Abstract

The purpose of this research paper was to analyze three anti-forensic techniques for potential methods of mitigating their impact on a forensic investigation. Existing research in digital forensics and anti-forensics was used to determine how altered metadata, encryption, and deletion impact the three most prominent operating systems. The common file systems for these operating systems were analyzed to determine if file system analysis could be used to mitigate the impact of the associated anti-forensic technique. The countermeasures identified in this research can be used by investigators to reduce the impact of anti-forensic techniques on an investigation. Also, the results could be used as a basis for additional research. File system analysis can be used to detect and mitigate the impact of the three methods of anti-forensics researched under the right circumstances. Some areas of anti-forensics and file systems have been relatively well-researched. However continued research is necessary to keep pace with changes in file systems as well as anti-forensic techniques. Keywords: Cybersecurity, Albert Orbinati, Windows, Linux, Macintosh, file table.

MITIGATING THE IMPACT OF ANTI-FORENSIC TECHNIQUES THROUGH FILE SYSTEM ANALYSIS

by

Gabriel A. Flynn

A Capstone Project Submitted to the Faculty of

Utica College

August 2012

in Partial Fulfillment of the Requirements for the Degree of

Master of Science
Cybersecurity – Intelligence & Forensics

Table of Contents

Statement of the Problem

Definition of the Problem

Anti-forensic techniques can remove the possibility of a complete and thorough investigation by even the most experienced investigator. Anti-forensic techniques have applications that are not associated with criminal activities, such as securely erasing banking documents. However, these same techniques are used in computer crimes to hide evidence of incriminating actions (Kedziora, 2011). One way to mitigate the impact of anti-forensic techniques is to understand the implications surrounding those anti-forensic methods. This understanding comes from thorough analysis of not only the technique used, but also of the environment in which it is used.

Anti-forensic techniques can be considered any action that attempts to discredit the efforts put forth by an investigator. However, these attempts are broken into attacks that aim to destroy evidence, anti-forensics, and those that attack the tools used to collect or analyze the evidence, counter-forensics (Hartley, 2007). By this definition, three anti-forensic techniques include *altering metadata*, *encryption*, and *file deletion*. These three methods can be accomplished with the use of simple programs that do not require a technical background.

The purpose of this research was to analyze how anti-forensic techniques impact the three most prominent operating systems. Specifically, this research addresses the questions: can the use of anti-forensic techniques be detected through analysis of the file system, and are there techniques an investigator can use to mitigate their impact on the investigation?

Justifying the Problem

Anti-forensic techniques detract from the accuracy of the evidence collected during an investigation. The increased utilization of anti-forensic tools and techniques are leading people to question the validity of digital evidence (Bayuk, 2010, p.116). Digital evidence that is collected during an investigation can be subjected to detailed analysis and therefore any doubt about its legitimacy can create problems in a court of law. For example, Trojans, viruses, and other malware are often blamed for questionable activities found in the course of an investigation. This defense is often supported with the use of anti-forensic methods in an attempt to make the evidence questionable.

Anti-forensic techniques can be used to hide, alter, replace, or destroy evidence that could be useful to an investigator. A failure to undeniably associate specific digital evidence to a person or persons lead some people to question its validity in a court of law (Strickland, 2008). The goal of an investigator is to overcome the impact of anti-forensics as much as possible if they are discovered during an investigation.

Understanding an anti-forensic technique is a critical component in mitigating the impact it has on the investigation. Anti-forensics will never be completely avoidable, but the susceptibility to these techniques can be reduced by focusing on each problem area (Harris, 2006). A greater understanding in each problem area increases the likelihood that an investigator will recognize signs of anti-forensics, and possibly even be able to overcome some attempts at hiding evidence.

Deficiencies in What We Know

There are numerous methods of anti-forensics that are used for both lawful and nefarious reasons. Four categories can be used to generally group the methods by their effect. These categories include *overwriting data and metadata, hiding data, obfuscation of data*, and *exploitation of bugs* in forensic tools (Barbara, 2008). For the purpose of this research three of these categories are represented by *altering metadata*, *deletion*, and *encryption*.

Altering Metadata. The altering of metadata can be any number of variables associated with a particular file on the volume. For example, *time stomping* is a term that is associated with the manual altering of a file's creation, modification, and access dates and times. When this type of anti-forensics method is utilized, it is more complicated for the investigator to prove when the files were in fact created, modified, or accessed. Altering metadata is comparable to counterfeiting in that the true evidence of the file still exists, but the origin is less clear (Harris, 2006). Altering the metadata is not always easily spotted.

The effects of time stomping can be identified in a number of different locations in the Microsoft Windows environment. One way to identify the use of time stomping is by comparing information found in the Prefetch files of Windows to that in the Master File Table (MFT) (Wade, 2010). This method assumes Prefetch is enabled, and that the file whose metadata has been altered is referenced in one of the entries. Also, it is unclear how the evidence may be compromised by incorrect system time. Additional research in the MFT alone shows that evidence of time stomping can be found by comparing two attributes of a single entry (Pogue, 2011). Either of these two methods can provide some leverage for an investigator in determining the true origin of a file.

File metadata is easily modified in the Linux Ext2/Ext3 file systems to a certain extent. Without any additional software or tools, a user can issue a command to manually change the access and modify times of a file (Altheide & Carvey, 2011). This change is not unlike the change that is found with NTFS, but there are some differences. In addition to the modify and access times for a file, Ext2/Ext3 stores a timestamp for the last change and the deleted time. However, the last change timestamp cannot be manually set, and the deleted timestamp is only populated once the file is actually deleted (Altheide & Carvey, 2011). Altheide and Carvey do not explain if or how the last change timestamp is altered during manual manipulation of the other timestamps. Also, specific information detailing how the metadata is altered when a file is deleted is not yet available. The limited amount of research in this area makes it difficult to accurately detect altered metadata.

The metadata for a file found on an HFS+ volume behaves similar to that of Ext2/Ext3 or NTFS. In addition to other metadata, a file's creation date, modified date, accessed date, backup date, and attribute modification date are all included as an entry in the catalog file for HFS+ (Craiger & Burke, 2006). Research did not uncover any tools specifically designed to alter this particular metadata, but this does not eliminate the possibility of it being altered. Unfortunately, different analysis tools do not consistently display or interpret the date/time stamps (Casey, 2010). As with any other file system, the reliability of the metadata on an HFS+ volume is useful in assigning ownership to a file.

Encryption. Encryption allows a user to hide, or obfuscate files, but still be able to use them as needed. Encryption can be applied to one or more individual files, or an entire volume can be encrypted (Carrier, 2005, p.142). As the files are needed, they can be

decrypted in the same manner of which they were encrypted. For example, files that are encrypted with one program are decrypted with the same program before they are used. The encrypted files or volume effectively render the information useless until they are decrypted. In some cases, it may not be apparent that the file even exists due to the lack of an organized structure.

The various methods that can be used to encrypt files in Windows have different effects on the file system. External applications encrypt a file after an application is through with it, but before it is written to the disc. In some cases only the content of the file is encrypted leaving the filename and metadata in plain-text form (Carrier, 2005, p.142). Carrier did not provide a detailed explanation of when filenames may remain in plain text. However, when an entire volume is encrypted there may not even be an obvious indication of existing files.

The basic process of encryption in Ext2/Ext3 is not unlike that of any other file system. Encrypting single files on in this file system is less complicated than full disc encryption since some Linux operating systems do not natively support the installation onto an encrypted drive (Petullo, 2004). In either situation, it is unclear what to expect of the state of the files system when encrypted files exist. Knowing whether Ext2/Ext3 behaves similar to that of NTFS, or if it is different all together would allow an investigator to recognize signs of encryption at a minimum.

Built-in and third-party utilities are available for the encryption of Macintosh files. This includes encryption for the entire volume, single files, and also container files. The key difference in these methods of encryption is when the encryption gets applied to the data.

Additional research is necessary to determine how encrypted files are stored in an HFS+ volume, and when they may be found in plaintext form.

Deletion. The deleted files an investigator may encounter range from simple deletion to those that have been overwritten with third-party software. A file that is simply deleted may still exist in raw form on the media. However, when the file is overwritten with meaningless data, or wiped, the actual bits that made up the original file are replaced by those unrelated to it. Overwriting a files information is not as simple as replacing the file with a new one of the same name. This means additional software is used to erase traces of the file. The software that is used for this process can offer the investigator new leads in an investigation (Harris, 2006). The more complex the deletion method that is used results in a more complex recovery effort.

The Windows operating system provides the user with methods to recover accidently deleted files. When a file is deleted in a normal manner on the more recent Microsoft Windows Vista systems, the file is simply moved to a hidden, system folder known as the Recycle Bin (Machor, 2008). Recovery in this case can be as simple as restoring the file from the Recycle Bin. Overwritten files are more complicated, but the fact that an overwritten file may not contain usable information for evidence is not necessarily a dead end to an investigation. Some overwriting programs can still leave information, such as the original filename, in the associated MFT entry (Russinovich, 2011). It is not quite understood what specific overwriting programs leave information behind. Likewise, it is unknown if any additional information in the MFT is left behind; such as any metadata.

Like any other system, a file that has been deleted from an Ext2/Ext3 file system is not necessarily gone. Initially, the belief was that a file that was deleted from an Ext3 file system could not be recovered (Haberland, 2004). The file system does not have a built-in undelete capability, but there are in fact ways to recover deleted files from the system. In order to recover any deleted files, third-party software must be used (Stonecypher, 2009). The process described by Stonecyper does not recover filenames as expected. Also, it is unknown if analysis of the file system could recover additional information about deleted files. Identifying what information can be found as part of a recovered file, or associated to the recovered file, is a critical piece of mitigating deleted or wiped files.

The methods of deletion found in Macintosh are very similar to those found in recent versions of Windows. Files that are simply deleted or moved to *Mac Trash*, are placed in a hidden system folder but still remain on the disc with all of the associated metadata intact (Craiger & Burke, 2006). Unless wiping methods are used, the actual data that makes up the file remains on the media while the reference to it in the catalog file is removed. Unlike NTFS however, the catalog file is a balanced tree structure that may require rebalancing after an entry is removed, and therefore overwriting the entry ("File deletion in," n.d.). The circumstances that cause the catalog file to be rebalanced are not explained. The limited window of opportunity to recover any information pertaining to a deleted file in HFS+ increases the complexity of an investigation involving deleted files.

Defining the Audience

This report will benefit digital forensic investigators, system and software developers, typical users, and also suspected criminals. Digital forensic investigators and suspected

criminals will benefit the most from the information for similar reasons. Each of these groups will benefit from the identification of shortcomings associated with the anti-forensic techniques. Similarly, system and software developers can use the information in this report to develop hardware and software to make anti-forensics easier or more difficult. Lastly, anyone concerned with increased privacy will benefit from the knowledge included in this report.

Literature Review

Introduction

The following literature review is intended to provide associated contextual information and an overview necessary to understand the research problem: *mitigating the impact of anti-forensics through file system analysis*. An overview of the three most common operating systems will focus on the file systems used by each one. Understanding the file system used by an operating system provides a basis for the knowledge needed to recognize the impact of anti-forensic techniques.

Next, this literature review will discuss some of the general information associated with computer forensics. A thorough explanation of the goals of forensic investigations will be given. Also, an introduction will be provided on some of the challenges an investigator may face. The background information on computer forensics illustrates how anti-forensic techniques complicate an investigation, and why it is important to mitigate their impact.

This literature review will also give a detailed explanation of *anti-forensics*. The methods of anti-forensics focused on in this research will be discussed in three main categories of altering metadata, encryption, and deletion. In addition to an overview of the anti-forensic techniques, this literature review will also cover some of the desired goals associated with these three categories of anti-forensics. This information provides the reader with a detailed understanding and intended outcome of an anti-forensic technique.

Finally, this literature review will focus on the information relative to the research questions: can anti-forensic techniques be detected through file system analysis, and how can the investigator mitigate their impact on the investigation? The goal of this literature review

is to provide the readers with a detailed understanding of the anti-forensic techniques as well as indications in the file system of them being implemented.

Operating Systems

The operating system (OS) of a computer provides the translation from user-level actions to instructions the hardware understands. Stallings (2009) defines an OS as "… a program that controls the execution of application programs and acts as an interface between applications and the computer hardware" (p.51). The OS makes it possible to run other programs by managing the use of hardware between system functions and the user's applications (Dhotre, 2009). A user's actions on the computer would not be understood at the hardware level if not for the OS. One capability that the OS makes possible is the ability to accurately interact with files stored by the computer.

The file system associated with an OS is the method by which files are stored, cataloged, and consistently retrieved. A file system permits users to create, delete, open, close, read, and write files to a storage device. The file system also generates and maintains metadata about each of the stored files (Stallings, 2009). The information maintained by the file system provides an investigator with additional details for each file. Since file management is unique to each OS, different operating systems sometimes implement different file systems.

The personal computer market is comprised of operating systems that fall into two main categories: Windows-based and UNIX-based. Microsoft Windows began as *Microsoft Disk Operating System*, and has changed over the years to include several core OS's with different versions of each. UNIX was developed in 1970, but has evolved over the years to

form numerous variants; two of which include Linux and Mac OS (Stallings, 2009). The variety of operating systems available creates a fluid environment for investigators. Fortunately, Windows, Linux, and Mac OS make up the vast majority of operating systems installed on desktop systems.

Windows. Windows has operated on top of two main file systems throughout its existence. Windows relied on the *File Allocation Table* (FAT) for its file system until the *New Technology File System* (NTFS) was implemented with Windows NT and later versions (Carrier, 2005). Some of the benefits associated with the newer NTFS include improved functionality, new capabilities, and increased capacity. The simplicity of the FAT file system has ensured it remains in use on external USB drives, and therefore is still prevalent in investigations.

FAT. The different versions of FAT have unique characteristics and applications. FAT12, FAT16, and FAT32 are the three versions of FAT and differ in the size based on the entries of their associated FAT structure (Carrier, 2005). FAT12 is used for volumes smaller than 16 megabytes (MB) in size while FAT16 can be used on volumes from 16MB up to 4 gigabytes (GB). FAT 32, on the other hand, can be used on volumes from 33MB up to 32GB ("File systems," n.d.). While there are some different characteristics of the FAT versions, the file system maintains files the same way in each version. The information for each file is stored as an entry in the *file allocation table* on the volume.

The file allocation table is the root source of information for all files stored on a volume formatted with the FAT file system. Files and directories on a FAT volume each have corresponding directory entries with file name, size, starting address of the content, and

metadata in the FAT (Hull, 2009). The information in an entry is used by the OS for file management, but can also be used by an investigator when necessary. While FAT is still being used primarily on removable media, the primary file system for Windows is NTFS.

NTFS. NTFS increases the maximum volume capacity, and adds new functionality from that of the FAT file system. The NTFS format is capable of handling volumes up to 16 exabytes (EB) *in theory* ("File systems," n.d.). In addition to the file information found in a FAT entry, NTFS is also capable of tracking the last access time for a particular file if enabled in Windows (Casey, 2010). The improvements to the file system required the development of a new way of maintaining file management. File management in NTFS is made possible by the *master file table* (MFT).

The MFT is a structured approach to maintaining a list of files, directories, and their associated attributes on an NTFS volume. Each entry in the MFT is exactly 1024 bytes in length, and is comprised of a number of attributes that have specific purposes (Casey, 2010). Attribute headers are used by the operating system to identify one entry from the next in the MFT (Medeiros, 2008). For example, the second portion of an MFT entry can contain date and time stamps, file contents, alternate data streams, and a number of other possible attributes. While some other operating systems can read and modify NTFS, the UNIX based operating systems employ a different type of file system.

Linux. The open source architecture of Linux allows a multitude of different versions of the operating system that supports a wide variety of file systems. Over 600 different Linux distributions are available for devices from game consoles to super computers (Goldsborough, 2011). The majority of the distributions default to the Third Extended (Ext3)

file system for the volume where the OS is installed (Siever, 2009). The Extended (Ext) file system serves as an appropriate base for Linux investigations due to the fact that the majority of the distributions support it. Like the FAT file system, the Ext file system has a few different versions with similar capabilities.

Ext. Compatibility concerns ensure that the different versions of the Ext file system are still encountered today. The first Ext file system was created in 1992 with three revisions following in 1993 for Ext2, 2001 for Ext3, and 2008 for Ext4. Ext4 was specifically designed to be backward compatible with more common Ext3 (Jones, 2009). The differences between the three versions are primarily increased capacity and additional features. However, Ext4 does alter the time stamps in the file system to mark down to the nanosecond. For the purpose of this research, the focus will be on the more common Ext2 and Ext3.

The Ext file system stores information about the files in a different manner than that of NTFS or FAT. The core of Ext is stored in three sections; data blocks containing the actual file contents, inodes containing file metadata, and directories containing filenames (Carrier, 2005). The metadata associated with a file in this case includes the modified, accessed, created, and deleted times (Casey, 2010). It is important to note that the deleted time stamps are not part of either Windows file system. Moreover, the filenames are not a part of the same structure, and therefore must be referenced in a different manner. While Macs are also UNIX based, they work on an entirely different file system.

Mac. Apple is based on an open source file system, but the distributions are more tightly controlled than those of Linux. The current version of software for Apple computers is based on Mac OS X that was first released in 2001 (Altheide & Carvey, 2011). The file

system associated with the Mac OS has remained relatively unchanged over the years. Much like Windows, improvements in technology pushed Apple to move from the *Hierarchical File System* (HFS) associated with earlier operating systems to the current HFS+ file system.

HFS. HFS+ is the default version of the file system that is found in all current Mac operating systems. With the release of Mac OS 8.1, HFS+ replaced HFS as the default file system for all future OS versions. The newer version of the file system incorporated improvements such as support for longer file names, larger files, and multiple data forks (Craiger & Burke, 2006). The enhancements associated with HFS+ have all but eliminated the use of HFS on newer systems. HFS+ approaches file management in a similar fashion as that of NTFS.

The *catalog* file on an HFS+ formatted volume contains the majority of the information necessary for file management. The records that make up the catalog file on Mac OS X are 8 kilobytes (KB) in length and include the file ID, the parent ID, time metadata, and information on the data of the file (Altheide & Carvey, 2011). Also, the catalog file is stored in a balanced tree (B-tree) structure thatkeeps the records sorted to improve searching ("File deletion in," n.d.). The catalog file is a good source of information during an investigation, but some information can be lost if the B-tree is rebalanced. The file system of an OS has a tremendous impact on how a forensic investigation is performed.

Digital Forensics

Digital forensics is an extension of the term computer forensics. Digital forensics refers to the process of acquiring evidence from any electronic device while computer forensics focuses on acquiring evidence from computer systems (Reith, Carr & Gunsch,

2002). The distinguishing line between these two methods is blurred by the interconnectivity of devices. For the purpose of this research, these two terms will be considered synonymous, and may be used interchangeably. Some common goals exist in investigations involving digital and computer forensics.

Goals. The goals in an investigation can vary depending on the circumstances surrounding the events. Corporate environments may use the results of an analysis to protect company interest, while law enforcement may use the results to prosecute a suspected criminal (Steel, 2006). Even though the ultimate goals may be slightly different, some procedures may be the same in every case. Some typical components of an investigation are *uncovering hidden files*, *recovering deleted files*, and *proving ownership* of data.

Uncovering hidden files. Hidden files exist in all computers, but not all hiding methods create the same amount of privacy. Most operating systems allow the user to hide a file by simply changing its attributes, but this method is very easy to uncover. However, files can also be hidden through the use of encryption, steganography, or Alternate Data Streams (ADS). ADS are associated with Windows NTFS, do not show up in *Explorer*, are not limited in size, and can be any form of binary data (Gupta, n.d.). Every method of hiding files requires a different approach in an investigation. Similarly, deleted files can exist in a number of different states in an investigation.

Recover deleted files. Deleting files from a computer system is a common way to remove signs of activity on the system. These deleted files can be as innocuous as a file that is no longer needed, or an attempt to remove signs of illegal content. Users that share access on a computer are encouraged to remove traces of Internet activity before leaving the system.

The same reasons users delete files are the same reasons an investigator recovers them. Like hiding files, files can be deleted in different ways. The apparent actions on a computer can vary drastically with the recovery of deleted files.

The deleted files on a suspect's machine are used to create a more accurate depiction of the true events on the machine. Deleted files can include Internet history, personal user files, event logs, and a number of other data types (Kruse & Heiser, 2001). Whether the files are deleted by the user, or through standard interaction with the computer, these files can contain the true events of the computer. Once the events of the computer are established, the analyst has to identify the user responsible for each action.

Prove ownership. Every action on a computer will have either a user account or a built-in system account associated with it. When an investigator learns what typical actions are associated to the built-in accounts, they will recognize those actions that need further investigation. For example, some system-driven actions are normal while others may be the result of a virus. According to Gerald King (2006), "The investigator should always attempt to prove ownership of the evidence" (p. 23). There are a number of ways to prove ownership in different cases, but it will depend on the events associated with each investigation to determine what approach to take.

Proving ownership is not always as straight forward as looking at the username associated with a file. Recovered files do not usually include the metadata originally associated with them. Also, simply because the file is in a user's personal folder does not necessarily mean the file is theirs. Some questions to help assign ownership include; who was using the computer, where the file was located, who owns the file, and is the file in a

secured location (King, 2006). It may take a combination of these questions to definitively assign ownership to a file or action, and even then some doubt may exist. The difficulty of producing an accurate analysis can be complicated by additional challenges.

Challenges. Forensic investigators encounter many challenges; some challenges can be compounded due to the actions of others. Certain actions by users are intended to increase the difficulty of an investigation. However, actions by inexperienced or untrained users can also inadvertently complicate an investigation. For example, a user deleting files from their computer and an inexperienced agent removing power from a running machine will destroy potential evidence. In some cases, the accidental damage can be more difficult to mitigate than a user's attempts at hiding their actions. Two areas of challenges that can drastically alter an investigation are *anti-forensics*, and the *environment*.

Anti-forensics. Any attempt by a user to discredit an investigator's analysis can be considered anti-forensics. Anti-forensic techniques attempt to manipulate, erase, or obfuscate data to the point where the investigation becomes difficult, time consuming, or near impossible (Barbara, 2008). These techniques can be applied to the data being analyzed, or to the tools used by the investigator. Some anti-forensic techniques leave traces of evidence indicating the data has been modified in some way.

Learning to recognize signs of anti-forensics is one step in being able to mitigate impacts. Some signs of anti-forensics can be subtle while others are more obvious. For example, the presence of certain software or mysteriously empty logs can be subtle indications. However, attempts at physical destruction or removal can be more obvious attempts at anti-forensics. Knowing what actions have been taken to hide data helps an

investigator develop methods to work with these challenges. In addition to anti-forensic techniques employed by users, the environment of the investigation provides additional challenges.

Environment. The environment for a digital investigation that an investigator needs to take into consideration during their analysis extends beyond the computer. The environment of an investigation can consist of physical and logical areas, and it can change during the course of the investigation (Steel, 2006). The logical areas include the operating system along with any third-party software that is found on the system. External devices must also be considered as part of the hardware in addition to the actual digital device. For example, USB drives, external DVD burners, and scanners can be an important addition to the investigation. One key component of the software portion of an investigation is the OS.

Operating system. The operating system installed on a computer will dictate several components of an investigation. Each operating system behaves differently, and even different versions of the same operating system can perform in dissimilar ways. It is important to note that the default file system changed from FAT to NTFS with Windows NT and later versions (Carrier, 2005). On the surface different versions of Windows may appear similar, but the detailed information may be completely different. Likewise, although Mac and Windows both provide a native way to recover deleted files, the approach to analyze the Windows Recycle Bin may not be the same as that of Mac Trash. The hardware involved in the investigation often compliments the software in that installed software can indicate hardware to look for.

Hardware. The physical devices that are part of the investigation require appropriate considerations during evidence gathering and analysis. Some hardware in the environment may not be directly connected to the computer, but will be implicated by unique software such as phone backup software. However, not all hardware included in the environment will have software associated with it. A solid state drive (SSD) is one relatively new piece of hardware that is part of the environment, but will not have unique software associated with it.

SSDs can effectively destroy evidence without any interaction if they are not given special attention during an investigation (Bell & Boddington, 2010). While the environment may change during the course of an investigation (Steel, 2006), understanding the hardware and the rest of the environment will ensure as much evidence is collected as is possible. Some techniques are being implemented which are detrimental to a forensic investigation.

Anti-forensics

A forensic investigation can be attacked through the evidence or through the tools used during the investigation. While sometimes considered synonymous, counter-forensic methods are attacks on the tools used to conduct an investigation, and anti-forensic techniques are attacks on the actual evidence (Hartley, 2007). An attack against the tools being used has limited applicability if multiple tools are used to accomplish the same goal. However, attacking the evidence will impact all of the tools in at least some way. This paper will focus on three methods of anti-forensics that can be encountered in an investigation.

Methods. There are several ways a user's actions can affect the outcome of an investigation. Anti-forensic techniques can include traditional methods such as deletion and encryption, and also methods that minimize the evidence left behind for analysis. Traditional

methods of anti-forensics are easier to spot when the investigator is aware of their presence (Garfinkel, 2007). Altering a file's metadata is one method of anti-forensics that complicates an investigation.

Altering Metadata. A file's metadata provides information that can be used to assign ownership, correlate events, and detect modifications. Altering metadata can corrupt the validity of the real evidence thereby rendering it worthless to an investigation (Harris, 2006). Metadata can be altered by wiping the entries out or by replacing the entries with valid, but incorrect, entries that are different from the originals. Basing an investigation on altered metadata can cause some evidence to be overlooked or even the wrong person to be associated with the actions. Another method of anti-forensics that can complicate an investigation is encryption.

Encryption. Legitimate user files can appear meaningless when encryption is used to mask the content. The appearance of a file or files to an examiner will depend on how the encryption is applied. Also, the encryption method will determine if or when the files may be in plain text. Some encryption programs can encrypt single files, virtual disks, or the entire volume (Yegulalp, 2008). When the entire volume is encrypted, the decryption occurs during the boot process and the content of the volume remains in plain text while the system is running. However, when individual files are encrypted, they are only in plain text when they are in use. Encryption is an effective method of securing data that is still needed, but data that is no longer necessary can be deleted.

Deletion. Deleting incriminating files from a computer can severely complicate an investigation when done properly. The contents of files, which have been deleted, are not

really removed from the storage volume (Max, 2009a). This includes files that have been removed from Windows' *Recycle Bin* and *Mac Trash*. However, advanced methods of deletion referred to as secure deletion involve writing over the contents of a file with irrelevant data. Secure deletion can include one or more passes of writing patterns of data over the contents of a file, multiple files, or an entire disk (Innes, 2005). Files that have been overwritten with other data are considered unrecoverable by traditional forensic methods. Deletion, like other methods of anti-forensics, is a method which users implement to attain one or more goals.

Goals. As previously mentioned, the goals associated with anti-forensic techniques are not necessarily nefarious in nature. Some goals of anti-forensics are designed to complicate the investigation process. However, privacy is one area where methods of anti-forensics can be applied without being considered an attempt to hide something. While some people make the argument that privacy is not necessary if there is nothing to hide, others also are concerned that their actions may be misconstrued if taken out of context (Caloyannides, 2004). Regardless of the reason, privacy is one of the main goals of anti-forensic techniques.

Privacy. There are a number of ways in which a user may employ anti-forensic techniques to increase their privacy. Each of the three methods of anti-forensics mentioned in this paper can be applied to establish a level of privacy for the user. The metadata associated with a digital photograph can be altered or removed to mask when and where the picture was taken. Encryption and secure deletion can each be applied to electronic banking documents to keep their content away from prying eyes. In addition to privacy, anti-forensic techniques are also used to destroy evidence before an investigator performs their analysis.

Destroying Evidence. The thorough analysis of a suspect's computer relies on the investigator having access to all of the evidence. When metadata is altered to the point it is removed or unrecognizable, the evidence can be considered destroyed. Obviously, the deletion of files is a form of anti-forensics that can be directly applied to the destruction of evidence. Encryption can also be considered a simple form of data destruction in that the data is unreadable when encrypted (Max, 2009a). However, the argument can be made that the evidence can be recovered if it is able to be decrypted. One other goal of anti-forensic techniques is to misdirect an investigator.

Misdirection. The complete destruction of evidence can sometimes raise suspicion during an investigation while misdirection can lead an investigator to the wrong conclusion. Altered metadata can be used to give the appearance a file was created at an incorrect time, or a picture was taken at an alternate location. Some encryption programs allow a user to place encrypted containers within other encrypted containers to establish plausible deniability ("Plausible Deniability, n.d.). Misdirection can give an investigator the illusion that all of the evidence was correctly collected; when in fact the evidence had been altered. It is up to the investigator to recognize the signs of anti-forensics, and mitigate their impact on the investigation.

Mitigating Anti-Forensics

The behavior of any anti-forensic technique can change when it is applied under different circumstances. Files that have been deleted from a volume formatted for Macintosh may appear differently from those deleted from Linux or Windows. Likewise, some tools are designed to completely remove traces of previously deleted files by overwriting the file's

entry in the file table (Kessler, 2007). Learning to identify the signs associated with an anti-forensic technique is one step to developing countermeasures for them. The file system determines where and how files are stored on a volume, and therefore is a logical source for identifying anti-forensic techniques.

File System Analysis. The file system is responsible for the location and state of every file on the volume. Even files hidden from the user will be referenced in areas such as the Master File Table (MFT) where they can be examined for non-standard NTFS flags (Harris, 2006). Also, some file systems retain a certain amount of information about files that have been deleted from the volume ("File deletion in," n.d.). The fact that the file system is relied on for interaction with the all the files on a volume makes it a possible source of evidence of anti-forensics. Each file system depends on a file, catalog, or table of some sort to store information about the files on that volume.

File Tables. The only way a computer is able to accurately store and recover files to and from a volume is by recording certain information about those files in a type of table. NTFS uses the MFT to track the files on a volume, and to store the metadata associated with those files. The Linux Ext2 and Ext3 file systems rely on an *inode* file for the same functionality. Lastly, the Apple HFS+ file system uses the *catalog* file to track files and metadata on its volume (Casey, 2010). The information found in each one of these tables can help indicate when certain anti-forensic methods have been used to hide user actions. The information necessary to mitigate the impact of the anti-forensic technique will vary from case to case.

Altering metadata, encryption, and file deletion all modify the associated file table in some way. Each of the file systems in this research paper retain date and timestamps for when a file is modified, accessed, and created. However, some of these features can be disabled by options in the OS. A simple utility can be used to modify the Windows Registry setting and disable the method used to update the LastAccess timestamp stored in the MFT (Max, 2009b). Knowing what metadata is expected to be in the file table, and how the data is updated will make it possible for the investigator to identify signs of anti-forensics. In addition to knowing what information should be available to an investigator, it is important to understand the anti-forensic techniques.

Understanding Anti-forensic Techniques. The implementation of an anti-forensic technique can change, but the underlying methodology remains the same. Full disk encryption and file encryption interact with the OS in very different ways, but the encryption methods used in each can be the same (Lubert, n.d.). While the user can apply anti-forensic techniques in a variety of ways, the investigator can anticipate certain activities with a thorough understanding of the technique. Altering metadata is one technique that can be implemented in different ways.

Altering Metadata. A user can alter the metadata for a file in two main areas. File system metadata is the date and timestamps that are associated with a file and stored in the file table (Buchholz & Spafford, n.d.). Some files also have metadata embedded as part of the actual file. For example, image files can contain metadata about the camera that took the original picture. While some files may contain additional metadata, all files will have file

system metadata associated with them. One way to alter the file system metadata for a file is through the use of external programs.

Third-party software can modify the file table information associated with a file and affect an investigation. TimeStomp can be used to change the file system metadata for a file to any specified date, or to blank the date entirely ("Anti-forensics with timestomp," n.d.). This particular method of altering metadata does not change the $FILE_NAME attribute, and therefore the investigator can use the $FILE_NAME attribute for a more accurate time (Carvey, 2010). Knowing what metadata values are changed under certain circumstances can give insight when the metadata is not in its original form. Encryption also is a method of anti-forensics that can have different artifacts in the file system.

Encryption. Encryption not only makes it difficult or impossible to view a file, but it can also mask the fact that a file even exists. Once initially setup, cryptographic methods can run with little maintenance making them perhaps the most troublesome anti-forensic method (Kessler, 2007). An encrypted file saved to a disk appears as meaningless data when viewed using a hex editor or similar tool. Signs of encryption software or techniques are an indication to an investigator that special attention needs to be given to the file system in an attempt to identify encrypted files. Once files are no longer needed on a volume, the user will often delete them to cover their tracks.

Deletion. An investigator cannot analyze files that no longer exist on a volume. Files that have simply been deleted by a user still exist on a volume until another file uses the physical location. An additional step in this anti-forensic technique is to overwrite the file with meaningless data (Korso, 2009). Once the files have been wiped from the volume they

are considered unrecoverable. Evidence of wiping a file may be found in the file table, and in some cases some of the file information may still be available. The more information known about a particular anti-forensic technique, the more likely an investigator will be able to limit its impact on an investigation.

Discussion of the Findings

Major Findings

A digital forensic investigation can become complicated or even impossible when anti-forensic techniques have been used on the computer. Anti-forensic techniques are aimed at destroying or corrupting possible evidence (Hartley, 2007). Some anti-forensic techniques can be detected by investigators; thereby mitigating their impact on the investigation. However, as the number of digital forensic cases involving the use of anti-forensic techniques increases, the legitimacy of digital evidence is being questioned (Bayuk, 2010, p.116). Understanding an anti-forensic technique and the associated environment will help an investigator produce a sound analysis. The environment is a key component to understanding the expected behavior of an anti-forensic technique.

Understanding the Environment. Its OS, other software, and the hardware define the functionality of a computer. The sources that were chosen for this research focused on the three most common operating systems installed on personal computers and their associated file systems. Nevertheless, other factors in the environment can be used to detect anti-forensic activities. Secure deletion software and encryption software can be indicative of anti-forensic activities on the system. Also, USB drives and solid state drives may change the approach necessary to mitigate the impact of some anti-forensic techniques. Identifying NTFS flags that are out of the ordinary is one countermeasure made possible through understanding the environment (Harris, 2006).

Understanding the Technique. The sources used in this research explain the goals of each anti-forensic technique as well as some countermeasures. The methods of anti-

forensics focused on in this research were altering metadata, encryption, and deletion. Privacy, destruction of evidence, and misdirection are goals associated with one or more of the researched anti-forensic techniques. Understanding how an anti-forensic technique alters metadata, encrypts data, or deletes a file makes it possible to circumvent the attempts at times.

Theme One: NTFS Anti-forensic Techniques

Anti-forensic techniques used in a modern Windows environment will impact the NTFS in some way. Evidence of the three common anti-forensic techniques included in this research can be seen in the file system if the investigator knows what to look for. Understanding what to look for is dependent upon the implementation of the particular method of anti-forensics. The primary place in the NTFS where evidence of anti-forensic techniques can be detected is in the MFT. However, each of the researched methods of anti-forensics had different indicators in the MFT. Altering metadata in NTFS resulted in a number of different indications depending on the implementation.

Altering Metadata. While it is possible to modify the metadata stored in the MFT manually, external programs make it easy enough for a non-technical user. TimeStomp is one program that can be used to alter the file system metadata. This utility allows the user to modify the stored metadata to anything they desire, or to remove it entirely ("Anti-forensics with timestomp," n.d.). This method of altering metadata will create anomalies in the attributes associated with a file as well as surrounding files in the MFT. For example, the $FILE_NAME attribute in the MFT will not match the modified entries, and can be used as a more accurate time (Carvey, 2010). Also, surrounding files in the MFT may indicate when

the intended file was actually created on the volume. When a new file is created on an NTFS volume an associated entry in the MFT is created at the first available location. While altering the metadata may be misleading, it does nothing to hide the true contents of a file. Encryption is commonly used to hide files the owner does not want anyone else to access.

Encryption. Encrypted files on a volume can be impossible to view, and can even go undetected in some cases. Different software and methods of encryption can impact the file system in different ways. Some encryption schemes will encrypt the entire volume when the computer is powered down and decrypt the contents at boot while others only encrypt and decrypt individual files as needed. Also, some methods of encryption store the filenames in plain-text (Carrier, 2005, p.142). The plain-text filename is not necessarily enough evidence for a conviction, but it can be used to corroborate other information. In some cases, the only thing the investigator can definitely state is that encryption methods were used. Once a user no longer needs a file, they will often delete it all together.

Deletion. Like encryption, deleted files can come in many forms with varying degrees of complexity. Simple file deletion provides a minimal amount of anti-forensic functionality. The file that is simply deleted still resides on the disk and has an entry in the MFT until each one is overwritten with new data (Max, 2009a). The window of opportunity for an investigator to recover the deleted file will depend on the activity on the system. Wiping files on the other hand, writes over the desired file with irrelevant data in the course of deletion. In cases of wiped files the data is considered to not be recoverable, but some wiping programs can leave some information behind in the associated MFT entry (Russinovich, 2011). Again, the MFT entry will be overwritten as enough files are created on

the disk. One common file system associated with the Linux operating system, Ext3, shares some similarities for mitigating anti-forensic techniques as those observed in NTFS.

Theme Two: Ext Anti-forensic Techniques

While there are a number of file systems that Linux can be installed on, most installations default to the Ext3 file system (Siever, 2009). Like NTFS, evidence of anti-forensic activities can be detected in the core of the Ext file system. The Ext file system maintains the content of the volume in three sections - the file contents are stored in data blocks, metadata about the files are stored in inodes, and the filenames are stored in directories (Carrier, 2005). The anti-forensic technique used will dictate which area needs to be examined. In terms of file system metadata, the inodes can indicate when a user has attempted to alter the metadata.

Altering Metadata. Linux operating systems include a built-in application that can be used to alter some of the metadata associated with a file. The Ext3 file system includes timestamps for modified, accessed, and created times as well as an additional deleted timestamp. The *touch* command can be used to manually set the modification or access timestamps associated with a file. However, this method of altering the metadata inadvertently updates the change timestamp of the file (Casey, 2010). A comparison between the changed timestamp and the modified or accessed timestamps can indicate when metadata has been altered. Based upon the method used for encryption, all three core sections of the Ext3 file system can be used to show when encryption has been used.

Encryption. The strength of encryption as an anti-forensic technique on an Ext3 volume is comparable to that of NTFS. Attempts to mitigate full disk encryption fall into two

categories: obtain the data while it is in plain text, or obtain the encrypted data and try to obtain the decryption key (Petullo, 2004). The same can be mitigation methods can be applied to individually encrypted files, but in some cases the metadata in the inode and the filename in the directory may be stored as plain-text. Again, this information can be used to corroborate other evidence when the plain-text data cannot be obtained. Deleted files are also an equal threat in Ext3 as they are in NTFS.

Deletion. Any files that have been deleted from an Ext3 file system cannot be recovered with a built-in function. When a file is deleted on an Ext3 file system, the links to the data and the file size in the inode is zeroed out. However, the Ext3 file system incorporates a *journal* that stores changes on the file system (Narvaez, 2007). The journal can be examined to recover some information pertaining to the deleted file. The amount of information that can be recovered will depend on the level of journaling enabled on the system. The journal is limited in size and will overwrite its entries as needed, and therefore the window of opportunity to identify or recover deleted files is still limited by the activity of the system.

Theme Three: HFS+ Anti-forensic Techniques

Most Apple operating systems rely on the HFS or HFS+ file system to manage interactions with the hardware. HFS+ became the default file system with the release of Mac OS 8.1, and has nearly eliminated the use of HFS. The core structure of the HFS+ revolves around the catalog file, and can be used similar to that of the MFT to identify the presence of anti-forensic techniques. The catalog file maintains a list of entries for all files and directories

on the volume along with their associated attributes. A user's attempt to alter the system metadata associated with a file can sometimes be seen in the catalog file.

Altering Metadata. During the course of this research there were no specific programs found to alter system metadata on HFS+ volumes. Likewise, there has been minimal research performed to identify when the user has altered the metadata. The research that has been performed resulted in different tools interpreting the date and timestamps in different ways (Casey, 2010). The limited research and tools complicate the process of detecting and mitigating the impact of altered metadata. However, some characteristics of the catalog file can be an indication that the metadata has been modified.

Every entry in the HFS+ catalog file is assigned a Catalog Node ID (CNID). The CNIDs are sequentially assigned to an entry as it is created, and none are reused until every available ID has been used (Altheide & Carvey, 2011). Since CNIDs are not reused, the relationship between CNIDs of two files should correspond to that of the creation dates found in the catalog file. Likewise, creation dates that are more recent than the associated CNID indicate manipulation of the metadata. Encrypted files on an HFS+ volume have the same complications as those on other volumes.

Encryption. Much like altered metadata, there has been little research into the impact of encrypted files on a HFS+ volume. Third party applications are capable of encrypting singular files as well as the entire disk. It is unknown if any applications leave the filename or other file attributes in plain-text while a single file is encrypted. However, like other file systems, whole disk encryption stores the entire volume in an encrypted form until the

correct key is entered for decryption. Files that have been deleted from an HFS+ volume have challenges beyond encrypted files on the system.

Deletion. The behavior of the catalog file when files are created and deleted can be used to identify when files have been deleted. Unlike the MFT, the balanced tree structure of the catalog file can cause the catalog file entries associated with a deleted file to be overwritten by the file system ("File deletion in," n.d.). The shorter window of opportunity to identify deleted files by their catalog file entries decreases the likelihood of mitigating the anti-forensic technique in this manner. Sequentially assigned CNIDs will indicate when a file has been deleted, but this does not necessarily mean the file was deleted to destroy evidence.

Recommendations and Conclusions

Anti-forensic techniques can remove the possibility of a complete and thorough investigation by even the most experienced investigator. One way the impact of these anti-forensic techniques can be mitigated is by understanding the implications surrounding those anti-forensic methods. An anti-forensic technique is a method designed to destroy evidence (Hartley, 2007). By this definition, three anti-forensic techniques include *altering metadata*, *encryption*, and *file deletion*. These three methods can be accomplished with the use of simple programs that do not require a technical background.

The purpose of this research was to analyze how anti-forensic techniques impact the three most prominent operating systems. Specifically, this research addresses the questions: can the use of anti-forensic techniques be detected through analysis of the file system, and are there techniques an investigator can use to mitigate their impact on the evidence?

Recommendations

This research has resulted in a number of possible methods that can be used to mitigate the impact of anti-forensic techniques an investigator may encounter. The methods discussed in the research are not the only ways to mitigate the associated anti-forensic techniques. Likewise, the anti-forensic techniques covered in this research are not the only ones that can complicate an investigation. In addition to direct application in an investigation, the results of my research should be used as a baseline for additional research into the mitigation methods as well as the anti-forensic techniques. Identifying and possibly mitigating a user's attempts at hiding, destroying, or creating misleading evidence is an important part of an investigation.

Investigation Application. The mitigation methods highlighted in this research can be applied to investigations involving the major file systems for Windows, Linux, and Mac OSX. Including the file system and associated files, as part of the investigation can help identify when these anti-forensic techniques have been used. Under the correct circumstances, analysis of the file system can completely negate the anti-forensic technique. In this research the focus was as broad as possible in order to achieve results that are applicable to numerous investigations. For example, the exact version of the OS and any third-party software was not specified, and the results therefore are not dependent on them. Continued research based on my findings could produce consistently reliable countermeasures under less generic circumstances.

Continued Research. The number of variables associated with any given anti-forensic technique are too great to cover in a single report. This research indicated that different methods of altering metadata could have a dramatically different impact on an investigation. The different environments encountered by an investigator can be researched to identify the specific indications that can be expected. This research should be focused on different encryption techniques to identify which ones leave the filename in plain text, as indicated by Brian Carrier (2005, p.142). Alternatively, additional research could identify specific indications associated with encountered software and OS combinations. In addition to the method used in this research, there are different methods that could have been incorporated.

Alternative Research Methods

This research was based entirely on the work performed by experts in the fields of digital forensics and anti-forensics. This method of research forces the analysis to be based on the findings of others. Also, the research is then confined to that which already exists in the forensic community. However, a benefit of this style of research is that it allowed analysis of techniques and environments that are otherwise inaccessible. One alternative to my style of research is to study the anti-forensic techniques in a controlled environment.

Analyzing an anti-forensic technique in a controlled environment can illustrate exactly what to expect from an investigation under similar circumstances. Unlike the method used in this research, the results of a controlled study are more detailed in some regards. The details associated with controlled study results do not come without a cost. Though, as the results become more detailed they can lose their applicability. For example, analyzing how Timestomp alters metadata in the MFT does not necessarily apply to other methods of altering metadata or to other file systems. Whether the research method is a controlled study or analysis of existing data, most methods have associated limitations.

Limitations

Over the course of this research a few limitations were discovered that can be associated to the applicability and the level of detail related to the results. The focus of this research was on the file systems associated with the three most common operating systems. Windows and Mac OS X are almost exclusively installed on the NTFS and HFS+, respectively. On the other hand, Linux is typically installed on Ext3, but the 600+ distributions of Linux support a wide variety of file systems (Goldsborough, 2011). The

results of this research can be applied to the associated file systems and may be applicable to others, but the behavior of these anti-forensic techniques is unknown in other cases. Similarly, the anti-forensic techniques were analyzed as a concept rather than through individual tools. The methods used by third-party tools to accomplish an anti-forensic technique can vary in their indications. The method of research in this report is also limited by the amount of available information.

According to the results of this research, the popularity of an OS is proportional to the amount of forensic research available. The likelihood of an investigation involving the Windows OS is significantly greater than that of an investigation involving Linux or Mac OSx. While the research focusing on NTFS did contain an abundant amount of information, continued research is necessary for further understanding of the techniques. The information available on HFS+ and the Ext3 file system was limited in this research. As these file systems continue to gain popularity, the likelihood of encountering them will increase. Research on these two file systems should increase to match their usage.

Conclusion

Under the right circumstances it is possible to identify, and sometimes mitigate, the usage of anti-forensic techniques. Of the methods of altering metadata covered in this research, a comparison to other values found in the file table can be used as a countermeasure against the anti-forensic technique. Encryption as an anti-forensic technique is very effective at hindering an investigation. File system analysis will provide little more than an affirmation that encryption has been used. Additional research is necessary to be able to determine if file system analysis can identify different methods of encryption. Files that have been deleted

from the researched file systems include additional complications. File table entries, for deleted files, are marked for deletion even when the data may remain on the volume. These entries are then overwritten as needed, effectively removing traces of the files. The results of this research are an additional step in the right direction to mitigating the impact of anti-forensics on digital investigations.

The link between a forensic investigation and the encountered anti-forensic techniques is that of a digital game of cat-and-mouse. Criminals improve and create new anti-forensic techniques in an attempt to hide their true actions in a digital environment. The forensic community is constantly faced with the challenge of mitigating the impact of the changing anti-forensic techniques. This project looks at three areas of anti-forensic techniques that are simple enough to be used with minimal skill. The goal of this research is to establish a basic level of understanding on these techniques that can then be used by the forensic community to diminish the gap between the criminals and the investigators.

References

Altheide, C., & Carvey, H. (2011). *Digital forensics with open source tools*. Burlington, MA: Syngress.

Anti-forensics with timestomp. (n.d.). Retrieved from http://www.isdpodcast.com/resources/training/anti-forensics

Barbara, J. J. (2008, December 01). *Anti-digital forensics, the next challenge*. Retrieved from http://www.dfinews.com/article/anti-digital-forensics-next-challenge

Bayuk, J. (2010). *Cyberforensics: Understanding information security investigations*. New York: Humana Press.

Bell, G. B., & Boddington, R. (2010). Solid state drives: The beginning of the end for current practice in digital forensic recovery?. *The Journal of Digital Forensics, Security and Law*, *5*(3). Retrieved from http://researchrepository.murdoch.edu.au/3714/1/solid_state_drives.pdf

Buchholz, F., & Spafford, E. (n.d.). *On the role of file system metadata in digital forensics*. (Master's thesis, Purdue University) Retrieved from http://homes.cerias.purdue.edu/~florian/publications/metadata_jdi.pdf

Caloyannides, M. A. (2004). *Privacy protection and computer forensics*. (2nd ed.). Norwood, MA: Artech House, Inc.

Carrier, B. (2005). *File system forensic analysis*. Boston: Addison-Wesley.

Carvey, H. (2010, May 21). [Web log message]. Retrieved from http://windowsir.blogspot.com/2010/05/analysis-tips.html

Casey, E. (2010). *Handbook of digital forensics and investigation*. London, England: Elsevier Academic Press.

Craiger, P., & Burke, P. (2006). Mac OSX forensics. In M. Oliver & S. Shenoi (Eds.), *Advances in Digital Forensics II*, pp. 159-170. New York, NY: Springer.

Dhotre, I. A. (2009). *Operating systems*. (8th ed.). India: Technical Publications Pune.

File deletion in various filesystems. (n.d.). Retrieved from http://www.reclaime.com/library/file-undelete.aspx

File systems. (n.d.). Retrieved from http://technet.microsoft.com/en-us/library/cc958073.aspx

Garfinkel, S. (2007). Anti-forensics: Techniques, detection and countermeasures. In L. Armistead (Ed.), *ICIW 2007: 2nd International Conference on i-Warfare and Security : Naval Postgraduate School, Monterey, California, USA : 8-9 March 2007,* pp. 77-84. Reading, UK: Academic Conferences Limited.

Goldsborough, R. (2011). Twenty years of Linux. *Tech Directions, 71*(3), 12-12. http://search.proquest.com/docview/896361626?accountid=28902

Gupta, C. (n.d.). *Disecting ntfs hidden streams*. Retrieved from http://www.forensicfocus.com/dissecting-ntfs-hidden-streams

Haberland, J. (2004, October 14). *Linux ext3 faq*. Retrieved from http://batleth.sapienti-sat.org/projects/FAQs/ext3-faq.html

Harris, R. (2006). Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem. *Digital Investigation*, *3S*, 44-49. Retrieved from http://dfrws.org/2006/proceedings/6-Harris.pdf

Hartley, W. M. (2007). Current and future threats to digital forensics. *ISSA Journal*, 12-14. Retrieved from https://www.issa.org/Library/Journals/2007/August/Hartley-Current and Future Threats to Digital Forensics.pdf

Hull, D. (2009, June 30). [Web log message]. Retrieved from http://computer-forensics.sans.org/blog/2009/06/30/fried-fat-fat-directory-entries-and-the-fat/

Innes, S. (2005). Secure deletion and the effectiveness of evidence elimination software. In C. Valli & A. Woodward (Eds.), *Proceedings of 3rd Australian Computer, Network & Information Forensics Conference*, pp. 24-44. Retrieved from http://igneous.scis.ecu.edu.au/proceedings/2005/forensics/2005_forensics_proceedings.pdf

Jones, M. T. (2009, February 17). *Anatomy of ext4*. Retrieved from http://www.ibm.com/developerworks/linux/library/l-anatomy-ext4/

Kedziora, M. (2011, January 11). [Web log message]. Retrieved from http://www.forensics-research.com/index.php/2011/01/anti-forensics-overview/

Kessler, G. C. (2007). *Anti-forensics and the digital investigator*. In *Proceedings of the 5th Australian Digital Forensics Conference*. Retrieved from http://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1000&context=adf&sei-redir=1&referer=http://scholar.google.com/scholar?hl=en&q=anti-forensics+for+privacy&btnG=&as_sdt=1%2C10&as_sdtp=

King, G. L. (2006). Forensics plan guide. *SANS Computer Forensics and e-Discovery*, Retrieved from http://computer-forensics.sans.org/community/papers/gcfa/forensic-investigation-plan-cookbook_283

Korso, M. (2009, December 1). [Web log message]. Retrieved from http://ezinearticles.com/?Two-Popular-Anti-Forensic-Techniques&id=3323749

Kruse, W. G., & Heiser, J. G. (2001). *Computer forensics: incident response essentials*. Boston, MA: Addison-Wesley.

Lubert, H. (n.d.). *Full disk encryption (fde) vs. file encryption technologies*. Retrieved from http://www.int2view.com/index.php?option=com_content&view=article&id=29:full-disk-encryption-fde-vs-file-encryption-technologies&catid=15&Itemid=46

Machor, M. (2008, January 28). [Web log message]. Retrieved from http://www.forensicfocus.com/forensic-analysis-vista-recycle-bin

Max. (2009a, March 17). [Web log message]. Retrieved from http://www.anti-forensics.com/disk-wiping-one-pass-is-enough

Max. (2009b, February 8). [Web log message]. Retrieved from http://www.anti-forensics.com/disable-the-lastaccess-timestamp-in-windows-xp

Medeiros, J. (2008). Ntfs forensics: A programmer's view of raw file system data extraction. Retrieved from http://grayscale-research.org/new/pdfs/NTFS forensics.pdf

Narvaez, G. (2007). Taking advantage of ext3 journaling file system in a forensic investigation. *SANS Institute Reading Room*, Retrieved from http://www.sans.org/reading_room/whitepapers/forensics/advantage-ext3-journaling-file-system-forensic-investigation_2011

Petullo, M. (2004). Encrypt your root file system. *Linux journal*, Retrieved from http://www.linuxjournal.com/article/7743

Plausible deniability. (n.d.). Retrieved from http://www.truecrypt.org/docs/?s=plausible-deniability

Pogue, C. (2011, February 23). [Web log message]. Retrieved from http://thedigitalstandard.blogspot.com/2011/02/time-stomping-is-for-suckers.html

Reith, M., Carr, C., & Gunsch, G. (2002). An examination of digital forensic models. *International Journal of Digital Evidence*, *1*(3), Retrieved from http://people.emich.edu/pstephen/other_papers/Digital_Forensic_Models.pdf

Russinovich, M. (2011, September 1). *Sdelete*. Retrieved from http://technet.microsoft.com/en-us/sysinternals/bb897443

Siever, E. (2009). *Linux in a nutshell*. (6th ed.). Sebastopol, CA: O'Reilly.

Stallings, W. (2009). *Operating system*. (6th ed.). Upper Saddle River, NJ: Pearson Prentice Hall.

Steel, C. (2006). *Windows forensics: The field guide for conducting corporate computer investigations*. Indianapolis, IN: Wiley Publishing, Inc.

Stonecypher, L. (2009, November 20). [Web log message]. Retrieved from http://www.brighthub.com/computing/linux/articles/34156.aspx

Strickland, J. (2008, February 25). *How computer forensics works*. Retrieved from http://computer.howstuffworks.com/computer-forensic.htm

Wade, M. (2010, December 08). Decoding prefetch files for forensic purposes: Part 1. *DFINews*, Retrieved from http://www.dfinews.com/article/decoding-prefetch-files-forensic-purposes-part-1

Yegulalp, S. (2008, March 28). 7 whole-disk encryption apps put a lock on data. *Information Week*,