WINDOWS HIBERNATION AND MEMORY FORENSICS

by

Amy L. Ayers

A Capstone Project Submitted to the Faculty of

Utica College

April 2015

in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Cybersecurity

ABSTRACT

The purpose of this capstone project was to research the hibernation file, its role in memory forensics and to explore current technology, techniques and concepts for analysis. This study includes an in-depth look at the Windows hibernation feature, file format, potential evidence saved to the file and its impacts in digital forensic investigations. This research was performed to demonstrate the importance of the hibernation file and to generate awareness for this forensic artifact. The research questions presented were designed to identify the properties of Windows hibernation and its significance in digital forensics. Additionally, these research questions were aimed at identifying the important concepts analysts should understand in selecting forensic software and in hibernation analysis. Through the literature review process, the hibernation file was identified as an essential part of digital forensics which provides analysts with snapshots of system memory from various points in the past. This data includes web, email and chat sessions in addition to running processes, login credentials, encryption keys, program data and much more. Beyond forensics, the hibernation file is useful in the fields of data recovery and incident response. A review of current hibernation file publications revealed incomplete and conflicting works culminating in the acknowledgment that more research is needed in order to close these research gaps. More awareness for hibernation forensics through its inclusion in future published works and in computer forensic educational courses is recommended. These inclusions will assist to arm practitioners with the ability to accurately utilize the hibernation file in order to obtain the highest quality forensic evidence. Keywords: Cybersecurity, hiberfil.sys, hybrid sleep, malware, slack space, Albert Orbinati.

ACKNOWLEDGMENTS

This capstone project was made possible because of the immense amount of support provided to me. First and foremost I would like to thank my husband, Clint. Without whose encouragement and support I would not have been able to complete my education. For several years now, he stood by me going above and beyond with household duties, not least of which included delivering hot plates of food to my desk as I sat feverishly trying to finish an assignment. His support, encouragement and love, not to mention cooking, carried me through and enabled my success. I would also like to thank my children for the sacrifices they have had to make as my time was divided between work, home and school. I would like especially to thank my daughter Amber who took on additional responsibilities and provided me with invaluable encourgagment. To my capstone advisor, Professor Albert Orbinati, thank you for your valuable feedback which helped to cultivate my writing into a Masters level project. A special thanks to my second reader Professor Vern McCandlish who unselfishly agreed to volunteer his time to be my subject matter expert. Finally, I would like to thank the Utica College Staff, most specifically all of my past instructors. Each of my professors at Utica College added their own unique instruction to my education which culminated in my ability to complete this capstone project.

TABLE OF CONTENTS

LIST OF ILLUSTRATIVE MATERIALS

STATEMENT OF THE PROBLEM

Definition of the Problem

Memory forensics is an essential part of a forensic investigation. The data contained in a systems memory is considered volatile because it is lost when a system is powered down. Memory provides valuable evidence including lists of running processes, network connections, program data, Internet history, chat sessions, login credentials, encryption keys, and clipboard contents (Carvey, 2007). Evidence from memory is only made available in an investigation if first responders find the system powered on, logged in and they are able to capture a memory image. Consequently, there are often cases where forensic analysts are not provided with this valuable data. In these cases, an analyst may benefit from stored memory dumps on the system such as the hibernation file.

Windows systems contain an energy saving feature called hibernation or hybrid sleep. This feature is activated when a system sits idle for a set time or if the laptop lid is closed. Upon activation, the systems memory is copied to the hibernation file, hiberfil.sys, in order for the system to be placed into a lower power state (Microsoft, 2015d). This file provides analysts with a memory image from the last time the system hibernated. Although this file can provide vital evidence, research has found that current studies are incomplete and sometimes contradictory. Additionally, there is a lack of studies on currently available forensic technology.

The purpose of this research is to demonstrate the importance of the hibernation file in memory forensics and to explore concepts, tools and techniques for analysis. This study will cover an in-depth look at the hibernation file binary structure, along with important memory forensic concepts. A comparative study of current forensic technology along with considerations for choosing forensic software will be explored.

Justifying the Problem

Memory forensics is crucial in many investigations due to of the vast amount of evidence it produces which does not exist on the hard drive. Memory images are not always available for analysis making the hibernation file the only source of this data. The hibernation file is also useful even when a memory image is available because it provides an additional data set from a previous point in time. When system backups are leveraged, memory snapshots can be examined from several points in the past. Examining these historical snapshots can help identify malware, encryption keys, login credentials and other valuable evidence contained in memory. The hibernation feature is complex and varies between operating systems and hardware configurations. Understanding these variations can help produce higher quality evidence. Advanced studies of hiberfil.sys can allow the analyst to perform deep forensics, carving evidence which may not have been obtained from forensic tools. Although this research has demonstrated the importance of the hibernation file and the importance of practitioner education, current studies show this topic is underrepresented, conflicting and incomplete.

Memory image is unavailable. There are many reasons a memory image may not be present in an investigation. In addition to first responders not collecting volatile data, a system may have already been powered off or locked when responders arrived on the scene. Memory acquisition could also fail for any number of reasons, or the image could become corrupt and unusable (Lee, 2014a). Without a memory image, investigators lose valuable evidence such as Internet history which was cleared or current session usernames and passwords. Encryption keys for full or partial disk encryption may also be extracted from memory (Casey, 2010). In the absence of a current memory image, the hibernation file may be the only access an investigator has to this valuable evidence.

Hibernation file as a supplement to a memory image. The hibernation file may be a valuable source of evidence even when a current memory acquisition is available because it offers evidence from another point in time. One example would be if a suspect were web browsing the last time their system went into hibernation and had later wiped the Internet history before a memory image was acquired. In this case, the hibernation file would contain web browsing history and the memory acquisition would not (Beverly, Garfinkel, & Cardwell, 2011). Another benefit to the hibernation file is that having two or more versions of memory to analyze can be useful in determining changes in the systems state (Lee, 2014a). Having a historical snapshot of memory increases the chances of finding evidence which was not contained in the current image.

Archived hibernation files. When a system hibernates, the contents of memory are written to hiberfil.sys. Hiberfil.sys is overwritten each time the system hibernates so that just one hibernation file will be present on the system (Carvey, 2007). System backups such as restore points, volume shadow copies and other external backups can include archived copies of the hibernation file. There are many reasons why memory data from additional points in history could be useful in an investigation. One example involves a user who closes their laptop every day for lunch, putting the system in hibernation mode. If that system has daily backups enabled, several days' worth of memory may be available. This scenario could prove invaluable in a case where the user wipes their history at the end of each day. It could also identify when a malicious process first showed up in memory (Ligh, Adair, Hartstein, & Richard, 2011). Historical memory data is useful in: identifying user behavior patterns; increasing the chance to find a particular credential set; narrowing a timeline; identification of system compromise; changes to system state and many other uses.

Hibernation and malware analysis. The hibernation file can be instrumental in identifying malware. The expanding size of memory which now comes standard on most systems has transformed memory into more of a secondary drive. The increased space gives malware the capability to download, extract and run all inside of memory. Memory is also less volatile than it was in the past which provides a more secure home for a malicious process. Some malware are designed to run in memory as an attempt at stealth leaving no traces of itself on the hard drive (Ligh et al., 2011). Incident responders may be investigating an intrusion after malware has been cleaned from a system. The hibernation file may have captured the malicious processes before it was removed. Often analysts need to acquire malware, particularly in zero-day advanced persistent threat (APT) attacks, in order to identify the malwares actions and what information the attackers may have been after. Examining hibernation files from system backups increases the chances of finding the malware and may aid in refining the incident timeline (Lee, 2014a). The hibernation file may also be useful in confirming whether a system was compromised. For example, a user under investigation for policy violation network traffic may be wrongfully accused if the malware which caused the traffic is not identified.

Extracting login credentials from the hibernation file. The hibernation file can be used to extract passwords for social media, online accounts and email. When a user enters their credentials into a local program or web-based account, the authentication data is stored, often in plain text, in memory (Anson, Bunting, Johnson, & Pearson, 2012). If the system hibernates during a login session, the hibernation file will contain this authorization data (Forensicmag.com, 2013). This evidence could be exceptionally valuable giving an investigator access to user accounts, providing their warrant grants permission to view these accounts (Brunty & Helenek, 2013). Even in cases where the investigator is not authorized access, the credentials retrieved can be useful in creating a keyword list for brute force attempts into other login sessions.

Using the hibernation file for decryption. In addition to login credentials, the hibernation file also can contain encryption keys for both full and partial disk encryption as long the encrypted container was open at the time of hibernation (Craiger, Pollitt, Swauger, 2005). Full disk encryption typically means system files are encrypted. However, the hibernation file is one of the exceptions to full disk encryption. In Windows XP particularly, the hibernation file could not be encrypted at all (Morello, 2007). For other versions of Windows, Microsoft does not provide third party developers with the ability to encrypt hiberfil.sys (Suiche, 2008a). Unless the system is Windows Vista or later and is encrypted with Microsoft's BitLocker, the hibernation file may be unencrypted and may contain the full disk encryption keys. For partial encryption, system backups can be leveraged to find a hibernation file created when the container was unlocked.

Understanding the hibernation feature. Hibernation mode was first seen in Windows 2000 and has since evolved and changed with each consecutive version of Windows. Hibernation mode, designed for laptops, has some significant differences from hybrid sleep mode designed for desktop systems. The differences in functionality between Windows versions and hardware configurations have significant impacts on the hibernation file. The factors impacted include file size, compression, software compatibility and encryption (Suiche, 2008a, 2008b, 2008c, 2010). The hibernation file is also configurable by the user and changes to these settings could impact analysis (Postinge, 2011). When an analyst understands the variations in the hibernation file and how these variations affect analysis, they can better extract complete and accurate evidence.

The hibernation file in deep forensics. The hibernation file is a complex proprietary format and current publications rely on its reverse engineering (Diablohorn, 2014; Kleissner, 2009a, 2009b; Suiche, 2008a, 2008b, 2008c, 2010). Studying advanced topics such as hibernation slack, which is slack space within the actual file, can provide an analyst with the ability to perform deep forensics and obtain evidence not typically retrieved from standard forensic methods. Advanced studies include an in-depth knowledge of the files format along with the variations between hibernation versions. While these topics are not required to obtain evidence from hiberfil.sys, some of the concepts researched brought into question the accuracy of evidence particularly when dealing with carved data (McCash, 2014). These advanced topics should be explored by analysts using hibernation artifacts carved during a deep forensics examination.

Understanding the hibernation file and best practices for analysis can improve on evidence quality. Additionally, in-depth knowledge of the hibernation file and available forensic technology can help prevent erroneous conclusions. Forensic professionals who blindly trust tools, without understanding the data being examined or how the tools function, can potentially have damaged, incorrect, or incomplete evidence (Ligh, Case, Levy, & Walters, 2014). The better educated an analyst, the better quality evidence they will produce. More research on hibernation forensics is required because the current studies are incomplete.

Deficiencies in What We Know

Hibernation forensics has come a long way in recent years but there is still much which remains unknown. The practice of memory forensics has grown tremendously over the past years. However, there are still many incident responders not trained to acquire a memory image. Many tools are now available for hibernation forensics; however a practitioner can improve the quality of their completed analysis with further education. Most analysts do not possess the time or resources to examine existing technology and would benefit from a comparative study. Another comparative study into the differing versions of the hibernation file would also profit the industry. Past publications on the hibernation file may be accurate for one version and inaccurate for another. These studies would require more research into the hibernation file format.

First responders and memory images. Memory forensics has come a long way in recent years, however not all forensic professionals make use of memory data. In fact, just a few years ago, the United States Secret Service and Department of Homeland Security's Best Practices for Seizing Electronic Evidence, included instructions to "pull the plug" from a running system (US Dept of Homeland Security, 2007). This method eradicates data contained in memory. Memory forensics has since become a more common practice and many investigators now acquire volatile data before pulling the plug. However, there are still some cases, such as Law Enforcement, where first responders may not have digital forensic training or the skills to acquire memory (Lee, 2014a). In the absence of a memory image, forensic professionals knowledgeable in hibernation forensics may still be able to benefit from memory data.

Hibernation forensic tools. Forensic tools allow analysts to acquire evidence without the need to fully understand the file being analyzed. These tools are available in a broad range of features and required skill level. An all-in-one solution such as AccessData's FTK may be able to parse the data from the hibernation file and present it to the user in an easily searchable format. This does not require the analyst to be educated in hibernation forensics. In fact, in the five hundred and twenty pages of the FTK User Guide, the hibernation file is not mentioned once (AccessData, 2014). An analyst may be able to identify a half-written email from the contents of the hibernation file using such a tool, but without understanding that it came from a previous memory capture and may or may not have been sent to the recipient. This artifact may have been pulled from hibernation slack and may have actually have been captured at an unidentified previous time. Appraising the value of evidence obtained from forensic tools become more accurate the more knowledgeable an analyst is.

Study of current forensic technologies. Several tools are available for examining the hibernation file. As of this writing, research did not locate any comparative studies which include cost, ease of use, time, immiscibility in court or usefulness of evidence. Analysts typically will not have time to test all of the tools available in order to determine the best one for their situation. A baseline comparison can drastically help a forensic professional in choosing the most efficient tool. It is important not only to choose the best tool but also to be familiar with other tools in case the analyst's first option fails or is unavailable. In the study guide provided by SANS for the GIAC Certified Forensic Analyst (GCFA) certification, Lee states "I have encountered multiple hibernation images that were not able to be successfully converted (or recognized) by Volatility. Remember the previous tool covered, hibr2bin.exe can be used as a potential backup" (Lee, 2014a, p. 127). It is important that an analyst have knowledge and access to more than one hibernation forensic tool.

Identifying variations in the hibernation file. The hibernation file has evolved with Windows operating systems. These changes can cause confusion when researching the hibernation file because a published work may be accurate for one hibernation version but not another. McCash of SANS noted that changes to the hibernation feature in Windows 7 resulted in facts published in a previous work being "manifestly untrue on the system I was investigating and on several other Win7 systems I checked as well" (McCash, 2014, para. 3). When Microsoft released Windows 8, a major change was seen to the compression format of hiberfil.sys (Ligh et al., 2014). Changes in the hibernation file need to be understood along with how they interact with current tools to maximize the accuracy in analysis.

Hibernation file binary structure. Having an advanced knowledge of the hibernation file includes an understanding of the files binary structure. Much of the published works on the hiberfil.sys do not include details of the files format. The problem is likely due to the hibernation file being compressed with a "proprietary format which has long impeded thorough understanding and applicability in forensic investigations" (Stefan & Freiling, 2011, p. 11). The current knowledge available on hibernation file format is the result of reverse engineering. The published data from this reverse engineering documents the features of the files structure but does not fully define each variable. Additionally, the reverse engineer acknowledges that hiberfil.sys is seen in several versions with structural differences (Suiche, 2008a, 2008b, 2008c). These discrepancies are not fully documented.

Defining the Audience

The intended audience for this research includes forensic, data recovery and security professionals, first responders and system administrators. Education in hibernation forensics increases the effectiveness of investigations and quality of evidence practitioners can produce. Security professionals benefit from this education as well, paticularly in response to security events involving malware which resides in memory. Incident responders with familiarity of the hibernation file may have circumstances when forcing a system into hibernation may be better than pulling the plug. Data recovery professionals can make use of the hibernation file when tasked with recovering an encrypted partition or in attempting to recover unsaved documents. System administrators should understand the hibernation files value when choosing to enable or disable the hibernation feature on a system.

LITERATURE REVIEW

Introduction

This literature review is intended to provide an in-depth study of the hibernation file and its role in forensics. This section contains an examination of the hibernation file format along with numerous changes to the hibernation feature in successive Windows versions. Several important concepts regarding the hibernation feature are covered followed by forensic uses of the hibernation file. This literature review concludes with a review of several forensic tools and considerations for selecting forensic technologies.

Windows Hibernation Mode

Hibernation mode is Microsoft's suspend to disk feature. Hibernation is a power conservation feature built into Windows operating systems beginning with Windows 2000. Prior to Windows 2000, suspend to disk was offered only as a hardware solution. The advancement of smart battery technology led to the implementation of the hibernation mode in Windows (Schwarz, 1998). In order to put a laptop on minimal power consumption, power to the memory must be cut. Cutting the power to memory, results in the loss of stored data. Hibernation mode copies the contents of memory to the hibernation file prior to entering the lowered power state. The system copies the data in the hibernation file back to the memory when it resumes, thus restoring the system to its previous state (Microsoft, 2015c). This feature is not exclusive to Microsoft. Other operating systems including mobile devices have their own suspend to disk features; however the scope of this study is limited to the Windows hibernation mode.

Hibernation File Basics

The hibernation file, named hiberfil.sys, is a binary file located in the root directory. The file format for hiberfil.sys is a Microsoft proprietary compression whose details have only been made available through reverse engineering (Suiche, 2008a, 2008b, 2008c, 2010). The file is also a protected, meaning it will only appear in Windows Explorer when the default folder options are changed to show both hidden and protected files (Microsoft, 2015d). The file size varies between operating systems and can either be the same size as memory with a maximum file size limit or compressed up to fifty percent of memory size with no limitation (Postinge, 2011). Only one copy of the hibernation file is maintained by Windows in the root directory. When hibernation mode is enabled, hiberfil.sys is created. When the system hibernates, the contents of memory are saved to hiberfil.sys and are later overwritten with successive hibernation. These features are explored in more depth later in this section.

Hibernation File Format

A thorough understanding of the file format for hiberfil.sys is not required to perform hibernation forensics. However, for forensic professionals interested in deep forensics this section reviews the binary structure of hiberfil.sys. Due to its proprietary nature, Microsoft has not supplied details on the hibernation files format. Hiberfil.sys was reverse engineered by Suiche, who has provided several publications regarding its structure (Suiche, 2008a, 2008b, 2008c, 2010). The ability to manually parse and examine the hibernation file can assist analysts who need to examine corrupted, damaged or partial sections of the hibernation file. The binary format of hiberfil.sys varies between operating systems and hardware architecture. A comprehensive study of the different versions of hiberfil.sys was not located. The research in this section covers the major concepts of the files format. Further research is needed to account for each version of hiberfil.sys.

Hiberfil.sys binary structure. The hibernation file is referenced as a “file”. However it may be better defined as a volume. “A volume is a collection of addressable sectors that an operating system (OS) or application can use for data storage" (Carrier, 2005, p. 70). The hibernation file is segmented into 4096 byte sections or pages. Memory contents are stored in the file in blocks of data called Xpress image blocks. The first page of hiberfil.sys contains the header. The second page contains the processor state and the third page is where the reserved memory map begins. The remaining pages of the file house the Xpress image blocks which are organized by memory tables. The exact organization of these blocks differs with Windows versions (Kleissner, 2009b). The hibernation files top level outline is shown in Table 1: Hibernation File Pages. Each of these segments is explained in more detail later in this section.

Hibernation File Pages			
Page	**First Byte**	**Ending Byte**	**File Section**
0	0	4095	Header
1	4096	8191	Processor State
2 - n	8192	EOF	Reserved Memory Map Memory_Tables Xpress Image Blocks

Table 1: Hibernation File Pages (Kleissner, 2009b)

Hiberfil.sys header. The first page of hiberfil.sys, page zero, contains the file header, PO_MEMORY_IMAGE. The header contains valuable information, however unless acquisition occurs while the system is hibernating, this data will not be available because it is zeroed out when the system resumes (Diablohorn, 2014). This provides a challenge for forensic tools because they must be able to parse the data in the hibernation file without the benefit of the header. The headers contents are displayed in Table 2: Hibernation File Header. Most of the variables within the header structure were not defined in existing publications and require further research.

Hibernation File Header - PO_MEMORY_IMAGE		
Byte Offset	**Structure**	**Attributes**
0x0	Signature	[‘String’, {‘length’: 4}] (PostVista) ‘WAKE’ ‘HIBR’ ‘RSTR’ (PreVista) ‘wake’ ‘hibr’ ‘RSTR’
0x4	Version	[‘unassigned long’]
0x8	CheckSum	[‘unassigned long’] Set to 0 = MS Boot Loader does not check compressed pages.
0xc	LengthSelf	[‘unassigned long’]
0x10	PageSelf	[‘unassigned long’]
0x14	PageSize	[‘unassigned long’]
0x18	SystemType (ImageType)	[‘unassigned long’]
0x20	SystemTime	[‘winTimeStamp’, {}]
0x28	InterruptTime	[‘unassigned long long’]
0x30	FeatureFlags	[‘unassigned long’]
0x34	HiberFlags	[‘unassigned char’]
0x35	Spare	[array’, 3, [‘unassigned char’]]
0x38	NoHiberPtes	[‘unassigned long’]
0x3c	HiberVa	[‘unassigned long’]
0x40	HiberPte	[‘_LARGE_INTEGER’]
0x48	NoFreePages	[‘unassigned long’]
0x4c	FreeMapCheck	[‘unassigned long’]
0x50	WakeCheck	[‘unassigned long’]
0x54	TotalPages	[‘unassigned long’]
0x58	FirstTablePage	[‘unassigned long’] Pointer to the first memory table
0x5c	LastFilePage	[‘unassigned long’]
0x60	PerfInfo	[‘_PO_HIBER_PERF’] Introduced in Windows XP
	NoBootLoaderLogPages	Introduced in Windows Vista
	BootLoaderLogPages [8]	Introduced in Windows Vista
	TotalPhysicalMemoryCount	

Table 2: Hibernation File Header (Hale, 2012; Kleissner, 2009b)

Header signature. The first four bytes of the hibernation files header contain the signature indicating whether the file is hibernating, restored or in the process of waking. In Windows 2000 through XP, the signatures used lowercase letters. Windows Vista through 8 used uppercase letters. This four-byte segment is set to “hibr” or “HIBR” when the system is hibernating. This flag prompts the system to copy the contents of hiberfil.sys to memory upon waking. During the time when the contents are being copied back to memory, the signature changes to “rstr” or “RSTR”. Once the system is restored, the signature is set to “wake” or “WAKE”. The wake flag prompts the system to start up fresh for future boots (Kleissner, 2009a, 2009b). The signature is the only portion of the header which is not zeroed out when the system resumes and can be useful to identify what in what state the system was in at the time of acquisition.

Other contents in the header. In addition to the signature, the header contains information on the system version and time. It also contains relevant information regarding control registers useful for memory management functions such as virtual address translation. The header contains previous Enterprise Information Portals, the Interrupt Descriptor Table base address and the Global Descriptor Table base address (Suiche, 2008a, 2008c). When it comes to manually parsing the contents of the hibernation file, the FirstTablePage is crucial as it points to the location of the first memory table (Diablohorn, 2014). Although the FirstTablePage could be very useful, it will only be available if the hibernation file is acquired when the system is hibernating.

Hiberfil.sys processor state. The second page of hiberfil.sys is the processor state. This page begins at byte offset 4096 and continues to 8191. This structure is platform specific. The processor state contains a set of kernel context and state registers (Kleissner, 2009b). This structure is exported in Windows debugging symbols and contains the GDT and IDT offsets along with control registers CR0 and CR3 (Suiche, 2008c). This portion of the hibernation file could provide valuable information about the registry.

How data is stored in hiberfil.sys. It is important to understand how data is written to the hibernation file utilizing the reserved memory map and memory tables. Current research does not detail the exact procedure for writing data to the hibernation file but it stresses that the data is not stored consecutively within the files pages. A large segment of data (over 4096 bytes) would logically require more than one page for storage. Likewise, several smaller blocks of data may be stored in the same page. Pages can contain a maximum of 255 Xpress image blocks (Suiche, 2008b). In order to organize this data, the hibernation file has a reserved memory map which keeps track of the available and reserved pages. The data is broken into Xpress blocks and stored in pages with open space. Each page contains its own memory tables to keep track of the Xpress blocks within (Kleissner, 2009). These segments of the hibernation file are explained in more detail below.

Reserved memory map. The reserved memory map begins at the third page of hiberfil.sys at byte offset 8192. This structure is PO_MEMORY_RANGE_ARRAY and it lists the available and reserved pages. The contents are exported in debugging symbols and vary across Windows platforms (Suiche, 2008c). This structure is also instrumental in reloading data into memory (Kleissner, 2009b). Essentially, this map tells the operating system what pages can be used to store data and it maps the data to be restored back to memory. This structure should not be confused with the memory tables which are maintained by each page.

Memory table. Since the data from memory is not written to the hibernation file in a consecutive order, memory tables are used to point to the next location of a data set. Each page contains its own memory table. The memory table construction is shown in Table 3: Hibernation Memory Table. As mentioned previously, each page can contain as many as 255 Xpress block entries. The memory table EntryCount tracks the number of Xpress blocks entries within the page. The NextTablePage variable points to the next page for consecutive data. For each block of data, there is a MemoryTableEntries segment which gives the physical start and end location for that block (Kleissner, 2009a). These tables are used to go from one data set to the next in order for its reconstruction back into memory. The memory tables can be instrumental in deep forensics for a manual reconstruction of a carved hibernation file.

Hibernation Memory Table MEMORY_TABLE		
Section	**Subsection**	**Type**
PointerSystemTable	-	DWORD
NextTablePage	-	UINT32
CheckSum	-	DWORD
EntryCount = 255	-	UINT32
MemoryTableEntries[EntryCount]	PageCompressedData	UINT32
	PhysicalStartPage (Location of the memory range)	UINT32
	PhysicalEndPage (Location of the memory range)	UINT32
	CheckSum	DWORD

Table 3: Hibernation Memory Table (Kleissner, 2009a)

Xpress memory blocks. System memory is compressed and stored in blocks using the Xpress algorithm. A single uncompressed Xpress block is sixty-four kilobytes and spans sixteen uncompressed pages (Suiche, 2008b). The Xpress block header contains a signature followed by the number of uncompressed pages, compressed size and the reserved field. This structure is shown in Table 4: Hibernation Xpress Block Header. Carving tools can utilize the Xpress header signature to extract and reconstruct the hibernation file (Diablohorn, 2014). An additional important note is that this file format varies. More research is needed in order to fully define each variable within the file and to differentiate between the hibernation versions.

Hibernation Xpress Block Header IMAGE_EXPRESS_HEADER		
Section	**Type**	**Attribute**
Signature	CHAR	81h, 81h, “xpress”
UncompresssedPages	BYTE	16 pages per image
CompressedSize	UINT32	Size of the block
Reserved	BYTE	

Table 4: Hibernation Xpress Block Header (Kleissner, 2009a)

Evolution of Hiberfil.sys in Windows

Suspend to disk feature began as a hardware only feature. The Windows software solution, hibernation mode, was first introduced in Windows 2000. Each version of Windows, from 2000 through 8, brought some change to the hibernation mode. Although Microsoft provides many details about hibernation mode, they have not provided details on the compressed proprietary algorithm. Current research relies on publications from reverse engineers. This section will outline some of the differences which have been published.

Suspend to disk hardware solution. Suspend to disk feature existed as a hardware solution prior to the introduction of hibernation mode in Windows 2000. System OEM's provided drivers to suspend to disk when a battery was low. In the mid-1990's, technology could not reliably predict when a battery was drained and suspend to disk occurred at around twenty-five to thirty percent battery life. By 1998, smart batteries could detect battery life to plus or minus one percent. However, Windows 98 did not support these drivers (Schwartz, 1998). Windows 2000 was the first to introduced the hibernation mode and discontinued its support of suspend to disk feature (Microsoft, 2015a). Hibernation mode has since evolved with each release of Windows.

Compression format. The hibernation file used the Basic Xpress compression format in Windows 2000 and continued with this format through Windows 7. In Windows 8, the compression format included Huffman and LZ encoding in addition to Xpress (Ligh et al., 2014). It is important to note that the change in compression format in Windows 8 affects the compatibility of hibernation forensic tools.

File size. The hibernation file in Windows 2000 through Vista required disk space equal to the size of the systems memory and was limited to four gigabytes. This limitation coincided with the supported memory in thirty-two bit systems. However, Windows XP and later, sixty-four bit systems, supported memory in excess of four gigabytes (Microsoft, 2015b). The operating system would detect if more than four gigabytes of memory was present and would suppress the hibernate tab in power options (Microsoft, 2015e). In Windows 7, this size limitation was removed and compression was introduced. By default, in Windows 7 and later systems, hiberfil.sys is compressed to seventy-five percent of memory capacity. A system with eight gigabytes of memory will have a six gigabyte hiberfil.sys file. These compression settings can be modified, however if misconfigured can result in hibernation failure (Microsoft, 2009). In current Windows 7 and 8 systems, it is common for memory to be much larger than four gigabytes resulting in much more extensive hibernation files sizes.

Hybrid sleep. Hybrid sleep is a feature designed for desktop computers. It was first introduced in Windows Vista and remains available through Windows 8 (Microsoft, 2013). Without battery concerns, a desktop's power can be significantly reduced without cutting power to the memory which eliminates the need to copy the memory's full contents to the hibernation file. It is unclear what specifically is copied to hiberfil.sys in hybrid sleep mode. The Microsoft website only explicitly states that open documents and program data are copied to the hibernation file (Microsoft, 2015c). A conflicting statement by Bunting says, "if a system is placed in hibernation or hybrid sleep with Windows Vista, the contents of RAM are written to the hiberfil.sys file" (Bunting, 2008, p. 435). No existing studies were found which explicitly define the differences between these two features. Hybrid sleep conserves power but its primary function is to provide data preservation in the event of a power failure. Although the data saved from hybrid sleep may or may not be as comprehensive as hibernation, the file could still contain valuable evidence and should not be overlooked in desktops.

Hiberfil.sys binary structure discrepancies. The hibernation file has several different versions between operating system releases and hardware architecture. Between Windows XP and 7, Suiche reported he found seven to eight different versions of hiberfil.sys (Suiche, 2010). There is currently no source for comparing each of these versions. However, a few significant differences are documented. One of the largest differences is the file format structure. Prior to Windows Vista, the hibernation file was structured beginning with the header, followed by the memory array table then the processor state. Windows Vista and later versions formatted the file starting with the header, processor state then memory array tables. Another variation is in the header structures. Windows XP introduced the HIBER_PERF structure into the header and Windows Vista introduced BootLoaderLog fields and removed the ImageType section (Suiche, 2008b). Signature bytes also saw changes. Windows Vista and later versions switched to capitalized letters in the signature bytes; ‘wake’ became ‘WAKE’ and ‘hibr’ became ‘HIBR’ (Kleissner, 2009b). While these discrepancies are far from comprehensive, it is important to note that there are many differing versions which could lead to inaccurate or incomplete data.

Important Concepts of Hibernation Mode

Several concepts regarding the hibernation file are significant in forensic investigations. This section will cover whether the hibernation file contains a full memory image, reasons hiberfil.sys may be missing from a system, how changing the compression may affect hibernation, wake events and hibernation slack space.

Memory image verses the hibernation file. There are conflicting reports as to whether the hibernation file contains the complete content of memory. Many of the researched works state the hibernation file is an entire memory image created when the system last went into hibernation mode (Anson et al., 2012; Bunting, 2008; Carvey, 2009; Hale, 2012; Lee, 2014a;

Suiche 2008c). Some statements were explicit in their meaning. “It turns out that “hiberfil.sys” is a complete copy of everything in RAM when that lid was closed” (Lee, 2014a, p. 11). Other statements were more ambiguous. “These files are basically the compressed contents of Windows memory from when the system (usually a laptop) “goes to sleep”” (Carvey, 2012, p. 101). However, there are contradictions to these statements claiming that systems with a DHCP configuration will release any active connections before a system hibernates resulting in incomplete network data within the hibernation file. There may also be a difference between a memory capture and the hibernation file in regards to malware. Advanced malware which can detect hibernation may be programmed to delete itself from memory before the system hibernates (Ligh et al., 2014). It is unclear whether these conflicting statements are caused by differences in particular versions of the hibernation file.

Hibernation file not found. There could be many reasons why the hibernation file does not appear on a system. The first and most plain reason will be if hibernation mode is disabled. If a system had hibernation enabled in the past and had hibernated at some point, the data may be carved from the system slack space. This is because disabling the file does not zero it out but had merely unallocated the space. Hiberfil.sys is not a preinstalled system file but is created when hibernation is enabled for the first time. Because hibernation is enabled by default on most systems, the file should be created the first time a system is booted up. Another reason for the hibernation files absence could be that the memory is larger than four gigabytes in a Windows Vista or earlier system (Microsoft, 2015e). There may also have been anti-forensic tools deployed to wipe the hibernation file.

Hibernation compression. In a Windows 7 system, the default hibernation file size is 75% of the systems memory capacity. This compression ratio can be configured by the user ranging from fifty to one hundred percent. Changing the hibernation file size should be done with care so as not to cause a failure due to insufficient space (Microsoft, 2009). The compression ratio of the hibernation file may also affect forensic tools. In the event of a failed acquisition, conversion or analysis, the analyst should check if the compression ratio settings was altered. The analyst should investigate whether the selected forensic technology automatically detects and accommodates for these changes.

Hibernation wake events. Hibernating systems can wake upon events other than user interaction. The Wake-on-LAN (WOL) feature will wake a hibernating system for particular Ethernet events. An Ethernet packet can be specially designed to wake a hibernating system (Microsoft, 2013). This concept should be considered especially in reviewing an event log timeline. A system waking from hibernation does not automatically mean it was a user-initiated action. Further research on this topic is required including how the hibernation file is affected by WOL.

Hibernation slack space. Hiberfil.sys contains its own slack space. Hibernation slack space is different from the system slack space, as it is built into the file. There is conflicting research as to how data is written to the hibernation file and how it relates to slack space. The concept of slack space in a hibernation file is important because it can lead to inaccurate evidence. An analyst who utilizes the hibernation file should be aware of hibernation slack space and how their forensic tools deal with it. Useful evidence can be extracted from hibernation slack space but it is imperative the analyst be able to differentiate between evidence from the current hibernation and remnants of the past.

When hibernation is first enabled, hiberfil.sys is created at its maximum size. The EnCase Computer Forensics official study guide indicates that if a system has never been in hibernation mode, the hiberfil.sys file will be the same size as memory but filled with zeros (Bunting, 2008). This information was based on hibernation prior to Windows 7. When a Windows 7 system was tested by McCash of SANS Institute, he found that the hiberfil.sys created from a freshly installed operating system contained data from a hibernation file prior to the operating system installation. These results showed two points, first, in Windows 7 the hibernation file only allocates the file size but does not overwrite the space with zeros as stated by Bunting. The second point was speculation that when Windows operating system is reinstalled, without performing a zero wipe, the hibernation file is created in the same physical location (McCash, 2014). This point could be crucial as it suggests that if a company's practice is to reformat a system without zero wiping the drive, the hibernation file could potentially contain evidence from a previous employee.

Diablohorn performed further testing of slack space in the hibernation file. This trial involved parsing and examining a Windows 7 hibernation file. Utilizing the Xpress block headers, the end position of the final Xpress block was located in the current hibernation file. The remaining space was identified as slack space. Searching this space for Xpress block signatures can reveal data from a previous hibernation where more of the systems memory was utilized (Diablohorn, 2014). Diablohorn did not indicate in his work if he found data in the slack space he examined. Both McCash and Diablohorn's work on hibernation slack is enlightening, however, there is still a need for a scholarly study of this topic.

The research into hibernation slack was limited to Windows 7 systems. It is unknown if Windows Vista or earlier systems contain hibernation slack. Research shows that the hibernation file is not erased from the system but merely unallocated in these Windows versions (Suiche, 2008c). This may indicate Windows 2000 through Vista hibernation files could also contain slack space. Slack space within the hibernation file could cause inaccuracies in evidence. For example, during a forensic analysis of a system utilized by more than one user, the analyst finds particular Internet history artifacts in the hiberfil.sys slack space. It would be unknown from which users those artifacts came. Likewise, an employee granted personal use of their work laptop after hours could be wrongfully accused of playing video games at work based on the contents found in hibernation slack. The evidence in hibernation slack could be extremely useful. However it is imperative that the analyst be able to differentiate what evidence came from the current hibernation image and what was carved from the files slack. An analyst should know if their forensic tools are providing just the most recent hibernation data or it evidence is being carved from slack.

Hibernation File in Forensics, Incident Response and Data Recovery

There are numerous applications for the hibernation file in forensics and incident response. This section will cover malware which hides in memory and how the hibernation file can be used for detection. Utilizing the hibernation file to obtain encryption keys from both full and partial encrypted systems is covered in addition to retrieving passwords and leveraging system backups for historical copies of the hibernation file. 0

Malware analysis. The hibernation file can be useful in malware analysis. Malware which hides in memory is becoming more common and poses a problem for incident responders due to the lack of evidence left on the hard drive. This type of malware has been around for several years. In fact, the SQL Slammer worm seen in 2003 was not only one of the fastest spreading but was also memory-only malware. More recently memory-only malware was seen in the 2013 Target breach which resulted in the theft of more than forty million credit cards. Although harder to detect, memory-only malware is typically easier to clean because a reboot will usually remove the infection (Grimes, 2014). While removing the infection is important, it may be crucial for incident responders be able to reverse engineer the malware in order to help assess the scope of the breach or to identify the attackers. With the volatile nature of memory-only malware, often an incident responder may find the malware no longer resides on a system by the time they begin their investigation. The hibernation file, however, may have captured the malicious process.

Rootkits. Rootkits are a type of malware which attempt to hide resources in memory in order to avoid detection. They are commonly paired with other malware such as backdoors. Rootkits manipulate existing services or create rogue services in order to avoid detection and run with higher privileges (Sikorski & Honig, 2012). API hooking is a method which rootkits use to hide processes, files, registry entries and network connections. There are several API hooking methods rootkits utilize including: Inline API hooks; Import Address Table; Export Address Table; Interrupt Descriptor Table Hooks; Driver I/O Request Packet Hooks; and System Service Descriptor Table (Ligh et al., 2011). The hibernation file can preserve the malicious or altered processes, files, registry and network artifacts providing a malware analyst the ability to evaluate the malware.

Memory-only malware artifacts. Memory-only malware uses several techniques to avoid detection. One technique is to manipulate or stop existing services or create rogue ones. Services are typically non-interactive and continually run in the background with elevated privileges making them a target for malware. Malware can also write a driver to the kernel and hide in kernel memory by unlinking the thread to the loaded module list. This creates survivability for the malware and results in detached or orphaned threads. Malware typically contains network capabilities such as the ability to contact a command and control server or to download additional malware from the Internet (Ligh et al., 2011). The hibernation file may contain artifacts to help identify memory-only malware such as the detached and orphaned threads, API functions, service lists and network connections.

Hibernation and encryption. A key challenge forensic professional's face is encryption. Whether it is a full disk, partial disk or a single encrypted file, the encrypted container may possess vital evidence which is not accessible to the analyst. Encryption is one of the major reasons why pulling the plug on a running system is no longer considered best practice. If an encrypted system is powered down, an analyst cannot access the data on the file without the encryption key (Reyes, 2007). Encryption keys are stored in memory meaning that the hibernation file will also contain the keys for any encrypted container open at the time of hibernation.

Full disk encryption. Full disk encryption can prevent an analyst access to a hard drive. True "full" disk encryption encrypts the entire drive including system files. The hibernation file may be able to be utilized to acquire encryption keys even from whole disk encrypted systems. Depending on the operating system and the encryption program, it is possible that the hibernation file is not encrypted. As of 2008, Microsoft did not supply a method for third party developers to encrypt the hibernation file resulting in Microsoft's BitLocker being the only software capable of encrypting the hibernation file (Suiche, 2008a). However, not even BitLocker could encrypt the hibernation file in Windows XP. Morello plainly states "Like the page file, the hibernation file cannot be encrypted in Windows XP" (Morello, 2007, Enabling EFS, para. 7). Windows XP systems and those encrypted with third party software are likely to contain an unencrypted copy of the hibernation file.

Partial disk encryption. Partial disk encryption typically just includes the data portion of a drive, a particular volume, folder or single file. System files including the hibernation file are not included in partial disk encryption allowing an analyst to extract encryption keys (Lowman, 2010). The caveat for examining the hibernation file in hopes of retrieving a partial encryption key is that the encrypted container must have been unlocked at the time of hibernation.

Exposing login credentials. Login credentials for websites and programs are stored in memory, often in plain text. Like with partial encryption keys, these credentials are stored only when the login session is active. A memory image may contain the user's credentials to a particular online account and the hibernation file may hold credentials to other accounts. Because of this, examining every available memory dump can maximize the credentials which can be found. If the forensic examiner is unable to find a particular credential set, the combined data from other logins may be leveraged to help guess a password or to create a keyword list for brute force attempts. Additionally, finding the login to one account may assist the analyst in unlocking other accounts through a forgotten password feature.

Benefits from archived hibernation files. The hibernation file can be extracted from system backups. System restore points, volume shadow copies and other external full system backups can be examined for historical copies of hiberfil.sys. These additional memory dumps could provide a great deal of insight into past events. "Sometimes I have several historical memory images that I can perform differential analysis on. This may help determine when a compromise occurred, particularly if anti-forensics techniques were employed to destroy timestamps on the disk" (Williams, 2013, Memory image file formats, para. 4). As discussed earlier, the more memory dumps available, the more chances an analyst will have to find a particular piece of evidence or login credentials. Historical memory images can also help identify events to improve the timeline accuracy.

Restore points. The Windows XP operating system has the restore point feature which is a regular system backup. This feature is enabled by default and creates a system backup every twenty-four hours, normally at night. Restore points are also generated at software installations and system updates. Restore points typically use twelve percent of the hard drive space and are kept as long as there is room on the disk. Once the twelve percent is reached, the oldest backups are overwritten with new ones (Lee, 2014b). An analyst would need to utilize a forensic tool which can extract the hibernation files from the restore points.

Volume shadow copy. Volume shadow copies replaced restore points in Windows Vista through Windows 8. The volume shadow copy service creates backups of the system and is enabled by default. Windows Vista default settings are to take a snapshot once a day. In Windows 7, the default setting is once every seven days. The time of day varies because the copy is made when the system is idle for a set time or is shutdown. Volume shadow copies are also triggered by software installations (Whitfield, 2010). Creation time of the volume shadow copy is of interest while looking at historical copies of the hibernation file. The volume shadow copy service is set to backup after a set idle time on a particular day. An idle system will also hibernate after a set time which is customizable by the user. It was unclear as to whether or not a system would backup if the hibernation settings are scheduled to activate earlier.

Considerations for Hibernation File Analysis

In this section, special considerations for acquiring and examining the hibernation file are discussed. These considerations include the state of the header at the time of acquisition and how it affects analysis. Another consideration is the limitations created by the protected state of the hibernation file and its proprietary format. These topics should be considered when choosing a hibernation forensic tool.

Acquisition of the hibernation file from a forensic disk image. Acquiring hiberfil.sys from a forensic disk image merely requires the mounting of the disk image and copying the file from the mounted drive (Hale, 2012). However, if the system was not hibernating at the time of acquisition, the header will be zeroed out and data needed to reconstruct the image will be missing (Ligh et al., 2014). For this reason, forensic software have been developed to rebuild the contents of the hibernation file without the header data.

Acquisition of the hibernation file from a live system. Acquiring the hibernation file from a live system is more complex because the file cannot just be copied. With hibernation mode active, hiberfil.sys is present in the root directory and can be seen in Windows Explorer if the options to show hidden and protected files are enabled. Because it is a protected system file, copy and paste functionality is blocked. Even with administrative privileges, the file cannot just be copied. The analyst will need to use a forensic tool designed to acquire the hibernation file. When hibernation mode is disabled, the file no longer exists in the file system and its space is unallocated. The file data is not overwritten and remains on the hard drive but will require a carving tool to acquire it.

Analyzing the hibernation file. The hibernation file is a proprietary file format which requires a tool capable of decompressing and converting the file into a readable format (Microsoft, 2015d). Thanks to the reverse engineering of the hibernation, many forensic tools are now available which are capable of both analyzing hiberfil.sys and converting it into a raw memory image. The raw memory format is the most widely supported format for current forensic tools (Ligh et al., 2014). This conversion allows the analyst to use the memory forensics tool of their choice.

Review of Current Forensic Tools

In this section, forensic tools which can be used to acquire, convert and analyze the hibernation file are reviewed. This research does not cover programs which analyze the hibernation file after it is converted to a raw memory image. Memory forensics is already widely covered and is beyond the scope of this research. The tools included in this section were selected based on price, functionality, user friendliness and the researchers' familiarity with them. These tools should by no means, be considered the best options for examining the hibernation file, nor does their inclusion indicate endorsement.

MoonSols Windows Memory Toolkit. MoonSols Windows Memory Toolkit was designed by Suiche, who was the first to publish his reverse engineering of hiberil.sys. This tool is capable of acquiring the hibernation file from a both a live system or a disk image. MoonSols decompresses and converts the hibernation file into a raw memory image. The image can then be analyzed by MoonSols or any other memory forensic tool. In addition to raw image conversion, MoonSols can also convert hiberfil.sys into a crash dump format (MoonSols, 2015b). MoonSols is capable of extracting orphaned memory chunks from the system slack space (MoonSols, 2013). The current build of the enterprise version, 2.0, has several significant features including a server component which allows users to transmit memory dumps across the network for remote acquisition. Another feature is the inclusion of win32dd.exe and win64dd.exe which can convert the hibernation file into a raw memory dump with a single click (Ligh et al., 2014). These are only some of the key features of MoonSols and merely scratch the surface of this tools capabilities.

This licensed version of MoonSols is compatible with Windows XP through 8, in both thirty-two and sixty-four bit versions. The full enterprise version is available for $7,500. A free version is available with full functionality, however it does not support Windows 8, any sixty-four bit system or product updates (MoonSols, 2015a). MoonSols is a command line tool meaning the analyst would need to be comfortable with the command line. Although the free version is fully functional, its limitation to thirty-two bit Windows 7 or earlier systems may discourage an analyst from taking the time to learn the tool.

Volatility. Volatility is a widely used memory forensics tool. Like MoonSols, it provides numerous functionalities required for examining the hibernation file. The Volatility plugin hibinfo is used to identify the hibernation file format. The imagecopy plugin can be used to convert the hibernation file into a raw memory dump. Once converted, volatility can thoroughly examine the content of the memory image. Volatility is capable of analyzing the hibernation file in its native format as well. Using brute force, Volatility locates the data within the hibernation file, even in the absence of the header data (Hale, 2012). Volatility is already renowned for its abilities in memory forensics and is mentioned in many memory forensic publications. With the addition of hibernation file plugins, Volatility could serve as a standalone tool when it comes to hibernation forensics.

In addition to several non-Windows systems, Volatility supports both thirty-two and sixty-four bit versions of Windows XP through 8 systems (Volatility Foundation, 2014b). Volatility is free to use and free of commercial restrictions. It is also open source allowing users access to its source code (Volatility Foundation, 2014a). Although Volatility is free and very powerful, the technical barrier to entry is its biggest disadvantage. Volatility is a command line tool requiring analysts to have advanced skills. Additionally, each function in Volatility requires the analyst to learn different plugins. Manually using each plugin to extract data can take longer than an all-in-one tool which will extract the evidence and present it to the analyst in an easy to read format. Although it takes longer, an analyst who is proficient with Volatility may be able to extract and understand more evidence from the hibernation file than someone relying on the forensic software to present the evidence to them.

WinHex. WinHex is a hex editor which contains several digital forensics features including the ability to acquire the hibernation file from a live system. WinHex offers a licensed version ranging in price from around fifty U.S. dollars for personal use to thousands of dollars for commercial packages. WinHex takes a snapshot of the content in a live system. The licensed version of WinHex allows the user to copy the hibernation file from that snapshot. The free version will not allow the saving of files larger than 200 KB. WinHex, being a hexadecimal editor, enables the user to view the contents of the hibernation file (X-ways Software Technology, 2015). Although the documentation on WinHex tool not explicitly say it decompresses the hibernation file, viewing the hexadecimal data of the file confirms plain text is seen in the contents. The free version of Winhex can be useful in viewing the hibernation file on a live system. However, a licensed version is needed for acquisition. The searching capabilities of the free version of WinHex are limited and may be useful for locating a specific piece of evidence but not for extracting all the evidence from the file.

FTK Imager. FTK Imager is memory acquisition software offered by AccessData and is free to use even in corporate environments. This software will not acquire the hibernation file from a live system, will not convert the file to a raw memory image nor is it capable of analyzing the hibernation file. FTK Imager can mount a forensic disk image allowing the hibernation file to be acquired. The tool provides an easy to use GUI interface eliminating the need for an analyst with command level skills. FTK Imager is also available in a portable version designed to fit on external media for incident response (AccessData, 2012). Although this tool lacks the diverse functionality of other tools discussed thus far, its price, user friendliness and portability makes this tool notable for hibernation forensics.

Other tools. There are numerous additional tools not examined in this study. For instance, two well-known all-in-one forensic solutions, AccessData's Forensic Tool Kit and Guidance Software's EnCase are both enterprise level packages which offer high functionality and corporate level costs. Blade Forensic Data Recovery software can convert the hibernation file into a raw memory dump from both thirty-two and sixty-four bit versions of Windows XP through 7 (Blade Forensics, 2013). Belkasoft Evidence Center 2012 also offers hibernation file analysis (Belkasoft, 2015). The Forensic Disk Decryptor by Elcomsoft utilizes the hibernation file in addition to other memory dumps to obtain encryption keys and to decrypt hard drives for Windows XP through 7 systems (Elcomsoft, 2015). This list is far from exhaustive. Analysts should perform further research into the available forensic technologies before choosing the tool best for their investigations.

Considerations for selecting forensic software. The choice of which software will best suit an analyst depends mainly on its functionality, speed, price and skill requirement. The tradeoffs are different for each forensic lab, individual analyst and each investigation. Other considerations include the tools immiscibility in court. It is important also to understand the User License Agreement (ULA). Free tools exist which are exclusively intended for personal use as defined by their ULA and which may or may not offer licensing for commercial purposes. The best option should never be limited to just one program. Having two or three options will help in cases where the first program fails or does not provide the best functionality for the situation.

DISCUSSION OF FINDINGS

Major Findings

The purpose of this capstone project is to demonstrate the value of the hibernation file in forensics and incident response. The research is intended to provide more comprehensive documentation on the hibernation file than what is currently available. This study encompasses the hibernation feature in Windows, the hiberfil.sys file format and its forensic uses. While there are numerous publications dedicated to memory forensics, the majority of the works reviewed dedicate no more than a few pages to the hibernation file. This research has shown the hibernation file can contain crucial evidence and is a significant artifact in forensic investigations. The study also demonstrates that existing published works on the hibernation file are incomplete and sometimes contradictory. The topics chosen for this study were designed to increase the understanding of the hibernation file and the quality of evidence obtained. The study demonstrates how a lack of knowledge regarding the hibernation file could lead to fragmented and inaccurate evidence.

The literature review section covered an advanced look at the hibernation mode including file format and its variations in software and hardware configurations. The sources chosen for this study were based on reverse engineering work previously published and documentation directly from Microsoft. Numerous uses for the hibernation file were examined along with concepts affecting how the file is reviewed and interpreted. The advanced topics of hibernation slack space and malware analysis were also covered. The sources for these topics included memory forensics and malware analysis publications. Forensic tools with hibernation file capabilities were reviewed utilizing the developer’s web pages. Some of the published work on hibernation forensics is contradictory. Sources were chosen to highlight these contradictions and demonstrate how the discrepancies can affect evidence interpretation.

This research culminated in an advanced look into hibernation forensics and how to maximize the effectiveness in the files examination. The concepts covered demonstrate how misinterpreting the hibernation file could potentially lead to inaccuracies or incomplete data. This study is far from complete as there are still many unknowns with regards to the hibernation file format, what data is written to the hibernation file and how it varies between software and hardware configurations.

Theme One - Understanding the Hibernation File

The hibernation file has a complex and proprietary format. Hiberfil.sys is referred to as a “file”. However, its properties such as how it organizes and stores data in the file and its pre-allocated file size demonstrates that it could be more accurately referred to as a volume. A thorough understanding of the hibernation feature and its resultant file can aid forensic analysts in obtaining more comprehensive and accurate evidence. This section outlines how the key features of hibernation effects data and how understanding these characteristics along with variations in software and hardware can affect the quality of evidence.

Hibernation feature. Mastering the hibernation file requires knowledge of the hibernation mode and how the features differ in subsequent versions of Windows. Among the concepts covered, an essential point is the significance of knowing what data is written to the hibernation file. The majority of research was in agreement that the hibernation file contains the full contents of memory (Anson et al., 2012; Bunting, 2008; Carvey, 2009; Hale, 2012; Lee, 2014a; Suiche 2008c). However, a contrary publication indicates that systems running in DHCP will release active connections before it hibernates resulting in missing network data.

Additionally, memory-only malware may be designed to removing itself before a system hibernates (Ligh et al., 2014). While this is not a feature of hibernation mode, it does result in an incomplete memory capture. If an analyst expects that active network connections will be present in the hibernation file, their absence may lead to the conclusion that there was no network connections. If an analyst expects that the hibernation file will capture memory-only malware, its absence may result in the conclusion that there was no malware. Knowing what is and what is not copied to the hibernation file can prevent inaccurate interpretations of the evidence.

Negative evidence. Understanding what is, or is not, saved to the hibernation file is critical. When an analyst is aware that certain evidence may or may not be written to the hibernation file, not finding particular evidence produces "negative evidence". Negative evidence is data which an investigator can not verify if its absence is circumstantial or conclusive proof that it never existed. This type of data can still be useful in a supportive capacity as long as it can be identified as such (Kertesz & Rakosi, 2012). The danger lies in evidence which is incorrectly interpreted as conclusive proof. A case like this, where an analyst incorrectly concludes that missing data is proof of nonexistence, could have disastrous results. By understanding what data is not written to the hibernation file an analyst could correctly interpret the lack of evidence as being "negative" evidence and use it only in a supportive capacity.

Hybrid sleep feature. Hibernation forensics is not limited to laptops. Many publications focus on hibernation mode, designed for laptops and miss hybrid sleep mode designed for workstations. This in itself can cause practitioners to assume incorrectly desktops will not have a hibernation file. There was very little found in current publications which detail the hibernation file created by hybrid sleep mode. There is no clear indication as to what data precisely is written to the hybrid sleep hiberfil.sys. The Microsoft website indicated that in hybrid sleep mode, open documents and program data is written to hiberfil.sys (Microsoft, 2015c). Another publication stated that the full contents of memory are saved to hiberfil.sys in both hybrid sleep and hibernation modes (Bunting, 2008). As previously mentioned, it is important to know what evidence can be expected from the hibernation file. Even if the data written in hybrid sleep is limited, the file should not be overlooked as it can still contain valuable evidence.

Hiberfil.sys binary structure. The primary structures of the hibernation file format were reviewed. This study only scratched the surface of the hiberfil.sys format. There is a deficiency in the amount of published works describing these structures (Chow & Shenoi, 2010). The file is composed of a header, processor state; reserved memory map and memory blocks called Xpress images. The structure of these files varies between operating systems. The header contains a signature which indicates whether a system is hibernating, waking or has already resumed (Suiche, 2008a, 2008b, 2008c, 2010). Published work on the hibernation file format is incomplete and requires further study. Understanding the file format is not required to produce reliable evidence. However, it will aid analysts in deep forensics particularly when examining partial, corrupted or damaged files.

Important concepts for hibernation forensics. The effects of the differences in hibernation features across software and hardware configurations are important concepts. One major concept to be considered is the way the hibernation mode zeros out the file header upon resume removing valuable data needed to reconstruct evidence (Ligh et al., 2014). Forensic tools such as Volatility are available to reconstruct this data, or a skilled analyst could manually reconstruct the data using the Xpress image headers (Diablohorn, 2014). The compression algorithm which changed in Windows 8 will affect compatibility for forensic tools. Other concepts which affect tooling compatibility are the size of the file and the compression ratio which changed in Windows 7 (Microsoft, 2015b). The compression ratio for Windows 7 and later hibernation files can be modified by the user and could affect forensic analysis if the tool chosen is not able to allow for the change.

The study of the hibernation file is currently incomplete. Due to its proprietary nature, current published works have had to rely on reverse engineering. Although reverse engineers have provided a wealth of information on hiberfil.sys, there is data missing such as definitions for all of the files variables. Additionally, references are made to several variations in the structure but there is no comparative study to define all of the deviations.

Theme Two - Important Concepts for Hibernation Evidence

Several central concepts regarding the hibernation file can affect forensic evidence. It is important to define what data should be expected in the hibernation file and how changes to default settings can affect evidence analysis. Additionally, the concept of hibernation slack should be considered in an investigation because wrongfully attributing evidence to the most recent hibernation file could result in incorrect conclusions.

Hiberfil.sys versus live memory capture. Research found discrepancies in whether or not the hibernation file always contains the full contents of memory. Many publications explicitly state the hibernation file contains the entire contents of memory. Carvey made this statement, “Something else to consider about the hibernation files is that this functionality may be the only option available for capturing the full contents of physical memory” (Carvey, 2009, p. 119). Karp also makes the statement “Hibernate saves a copy of everything in your PC’s memory (RAM) onto your hard drive before it shuts down” (Karp, 2010, p. 276). Many other publications claim the full contents of memory are copied to the hibernation file. These claims are contradicted by Ligh, who warns “Before a system hibernates, the DHCP configuration (if any) is released and any active connections are terminated. As a result, network data in hibernation files might be incomplete" (Ligh et al., 2014, Chapter 4, Windows Hibernation File, para 5). Ligh goes on to speculate that malware could also remove itself from memory prior to hibernation. The key takeaway here is that in the absence of extensive experimentation, the claim as to whether or not the hibernation file contains the full contents of memory is not confirmed. Analysts should consider these discrepancies before making assumptions about missing data.

Hiberfil.sys compression ratio. Windows 7 introduced some of the biggest changes to the hibernation file. In addition to removing the four gigabyte file size limit, hiberfil.sys is also compressed by default to seventy-five percent. With administrative privileges, this default compression ratio can be changed to anywhere between fifty to one hundred percent (Microsoft, 2009). Forensic tools which support Windows 7 and later formats may be preconfigured to decompress based on the default settings or may be capable of detecting the compression ratio. If a forensic tool fails, the analyst could investigate into whether or not these default settings were changed. Another consideration is that changing the default settings may cause the hibernation to fail and could be identified as the reason for the absence of a hibernation file.

Hibernation slack space. Hibernation slack space is a concept that could be covered in an independent study by itself. This concept has not been fully explored or documented. Slack space in a hard drive exists because the drive is a set size and is typically not one hundred percent full. Evidence is found in hard disk slack space because when a file is deleted the space is merely unallocated not overwritten (Marcella & Guillossou, 2012). Files do not typically contain slack space but as discussed earlier hiberfil.sys is more like a volume than a traditional file. Hiberfil.sys is created at its maximum size when hibernation mode is enabled. When a system resumes, the header is zeroed out but not the entire file. If a system hibernates while using less memory than the previous hibernation, there will presumably be remnants of the past hibernation left in the slack space.

There were some discrepancies in current research regarding how the hibernation file is written. When hibernation is enabled, hiberfil.sys is created at its full size. According to the EnCase computer forensics official study guide, the hiberfil.sys in Windows XP is generated and the space it occupies is filled with zeroes (Bunting, 2008). McCash of SANS published his claims that in examining Windows 7 hibernation files, he found the hiberfil.sys created on a freshly installed Windows operating system did not contain zeros. Interestingly, he found the contents of the hibernation file to contain data from hibernation prior to the reinstallation of Windows. McCash speculated that the new install of the operating system had placed the hibernation file in the same location it was in previously (McCash, 2014). The presence of the old data indicates that the file does not fill with zeros when it is created but rather allocates space. These discrepancies are likely due to the variation of Windows systems; however more research is needed to verify this assumption.

The implications of this study by McCash could be profound. For example, in corporate environments, when a laptop is passed on to another employee without zero wiping the hard drive, the hibernation file may contain data in slack space from the previous employee. If the former employee utilized most of the systems memory and the new employee utilized much less memory, there could be considerable remnants left on the system. Additionally, in cases where a computer is shared between users, evidence pulled from hibernation slack may not be attributed accurately to a particular user. Although slack space can have an adverse effect on evidence integrity, it could also prove to be an asset. Slack space can be responsible for locating more data than just the latest hibernation. The important consideration for hibernation slack space is determining how reliable the data is. If the evidence is found in hibernation slack, the analyst must be able to confirm reliably as to whether or not it can be attributed to the current user. An analyst must also consider whether or not their chosen forensic tool is providing data exclusively from the last hibernation or carved from slack. This study is far from complete and warrants additional research due to the serious implications it can have on the accuracy of evidence interpretation.

Theme Three – Practical Uses of the Hibernation File

The role hiberfil.sys plays in forensics is a major theme of this research. Hiberfil.sys also plays a significant role in incident response and data recovery. For incident response, the hibernation file can be a powerful tool in the identification of malware. For data recovery, hiberfil.sys is useful due to the presence of encryption keys stored in memory. The hibernation file provides a window into the past. Together with system backups, a practitioner can access several windows from the past, refining timelines.

Hiberfil.sys in malware analysis. Advanced malware can hide itself inside system memory. The hibernation file can be central to the identification of this memory-only malware. Incident responders often need to acquire malware, particularly in cases of zero-day, advanced persistent threat (APT) attacks (Nathans, 2015). Although incident responders may be able to obtain a memory image from the system, the malware may have already been cleared. It is possible the hibernation file could have captured the malware before it was removed. Archived hibernation files may provide additional information as well, to help narrow down the incident timeline.

Hiberfil.sys and encryption. The hibernation file can be a valuable tool in data recovery particularly because it may contain encryption keys for both full and partial encryption (Anson et al., 2012). Microsoft explicitly states that the hibernation file in Windows XP can not be encrypted. Additionally, Microsoft does not provide third parties with a method to encrypt the hibernation file (Morello, 2007). As a result, Windows XP systems can be unlocked by utilizing the hibernation file. Additionally, systems encrypted with any software other than Microsoft's BitLocker may not have an encrypted hiberfil.sys.

Hiberfil.sys archives. In addition to the main hiberfil.sys located in the root directory, system backups such as the volume shadow copy, restore points and remote backups will contain archived copies of the hibernation file. Windows XP and later versions have default settings to backup either daily or once a week depending on the system. Backups are kept as long as there is room on the disk (Lee, 2014b). By acquiring and examining the hibernation file from backups, an analyst can investigate system memory from several points in the past providing valuable data for timelining events or in determining regular user behavior.

The hibernation files value in forensics goes far beyond what is covered in this research. Having a snapshot or several snapshots from points in the past of Internet history, registry states, clipboard contents, email, chat logs and more can play a vital role in constructing a picture of past events.

Theme Four – Hibernation Forensic Tools

The choice of forensic program entails several considerations. Because it is a proprietary and protected system file, hiberfil.sys requires a forensic tool capable of decompressing and parsing the data. The best choice for forensic software will depend on many factors including budget, analyst's skill level and the individual situation. A number of tools were examined in the literature review. These tools were selected for their functionality and the researcher's familiarity. Their inclusion in this study should not be viewed as an indication of their value or as an endorsement. Additionally, to maximize the effectiveness of investigations, analysts should have access to and skill in using more than just one program.

Why forensic software is needed. The acquisition and examination of the hibernation file requires forensic software. Hiberfil.sys is a protected system file which cannot be moved or copied from a live system using the copy and paste function in Windows. The file cannot be copied while "in-use" which is its state when the hibernation mode is enabled. When hibernation mode is disabled, the file is no longer allocated in the file system and will require a carving tool. The file format is proprietary and in Windows 7 and 8; the file is also compressed (Microsoft, 2009). A forensic tool must be able to parse and decompress the data from the proprietary file. Another challenge to hibernation forensics is that the hiberfil.sys header is zeroed out at any point other than when the system is hibernating (Ligh et al., 2014). The forensic software must be able to able to reconstruct the data without the benefit of the files header.

Comparison of current technologies. The tools which were included in the literature review are just a few of many available forensic programs. An analyst should investigate many tools before considering the right one in an investigation. These tools can be categorized by price, licensing requirements, user skill requirements and capabilities.

When choosing the best program, cost plays a significant role. A tool like Volatility is free and has many features such as the ability to convert the file into a raw memory image, or to be able to analyze the file in its native format. Volatility is also a full memory analyzer (Volatility Foundation, 2014b). The biggest disadvantage of Volatility is the skill required by analysts who utilize it. Volatility is command line based and each feature requires the analyst to learn the different plugins.

The commercial software, MoonSols Windows Memory Toolkit, is rich in capabilities and also offers a free version. However, the restrictions of the free version exclude it from being useful in many investigations. Although costly, the licensed version provides many valuable features and customer support (MoonSols, 2015b). Support is a feature which should not be overlooked while evaluating forensic programs.

Other tools are not quite as rich in features but are still valuable. WinHex is capable of acquiring the hibernation file from a live system and allows practitioners to view and search the file in hexadecimal format (X-ways Software Technology, 2015). Although WinHex is a simpler tool, it is also much less expensive than other commercial programs. WinHex is a simple to learn Graphical User Interface (GUI) software.

Other tools available include specialized software which provides a single function such as the Forensic Disk Decryptor, which uses the hibernation file to find encryption keys (Elcomsoft, 2015). An analyst may be able to find an all-in-one solution or may choose to select several specialized tools as their preferred forensic method.

One of the biggest challenges in forensics is time constraints. Specifically with the growth in hard drive and memory sizes, investigators need to examine an immense amount of data in minimal time (Cohen, 2015). The speed in which a tool can acquire and analyze data is a main consideration. Of the tools reviewed, Volatility may look like the obvious choice to some analysts based on price and features. However, even for analyst's already proficient with command line, deadlines may make an all-in-one tool such as AccessData's Forensic Toolkit more desirable. An all-in-one solution may be able to parse all evidence and present it in an easily searched format which speeds up the analysis process. Volatility requires a separate plugin for many of its features and the output requires individual examination. All-in-one solutions typically have convenient reporting features which can also speed up an investigation.

Analysts should be familiar with as many forensic tools as possible. Cost will play a factor in any forensic lab but so too will deadlines. The analyst's skill level is another major factor making GUI interfaces desirable in labs with a high employee turnover rate. The tool chosen must also be compatible with any operating system which may need analysis. Law enforcement labs must also take into consideration the reputation of forensic software and whether the courts have previously accepted it. These are just some of the major considerations for selecting the best forensic technology.

Comparison of Findings with Existing Studies

This study is a broad look at the hibernation file. Much of the published works researched for this study were vague or focused on a single aspect of the hibernation file. This study combined the individual facets of several studies previously published to provide a complete accounting of hiberfil.sys. This project was not limited to just the hibernation file but also its usefulness in forensics and significant concepts for analysis. A wide range of topics were covered, which range from understanding the hibernation feature to considerations for choosing a forensic tool.

Unlike the work performed by reverse engineers and other experimental publications, the scope of this project did not include an experimental section. Instead, this research focused on why the hibernation file is useful and on reporting discrepancies in current research. This work was research-based, combining the findings in many other publications to provide a more in depth accounting of the hibernation file.

This study focused on how the hibernation file differs between Windows versions. Some existing studies of the hibernation file do not explicitly define which Windows version of the hibernation file is covered. This project not only highlights these differences but also discusses how they could affect evidence. Not all differences could be identified, this research stresses the fact that while much is known about the hibernation file, extensive research could be performed to expand the current knowledge set and to expose misconceptions.

Limitations of This Study

This study was limited by the proprietary nature of the hibernation file. Existing publications rely on reverse engineering by Suiche. Other professional publications regarding the binary format all referenced back to Suiche's work. The absence of more than one experimental study of the file format means there is less of a chance to identify mistakes or to fill in the gaps.

The software review section of this project was limited by the lack of access to fully licensed software. Technology appraisal relied on research and not firsthand experience with the tools. While this study was able to report on the capabilities of each tool, it lacks the intuitive comparison which comes only from direct experience with the software. As a result, user-friendliness, ease of learning and speed was unable to be reported on beyond identifying the software as GUI or command line based.

Research in this study included contradictory publications. The inclusion of these contradictions was intended emphasize the need for further research and the dangers of interpretations based upon assumptions. This study was limited by the lack of an experimental section with the ability to prove or disprove contradictions.

The scope of the research was limited to covering just Windows operating systems. This study could be expanded to include other operating systems such as Apple and Linux distributions which have their own hibernation modes. Additionally, mobile computing such as tablets and smartphones contain their own suspend to disk features which could be examined as a source of digital evidence. Specifically with the growing use of mobile computing, a study of suspend to disk features would be beneficial.

RECOMMENDATIONS

This capstone project distinguished the hibernation file as being an important part of digital forensics and incident response. Academics in the field of memory forensics regard the hibernation file as a valuable source of evidence (Beverly et al., 2011; Lee, 2014a; Postinge, 2011; Stefan & Freiling, 2011; Suiche, 2008c). This research found that notwithstanding the files importance, there are considerable holes and conflicting inferences in current research. The significance of the hibernation file merits a bigger presence in future memory forensics publications as well as in academic settings. The quality of evidence produced from an investigation can be improved with education on the hibernation file. However, further research is needed to close gaps and resolve conflicts in current studies regarding the hibernation and hybrid sleep modes and their resultant hiberfil.sys file. The scope of this project did not allow for a more in-depth look at current forensic tools or at other operating systems outside of Windows. Further research on these topics would advance the field of memory forensics.

Recommendations from Professionals

Practitioners in the area of forensics agree that further research into the hibernation file is needed. Chow and Shenoi state that in regards to hibernation forensics "difficulties include the lack of published descriptions of hibernation file formats" (Chow & Shenoi, 2010, p. 197). This idea is shared with Stefan and Freiling who point out that the proprietary format of the hibernation file has "long impeded thorough understanding and applicability in forensic investigations" (Stefan & Freiling, 2011, p. 11). Thanks to the reverse engineering, first published by Suiche, details of the hibernation file format are now available. However, these publications do not cover every variable or every version of hiberfil.sys (Beverly et al., 2011; Diablohorn, 2014; Kleissner, 2009a, 2009b; Suiche, 2008a, 2008b, 2008c, 2010). Further research could identify and define the missing aspects of the hibernation file not described in current studies.

Recommendations for Awareness and Education

The significance of the hibernation file justifies a greater awareness and representation within future published forensics works. Several sources which were researched included digital and memory forensic books. These works either briefly mentioned the hibernation file or dedicated no more than a couple pages to the topic (Anson et al, 2012; Bunting, 2008; Carvey, 2009; Karp, 2010; Ligh et al. 2014; Marcella & Guillossou, 2012). This mention is not to criticize these works but purely to point out that, in the large range of relevant forensic topics covered, the hibernation file was only a small part. As awareness of the hibernation files value develops within the industry, future works may well choose to provide a more in-depth accounting of hiberfil.sys and its uses.

The quality of evidence obtained during a forensic investigation can be improved when the analyst is skilled in hibernation analysis. A practitioner should be able to identify the types of evidence which can be retrieved from the hibernation file and how to extract the data. Education in extracting archived hibernation files from system backups will help improve timelining and could assist in the identification of malware. Further education should include how to pull user credentials and encryption keys from the hibernation file. Professionals who are educated in and have access to several hibernation forensic tools will reduce the risk of tooling error impeding an investigation. Advanced skills in both carving hibernation segments from system slack space and carving data from hibernation slack space can provide evidence which may not have been acquired by other means. However, further research is needed before an analyst can gain a complete understanding of the hibernation file.

Recommendations for Further Research

There are many unanswered questions about the hibernation file requiring further study. Many of these issues can alter how acquired evidence is perceived. Chief among these problems is defining what data specifically is written to hiberfil.sys. Many publications have statements which claim that the entire contents of the systems memory is saved (Anson et al., 2012; Bunting, 2008; Carvey, 2009; Hale, 2012; Lee, 2014a; Suiche 2008c). A conflicting publication claims the hibernation file may not, in all actuality, contain the complete contents of memory citing DHCP connections (Ligh et al., 2014). Further research could contain the particular methodology that includes the acquisition of a memory image followed by an immediate hibernation and the comparison of the two. This methodology would need to be repeated in different Windows versions and varying situations. Further research can also be made by defining what happens when the hibernation file is created, deleted or overwritten and how slack space is handled. Although McCash' s publication on hibernation slack space was informative, there is a need for a scholarly study that provides a detailed experimental section (McCash, 2014). These studies can be performed both for the hibernation mode and hybrid sleep mode.

Current publications contain conflicting statements as to whether or not hybrid sleep mode writes the full contents of memory to hiberfil.sys. Microsoft's website indicates that open documents and program data is saved to hiberfil.sys in hybrid sleep mode (Microsoft, 2015c). However, another publication states that the entire contents of memory are copied to hiberfil.sys in Hybrid sleep mode (Bunting, 2008). Most of the books researched in this capstone did not even mention hybrid sleep mode. There were no significant sources located for this topic. This topic is of importance because knowing what data an analyst should expect can make the difference between negative evidence and evidence which is incorrect. For example are network connections missing because they were not saved, or because the system was not connected? The study of hibernation forensics would benefit from a study comparing hibernation mode to hybrid sleep mode and the resultant data contained within hiberfil.sys.

Comparison of Current Forensic Technologies

The forensic industry would benefit from a comparative publication of current forensic tools. A specific study of hibernation forensic technology could assist in better-educated decisions on which software analysts keep in their toolkits. Most analysts do not have the time or resources to learn and evaluate each tool available. This capstone project made a comparison of just a few of the tools currently available and the important considerations to be assessed. Further research which includes an experimental comparison of these technologies outlining user-friendliness and the time to acquire and analyze evidence would be valuable to forensic professionals. This research merely scratched the surface of current technologies. There are two major disadvantages to this type of study. One drawback is that even if researchers attempt to exclude bias, the simple fact is that one tool may be best for one analyst and not for another. This study would need to be made by several researchers combining their findings. The second disadvantage is that this type of study would require a constant refresh as technology evolves.

How This Research Could Be Improved

This capstone project had a scope which did not include an experimental analysis of the hibernation file format or current forensic tools. The topic of hibernation forensics is still in its infancy resulting in limited scholarly resources. This study could have been improved with an experimental section and the ability to acquire commercial forensic tools. An experimental section may also have been able to close some of the holes in current publications and possibly address the conflicting research identified in this project. Further, collaboration with other professionals who have studied the hibernation file could have also enhanced this capstone project.

This study was limited to the Windows platform. Other operating systems such as Apple and Linux not to mention mobile devices such as Android contain their own suspend to disk feature. The recommendations for future study made here can be expanded to other operating systems. Particularly, with the recent explosive growth in mobile computing, more and more evidence may be located on a smartphone or tablet as opposed to a computer. Because of this, it is increasingly important for forensic professionals to understand and be able to analyze the artifacts from these devices.

There is still much research to be performed in the field of hibernation forensics. The importance and complexity of the hibernation file could easily result in an entire book dedicated solely to hibernation forensics. However, before such a book could be written, further research is required to fill the current gaps and to settle discrepancies. The field of digital forensics is still a relatively new one and scholarly references, even on significant topics such as hibernation, are lacking.

CONCLUSION

Memory forensics is a dynamic and significant part of digital forensics. Due to memories volatile nature and the vast wealth of evidence it can provide, forensic professionals study memory forensics as an autonomous piece of their education. Windows Systems copy the contents of memory to the hibernation file when the computer enters hibernation mode. Forensic tools are available which convert the proprietary hibernation file into a raw memory image so it can be examined with traditional memory forensic methods. This file is a valuable source of evidence both in the absence of and in addition to a current memory capture. Although the hibernation file can provide an immense wealth of data, the existing published works on the hibernation file are far from comprehensive and require further research.

The hibernation file format is proprietary and not wholly documented. Although referred to as a “file”, hiberfil.sys behaves more like a volume. Current studies rely on the publications from the reverse engineering of hiberfil.sys. These documents do not define every variable within the structure. Additionally, the structure varies between Windows versions and hardware configurations. Current studies contain conflicting reports of hibernation mode behavior and the resultant hiberfil.sys file. These factors demonstrate that current research is inadequate to allow forensic professionals to understand the hibernation file entirely. Further study is needed to accurately define the contents of the hibernation file and to identify variations in its format.

This capstone project demonstrated the usefulness of the hibernation file for enhancing timelines, detecting malware and in retrieving encryption keys, usernames and passwords. The software review section explained that analysts should consider, in addition to pricing, required skills, time, features, immiscibility in court and specific investigation needs. Additionally, an analyst should not limit themselves to just one tool for all jobs but strive to learn as many as possible.

The recommendations of this study are for further research into the hibernation file in order to fill existing gaps and resolve contradictions in current works. Hibernation forensics is a dynamic and complex study which could in itself fill an entire book or be the topic of future forensic courses. This subject should be expanded past Windows hibernation and encompass all other operating systems and especially mobile devices.

The field of digital forensics is still relatively new although it has grown in leaps and bounds in recent years. Because of the significance of the hibernation file in digital forensics, its complex nature and the limitations of current studies, further research on this topic is necessary. A digital forensic investigation is often performed to determine a person's guilt or innocence. The consequences of the forensic analysis can result in loss of job or reputation, incarceration, the location of a missing child or even saving thousands of lives from a terrorist attack. This study demonstrated that understanding and utilizing the hibernation file can result in more thorough and accurate evidence. The expanding study of this subject can lead to better quality evidence and the most accurate resolution of forensic cases. Although not all investigations result in the saving of lives, justice for victims or vindication of the innocent, the need to obtain complete and accurate evidence is always essential.

REFERENCES

AccessData. (2012, March 21). *AccessData FTK Imager user guide*. Retrieved from https://ad-pdf.s3.amazonaws.com/Imager%203_1_4_UG.pdf

AccessData. (2014, August 21). *AccessData's Forensic Toolkit user guide.* Retrieved from https://ad-pdf.s3.amazonaws.com/ftk/ftk%205.5/FTK_UG.pdf

Anson, S., Bunting, S., Johnson, R., & Pearson, S. (2012). *Mastering Windows network forensics and investigation, second edition*. Indianapolis, IN: John Wiley & Sons, Inc.

Belkasoft. (2015). *Belkasoft Evidence Center 2015*. Retrieved from Belkasoft.com: http://belkasoft.com/en/ec

Beverly, R., Garfinkel, S., & Cardwell, G. (2011). *Forensic carving of network packets and associated data*. Retrieved from http://rbeverly.net/research/papers/ipcarving-dfrws11.pdf

Blade Forensics. (2013). *Blade forensic data recovery*. Retrieved from http://www.bladeforensics.com/

Brunty, J., & Helenek, K. (2013). *Social media investigation for law enforcement*. Waltham, MA: Elsevier Inc.

Bunting, S. (2008). *EnCE The official EnCase certified examiner study guide second edition.* Indianapolis, IN: Wiley Publishing, Inc.

Carrier, B. (2005). *File system forensic analysis.* Boston, MA: Pearson Education, Inc.

Carvey, H. (2007). *Windows forensic analysis DVD toolkit*. Burlington, MA: Syngress Publising, Inc.

Carvey, H. (2009). *Windows forensic analysis DVD toolkit 2E*. Burlington, MA: Syngress Publishing, Inc.

Carvey, H. (2012). *Windows forensic analysis toolkit advanced analysis techniques for Windows 7*. Waltham, MA: Syngress Publising, Inc.

Casey, E. (Ed.). (2010). *Handbook of digital forensics and investigation*. Burlington, MA: Elsevier Academic Press.

Chow, K., & Shenoi, S. (Eds.). (2010). *Advances in digital forensics VI.* Berlin, Germany: Springer.

Cohen, C. L. (2015, April). *Growing challenge of computer forensics.* Retrieved from http://www.policechiefmagazine.org/magazine/index.cfm?fuseaction=display_arch&article_id=1136&issue_id=32007

Craiger, J.P., Pollitt, M., & Swauger, J. (2005, April 1). *Law enforcement and digital evidence.* Retrieved from http://euro.ecom.cmu.edu/program/law/08-732/Evidence/Craiger.pdf

Diablohorn. (2014, December 10). *Parsing the hiberfil.sys, searching for slack space*. Retrieved from https://diablohorn.wordpress.com/2014/12/10/parsing-the-hiberfil-sys-searching-for-slack-space/

Elcomsoft. (2015). *Elcomsoft forensic disk decryptor*. Retrieved from https://www.elcomsoft.com/efdd.html

Forensicmag.com. (2013, February 20). *Passware Kit exposes passwords for social networks*. Retrieved from http://www.forensicmag.com/articles/2013/02/passware-kit-exposes-passwords-social-networks

Grimes, R. A. (2014, February 4). *Should you worry about memory-only malware?* Retrieved from http://www.infoworld.com/article/2608848/security/should-you-worry-about-memory-only-malware-.html

Hale, M. (2012, October 12). *Volatility an advanced memory forensic framework*. Retrieved from https://code.google.com/p/volatility/wiki/HiberAddressSpace

Karp, D. A. (2010). *Windows 7 annoyances*. Sebastopol, CA: O'Reilly Media Inc.

Kertesz, A., & Rakosi, C. (2012). *Data and evidence in linguistics: A plausible argumentation model*. New York, NY: Cambridge University Press.

Kleissner, P. (2009a). *Hibernation file attack*. Retrieved from http://stoned-vienna.com/downloads/Hibernation%20File%20Attack/Presentation.pdf

Kleissner, P. (2009b). *Hibernation file format*. Retrieved from http://stoned-vienna.com/downloads/Hibernation%20File%20Attack/Hibernation%20File%20Format.pdf

Lee, R. (2014a). *SANS Forensics 508.2: Memory Forensics*. The SANS Institute.

Lee, R. (2014b). *SANS Forensics 508.4: Deep-dive forensics and anti-forensics.* SANS Institute.

Ligh, M. H., Adair, S., Hartstein, B., & Richard, M. (2011). *Malware analyst cookbook and DVD.* Indianapolis, IN: Wiley Publishing, Inc.

Ligh, M. H., Case, A., Levy, J., & Walters, A. (2014). *The art of memory forensics* [Kindle edition]. Indianapolis, IN: John Wiley & Sons, Inc.

Lowman, S. (2010, January 1). *The effect of file and disk encryption on computer forensics*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.391.3217&rep=rep1&type=pdf

Marcella, A. J., & Guillossou, F. (2012). *Cyber forensics: From data to digital evidence*. Hoboken, NJ: Wiley & Sons Inc.

McCash, J. (2014, July 1). *Hibernation slack: Unallocated data from the deep past*. Retrieved from http://digital-forensics.sans.org/blog/2014/07/01/hibernation-slack-unallocated-data-from-the-deep-past

Microsoft. (2009, September 15). *Reducing the disk footprint for Windows 7 hibernation.* Retrieved from http://download.microsoft.com/download/7/E/7/7E7662CF-CBEA-470B-A97E-CE7CE0D98DC2/HiberFootprint.docx

Microsoft. (2013, January 5). *System power states*. Retrieved from https://msdn.microsoft.com/en-us/library/windows/desktop/aa373229%28v=vs.85%29.aspx

Microsoft. (2015a). *Drive letters assigned to unsupported partition types*. Retrieved from https://support.microsoft.com/kb/KbView/221799

Microsoft. (2015b). *Memory limits for Windows and Windows server releases*. Retrieved from https://msdn.microsoft.com/en-us/library/windows/desktop/aa366778%28v=vs.85%29.aspx

Microsoft. (2015c). *Sleep and hibernation: frequently asked questions*. Retrieved from http://windows.microsoft.com/en-us/windows7/Sleep-and-hibernation-frequently-asked-questions

Microsoft. (2015d). *To display hidden files and folders*. Retrieved from https://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/win_fcab_show_file_extensions.mspx?mfr=true

Microsoft. (2015e). *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*. Retrieved from http://support.microsoft.com/kb/888575

MoonSols. (2013, November 10). *MoonSols releases MoonSols Windows Memory Toolkit 2.0 for forensics, incident response investigators*. Retrieved from http://www.moonsols.com/wp-content/uploads/2011/02/PRESS-RELEASE-10Nov2013-EN-v1.1.pdf

MoonSols. (2015a). *MoonSols delivers high-end utilities to respond to advanced cyber-security attacks*. Retrieved from http://www.moonsols.com/#pricing

MoonSols. (2015b). *MoonSols Windows Memory Toolkit*. Retrieved from http://www.moonsols.com/windows-memory-toolkit/

Morello, J. (2007, March). *Deploying EFS: part 2*. Retrieved from https://technet.microsoft.com/en-us/magazine/2007.03.securitywatch.aspx

Nathans, D. (2015). *Designing and building a security operations center*. Waltham, MA: Syngress Publising, Inc.

Postinge, M. (2011). *Forensic examination of encrypted systems*. Retrieved from http://postinger.com/publications/encrypted_forensics.pdf

Reyes, A. (2007). *Cyber crime investigations bridging the gaps between security professionals, law enforcement, and prosecutors*. Rockland, MA: Syngress Publising, Inc.

Schwartz, E. (1998, July 27). One-chip solution powers up battery life. *Infoworld, Twentieth Anniversary 1978-1998,* 20(30).

Sikorski, M., & Honig, A. (2012). *Practical malware analysis.* San Fransico, CA: No Starch Press, Inc.

Stefan V., & Freiling, F. C. (2011, June 11). *A survey of main memory acquisition and analysis techniques for the Windows operating system*. Retrieved from http://www.ccse.kfupm.edu.sa/~ahmadsm/coe589-121/vomel2011-memory-forensics-survey.pdf

Suiche, M. (2008a). *Exploiting Windows hibernation*. Retrieved from http://www.msuiche.net/con/euro2008/Exploiting_Windows_Hibernation_File.pdf

Suiche, M. (2008b). *Sandman project*. Retrieved from http://sandman.msuiche.net/docs/SandMan_Project.pdf

Suiche, M. (2008c). *Windows hibernation file for fun 'n' profit*. Retrieved from http://www.blackhat.com/presentations/bh-usa-08/Suiche/BH_US_08_Suiche_Windows_hibernation.pdf

Suiche, M. (2010). *Blue screen of death is dead*. Retrieved from https://digital-forensics.sans.org/summit-archives/2010/eu-digital-forensics-incident-response-summit-matthieu-suiche-blue-screen-of-death-is-dead.pdf#__utma=216335632.595579441.1388420344.1422815802.1423691039.23&__utmb=216335632.5.8.1423691144814&__utmc

US Dept of Homeland Security. (2007, March 8). *Best practices for seizing electronic evidence v.3: A Pocket guide for first responders.* Retrieved from https://www.ncjrs.gov/App/Publications/abstract.aspx?ID=239359

Volatility Foundation. (2014a). *About the Volatility Foundation*. Retrieved from http://www.volatilityfoundation.org/#!about/cmf3

Volatility Foundation. (2014b). *Volatility*. Retrieved from http://www.volatilityfoundation.org/#!faq/c1q11

Whitfield, L. (2010, April 19). *Into the shadows*. Retrieved from https://forensic4cast.com/2010/04/into-the-shadows/

Williams, J. (2013, December 6). *Memory image file formats*. Retrieved from http://malwarejake.blogspot.com/2013/12/memory-image-file-formats.html

X-ways Software Technology. (2015). *WinHex: Computer forensics and data recovery software*. Retrieved from http://www.x-ways.net/winhex/index-m.html