other information.The KFF component can compare file hashes against a database of hashes from known files. Here we will mention the capabilities of FTK relating to memory analysis. The FTK can enumerate all running processes, including those hidden by rootkits and their associated DLLs. The most powerful feature of FTK is its powerful index and live search. The live search involves an item-by-item comparison with search terms specified by the investigator, while the indexed search involves the use of a powerful search engine. This search method creates a full index of the data and greatly speeds up keyword searches. This feature was very helpful in the analysis part of our experiments in this report. FTK also supports email and graphics searching facility. The email feature help investigator to

view email messages from many mail clients like outlook express, Microsoft Internet Mail (EML), Earth link, Thunderbird and Quick mail. The graphics facility can display thumb nails including deleted and undeleted graphical images.

3.5.2 Free Tools

1. Volatility

   - Background

   Volatility is a collection of open source tools implemented in python for the analysis of memory. This tool was developed by Volatile systems [110]. The the tool can be run on any opearting system where python is already installed. This tool support Linux, Cygwin and Windows (32 bits) opearting systems.

   - Capabilities

   According to the specifications mentioned by the Volatility it can extract the following types of information from memory images:
   Image date and time, Running processes, Open network connections, DLLs loaded for each process, Open files for each process, Open registry handles for each process, A process addressable memory, OS kernel modules, Mapping physical offsets to virtual addresses, Virtual Address Descriptor information Extracting executables from memory samples, Transparently supports a variety of sample formats (ie, Crash dump, Hibernation, DD).

2. Memoryze

   - Background

   Memoryze is free memory forensic software from MANDIANT [68] that can do the analysis of memory on Windows systems (32 bit) both in live state and its images in off line mode just like disk images. On live systems Memoryze can include the paging file in its analysis as well.

   - Capabilities

   According to the specifications from MANDIANT, memoryze has the following functionalities:
   Enumerating all running processes (including those hidden by rootkits). Listing the virtual address space of a given process. Displaying all strings in memory on a per process basis Identifying all loaded kernel modules by walking a linked list

3. PTFinder and PTFinderFE
   These are source free tools developed by Andreas Schuster [94] for the analysis of memory images from Windows systems. PTFinder (short for "Process and Thread" Finder) is used for the analysis. It accepts many forms of dumps. They are primarily used to list processes (and pool tags) from within Windows memory dumps. It outputs the processes in text or XML compatible formats. PTFinderFE is simply a front end for PTFinder, providing GUI and on the-fly generation of complete process visualizations.

4. The Sleuth Kit and Autopsy

- Background

The Sleuth Kit and its browser-based GUI Autopsy is a collection of source free tools developed by Brian Carrier [32] for performing a disk-based investigation. The tool is cross-platform both in operation and execution, and can be run from a CD-ROM or flash drive for live and dead analysis of memory images.

- Capabilities

This tool kit features searching for deleted files, time line creation, file system and meta data analysis, hash database search for identification of malicious software and thumbnail listings for easy inspection of evidence files. Autopsy provides a handy front end to the underlying powerful analysis abilities of Sleuth Kit.

4 Methodology

This chapter presents methodology for the experimental work of the thesis that shows practically the importance of physical memory and page file forensics in a digital crime incident. The environment of the experiment and the scenarios are explained in detail. The forensic process model and other standards and practices discussed in chapter 2 will be followed during this process.

4.1 Virtual Setup for the Experiments

We had the option to execute our experiments in controlled environment or to execute them without any constraints. We decided to use virtual environment. VMware version 6.0.4 was chosen for this purpose. Though such environment will not show the actual incidents but there were many reasons for using virtual environment. The 1st main reason was that we did not wanted the acquisition methods to put any constraints on our experiments. We were able to get a clean and unchanged state of both memory and page file at the same time. VMware Fusion has the ability to suspend (write memory to disk) and take snapshots of Virtual Machines and supporting many different operating systems. There are many other advantages of virtual environment. They are detailed in memory acquisition section of chapter 3.

The Virtualized environment works on the concept of host and guest machines. The terms host and guest describe physical and virtual machines:

Host: The physical computer on which Virtual software is installed is called the host computer, and its operating system is the host operating system

Guest: The operating system running inside a virtual machine is called a guest operating system

We used Windows XP professional (32 bits) for the experimental work. Windows XP is the dominant operating system in consumers today and as of September 2009, Windows xp share 71.79% of market share in personal computers [1]. We used a single laptop computer as the test environment for the execution of the experiments with the following details:

Manufacturer: Acer
Model: Aspire 5720Z
Processor: Intel(R) Pentium(R) Dual CPU T2310 @ 1.46 GHz.1.47 Ghz
Physical memory: 2038 MB
Operating System Installed: Windows Vista Home Premium
System Type: 32 bit operating system

4.2 Different States of the Machine

When a digital crime incident occurs, the investigator will find the computer in a certain state. Keeping this scenario in mind we have defined several states of the system. This is not an exhaustive list of all the available states but still it gives enough scenarios to cover a given digital crime. Mostly these system states have been defined by Andre Årnes and his co researchers [20] in their paper. We will be using some of these system

states in addition to some more. In this section we are going to define those possible states that will become the basis for our experiments. Since we would be conducting our experiments on a laptop system, some of the states might not be available on the common desktop computers.

4.2.1 Ready State

This is also called live state. In ready state the computer is fully powered up and the operating system and application are running. In our experiments, in ready state the machine will be powered up, the user logged on and the chat tools and other application softwares running.

4.2.2 Screen Saver State

A screen saver is a picture or animation that covers the screen and appears when the machine is idle for a set period of time. In our experiments we will set this time to 10 minutes. In our case the screen saver will be password protected and the virtual machine will be suspended after the screen saver is activated.

4.2.3 Log off State

Log off is a state of computer where all the running programs are closed but the computer is not turned off. This means the computer can be used by the same user or another user without restarting the computer. In our case all open programs, chat clients and browsers will be closed, the user logged out from computer but still running.

4.2.4 Reboot State

Reboot is a state of computer where all currently open programs are closed and the machine is restarted. When the machine reboots, depending on the machine setting it can go into several sub states like asking for user name and password or waiting for a hardware interrupt like pressing a key from keyboard. The reboot state has further two sub states.

1. Hard Reboot
   This reboot is also called cold reboot. In hard reboot the machine is powered down and then powered up without any clean shut down procedure. But this reboot leaves the system in unclean state and an automatic scan of the file system is started upon reboot. Hard reboot may be caused by sudden power failure, accidently or be done deliberately. When done deliberately, it can be initiated by pressing the reset button, by pushing the computer's power button and holding it until the computer shuts down and then restarting the computer by pushing the power button again or by pulling the power cable and starting the machine again. Most modern desktop machines and laptops have no reset button and hard reboot is accomplished by the power button or by pulling the power cable when there is direct power supply to the machine. This technique is specially used by intruders to access cryptographic keys and passwords from RAM, which is called a cold boot attack.
2. Soft Reboot
   This reboot is also called warm reboot. Soft reboot is under the control of operating system. Unlike hard reboot this is a controlled reboot of the machine without turning off the machine's power. Soft reboot is executed by accessing the restart command through start menu in Microsoft operating systems. Both of the sub states of reboot

will be tested by thus establishing further sub cases.

4.2.5 Sleep Mode

Sleep is a power-saving state of the computer where power to the machine hard drive, screen and other peripheral devices is disconnected except physical memory. Thus all open programs are saved in the physical memory. When the power button is pressed the computer resumes its state where it was put to sleep. In some machines sleep mode is also know as standby. It is important to provide continuous power to the physical memory to keep the currently running programs active. The movement power is disconnected to physical memory, all data is lost. To avoid such situation many desktop computers have a new feature called hybrid sleep. In this mode the operating system automatically saves the currently running applications both to memory and to hard disk. This in case of a sudden power failure, operating system can restore the status of physical memory from hard disk [103] and [73].

4.2.6 Hibernate Mode

In Hibernate an image of the currently open programs in physical memory are saved to the hard drive. Then the computer is powered down just like turning it off. All this is done by the operating system once the sleep button is triggered. Next time when ever the computer is started, it returns to the last state it was in before the computer was put into hibernation mode [14].

4.3 Scenarios of the Experiment

In this section we describe the scenarios for the experiments. Detailed explanation of the scenarios will be discussed in chapter 5. This will be the prior information available to the investigator. In real investigation such information is not available to the forensic examiners. Here we briefly define the main components involved in the experiments. The information given in these scenarios will be the success criteria for our analysis work.

4.3.1 Users and Applications

Figure 8 is showing the test users and applications that will be used in experimental work of this thesis.

Opearting System	Windows XP professional SP3 (32 bits)	
Web browsers	1. Internet Explorer Version8 2. Opera version 9.6 3. Google Chrome 4. Fire Fox	
Chat clients	1. Windows Live messenger (2009) 2. Yahoo Messenger Version 9.0 3. Google talk	
Users on (Recieving side)	**User Name**	**Password**
	forensictest3@yahoo.com	18NPynoi07
	forensictest3@gmail.com	18NPynoi07
	forensictest3@hotmail.com	18NPynoi07
Users on (Sender side)	**User Name**	**Password**
	forensictest2@yahoo.com	18NPynoi07
	forensictest2@yahoo.com	18NPynoi07
	forensictest2@yahoo.com	18NPynoi07

Figure 8: Test Users and Applications

4.4 General Procedure for the Experiments

The following step by step procedure was used to generate different scenarios according to the system states mentioned in section 4.2.

1. The host machine was formatted and a fresh copy of windows Vista Home premium basic (32 bits) was installed on drive C.
2. VMware workstation version 6.5 was installed on this host machine to be the virtual environment for the guest operating system.
3. Windows XP SP3 was installed in VMware with all security patches and required drivers for the hardware devices. This is the guest operating system.
4. The necessary web browsers and chat softwares were installed inside guest machine.
5. A snapshot was taken and saved to an external drive. This snapshot will be the basis for the analysis on each state of the machine described in section 4.2.
6. Normal operation of the guest operating system was started. Chat sessions were established and email accounts were accessed by using the information given in section 4.3. Some other applications were opened to create a normal simulated environment. The usage of the physical memory was monitored by the Windows Task Manager and Process Explorer tools [87] to ensure the usage of available physical memory and page file.
7. A snapshot was taken and the resulting memory and page file images were seized and hashed using SHA-256 and MD5.
8. The duplicate copies of the images acquired in step 5 were analyzed using the selected

tools. After analysis the images were again hashed using SHA-1 and compared with the hashes computed in step 5. This ensures the integrity of the images and follows the rule of chain of custody.

9. The guest machine was reverted to the snapshot created in step 5 and all the system states explained in section 4.2 were utilized and the analysis was repeated from step 6 to 7.

4.5 Tools used in the Experiments

As mentioned earlier in chapter 1, we have chosen to utilize easily and freely available software tools for the experimental part of this thesis. This section will provide a description of these tools and their usage in the experiments. More details are given in chapter 3 about the tools and techniques for the acquisition and analysis of the physical memory and page file.

4.5.1 Tools Used for the Acquisition of Evidence

We used the VMware for the acquisition of both physical memory and page file. The snapshot feature was used to get a clean and undisturbed state of memory and page file. This technique was used for all the experiments. The acquired images of both sources were cryptographically hashed and saved in read only mode to an external USB drive for analysis. The calculated hash values were saved in a safe place. This will keep the integrity of acquired images and follow the chain of custody. "FTK Imager" and the "dd" utility were used for this process.

4.5.2 Tools Used for the Analysis of Acquired Data

We used three tools for the extraction of sensitive data from both sources.

1. Forensic ToolKit (FTK) :
   This is a commercial tool but it allows the analysis of 5000 files in demo version. Since in our experiments the total number of files in both sources (memory and page file) were less than 50, so it was useful to do analysis of data with this tool. The features of this tool are detailed in chapter 3.
2. Winhex:
   This is a free tool allowing both the acquisition and analysis of memory images. In our experiments it was used for key word and string searches of emails, passwords and chat logs.
3. Access Data FTK Imager:
   This tool was also used in the analysis part. It has the ability of both imaging and key word search.

5 Experiments Results

This chapter presents the findings encountered during our experiments. The results of all scenarios are presented in the form of screen shots that were obtained during the analysis phase. During this phase the standard methodology of a forensics process discussed in chapter 1 will be applied to all cases. It is important to document the results of the experiments. The cryptographic checksums of all the acquired memory images can be found in the appendix section of this report. Since during searching and analysis phase of the experiments we got a large number of hits for the user IDs, therefore here we will show the detailed screen shots only from "Ready State" ,while the results for the rest of the states are summarized in tables and will be discussed briefly in their respective sections.

5.1 Ready State

To establish the scenario of ready state the virtual machine was configured, a fresh copy of Windows XP professional installed with all the security patches and necessary updates. Then we installed all the mentioned web browsers (Internet Explorer Version 8,Opera version 9.6, Google Chrome and Fire Fox) and chat clients (Windows Live messenger 2009, Yahoo Messenger Version 9.0 and Google talk) on the guest machine. At this stage a snapshot was taken and saved. This snap shot was the basis for the all the states of the machine. When one the experiments with one state are over, we will revert the machine to this state to start with the next state.

Next we created user accounts with these chat clients. Two types of user accounts were created.

1. Receivers
   These test users were used to do instant messaging from the observed computer. At this stage for these users we put two conditions on chat clients Yahoo Messenger and Windows Messenger. We affirmed the questions about remembering user names and passwords. And the conversations were not saved locally. During normal usage of the computer the "Receiver" users also accessed their mail box. All the mentioned browsers were used for this purpose. This gave us an idea about the security features of those browsers. At this stage we also asked the browsers to remember the user names and passwords when the users accessed their email accounts. These conditions will apply to all the states of the system in the esperiments.
2. Senders
   These test users were used to do instant messaging with the "Receivers". from an independent computer. The conversation on their side will be saved so that we can compare them with conversations recovered during analysis phase from the "Receivers" side.

After this normal usage of the computer was initiated. We kept observing the usage of physical memory with Windows Task Manager built in facility and Process Explorer

tools [87] to ensure the usage of available physical memory and page file. After making sure that physical memory has been sufficiently used, all chat applications are running, conversations are in progress and the corresponding email accounts in the web browsers are open, at this point a snapshot was taken by using the VMWare built in facility. Then by booting FTK Imager from HELIX package through the host opearting system, we imaged the snapshots of both memory and page file to an external USB device (Figure 9). To keep the integrity of the acquired images we loaded them in read only mode. FTK Imager has the advantage of calculating the checksums during imaging process. Both SHA-1 and MD5 hashes were obtained during this process and saved to continue the chain of custody. Figure 8 below is showing the process of imaging the memory for the Ready state of physical memory. The acquired images were saved in an external machine. Their copies will be used in the analysis phase of our investigation.

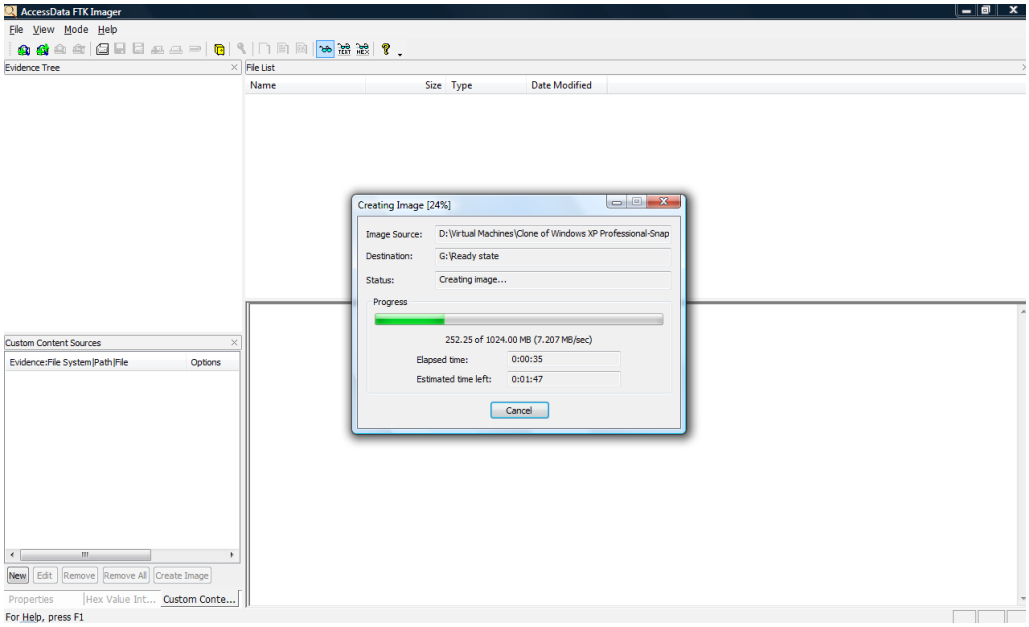

Figure 9: Imaging process of physical Memory by FTK Imager

After securing the images we proceeded to the evidence searching phase. Since as already described in chapter 4 we already know what to look for. Our success criteria was to look for the email IDs and passwords of the users in both memory and page file images. for the evidence searching phase we used Winhex, FTK Imager and Forensic ToolKit(FTK). We selected thses tools since all of them have "Keyword" features. We 1st loaded a copy of the acquired memory image was in FTK. It was not an issue for us to use the evaluation version of this tool which allows the analysis of maximum 5000 files at a time. While our acquired memory images had 41 files and having the size of 1 GB in every case. The same was the case with the images of page file. The number of files were varied but the size was 1.5 GB in every case which is the default size allocated by the Windows. Next we launched key work searches on the loaded memory image. At this stage we faced some problems regarding user IDs. Since our designated user IDs for both "Receivers" and "Senders"' were different from each others but they were the same for all chat clients. This fact can be seen in chapter 4 in section 4.3.1. This made our searching

process tedious and time consuming. In "Ready state" when we searched for the keyword "forensictest3" which is the receiver ID in communications, we got 1676 hits (shown in figure 10). This number is the combination of 5 types of IDs. These are:

1. forensictest3@hotmail.com
   This is the user ID for the email account in hotmail. Since we logged on the user email inbox in all web browsers at the same time.
2. forensictest3@hotmail.com
   Again this is the same user ID but this time it was used by the user to log on to Windows Live messenger.
3. forensictest3@yahoo.com
   This is the user ID for the email account in Yahoo. Since we logged on the user email inbox in all web browsers at the same time.
4. forensictest3@yahoo.com
   Again this is the same user ID but this time it was used by the user to log on to Yahoo Messenger.
5. forensictest3@gmail.com
   This is the user ID for the email account in Gmail. Since we logged on the user email inbox in all web browsers at the same time. We did not install a separate chat client of Gmail since the user established chat sessions from the Ggmail inbox.

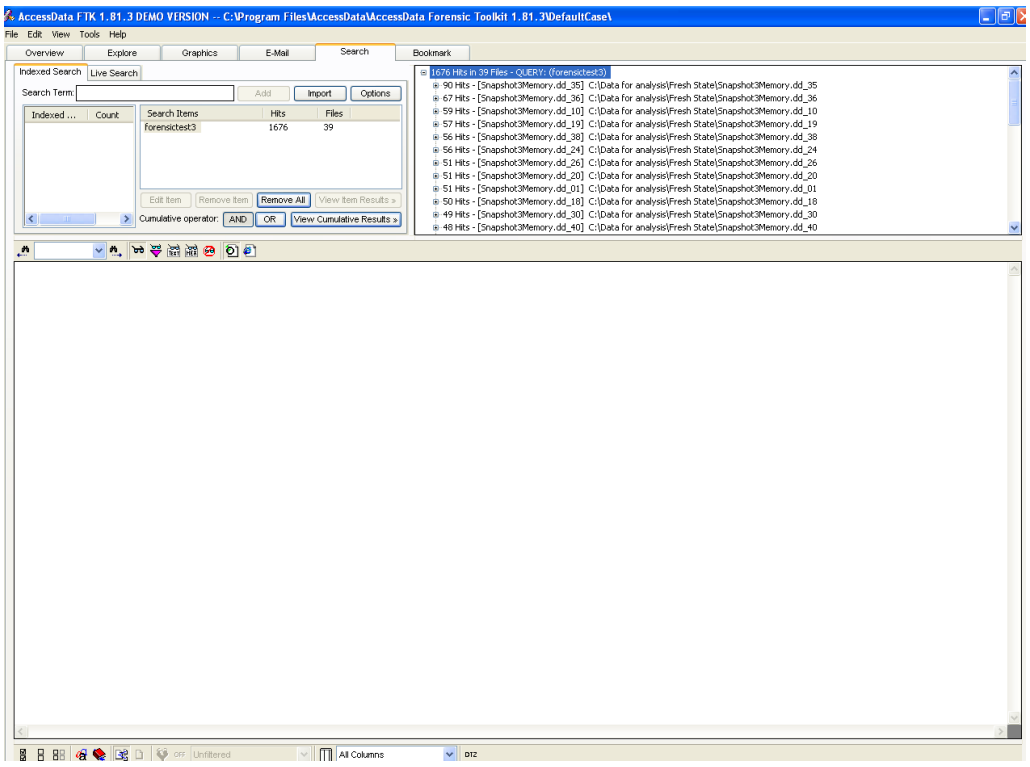

Figure 10: User IDs found by the keyword "forensictest3"

Now to separate these IDs and associate them with the corresponding chat or email

accounts we had to scan through the whole memory image by using different keyword combinations or doing manual search. To make our task easier we decided not to separate chat and email account into two different categories. For example the we decided to put the user IDs for hotmail email and chat client (Windows Live messenger) into one group. The same rule was applied to the user account in Yahoo and Yahoo Messenger. We made this categorization since it was not possible for us to link the user IDs to a particular browser application. We found it on very rare occasions that a user ID was found with the name of a browser or some other keywords of that browser. Therefore we decided to put a user email and chat ID into one category. This categorization was applied to all states of the system. The same principle was applied to the page file images of all states.

Now that we narrowed our searching criteria for user IDs, it was easier to associate the user IDs with their applications (Hotmail, Yahoo and Gmail). Using many different combinations of keywords and manual searching we were able to link the 1676 IDs found to the corresponding applications. For example when we searched for the keyword "forensictest3@yahoo.com" in FTK, we found only 114 hits. This is shown in Figure 11 below. While the remaining hits for this ID were traced by using other combinations of keywords like "YMSG" and by manual searching. In the same way when we searched for the keyword "forensictest3@hotmail.com", we found 504 hits (Figure 12) while the rest of the IDs associated with this account were found by manual search. While for Gmail account, the keyword "forensictest3@gmail.com" produced 249 hits (Figure 13) while the rest of the IDs associated with this ID were traced by other combinations of keywords, manual search and using the other two analysis tools i.e FTK Imager and WinHex. All these findings have been summarized in figure 18.

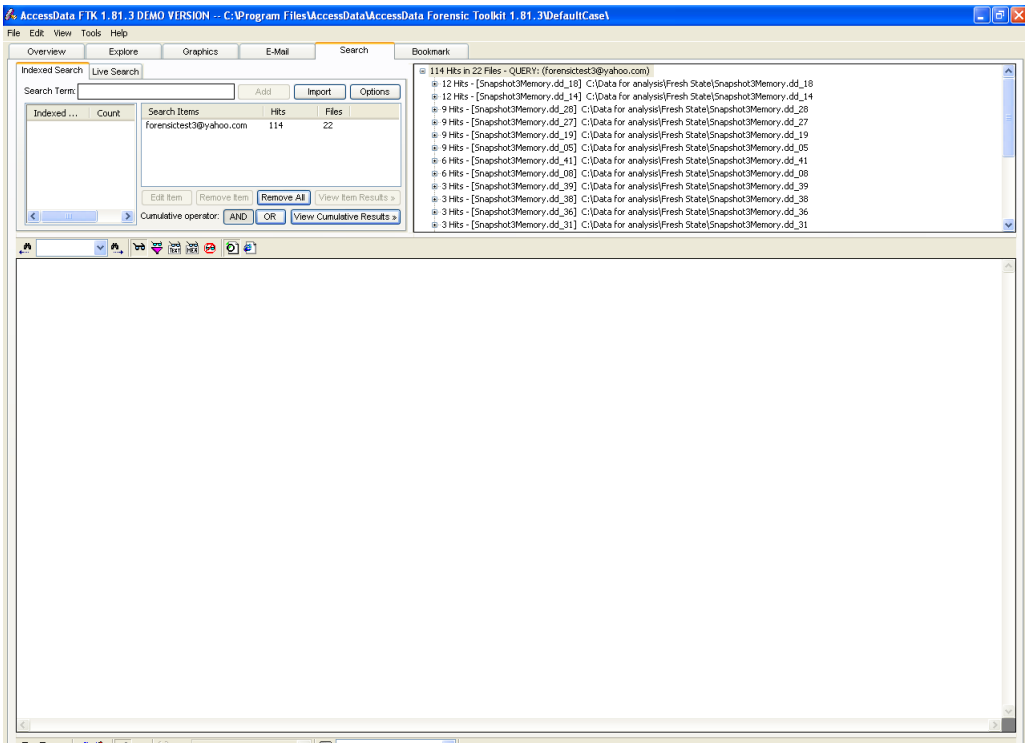

Figure 11: Hits found for forensictest3@yahoo.com

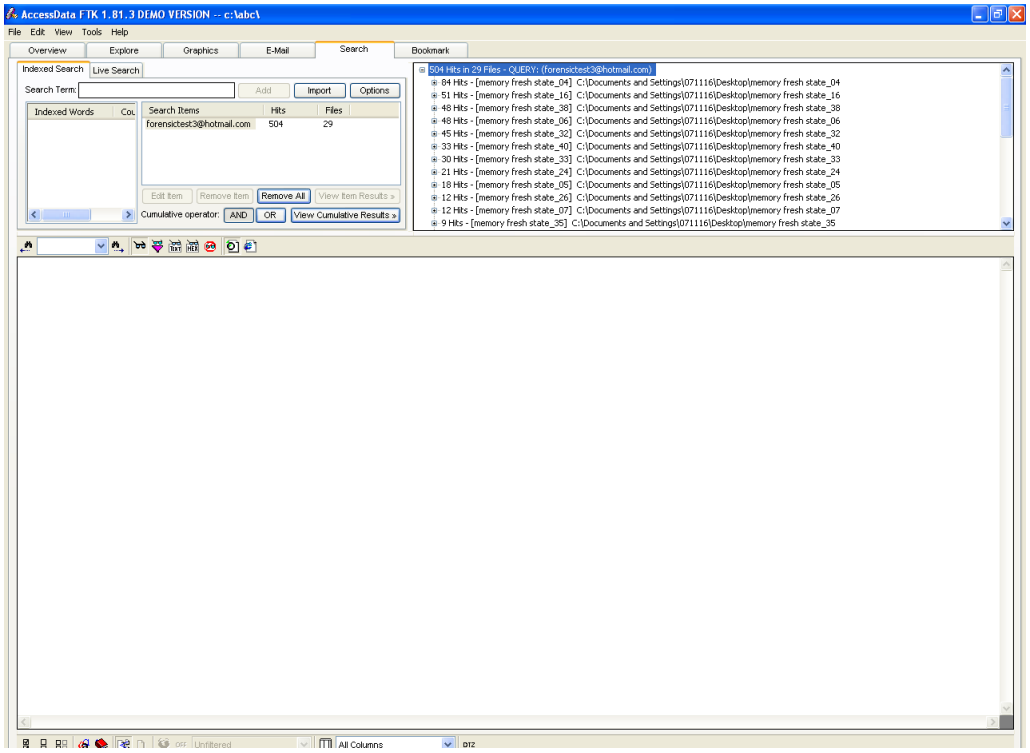

Figure 12: Hits found for forensictest3@hotmail.com

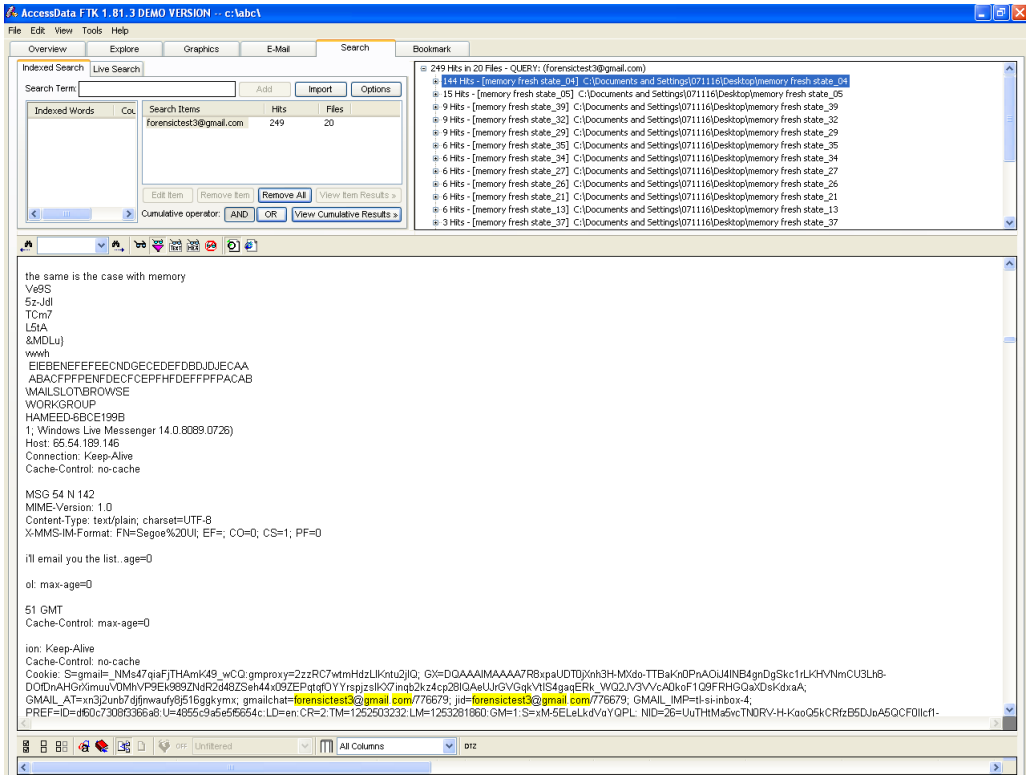

Figure 13: Hits found for forensictest3@gmail.com

Next we searched for the chat traces of "Receivers" and "Senders" IDs and we found the chat traces in many places in memory images. Figure 14 and 15 are showing one some chat conversation fragments.

Next we performed a search for the passwords of the "Receiver" IDs. We were able to trace two passwords. One is from the user ID in Yahoo and the other in Gmail. Figures 16 and 17 are showing the results.

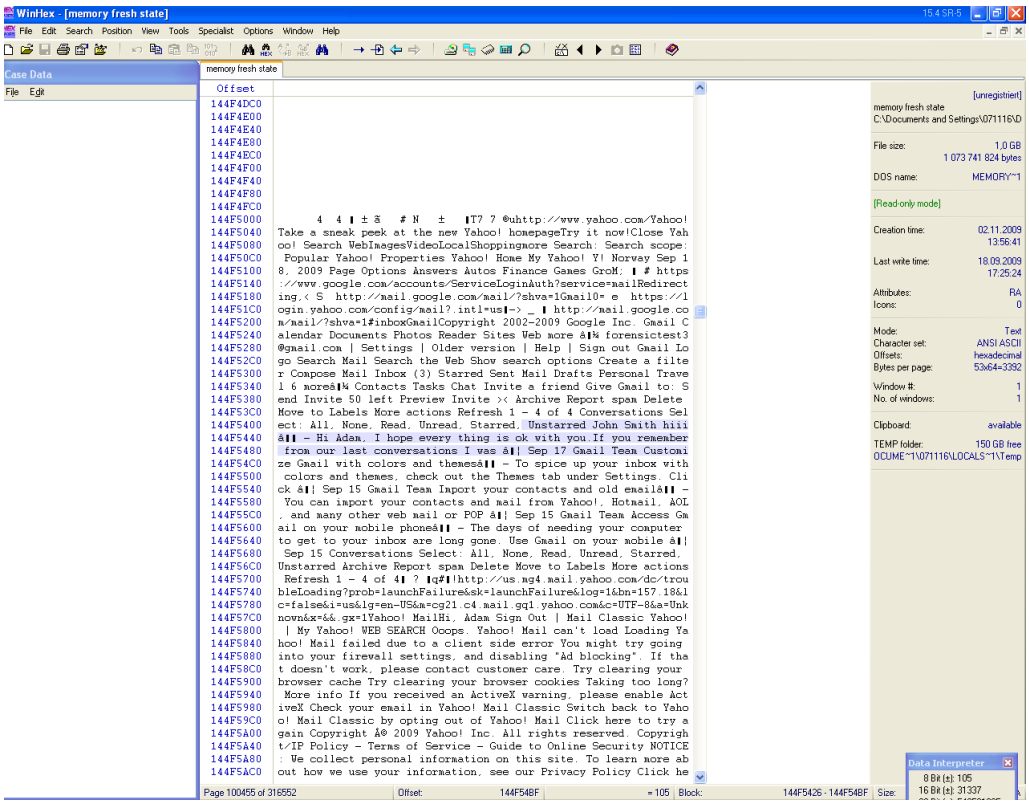

Figure 14: Chat Fragments

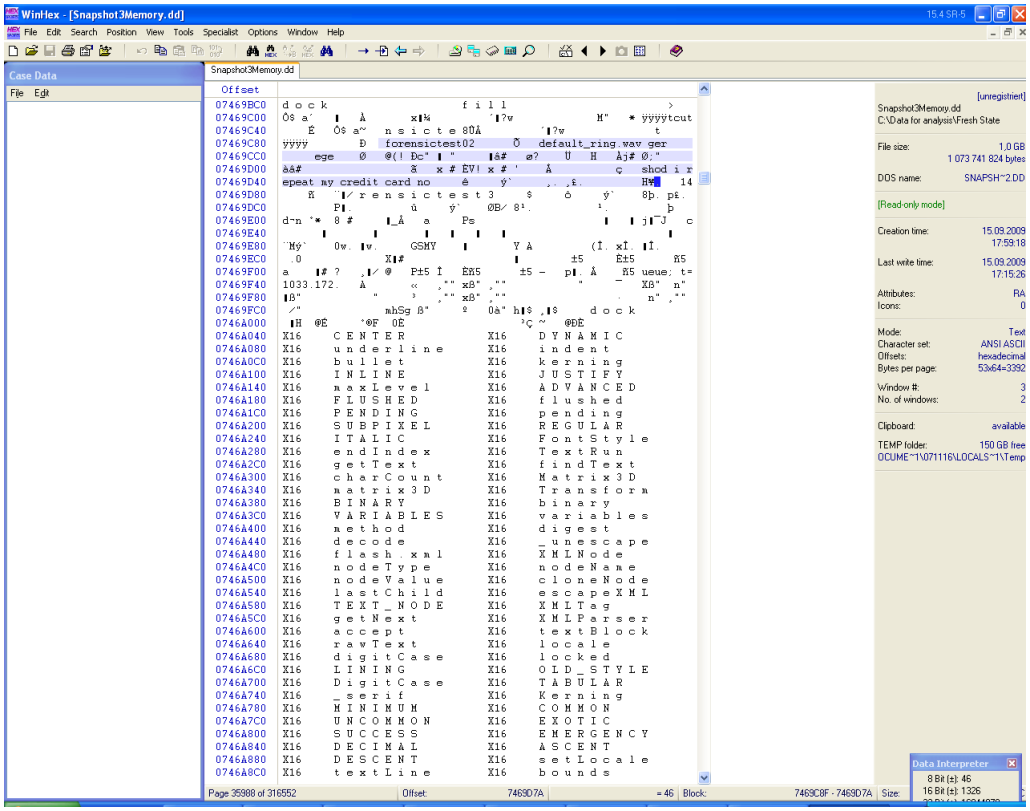

Figure 15: Chat Fragments

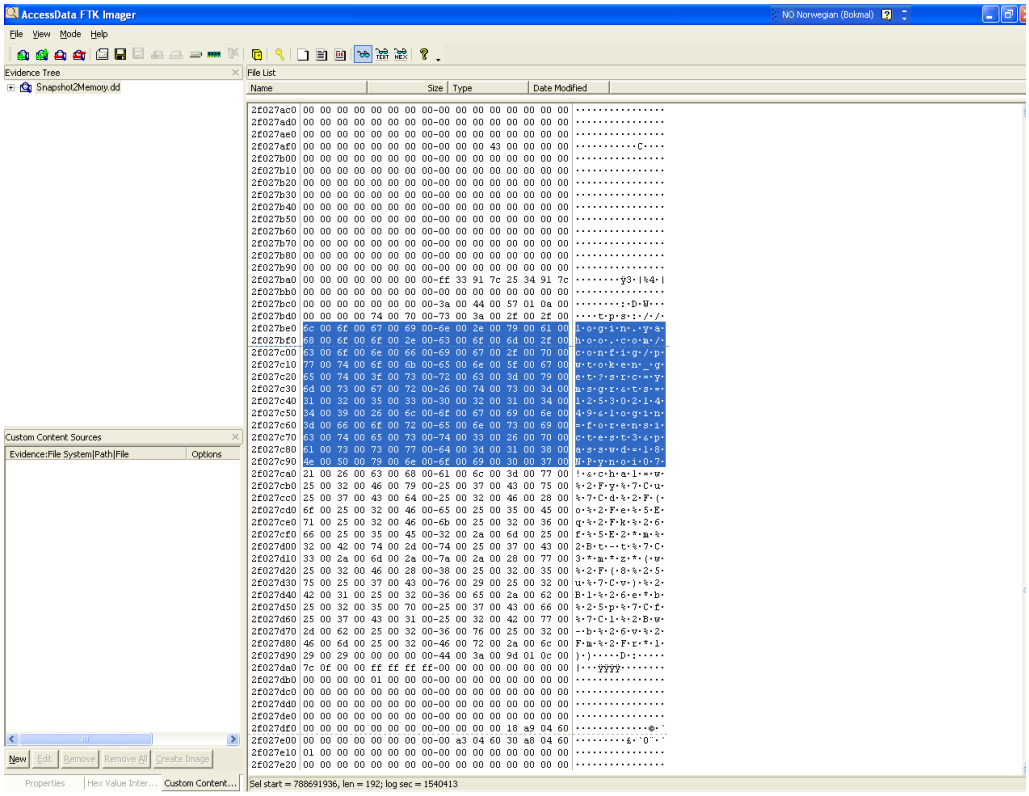

Figure 16: Receiver password on Yahoo from Ready State

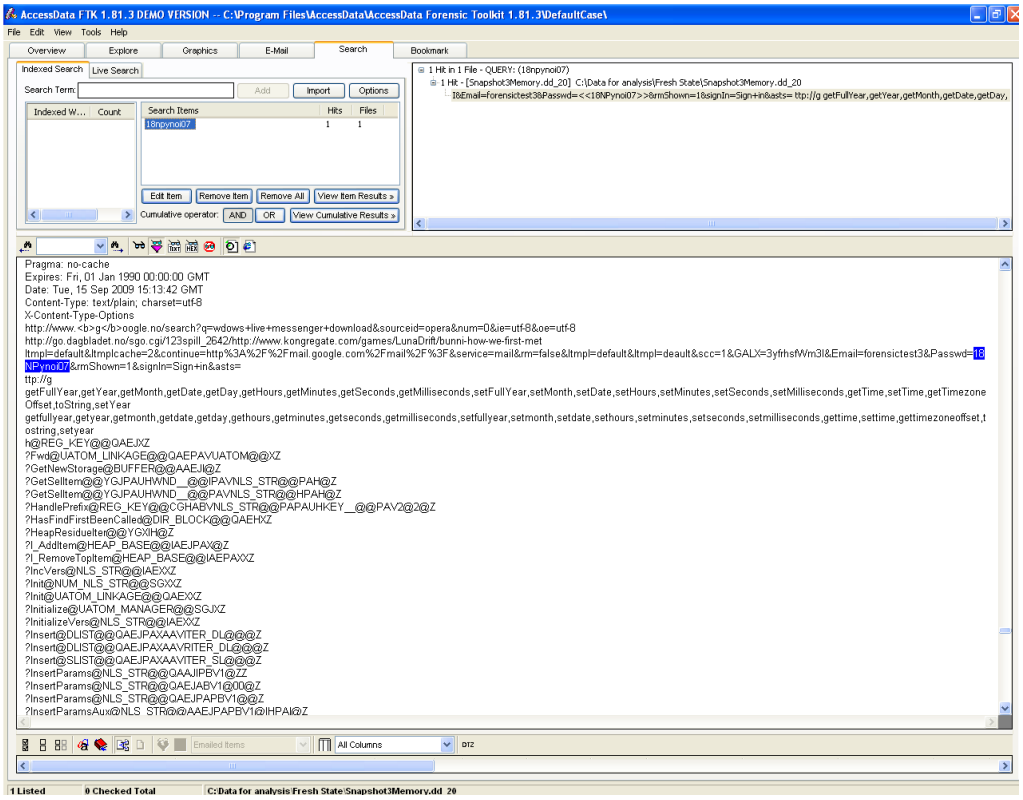

Figure 17: Reciever password on gmail from Ready State

We repeated the same procedure of searching and analysis for the page file image of the "Ready State". The page file was acquired from the test machine by using FTK Imager. The MD5 and SHA-1 were calculated during acquisition to keep the integrity of the evidinec. Then a copy of the original evidence page file image was loaded into FTK and search for the predefined keywords was commenced.

The findings of "Ready State" are summarized in figure 18. From the summary we can see that for the "Receiver" category the most occurring ID is hotmail and MSN with the total hits of 686 followed by Gmail with 564 and then Yahoo with 426. The next column shows the number of passwords found for each ID. In this state we traced only 2 passwords one for Gmail and the other for Yahoo account. The next column is showing the chat conversations traced for each account. Since there were numerous traces of chat conversations identified, therefor it was not possible to count them. Some times we found a single word of conversation and some times a single sentence. Therefore we put just "yes" since we were not able to count the no of chat conversations traced in the memory image. The next column is showing the the IDs and chat conversations found for the "Sender". In this case Gmail was the most occurring ID followed by Hotmail and then Yahoo.

The lower portion of the figure is showing the results summary for the user IDs and passwords recovered from the page file image of the "Ready State". In this case for the "Receiver" ID again Hotmail was the most found ID, followed by Yahoo this time and then Gmail. Two passwords were found; one for Hotmail account and the other for Yahoo. Like

the memory part of this state numerous chat fragments were found in this file image also. While for "Sender" ID this time Gmail was the most occurring ID followed by Hotmail and then yahoo.

Source : Physical Memory

Status: Ready State

Chat Application	Sensitive Data				
	Reciever Data			Sender Data	
	No of User e-mail IDs	No of Paswords	Chat History	No of User E-mail IDs	Chat History
MSN	686	No Hits	Yes	554	Yes
Gmail	564	1 Hit	Yes	1572	Yes
Yahoo	426	1 Hit	yes	459	Yes
Total	1676	2	Yes	2585	yes

Source: Page File

Status: Ready State

Chat Application	Sensitive Data				
	Reciever Data			Sender Data	
	No of User e-mail IDs	No of Paswords	Chat History	No of User E-mail IDs	Chat History
MSN	129	1 Hit	Yes	161	Yes
Gmail	39	No Hits	Yes	2011	Yes
Yahoo	121	1 Hit	Yes	26	Yes
Total	297	2	yes	2198	yes

Figure 18: Ready State

5.2 Screen Saver State

We did not kept track of the user IDs recovered from the remaining states of the experiments. Since we got many hits during the analysis but some passwords were saved and their screen shots are added in their respective sections.

To simulate a real environment for the screen saver state we reverted the test machine to the clean state as mentioned as mentioned in chapter 4 and at the beginning of this sections. Then normal usage of the test machine was commenced. When the machine had been used sufficiently we activated the screen saver. In this case we left the machine untouched for 10 minutes for the screen saver to be activated. During this all the chat clients and email accounts were left open. when the screen saver was activated, a snapshot was taken and the images of memory and page file were saved to external USB for further analysis. Figure 19 below is showing the summarized results of the screen saver state.

Source : Physical Memory

Status: Screen Saver State (Activated after 10 Minutes)

Chat Application	Sensitive Data				
	Reciever Data			Sender Data	
	No of User e-mail IDs	No of Paswords	Chat History	No of User E-mail IDs	Chat History
MSN	915	2 Hits	Yes	563	Yes
Gmail	205	No Hits	Yes	5370	Yes
Yahoo	318	1 Hit	yes	211	Yes
Total	1438	3 Hits	Yes	6144	yes

Source: Page File

Status: Screen Saver State (Activated after 10 Minutes)

Chat Application	Sensitive Data				
	Reciever Data			Sender Data	
	No of User e-mail IDs	No of Paswords	Chat History	No of User E-mail IDs	Chat History
MSN	291	No Hits	Yes	246	Yes
Gmail	139	1 Hit	Yes	498	Yes
Yahoo	203	1 Hit	Yes	126	Yes
Total	633	2 Hits	yes	870	yes

Figure 19: Screen Saver State

From the figure we can see that though the machine was left unattended for 10 minutes but still memory kept the User IDs of Receivers and Senders alive. Three passwords were recovered from the memory image of screen state. Figure 20 is showing the recovered yahoo password from the memory.

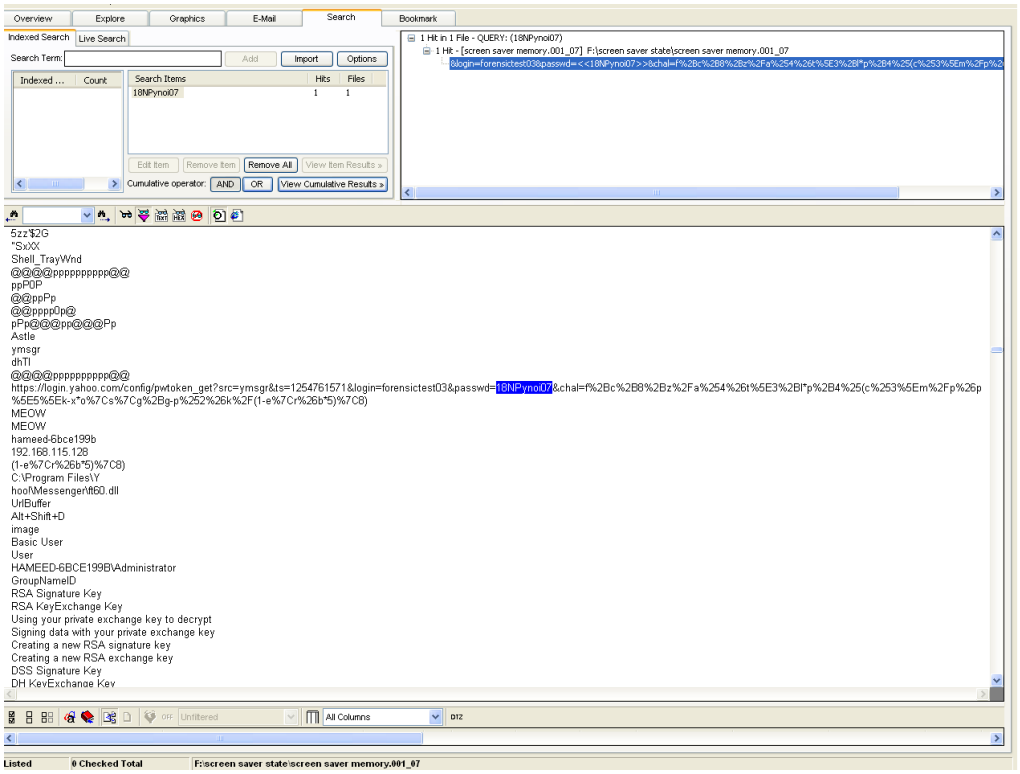

Figure 20: Yahoo Password from memory image of Screen Saver state

5.3 Standby State

Standby state is also called sleep state. Normal usage of the machine was started from the clean state. Then the test machine was put to sleep after 20 minutes of no activity. Again we were able to get manny user IDs and their corresponding passwords. Figures 21 and 22 are showing the passwords recovered from memory of the standby state. While figure 23 is showing a summary of user IDs and passwords recovered from the memory and page file of this state.

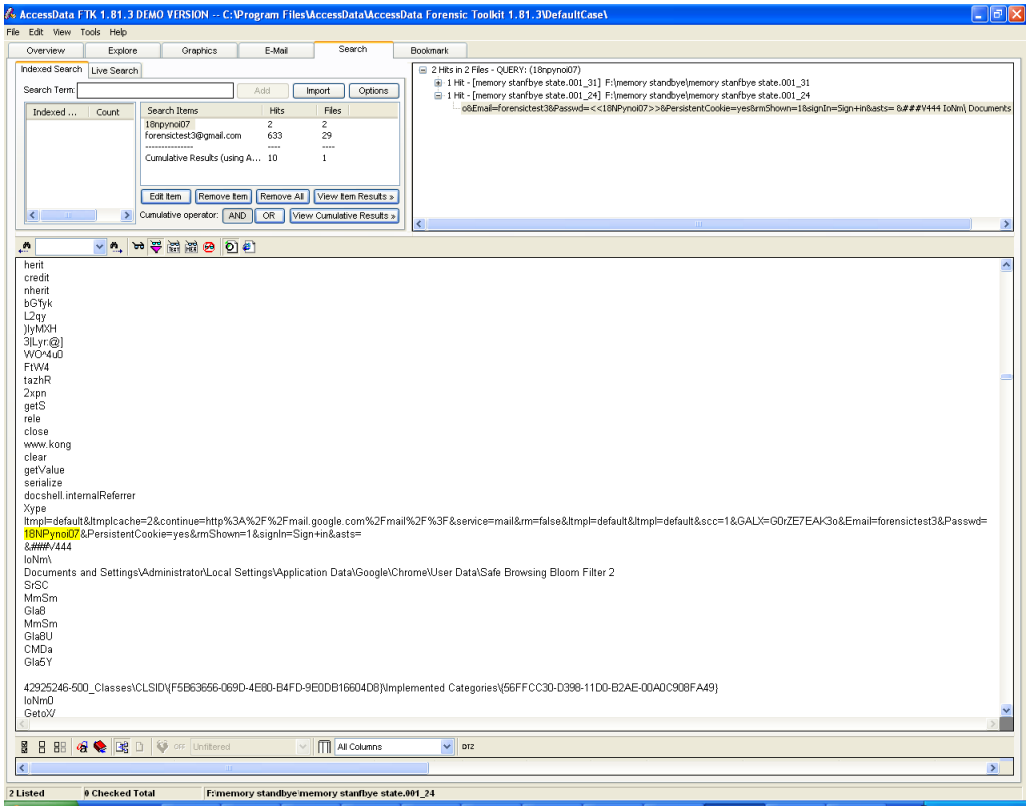

Figure 21: Gmail Password from memory image of Standby state

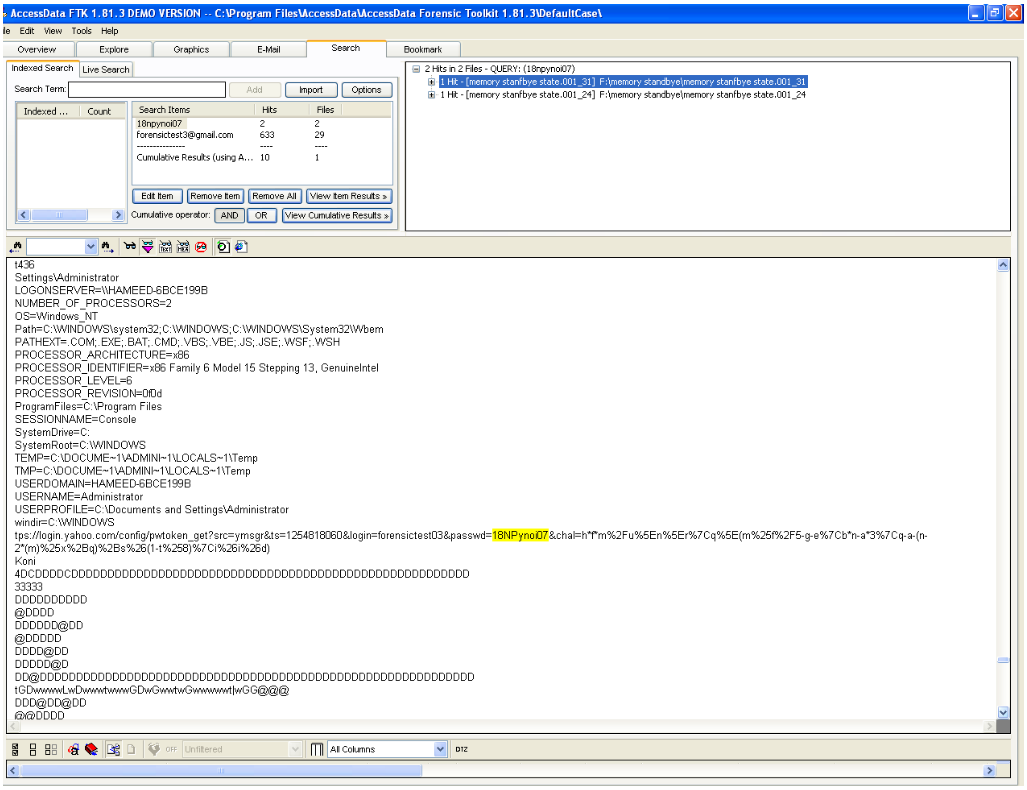

Figure 22: Yahoo Password from memory image of Standby state

Source : Physical Memory					
Status: Standby (Activated after 20 Minutes)					
Chat Application	**Sensitive Data**				
	Reciever Data			**Sender Data**	
	No of User e-mail IDs	**No of Paswords**	**Chat History**	**No of User E-mail IDs**	**Chat History**
MSN	454	No Hits	Yes	1397	Yes
Gmail	239	1 Hit	Yes	6536	Yes
Yahoo	364	1 Hit	yes	176	Yes
Total	1057	2 Hits	Yes	8109	yes
Source: Page File					
Status: Standby (Activated after 20 Minutes)					
Chat Application	**Sensitive Data**				
	Reciever Data			**Sender Data**	
	No of User e-mail IDs	**No of Paswords**	**Chat History**	**No of User E-mail IDs**	**Chat History**
MSN	209	1 Hit	Yes	302	Yes
Gmail	112	No Hits	Yes	411	Yes
Yahoo	172	2 Hits	Yes	244	Yes
Total	493	3 Hits	yes	957	yes

Figure 23: Standby State

5.4 Hibernation State

In hibernation state the test machine was put into hibernation after 1 hour of no user activity. Again we were able to extract many user IDs and and some passwords from the memory and page file images. Figure 24 is showing the yahoo password recovered during analysis of the hibernated state memory. Figure 25 is showing the summarized results for this state of the machine.

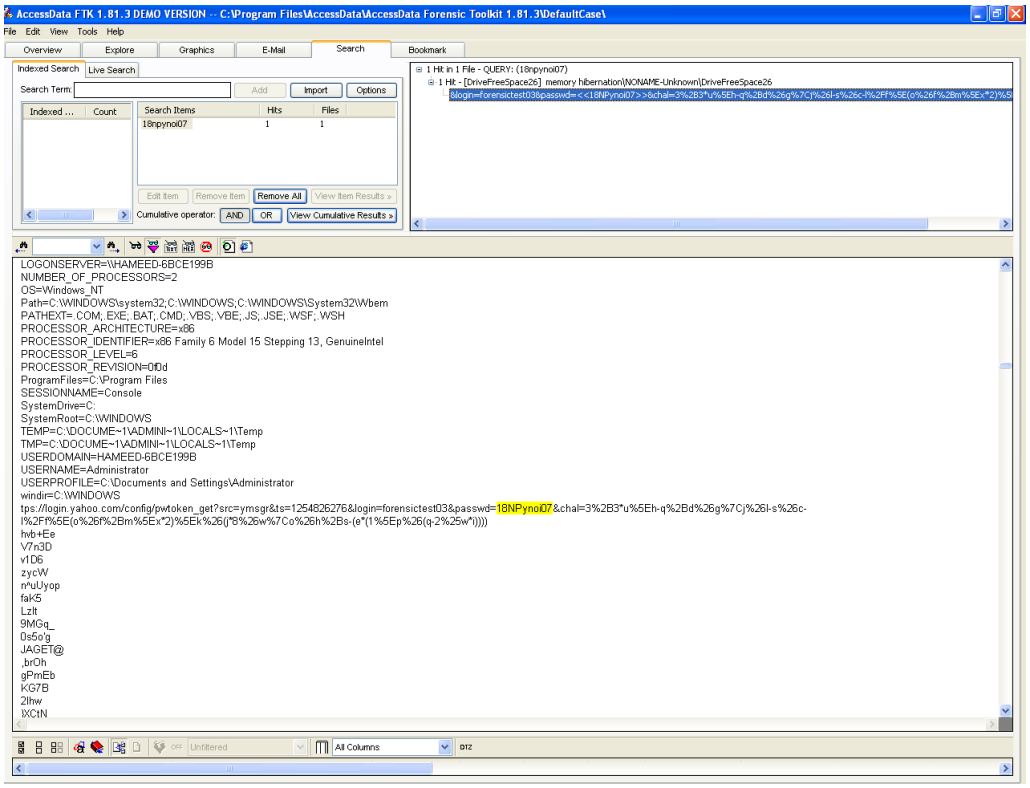

Figure 24: Yahoo Password from memory image of Hibernation state

Source : Physical Memory

Status: Hibernation (Activated after 60 Minutes)

Chat Application	Sensitive Data				
	Reciever Data			Sender Data	
	No of User e-mail IDs	No of Paswords	Chat History	No of User E-mail IDs	Chat History
MSN	708	1 Hit	Yes	296	Yes
Gmail	318	No Hits	Yes	196	Yes
Yahoo	459	1 Hit	yes	182	Yes
Total	1485	2 Hits	Yes	674	yes

Source: Page File

Status: Hibernation (Activated after 60 Minutes)

Chat Application	Sensitive Data				
	Reciever Data			Sender Data	
	No of User e-mail IDs	No of Paswords	Chat History	No of User E-mail IDs	Chat History
MSN	133	1 Hit	Yes	123	Yes
Gmail	42	1 Hit	Yes	355	Yes
Yahoo	87	No Hits	Yes	111	Yes
Total	262	2 Hits	yes	589	yes

Figure 25: Hibernation state

5.5 Log off State

In log off state all the chat clients were running and user web accounts were open in the browsers. Then without closing those applications we logged off the machine to see the discarding ability of the physical memory and operating system. After few minutes we logged on to the test machine to see the remanence effects of the memory. Still were able to extract user IDs and passwords from this state. Surprisingly we were able to get 4 hits of the hotmail password. Figure 26 is showing the results of the of those occurrences. Figure 27 is showing the summarized results of log off state.

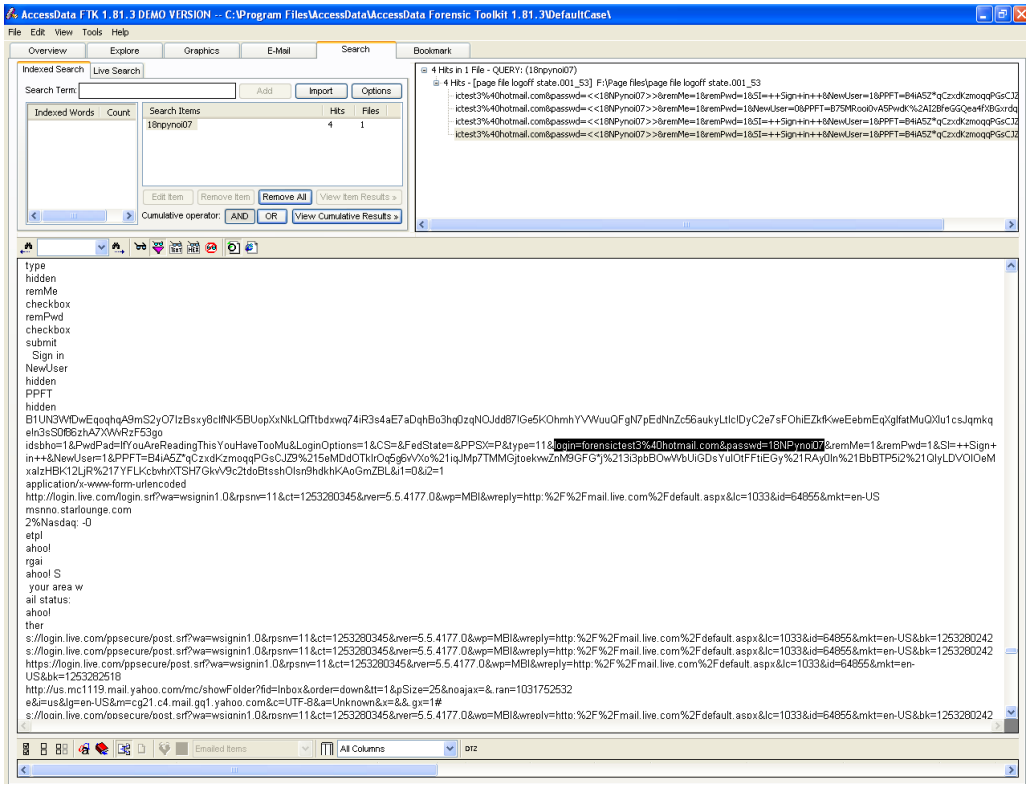

Figure 26: Hotmail Passwords recovered from page file of Log off State

Source : Physical Memory					
Status: Logoff State					
Chat Application	**Sensitive Data**				
	Reciever Data			**Sender Data**	
	No of User e-mail IDs	**No of Paswords**	**Chat History**	**No of User E-mail IDs**	**Chat History**
MSN	168	1 Hit	Yes	89	Yes
Gmail	131	1 Hit	Yes	109	Yes
Yahoo	107	No Hits	yes	76	Yes
Total	406	2 Hits	Yes	274	yes
Source: Page File					
Status: Logoff State					
Chat Application	**Sensitive Data**				
	Reciever Data			**Sender Data**	
	No of User e-mail IDs	**No of Paswords**	**Chat History**	**No of User E-mail IDs**	**Chat History**
MSN	433	4 Hits	Yes	275	Yes
Gmail	56	No Hits	Yes	34	Yes
Yahoo	85	No Hits	Yes	254	Yes
Total	574	4 Hits	yes	563	yes

Figure 27: logoff state

5.6 Soft Reboot State

In soft reboot the test machine was restarted under the control of operating system. This state gives us an idea of the data curbing ability of the operating system when it is warm and then retatred. Figure 28 is showing the summary of findings from this state of the machine.

Source : Physical Memory

Status: Soft Reboot

Chat Application	Sensitive Data				
	Reciever Data			Sender Data	
	No of User e-mail IDs	No of Paswords	Chat History	No of User E-mail IDs	Chat History
MSN	5	1 Hits	Yes	5	Yes
Gmail	3	No Hit	Yes	7	Yes
Yahoo	3	No Hit	yes	3	Yes
Total	11	1 Hit	Yes	15	yes

Source: Page File

Status: Soft Reboot

Chat Application	Sensitive Data				
	Reciever Data			Sender Data	
	No of User e-mail IDs	No of Paswords	Chat History	No of User E-mail IDs	Chat History
MSN	282	2 Hits	Yes	223	Yes
Gmail	122	1 Hit	Yes	77	Yes
Yahoo	160	No Hits	Yes	375	Yes
Total	564	3 Hits	yes	675	yes

Figure 28: Soft Reboot state

5.7 Hard Reboot State

In hard reboot we restarted the test machine without allowing the operating system to shut down cleanly. Still we were able to extract a couple of user IDs from the physical memory of this state. Figure 29 is showing the results of the this state.

Source : Physical Memory

Status: Hard Reboot

Chat Application	Sensitive Data				
	Reciever Data			Sender Data	
	No of User e-mail IDs	No of Paswords	Chat History	No of User E-mail IDs	Chat History
MSN	1	No Hit	Yes	No Hit	Yes
Gmail	No Hit	No Hit	No	1	Yes
Yahoo	No Hit	No Hit	No	No Hit	Yes
Total	1	0	Yes	1	yes

Source: Page File

Status: Hard Reboot

Chat Application	Sensitive Data				
	Reciever Data			Sender Data	
	No of User e-mail IDs	No of Paswords	Chat History	No of User E-mail IDs	Chat History
MSN	1325	2 Hits	Yes	644	Yes
Gmail	385	1 Hit	Yes	872	Yes
Yahoo	749	1 Hit	Yes	302	Yes
Total	1759	3 Hits	yes	1818	yes

Figure 29: Hard Reboot state

5.8 Passwords Summary

In this we will provide a summary of passwords from all defined states of the experiments.

From figure 30 we can see that in all defined states the most occurring password was from Hotmail. From column 1 we can see that It was found in the memory images of all states except "Ready", "Standby" and "Hard Reboot". While in the case of page file images it was not found just in "Screen Saver" state. Yahoo was 2nd by occurrence in the defined states. Its occurrence was frequent until "Standby" state but for the rest of the states we did not found any password for except in the page file of "hard Reboot" where it was traced one time. Gmail was in 3dr position by occurrence. Its distribution is random in all the defined states. Some times it was traced in the memory image and some times in the page file images. From the summary we also can get the idea that password padding is the function of time. We still have chance to trace passwords but with the increase in time the padding factor also increases. We also see that password was traced in the page file of every state at least for two applications. It means once the system start paging, there is quite enough probability to trace password. The most found password in any state was that of Hotmail found in the page file of "log off" state. We found 4 hits. This shows the importance of page file in a digital investigation process.

State	Source	Application		
		MSN Password	Yahoo password	Gmail password
Ready	Memory	-	1	1
	Page File	1	1	-
Screen Saver	Memory	2	1	-
	Page File	-	1	1
Standby	Memory	-	1	1
	Page File	1	2	-
Hibernation	Memory	1	1	-
	Page File	1	-	1
Logoff	Memory	1	-	1
	Page File	4	-	-
Soft Reboot	Memory	1	-	-
	Page File	2	-	1
Hard Reboot	Memory	-	-	-
	Page File	2	1	1

Figure 30: Summary of passwords recovered in all states

5.9 Receivers ID Summary

In section we give a summary of the users IDs found at the "Receiver" side of the communications application wise. If we look at the memory column of all the defined states in figure 31, we get a clear idea about the behavior of applications. The occurrence of Hotmail Ids were dominant in all states, therefore it is at the 1st position in the figure. While Yahoo and Gmail showed unpredictable behavior. In half states yahoo IDs were dominant while in the remaining half states Gmail IDs were dominant. Therefore they both share position 2 and 3 in the figure. Now if we consider the column of page file, we get a clear picture for all the states of the observed system. Again like memory, here also Hotmail was the dominant ID found in all states. Yahoo is at 2nd position and Gmail at 3rd position. From these results we can get an idea about the behavior of opearting system.

Receiver Data						
Source	Memory			Page File		
Order of occurrence	1st	2nd	3rd	1st	2nd	3rd
Ready State	MSN	Gmail	yahoo	MSN	Yahoo	Gmail
Screen Saver State	MSN	Yahoo	Gmail	MSN	Yahoo	Gmail
Standby State	MSN	Yahoo	Gmail	MSN	Yahoo	Gmail
Hibernation State	MSN	Yahoo	Gmail	MSN	Yahoo	Gmail
Logoff State	MSN	Gmail	Yahoo	MSN	Yahoo	Gmail
Soft Reboot state	MSN	Gmail	Yahoo	MSN	Yahoo	Gmail
Hard Reboot State	MSN	-	-	MSN	Yahoo	Gmail

Figure 31: Summary of receiver IDs recovered in all states

5.10 Senders ID Summary

In this section we give a summary of the results for the "Sender" IDs that we found on the observed system. From figure 32 we can see that opearting system and applications have mostly shown a consistent approach regarding handling of the "Sender" IDs in memory section. In this case we we found Gmail as the dominant occurring ID in all the defined states of the system except hibernation state where MSN is the most coouring ID. Where as MSN is the second dominant occurring followed by Gmail. If we consider the page file section of figure ABC, we see different results. In all the states until "hibernation" state we can see a uniform approach by the applications and opearting system. Again just like memory Gmail was the dominant occurring ID followed by MSN and then Yahoo. But from "Log off" state the results were mix. IDs from all three applications occurred in different order for these states.

Sender Data						
Source	Memory			Page File		
Order of occurrence	1st	2nd	3rd	1st	2nd	3rd
Ready State	Gmail	MSN	Yahoo	Gmail	MSN	Yahoo
Screen Saver State	Gmail	MSN	Yahoo	Gmail	MSN	Yahoo
Standby State	Gmail	MSN	Yahoo	Gmail	MSN	Yahoo
Hibernation State	MSN	Gmail	Yahoo	Gmail	MSN	Yahoo
Logoff State	Gmail	MSN	Yahoo	MSN	Yahoo	Gmail
Soft Reboot State	Gmail	MSN	Yahoo	Yahoo	MSN	Gmail
Hard Reboot State	Gmail	-	-	Gmail	MSN	Yahoo

Figure 32: Summary of sender IDs recovered in all states

6 Results Discussion

In this section we will elaborate on the results found in chapter 5. We will discuss our findings in context of physical memory and page file, effect of increased size of physical memory, the influence of opearting systems, role of applications and the state of a system at a given time and finally role of the currently available tools for the analysis. All of them play their role while investigating volatile memory. While discussing these factors our focus will be on the answer to the research questions of this thesis work.

6.1 State of the System

Our experiments tried to answer to the question whether we can extract sensitive data from memory and page file when the computer is in a particular state. The results confirms that it is possible to extract sensitive data from memory and page file in all the mentioned states. One main factor in all these states is time. We increased the time in ascending order to see the fading effect of data in memory. This was specially observed for screen saver state (10 minutes), Standby state (20 Minutes) and Hibernated state (60 minutes). For the rest of the cases we did not kept a strict time constraint. On average the system was used for at least one hour before it was imaged for analysis. In all cases we observed a predictable behavior in memory regarding the fading of data. The fading effect of data increased with the increase of no user activity.

The factors for page file were little different in this case. Swapping depends on the usage of physical memory. The more the usage of physical memory, the more should be swapping. Though in our experiments we observed unpredictable behavior of swapping. In some cases we found more hits and in in others we found less hits. There was no ascending or descending order of number of hits found.

6.2 Behavior of the Operating System and Applications

Our experiments tried to see the behavior of operating system and application in handling sensitive data. From results we can see that it is possible to expect careless behavior from the opearting system and applications we have utilized. We got quite alot of hits for both user names and passwords in all states of the system. The same is the case with handling passwords except hard reboot where we could not trace any password in the memory image. Passwords were lying in plain text both in memory and page file in most cases.Our experiments also reveal that sensitive data is swapped to page file with no discrimination by these applications. The same is the case with browsers applications.

6.3 Ability of Available Tools

We used three memory forensic tools for our experiments. Since our the required ability from these tools was to do a keyword search which is not a big challenge for a forensic tool. String and keyword search is an established searching technique in digital forensic so it was quite easy for these tools to extract the required results from the evidence. The indexing feature of Access Data's Forensic Toolkit was very handy in our experiments. Using this feature we were able to get the total hits for a given keyword in one cycle. But

still there were some disagreements between the results of the three tools that we used. In case of password search the results of the tools were diffenet from each other. This means the tools need standardization and more development in memory analysis.

6.4 The Importance of Page File

We included page file in our experiments. The numbers of user names and password recovered from this source shows its importance in forensics. But if we want to get a complete picture of the what is going on in the system then it should be acquired at the same time when memory acquisition. Though it can also be recovered later from hard disk image. But alone page file or memory will reveal incomplete analysis results. As discussed in chapter 3, page file is not cleared by default by windows systems, so data once swapped to page file can reside there for an unlimited time.

6.5 Effect of the Memory Size

Our allocated memory size in all stats and scenarios was 1 GB. Though we did not measure the influence of VMware on memory in the experiments. But the results are making it clear that as the size of memory is increasing in computers, the chance of getting sensitive data are also increasing. We can say that we have the luxury of more memory but on the other hand it is also exposing the sensitive data. This matter should be taken care of, both from opearting system and application softwares.

6.6 Limitations and Caveats

in this section we would like to mention some of the constraints and limitations of the work performed in this thesis report. All the experiments conducted for this report were executed in an ideal virtual environment. The purpose was to show the importance of this emerging field of memory forensics and serve the forensic research community and crime digital investigators. In actual crime scene the status of the system might be diffenet resulting in different analysis outcomes. One of the concern is digital forensics is to test the tools and techniques on a common evidence to compare the analysis results and make the process of forensics more standardized. In our experiments we have taken full care to do the analysis and investigations by following the standards rules of forensics. All the acquired images were hashed and saved to be used in future references.

The memory dump techniques used in this report are also simplified compared to the real world. Suspending a VMware session is of course a dream come true for an investigator, giving him a picture of an entire machine in seconds. Though this technique is no more uncommon and is getting popularity. Since most memory imaging techniques available leaves a footprint, these should be investigated and documented so that they may be identified and excluded from the final evidence.

7 Conclusions

This thesis work has attempted to show the usefulness of the forensic analysis of physical memory and page in a typical digital investigation. Sensitive data was extracted from the physical memory and page file of a live computer running under Windows operating system. Several states of the computer were investigated running different applications. In the next section we will summarize the findings of this report in the form of a chapter wise summary. After the research has been summarized, possible future work will be suggested.

7.1 Summary

In chapter one we tried to show the importance of volatile data forensics. The main issue is the increasing capacity of both volatile memory and permanent storage devices. The investigators are reluctant to add another disk to who are already finding it hard to analyze the huge capacity had disks. But on the other hand this volatile memory can give very vital information that can really increase the finding the desired results fast. Based on this theory we found several questions whose answers proved the importance of this volatile data source. The answers to these questions were covered in the remaining chapters. chapter 2 highlighted the state of the art in digital forensics showing where the forensics of volatile memory is standing. digital forensics has traveled its journey around 80s and by 2000 volatile memory was included in the 1st responder investigations. While physical memory was seriously taken as separate source for investigations in around 2005. It all started from the 1st digital forensic work shop (DFRWS) when memory was investigated and tool development started with rigorous speed. The internal working of the memory and page file was discussed in chapter 2 giving some more information for the interested users in that direction. The 1st task is the acquisition of memory and page file therefore we also included this section in our work. The currently available methods for the acquisition of memory and page file were highlighted with their strengths and weaknesses. The next step is to know the available techniques for the analysis of memory. This was also discussed with sufficient depth. We should also know the current state of the available tools for the analysis of memory and page file. therefore we included this section in our report to get an idea of the capabilities of the available tools in this direction. We should also know the current state of research in volatile data forensics. This will be a useful reference to get an understanding of what has been until now in memory and page file forensics. In the remaining section we tried to find answers to the questions if we can recover sensitive data from memory and page file. The methodology section explains the experiments setup for the experiments. We established several states of the system that would reflect real world situations. Our experiments revealed that we can get sensitive data like user names and passwords from both memory and page file if we investigate the computer in the defined running state. The results of our experimental work were discussed in chapter 5 and 6.

From the discussion of results in chapter 6 it is obvious that physical memory and page are important sources of information in a digital investigation process. Once data is in

physical memory, it can lie there for quite enough time. This is specially important when drive encryption is being used. The encryption keys line in plain text in memory. Now if memory is not captured during crime scene preservation then it might be very hard and time consuming for the examiner to break the encryption who is already overwhelmed by the huge amount of hard disk analysis. We also recommend that the memory page file should be integrated into the existing classical method of disk investigation. This will give a whole synchronized picture of the digital incident. We also suggest that progress in several areas will be essential to increase the usefulness of live forensics, including tools to automate and standardize the process of evidence acquisition, preservation and presentation that allow an investigator to present the facts clearly to a court.

7.2 Future Work

Due to time constraints we could not add some of the following extensions to our work. But they can be added as future work to our report. In additions to those some more future work is also suggested in this section.

- we have covered only Windows XP professional SP3 in our experimental work. This work can be extended to more operating systems like Vista and windows 7 which is quite new to the market. This would serve the forensic community quite alot.
- Another extension of our work would be to extend it to 64 bit operating systems. We might find different behavior in those systems.
- We tried to cover the most common states of the system but still there are many system states of the computer that can be considered. Safe mode can be added to the system states. Research shows that if the system has been shut down is recently recently then there are still good chances that data in the memory can be recovered. Therefore adding safe mood could be a nice option in this case since the operating system is loaded with minimum files in that state.
- In our experiments we used the most common that are selected by end users. Our test users asked the application to remember their user names and passwords. Also page file and memory were not cleared at system start up. Experiments on thses options could yield interesting results if we ask the applications not to remember user names and passwords. In the same way if we choose to clear page file and memory by the operating system at startup.
- Very little has been done on combining both memory and page file in digital forensics. This could be of great value to the forensic community.
- Incorporating memory and page file with classical disk forensics will give a complete picture of the whole crime scene. This could be a very interestinga and valuable work.
- There is no standard way of memory and page file acquisition. The impact of the available tools is varied. A standardized solution to this problem could be a valuable contribution. Also assessing the impact of the currently available tools and techniques will serve the forensic community alot.

Bibliography

[1] Comparison of microsoft windows versions. Last visisted on 20 October 2009.

[2] Debugging tools for windows - overview. Last visited on 08/09/2009.

[3] Deviceobject. Last visited on 06/09/2009.

[4] Disk explorer. Last visited on 15/09/2009.

[5] F-response. Last visited on 08/09/2009.

[6] Fastdump - a memory acquisition tool. Last visited on 05/09/2009.

[7] Forensic tool kit. Last visited on 15/09/2009.

[8] How to overcome the 4,095 mb paging file size limit in windows. Microsoft KB Article 237740. Last visited on 08/08/2009.

[9] ilook. last visited on 15/09/2009.

[10] Microsoft virtual pc. Last visited on 07/09/2009.

[11] Overview of memory dump file options for windows vista, windows server 2008, windows server 2003, windows xp, and windows 2000. Last visited on 10/09/2009.

[12] Quinn's web of horrors. Last visited on 08/09/2009.

[13] Vmware. Last visited on 06/09/2009.

[14] What is apm mode? Last visited on 10/10/2009.

[15] Worldwide virtualization services 2009Ű2013 forecast. Last visited on 05/09/2009.

[16] X-ways forensics. Last visited on 15/09/2009.

[17] You cannot put a computer that has more than 4 gb of memory into hibernation in windows xp, in windows server 2003, in windows vista, or in windows server 2008. Last visited on 05/09/2009.

[18] Jr AAron Walters, Nick L.Petroni. Volatools: integrating volatile memory forensics into the digital investigation process. 2007.

[19] Frank Adelstein. Live forensics: Diagnosing your system without killing it first. 2006.

[20] Carsten Maartmann-Moe André Årnes and Steffen E. Thorkildsen. The persistence of memory: Forensic identification and extraction of cryptographic keys. *Volume 6, Supplement 1, September 2009, Pages S132-S140 The Proceedings of the Ninth Annual DFRWS Conference*.

[21] 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.

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

[23] Daniel Ayers. A second generation computer forensic analysis system. *The Proceedings of the Ninth Annual DFRWS Conference*, Volume 6, Supplement 1, 2009.

[24] Venansius Baryamureeba and Florence Tushabe. The enhanced digital investigation process model. *Proceedings of DFRWS*, 2004.

[25] Chris Betz. memparser. www.dfrws.org/2005/challenge/memparser.html.

[26] Adam Boileau. Hit by a bus: Physical access attacks with firewire.

[27] D. Brezinski and T. Killalea. Guidelines for evidence collection and archiving. *RFC 3227, IETF*, 2002.

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

[29] Richard Nolan Cal Waits, Joseph Ayo Akinyele and Larry Rogers. Computer forensics: Results of live response inquiry vs. memory image analysis. 2008.

[30] Immunity CANVASS. last visited on 27.08.2009.

[31] Director Carrie Morgan Whitcomb. An historical perspective of digital evidence: A forensic scientist view. *International Journal of Digital Evidence*, Volume 1, Issue 1, 2002.

[32] Brian Carrier. The sleuth kit and autopsy. last visited on 27/08/2009.

[33] Brian Carrier and Eugene H. Spafford. Getting physical with the digital investigation process. *International Journal of Digital Evidence (IJDE), 2(2)*, 2003.

[34] Brian D. Carrier and Eugene H. Spafford. An event-based digital forensic investigation framework. *DFRWS 2004*.

[35] Brian D. Carrier and Eugene H. Spafford. Defining event reconstruction of a digital crime scene. *Journal of Forensic Sciences*, 2004.

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

[37] Harlan Carvey. *Windows Forensics and Incident Recovery*. Addison Wesley, 2004.

[38] Harlan Carvey. Windows memory analysis (chapter 3). *Windows Forensic Analysis DVD Toolkit, Second Edition*, pages 87–123, June 2009.

[39] AccessData Corporation. Forensic toolkit. Last visited on 22/08/2009.

[40] Mantech International Corporation. Mantech memory dd. Last visited on 08/09/2009.

[41] deroko. Ultimate way to hide rootkit. last visited on 28.08.09.

[42] DFRWS. Road map for digital forensic research. 2001. Report from the first digital forensic research work group (DFRWS).

[43] Digital Forensic Research Workgroup (DFRWS). Memeory analysis challenge, 2005.

[44] E-Fense. Helix. Last visited on 20/09/2009.

[45] Dan Farmer and Wietse Venema. *Forensic Discovery*. Pearson Education, Inc, 2004.

[46] Espen André Fossen and André Årnes. Forensic geolocation of internet addresses using network measurements. Master's thesis, Institutt of Telematics, Norwegian University of Science and Technology, 2005.

[47] Katrin Franke and Sargur N. Srihari. Computational forensics: An overview. Volume 5158/2008, 2008.

[48] Gabriela Limon Garcia. Forensic physical memory analysis: An overview of tools and techniques.

[49] Peter Gutmann. Data remanence in semiconductor devices. 2001. IBM T.J.Watson Research Center.

[50] HbGarry. Responder professional. Last visited on 02/09/2009.

[51] Vassil Roussev. Golden G. Richard III. Breaking the performance wall: The case for distributed digital forensics. *The Proceedings of the 2004 Annual DFRWS Conference*, 2004.

[52] Core Impact. Last visited on 26.08.09.

[53] Nadia Heninger William Clarkson William Paul Joseph A. Calandrino Ariel J. Feldman Jacob Appelbaum and Edward W. Felten J. Alex Halderman, Seth D. Schoen. Lest we remember: Cold boot attacks on encryption keys. *Proc. 2008 USENIX Security Symposium*, Feb 2008.

[54] Derek Bem Jason Solomon, Ewa Huebner, , and Magdalena Szezynska. User data persistence in physical memory. *Digital Investigation, Volume 4, Issue 2, June 2007, Pages 68-72*, March 2007.

[55] Tal Garfinkel Kevin Christopher Mendel Rosenblum Jim Chow, Ben Pfaff. Understanding data lifetime via whole system simulation. 2004. Standford University Department of Computer Science.

[56] Tal Garfinkel Mendel Rosenblum Jim Chow, Ben Pfaff. Data lifetime is a system problem. 2004. Standford University Department of Computer Science.

[57] Tal Garfinkel Mendel Rosenblum Jim Chow, Ben Pfaff. Shredding your garbage: Reducing data lifetime through secure dealocation. 2004. Standford University Department of Computer Science.

[58] George M. Garner Jr. Forensic acquisition utilities. Last visited on 04/09/2009.

[59] Garner jr GM. Knottools. http://www.dfrws.org/2005/challenge/kntlist.shtml.

[60] Randy Kath, 1992. Last visited on 10.10.2009.

[61] Chris prosise Kevin Mandia and Matt pepe. *Incident Response and Computer Forensics, 2nd edition*. McGraw Hill Osborne Media, 2003.

[62] Jesse D. Kornblum. Using every part of the buffalo in windows memory analysis. *In proceedings of the Annual DFRWS research conference*, 4, 2007.

[63] Marthie Lessing. Live forensic acquisition as alternative to traditional forensic processes. *IMF Conference September 2008*.

[64] Ted Lindsey. Challenges in digital forensics. *In proceedings of the 2004 Annual DFRWS Conference*, 2006.

[65] Eoghan Casey BS MA. *Digital Evidence and Computer CrimeŮForensic Science, Computers and the Internet, Second Edition*. ACADEMIC PRESS, An imprint of Elsevier, 2004 (2nd edition).

[66] Carsten Maartmann-Moe. Digital evidence and cryptography. 2007. Master Thesis, NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY, MATHEMATICS AND ELECTRICAL ENGINEERING.

[67] Douglas MacIver. System integrity team blog. Last visited on 12/09/2009.

[68] MANDIANT. Memoryze. Last visited on 25/09/2009.

[69] Clint Carr Mark Reith and Gregg Gunsch. An examination of digital forensic models. *International Journal of Digital Evidence*, Volume 1, Issue 3, 2002.

[70] R. McKemmish. 'what is forensic computing?'. 1999. Australian Institute of Criminology.

[71] David Collett Michael Cohen. Pyflag. Last visited on 10.06.2009.

[72] Diane Barrett Michael G. Solomon and Neil Broom. *Computer Forensics JumpStart*. Sybex, 2005.

[73] Microsoft. Turn off a computer: frequently asked questions. Last visited on 10/10/2009.

[74] Microsoft. Windows feature lets you generate a memory dump file by using the keyboard. Last visited on 05/09/2009.

[75] M. Miller and J. Turkulainen. Remote library injection, 2004.

[76] S. Mukkamala and A.H. Sung. Identifying significant features for network forensic analysis using artificial techniques. *International Journal of Digital Evidence, vol. 1, no. 4.*, 2003.

[77] Timothy Fraser William A. Arbaugh Nick L. Petroni Jr., AAron Walters. Fatkit: A framework for the extraction and analysis of digital forensic data from volatile system memory. *In proceedings of the Annual DFRWS research conference*, 3, 2006.

[78] U.S. Department of Justice. Electronic crime scene investigation: A guide for first responders, second edition. 2001.

[79] National Institute of Standards and Technology (NIST). Computer forensics tool testing (cftt) project. Last visited on 02/10/2009.

[80] Scientific Working Group on Digital Evidence (SWGDE). Digital evidence: Standards and principles, 2000. Last visited on 05/09/2009.

[81] The Metasploit Project. Last visited on 28.08.09.

[82] Tianjie Cao Qian Zhao. Collecting sensitive information from windows physical memory. *JOURNAL OF COMPUTERS, VOL. 4, NO. 1*, 2009.

[83] A.R. van Ballegooij R.B. van Baar, W. Alink. Forensic memory analysis: Files mapped in memory. *In proceedings of the Annual DFRWS research conference*, 5, 2008.

[84] G.G. Richard III and V. Roussev. Digital forensics tools: The next generation. 2006.

[85] Robert Richardson. Csi computer crime & security survey 2008. 2008.

[86] Nicolas Ruff. Windows memory forensics. October 2007.

[87] Mark Russinovich. Process explorer, February 4, 2009. Last visited on 02/10/2009.

[88] Mark E. Russinovich and David A. Solomon. *Microsoft Windows Internals, Fourth Edition: Microsoft Windows Server 2003, Windows XP, and Windows 2000*. Microsoft Press, 2004.

[89] Joanna Rutkowska. Subverting vista kernel for fun and profit. 2006.

[90] M. Debbabi S. M. Hejazi, C. Talhi. Extraction of forensically sensitive information from windows physical memory. *digital investigations*, 6, 2009.

[91] Bruce Sanderson. General windows information ram, virtual memory, pagefile and all that stuff.

[92] Antonio Savoldi and Paolo Gubian. Towards the virtual memory space reconstruction for windows live forensic purposes. *Third International Workshop on Systematic Approaches to Digital Forensic Engineering*, 2008.

[93] Andreas Schuster. Dmp file structure. Last visited on 08/09/2009.

[94] Andreas Schuster. Ptfinder and ptfinderfe. last visited on 22/08/2009.

[95] Andreas Schuster. Pool allocations as an information source in windows memory forensics. *Third international conference on incident management & IT-forensics*, 2006.

[96] Andreas Schuster. Searching for processes and threads in microsoft windows memory dumps. *In proceedings of the annual DFRWS conference*, 3, 2006.

[97] Andreas Schuster. The state of the art in windows memory forensics. 2008.

[98] Andreas Schuster. Acquisition (5): Firewire. Andreas Schuster blog, February 2008.

[99] Douglas Schweitzer. *Incident Response: Computer Forensics Toolkit*. Wiley Publishing Inc., 2003.

[100] Sangjin Lee Seokhee Lee, Antonio Savoldi and Jongin Lim. Digital evidence collection process integrity and memory information gathering. 2005.

[101] Sangjin Lee Seokhee Lee, Antonio Savoldi and Jongin Lim. Password recovery using an evidence collection tool and countermeasures. 2007.

[102] Sangjin Lee Seokhee Lee, Antonio Savoldi and Jongin Lim. Windows pagefile collection and analysis for a live forensics context. 2007.

[103] Ryan Single. Encryption still good; sleeping mode not so much, pgp says, February 2008. Last visited on 10/10/2009.

[104] Sergei Skorobogatov. Low temperature data remanence in static ram. June 2002.

[105] Andrew S.tanenbaum and Albert S.Woodhull. *Operating Systems Design and implementation 3rd Edition*. pearson Printice Hall.

[106] Matthieu Suiche. Windd, windows physical memory imaging utility. Last visited on 08/09/2009.

[107] Matthieu Suiche. Windows hibernation file for fun n profit.

[108] I. Sutherland, J. Evans, T. Tryfonas, and A. Blyth. Acquiring volatile operating system data tools and techniques.

[109] GMG Systems. Knttools with kntlist. Last visited on 08/09/2009.

[110] Volatile Systems. Volatility framework. Last visited on 15.07.2009.

[111] X-Ways Software Technology. Winhex. Last visited on 20/08/2009.

[112] Brian Kim Tim Grance, Karen Kent. *Computer Security Incident Handling Guide, Recommendations of the National Institute of Standards and Technology*. National institute of standarad and technology US department of commerece, 2004.

[113] Blom M van der Steen, M. A roadmap for future forensic research. technical report. 2007.

[114] Tim Vidas. Post-mortem ram forensics. *CanSecWest 2007*.

[115] Arne Vidstrom. Memory dumping over firewire uma issues. Last visited on 14/09/2009.

[116] MS W. Jerry Chisum, BS; Brent E. Turvey. Evidence dynamics: Locard's exchange principle & crime reconstruction. *Journal of Behavioral Profiling, January, 2000, Vol. 1, No. 1*. Last visited on 12/09/2009.

A SHA-1 and MD5 Sums

This section contains the cryptographic check sums of all the acquired images for physical memory and page files of the defined states during the experinemts.

State of the System	**Physical Memory**	
Cryptographic Checksums	**SHA1**	**MD5**
Configured Syatem	69f35eee0fa81ac433098dd5b6fd3798d6daee87	G6108nva0giop489c7eka70ze3jf38jc7
Ready State	76fg96vd98br4xh9ashdelhy64ecd807bgq1f74la5	70ba86f7c8b756a444abb58321406c62
Screen Saver State	3eb34d3a276740d672da84efa000135a9d636b16	8187422b85f34ea48e4c35effb2bdf2c
Standby	1fd2961bda35ed9c4bee4e663e9a27c9fb37fb72	a9d8659f1b7a773d40a4e2b2c85933f8
Hibernation	ae82868c7d93e79887dafc5c16a3cd2eaa0976fb	22cabea1d387faa84d478e8d99474d98
Logoff State	8b88d6da46792267e9f21e21cf0416ef832e0f42	bb98f117a095fc031585aed818dcb1cb
Soft Reboot	48d33b807d7b9483a16abee87e567243e675abfd	eac9eefa281f28e2efdaa14579ab6fc3
Hard Reboot	b408xts8rnx20mlzce84d1ookp715gdai93hfe67ip	08d7tioe753i4od1gt4z92hqfdy5323bx9

Figure 33: Cryptographic check sums of the memory images

State of the System	Page File	
Cryptographic Check Sums	SHA1	MD5
Configured Syatem	1cak84sloqa74cnzlsxt263halop01cbnn3l6azx7ic	35b10815c7b59447ebec77a711aa0752
Ready State	587bge7zl95d1k106dgafhw962rv190d9j74vzkjas	V6487jhdkgtfjj58219cgtrlo2e36vqwhdg
Screen Saver State	cc4e7a4e3a184b2fe2be884b861b9ca9a4afd737	bea377cbfc35707c5e662dfe871dbb19
Standby	09jnc20nrt10max6js291bw6nv1x05cb60lxef9kj	b9e0e472ad45a378175517b3ac3487ac
Hibernation	O801fklnrc5c900xn3pl1ei66llpc2suivx27zn234	A3nv9kjgs5hdo9225vbwk9610mlhfc52
Logoff State	47e7c27ex6p929c978de579d8ffd79f448cc1e7	916a135384a7b853502cf2a5f41fa262
Soft Reboot	9xc12nmlxc70vr4tew166miww01ve899hgc2a1	70ba86f7c8b756a444abb58321406c62
Hard Reboot	bpo6vsdyu40891263oixb fqjd7623iobl12K97vj4	754nvxqolr64vjk8dtr29kq6d7bu92id1p

Figure 34: Cryptographic check sums of the page file images