


Summer 8-2017

Digital Anti-Forensics: An Implementation and Examination

Stephanie Dachs

CUNY John Jay College, srdachs1@gmail.com

Follow this and additional works at: http://academicworks.cuny.edu/jj_etds

 Part of the [Computer Sciences Commons](#), and the [Forensic Science and Technology Commons](#)

Recommended Citation

Dachs, Stephanie, "Digital Anti-Forensics: An Implementation and Examination" (2017). *CUNY Academic Works*.
http://academicworks.cuny.edu/jj_etds/31

This Thesis is brought to you for free and open access by the John Jay College of Criminal Justice at CUNY Academic Works. It has been accepted for inclusion in Student Theses by an authorized administrator of CUNY Academic Works. For more information, please contact AcademicWorks@cuny.edu.

Digital Anti-Forensics: An Implementation and Examination

A Thesis

Presented in Partial Fulfillment of the Requirements

for the degree of Master of Science in

Digital Forensics and Cybersecurity

John Jay College of Criminal Justice

City University of New York

Stephanie Dachs

August 2017

Abstract

The rise of computer use and technical adeptness by the general public in the last two decades are undeniable. With greater use comes a greater possibility for misuse, evidenced by today's incredible number of crimes involving computers as well as the growth in severity from that of cyber hooliganism to cyber warfare. Although frequently utilized for privacy and security purposes, the vast range of anti-forensic techniques has contributed to the ability for hackers and criminals to obstruct computer forensic investigations.

Understanding how anti-forensics may alter important and relevant data on an electronic device will prove useful for the success and continued advancement of computer forensic investigations. This paper will amalgamate the academic literature on anti-forensics as well as test four of the most accessible anti-forensic tools available online to reveal at what degree they confound traditional computer forensic tools and techniques. Strategies for detecting and mitigating the effects of anti-forensic efforts will be put forth to help inform the future of computer forensic investigative techniques.

Keywords: Digital Forensics, Computer Forensics, Digital Anti-Forensics, Anti-Forensic Tools

Table of Contents

Introduction.....	1
Computer Crime and Computer Forensics	2
Anti-Forensics.....	3
Tool Testing and Evaluation.....	4
Methodology	5
Timestomp	6
Tool Description	6
Tool Implementation and Evaluation.....	6
Summary	11
SDelete.....	11
Tool Description	12
Tool Implementation and Evaluation.....	12
Summary	17
Eraser	18
Tool Description	18
Tool Implementation and Evaluation.....	19
Summary	23
CCleaner	24
Tool Description	24
Tool Implementation and Evaluation.....	24
Summary	28
Anti-Anti-Forensics	28
Conclusion	30
References.....	33
Appendix A.....	36

Table of Figures

Figure 1. Core Anti-Forensic Strategies	4
Figure 2. Timestomp Command	7
Figure 3. Additional Timestomp Command	7
Figure 4. File Attributes Before Timestomp Implementation- Autopsy.....	7
Figure 5. File Attributes After Timestomp Implementation- Autopsy	7
Figure 6. Picture File Attributes After Timestomp Implementation- FTK Imager	8
Figure 7. \$MFT, \$LogFile and USN JOURNAL Timestamp Entries for PDF file- ANJP.....	9
Figure 8. LNK File After Timestomp Implementation- Autopsy	10
Figure 9. Timestomp Prefetch File- Autopsy	10
Figure 10. Timestomp Executable- Autopsy	11
Figure 11. Timestomp Internet History- ESEDatabaseView.....	11
Figure 12. SDelete Commands	12
Figure 13. PNG File Preview Before SDelete Implementation- FTK Imager.....	13
Figure 14. PNG File Preview After SDelete Implementation- FTK Imager	13
Figure 15. JPG Hexadecimal Representation Before SDelete Implementation- FTK Imager	13
Figure 16. JPG Hexadecimal Representation After SDelete Implementation- FTK Imager.....	14
Figure 17. Artifact Found for Erased PNG- Autopsy	14
Figure 18. Thumbnail Found for Erased PNG File- Thumbcache Viewer.....	15
Figure 19. Thumbnail Found for Erased JPG File- Thumbcache Viewer	15
Figure 20. \$LogFile contents for PNG before SDelete Implementation	16
Figure 21. \$LogFile contents for PNG after SDelete Implementation	16
Figure 22. LNK Files After SDelete Implementation- Autopsy	16
Figure 23. SDelete Prefetch File- Autopsy	17
Figure 24. SDelete ZIP File Downloaded- Autopsy	17
Figure 25. SDelete Internet History- ESEDatabaseView	17
Figure 26. Target Type and Erasure Options- Eraser	18
Figure 27. Eraser Tasks.....	19
Figure 28. Metadata Before Eraser Implementation- Autopsy	19
Figure 29. Metadata Unchanged After Eraser Implementation- Autopsy	20
Figure 30. Location of Artifact Found for Erased PPT- Autopsy.....	20
Figure 31. Part of Artifact Found for Erased PPT- Autopsy	21
Figure 32. Location of Artifact "tandi215[1].dat" Found for Erased PDF- Autopsy	21
Figure 33. Contents of Artifact "tandi215[1].dat"- Autopsy	21
Figure 34. LNK Files After Eraser Implementation- Autopsy	22
Figure 35. Eraser Prefetch File- Autopsy	22
Figure 36. Eraser Executable File- Autopsy	23
Figure 37. Eraser Internet History- ESEDatabaseView.....	23
Figure 38. CCleaner Files Deleted.....	25
Figure 39. Documents Folder Before CCleaner Implementation- Autopsy	25
Figure 40. Documents Folder After CCleaner Implementation- Autopsy.....	26
Figure 41. CCleaner Prefetch Files- Autopsy	27
Figure 42. CCleaner Executables- Autopsy	28

Introduction

For every use a computer provides to a business, researcher or common citizen there lays the potential misuse of that same technology. The recent and rapid rise of technical adeptness by the general public as well as the level of anonymity and information accessibility the internet offers seem to intersect to encourage crimes using computer systems (Reith, Carr, & Gunsch, 2002). Law enforcement agencies and researchers around the world have come together to develop tools and techniques, known as computer forensics, to fight the ever-increasing occurrence of computer crime.

The last two decades have seen an exponential rise in the recognition and use of computer forensics as a ripened forensics discipline. As with any crime, the culprits of computer crime want to avoid being caught. Many built-in aspects of computer systems aid the offender in that respect, but there is a field of forensics employed by cyber-criminals that intentionally conceals, alters or makes inaccessible evidence of a crime: anti-forensics. This paper will refer to anti-computer forensics as “anti-forensics”. Anti-forensic techniques take many shapes and forms; disk sanitizers, encryption, anti-debugging techniques, rootkits and targeting computer forensic tool vulnerabilities are a few of the tactics used by individuals trying to conceal digital evidence (Blunden, 2009; Garfinkel, 2007).

It is easy to assume the people most interested in researching, discovering and using anti-forensic tools and techniques are offenders trying to avoid prosecution, but it is important to note that the research can help forensic tool developers create better products and help forensic investigators understand what they are up against. Anti-forensic techniques are also commonly used for privacy and security. A business may employ encryption in order to keep important

records confidential and “secure deletion” is recommended when discarding old machines or destroying financial records and classified documents.

The purpose of this paper is to provide an examination of anti-forensic techniques that are at the disposal of any individual with a computer. Four of the most accessible anti-forensic tools will be implemented on a Windows 10 operating system to reveal at what degree they successfully confound traditional computer forensic tools that may be used in an investigation. Using the results of the anti-forensic tool tests, this paper will provide recommendations for detecting, mitigating and combatting effects of anti-forensic techniques.

Computer Crime and Computer Forensics

Since the introduction of computers in the early 1900s and ARPANET in the 1970s computer crime has existed and evolved. Computer crime, as referenced in this paper, will include any action that breaks the law by use of a computer. Crime by use of a computer has expanded from physical damage to systems, to modification of data for financial gain or revenge, to diverse types of international fraud, theft, intrusion, trafficking and pornography. As technology evolves and the public becomes more tech-savvy, computer crime becomes more sophisticated. As a result, the tools and techniques used to investigate and prosecute these crimes, known as computer forensics, must equally evolve.

Sometimes referred to as digital forensics or media analysis, computer forensics is the collection, preservation, analysis and court presentation of electronic evidence (Patzakis, 2002; USCERT, 2006). There is a plethora of both commercial and open source computer forensic tools (CFTs) that help an investigator analyze a digital system. CFTs make a verifiably accurate copy of a computer system and analyze data to uncover information about the use of that system. These tools can perform tasks including disk and data capture, data recovery, file viewing, file

analysis, metadata extraction, memory analysis, registry analysis, email analysis, mobile device analysis, live analysis and network forensics. The list is extensive, but some of the most widely used tools include Guidance Software's EnCase Forensic, Access Data's Forensic Toolkit (FTK), The Sleuth Kit (TSK), Autopsy and Wireshark.

Anti-Forensics

Anti-forensics is exactly as it sounds - the use of tools, methods, and procedures to obstruct the forensic recovery of evidence (Erasani, 2010; Garfinkel, 2007; Rekhis & Boudriga, 2012). Goals of anti-forensics include avoiding detection, disrupting the collection of evidence, increasing the time an examiner spends on a case, casting doubt on a forensic report or testimony and subverting or directly attacking a forensic examination (Garfinkel, 2007; Blunden, 2009).

In order to lead an investigator astray, anti-forensic techniques may utilize one or more of these five general strategies: data destruction, data concealment, data transformation, data fabrication and data source elimination (Bilby, 2006; Blunden, 2009). These strategies are used to buy time, leave behind evidence that is difficult to capture and/or understand, force an investigator to follow false trails and destroy evidence altogether. The most common data destruction and concealment methods include overwriting data and metadata, data hiding, encryption and steganography (Blunden, 2009). Tactical implementations of core anti-forensic strategies are described in Figure 1.

Strategy	Implementation
Data Destruction	Degaussing, Data Cleansing, Meta-data Wiping, Registry Wiping
Data Concealment	HPA, DCO and Slack Space Concealment, Steganography
Data Transformation	Compression, Encryption, Obfuscation, Transmogrification
Data Fabrication	Introduce Known Files, False Audit Trails
Data Source Elimination	Data Contraception, In-Memory DLL injection

Figure 1. Core Anti-Forensic Strategies

There exist many open source and commercial software tools to perform one or more of the techniques mentioned above. A quick google search for anti-forensic software immediately shows a list of more than twenty tools that could be used to thwart a computer forensic investigation, many of which are available as a free download. Anti-forensic software includes many programs for artifact wiping, overwriting, data hiding and encryption.

Tool Testing and Evaluation

The purpose of this research was to determine and demonstrate the ease at which any computer user can implement anti-forensics as well as to provide recommendations to detect and mitigate the effects of anti-forensic tool use. The anti-forensic tools used in this research do not require knowledge of computer architecture or computer investigations. They were found by searching for “easy, free anti-forensic tools” on multiple search engines. In each tool section below, there is a summary of what the version of the tool claims to do as well as the procedure for downloading and implementing the tool. Results of the tests are also in their own sub-section.

Methodology

A baseline file system was created on a virtual machine running Windows 10 Home. VirtualBox Graphical User Interface Version 5.1.16 was used to host the Windows 10 virtual machine with 2048 MB base memory and 32.00 GB storage. This machine was loaded with a range of document and image file types. Web history was accumulated using internet browser Microsoft Edge. Files were downloaded, opened and modified to mirror typical user activity. Certain files were placed in a zipped folder, certain files were deleted and sent to the recycle bin and certain files were removed from the recycle bin. This base machine will serve as the control group, allowing the performance of each tool to be tested on identical data and activity. Since forensic tool Autopsy does not accept the VMDK file type, a raw DD file was created of the base machine using the “Create Disk Image” tool in FTK Imager.

Multiple “clones” of the full base machine were made in VirtualBox. Oracle VirtualBox’s clone feature creates an identical, fully operational machine from the source machine (Oracle Corporation, 2017). The first anti-forensic tool was downloaded, installed and utilized on the clone of the base machine named “Test1”. An image of this machine was saved after the anti-forensic tool had been implemented. Each successive anti-forensic tool test was performed on a new clone of the base machine.

The images of the machines after the anti-forensic tools were implemented were examined to assess the degree at which each tool successfully confounded traditional computer forensic tools. The assessment of success may differ between each anti-forensic tool depending on what the tool attempts to accomplish.

Timestomp

Altering MACE times ('M'odified, 'A'ccessed, 'C'reated, 'E'ntry Modified) and checksums can impede an investigator's ability to create an accurate and plausible timeline. Invalid times and dates make combining information from multiple evidentiary sources difficult or impossible. In addition, some computer forensic tools may not function with invalid or missing dates and times (Harris, 2009; Offensive Security, 2017).

Tool Description

The Metasploit Anti-Forensics Project aims to develop tools and techniques to remove forensic evidence from computer systems. The project includes tools such as Timestomp, a command line tool that allows a user to delete or modify MACE times in an NTFS filesystem (Bishop Fox, 2006). There is also a free Timestomp-GUI executable file that makes it even simpler to run. The command line tool offers a wide range of options from altering the "created" time of one file to altering the MACE times of an entire file system. Timestomp offers an option to "set the MACE timestamps so that EnCase shows blanks". The validity of that claim will not be tested in this paper; however, this option will be used and tested with forensic tools FTK Imager and Autopsy.

Tool Implementation and Evaluation

Timestomp was downloaded from <https://www.jonrajewski.com/resources/> on test machine "Test1" using Microsoft Edge. The commands in Figures 2 and 3 were run in the command line; the first to change all four MACE times of the file "stolen_car_parts.pdf" to Monday 01/01/2080 01:00:00 PM and the second to clear the MACE times of the entire Pictures folder. The '-r' option claims to "set the MACE timestamps recursively on a directory so that EnCase shows blanks".

```
>timestomp.exe C:\Users\John\Documents\stolen_car_parts.pdf -z "Monday 01/01/2080 01:00:00 PM"
```

Figure 2. Timestomp Command

```
>timestomp.exe C:\Users\John\Pictures -r
```

Figure 3. Additional Timestomp Command

The base machine and the Test1 machine were viewed in FTK Imager and Autopsy. On the base machine, before Timestomp implementation, the files and correct timestamps were visible for the file in the Documents folder and pictures in the Pictures folder.

/img_BaseRaw.001/vol_vol3/Users/John/Documents					
Table Thumbnail					
Name	Modified Time	Change Time	Access Time	Created Time	Size
stolen_car_parts.pdf	2017-04-10 19:52:58 EDT	2017-04-10 19:56:20 EDT	2017-04-10 19:52:57 EDT	2017-04-10 19:52:57 EDT	611327

/img_BaseRaw.001/vol_vol3/Users/John/Pictures					
Table Thumbnail					
Name	Modified Time	Change Time	Access Time	Created Time	Size
Jane_Doe.png	2017-04-10 18:44:12 EDT	2017-04-10 19:56:40 EDT	2017-04-10 18:44:11 EDT	2017-04-10 18:44:11 EDT	156135
car_thief.jpg	2017-04-10 18:48:20 EDT	2017-04-10 19:56:40 EDT	2017-04-10 18:48:20 EDT	2017-04-10 18:48:20 EDT	29840
car_theft.png	2017-04-10 18:49:24 EDT	2017-04-10 19:56:38 EDT	2017-04-10 18:49:24 EDT	2017-04-10 18:49:24 EDT	10388
John_Doe.jpg	2017-04-10 18:43:32 EDT	2017-04-10 19:56:40 EDT	2017-04-10 18:43:31 EDT	2017-04-10 18:43:31 EDT	3189

Figure 4. File Attributes Before Timestomp Implementation- Autopsy

/img_Test1Raw.001/vol_vol3/Users/John/Documents					
Table Thumbnail					
Name	Modified Time	Change Time	Access Time	Created Time	Size
stolen_car_parts.pdf	2080-01-01 15:00:00 EST	2080-01-01 15:00:00 EST	2080-01-01 15:00:00 EST	2080-01-01 15:00:00 EST	611327

/img_Test1Raw.001/vol_vol3/Users/John/Pictures					
Table Thumbnail					
Name	Modified Time	Change Time	Access Time	Created Time	Size
Jane_Doe.png	2076-11-29 10:54:34 EST	2076-11-29 10:54:34 EST	2076-11-29 10:54:34 EST	2076-11-29 10:54:34 EST	156135
car_thief.jpg	2076-11-29 10:54:34 EST	2076-11-29 10:54:34 EST	2076-11-29 10:54:34 EST	2076-11-29 10:54:34 EST	29840
car_theft.png	2076-11-29 10:54:34 EST	2076-11-29 10:54:34 EST	2076-11-29 10:54:34 EST	2076-11-29 10:54:34 EST	10388
John_Doe.jpg	2076-11-29 10:54:34 EST	2076-11-29 10:54:34 EST	2076-11-29 10:54:34 EST	2076-11-29 10:54:34 EST	3189

Figure 5. File Attributes After Timestomp Implementation- Autopsy






Name	Date Modified
 John_Doe.jpg	1/1/1601 07:00:00
 Jane_Doe.png	1/1/1601 07:00:00
 desktop.ini	1/1/1601 07:00:00
 car_thief.jpg	1/1/1601 07:00:00
 car_theft.png	1/1/1601 07:00:00

Figure 6. Picture File Attributes After Timestomp Implementation- FTK Imager

The desired, altered time and date was displayed in Autopsy after Timestomp was executed on the single PDF file. The recursive clearing option that Timestomp offers did not show blanks in FTK Imager or Autopsy, but displayed the date 1601-01-01 in FTK Imager and the date 2076-11-29 in Autopsy for each of the files in the Pictures folder. All Microsoft Windows operating systems after Windows 95 count units of one hundred nanoseconds from the epoch 1601-01-01, so FTK Imager displays a timestamp equivalent to a zero value. NTFS timestamps outside the range 1970-01-01 00:00:01 -- 2106-02-07 06:28:00 are translated to timestamps inside the range with a many-to-one correspondence in Autopsy (SleuthKitWiki, 2017; WPATHULIN, 2013).

After further testing, it appears Timestomp's recursive clearing option does not successfully descend into non-password-protected zipped folders. The selected zipped folder's MACE times were successfully altered in FTK Imager and Autopsy however the contents of the folder display their original timestamp information.

In an NTFS system, the MFT stores two sets of MACE times for a file; one set in \$STANDARD_INFORMATION, or Standard Information Attribute (SIA), and one set in \$FILE_NAME, or Filename Attribute (FNA). The timestamps in the SIA are the ones displayed in File Explorer and in most forensic tools. Timestamps in the FNA are more difficult to alter and as of 2009, there were no known automated tools that alter the FNA (Mueller, 2009).

Therefore, an examination of MACE times in the FNA can be used to subvert timestamp alteration efforts.

To locate original timestamps for the file “stolen_car_parts.pdf”, a tool named ANJP was used to parse system activity using the MFT, LogFile and USN Journal from the test machine. The log2timeline table of the database created by the parsing tool revealed the real MACE times of the Timestamp-altered file in each the MFT, LogFile and USN Journal. The MFT and LogFile both make it known that the timestamps were changed by displaying both the altered and unaltered times as different entries. Equivalent results were found for the picture files whose timestamps were altered with Timestamp. Examining these entries for inconsistencies can alert an investigator to potential timestamp alterations.

l2t_source	l2t_type	l2t_date	l2t_time	l2t_filename
(empty)	(empty)	(empty)	(empty)	(empty)
\$MFT	fna_ctime	2017-04-10	23:52:57.539	\Users\John\Documents\stolen_car_parts.pdf
\$MFT	fna_mtime	2017-04-10	23:52:57.742	\Users\John\Documents\stolen_car_parts.pdf
\$MFT	fna_mftmtime	2017-04-10	23:52:57.742	\Users\John\Documents\stolen_car_parts.pdf
\$MFT	fna_atime	2017-04-10	23:52:57.742	\Users\John\Documents\stolen_car_parts.pdf
\$MFT	sia_ctime	2080-01-01	20:00:00.000	\Users\John\Documents\stolen_car_parts.pdf
\$MFT	sia_mtime	2080-01-01	20:00:00.000	\Users\John\Documents\stolen_car_parts.pdf
\$MFT	sia_mftmtime	2080-01-01	20:00:00.000	\Users\John\Documents\stolen_car_parts.pdf
\$MFT	sia_atime	2080-01-01	20:00:00.000	\Users\John\Documents\stolen_car_parts.pdf
\$LogFile	sia_ctime	2080-01-01	20:00:00.000	\Users\John\Documents\stolen_car_parts.pdf
\$LogFile	sia_mtime	2080-01-01	20:00:00.000	\Users\John\Documents\stolen_car_parts.pdf
\$LogFile	sia_mftmtime	2080-01-01	20:00:00.000	\Users\John\Documents\stolen_car_parts.pdf
\$LogFile	sia_atime	2080-01-01	20:00:00.000	\Users\John\Documents\stolen_car_parts.pdf
\$LogFile	sia_ctime	2017-04-10	23:52:57.539	\Users\John\Documents\stolen_car_parts.pdf
\$LogFile	sia_mtime	2017-04-10	23:52:58.789	\Users\John\Documents\stolen_car_parts.pdf
\$LogFile	sia_mftmtime	2017-04-10	23:56:20.508	\Users\John\Documents\stolen_car_parts.pdf
\$LogFile	sia_atime	2017-04-10	23:52:57.742	\Users\John\Documents\stolen_car_parts.pdf
USN JOURNAL ENTRY	ur_datetime	2017-04-18	02:28:00.564	\Users\John\Documents\stolen_car_parts.pdf
USN JOURNAL ENTRY	ur_datetime	2017-04-18	02:28:00.564	\Users\John\Documents\stolen_car_parts.pdf
USN JOURNAL ENTRY	ur_datetime	2017-04-10	23:56:20.508	\Users\John\Documents\stolen_car_parts.pdf
USN JOURNAL ENTRY	ur_datetime	2017-04-10	23:56:20.508	\Users\John\Documents\stolen_car_parts.pdf

Figure 7. \$MFT, \$LogFile and USN JOURNAL Timestamp Entries for PDF file- ANJP

Additional, more common and easily accessible artifacts existed for the files whose MACE times were altered. A LNK file¹ existed for the file “stolen_car_parts.pdf”. LNK files can give a more accurate timeframe for when a file was accessed from a certain location and the mismatch of this time to the file’s metadata can indicate something suspicious has occurred on the system. The Edge Browser Automatic Jump List² and File Explorer Automatic Jump List contained the file “stolen_car_parts.pdf”. These artifacts can be used to inform timeline information for timestamp-altered files. Similar evidence was found for the picture files; the Microsoft Photos App Automatic Jump List contained each of the four image files from this test.

Name	Modified Time	Change Time	Access Time	Created Time	Size
stolen_car_parts.lnk	2017-04-10 19:56:20 EDT	2017-04-10 19:56:20 EDT	2017-04-10 19:56:20 EDT	2017-04-10 19:52:57 EDT	638

Figure 8. LNK File After Timestamp Implementation- Autopsy

A prefetch file³ existed for the Timestamp executable file and a hash search for the Timestamp executable file in Autopsy returned the file in the Downloads folder. Evidence of the Bing search for “timestamp” and the site from which Timestamp was downloaded existed in the Microsoft Edge history database⁴. These artifacts can be used to show the program existed as well as the times and dates it was run on the system under analysis.

Name	Modified Time	Change Time	Access Time	Created Time	Size
TIMESTAMP.EXE-503A8BE9.pf	2017-04-17 22:35:37 EDT	2017-04-17 22:35:37 EDT	2017-04-17 22:28:00 EDT	2017-04-17 22:28:00 EDT	2976

Figure 9. Timestamp Prefetch File- Autopsy

¹ LNK files are shortcut files that link to an application and are created when a user opens a local or remote file. LNK files are located at \Users\\AppData\Roaming\Microsoft\Windows\Recent\ (McQuaid, 2014).

² Created by a user or automatically by the operating system, Jump Lists give a user quick access to recently opened application files. A Jump List file is automatically saved as an *.automaticDestination-ms or a *.customDestination file at the locations \Users\\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations and \Users\\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations (Antonovich, 2014).

³ Windows creates a prefetch file when an application is run from a particular location for the first time to help speed up the loading of the application. Prefetch files are located at \Windows\Prefetch.

⁴ The Microsoft Edge history database is located at \Users\\AppData\Local\Microsoft\Windows\WebCache\WebCacheV01.dat.

Source File	MD5 Hash	File Path
🔗 timestomp.exe	e72a495a8e4af5d09b9400caf014d3d7	/img_Test1Raw.001/vol_vol3/Users/John/Downloads/timestomp.exe

Figure 10. Timestomp Executable- Autopsy

Container_11 [Table ID = 29, 25 Columns]
Url
Visited: John@https://www.bing.com/search?q=download+timestomp&form=EDGHPC&q&qs=PF&cvid=9
Visited: John@https://www.jonrajewski.com/resources/

Figure 11. Timestomp Internet History- ESEDatabaseView

Summary

Although Timestomp offers a quick and straightforward way to frustrate forensic tools, missing or unlikely timestamp values make it clear that unfavorable activity has been performed on the system. A tool like ANJP that analyzes the MFT of a system will display original timestamps and allows an investigator to spot inconsistencies between entries in the MFT, LogFile and USN Journal of a Windows 10 system. More simply, LNK files and Jump Lists can help pinpoint when and from where certain files were accessed. If there is suspicion that MACE times have been altered, that a program like Timestomp has been executed or simply as a precaution, hash or keyword analysis for known metadata altering tools may prove beneficial in locating prefetch files and internet history artifacts.

SDelete

Many programs exist that purposefully overwrite useful or deleted information on a storage device. Software-based tools that offer “sanitizing”, or “wiping” or “clearing”, of a disk often implement this by overwriting data. This can be done with a single pass of bytes with a chosen pattern or multiple passes of differing patterns of bytes on either an entire disk or selected individual files. Wiping tools commonly offer the cleaning of web browser cache and history,

chat logs, files and sometimes Jump Lists, thumbnails and registry items (Afonin, Nikolaev, Gubanov, 2015).

Tool Description

Typically, when a file is deleted in an NTFS file system, the system removes the reference to that file from the master file table (MFT) but the data still exists on the disk until it is written over. SDelete deletes existing files as well as erases any file data that exists in the unallocated portions of a disk. It is a command line utility with options that allow a user to delete one or multiple files or directories, clean free space on a disk and specify number of overwrite passes.

Tool Implementation and Evaluation

SDelete v2.0 was downloaded from technet.microsoft.com/en-us/sysinternals/sdelete.aspx using Microsoft Edge. The contents of the downloaded zipped file were extracted to the folder “SDelete” in Downloads on test machine “Test2”. The application `sdelete64.exe` was run and the License Agreement was accepted. From the Windows command prompt, `sdelete64.exe` was run twice; first, to erase the picture file “`car_theft.png`” with 1 pass and second, to erase the picture file “`car_thief.jpg`” with 3 passes.

```
C:\Users\John\Downloads\SDelete>sdelete64.exe -p 1 C:\Users\John\Pictures\car_theft.png
```

```
C:\Users\John\Downloads\SDelete>sdelete64.exe -p 3 C:\Users\John\Pictures\car_thief.jpg
```

Figure 12. SDelete Commands

The base machine and the Test2 machine were viewed in FTK Imager and Autopsy. On the base machine, before SDelete implementation, the picture files in the Pictures folder could be previewed, exported and viewed on a forensics machine. After the execution of SDelete, the picture file names were still visible in the Pictures folder along with the correct metadata.

Although the file sizes and metadata remained the same, the files were not viewable and the hexadecimal representation of the data displayed all zeros.

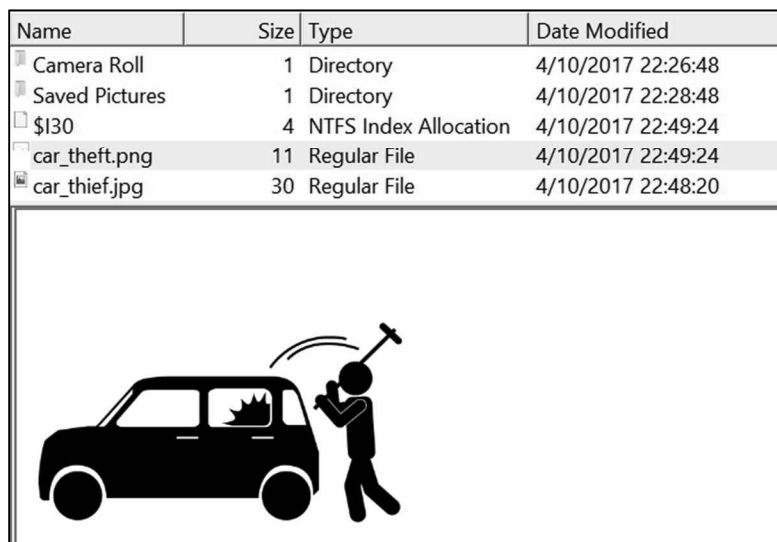


Figure 13. PNG File Preview Before SDelete Implementation- FTK Imager

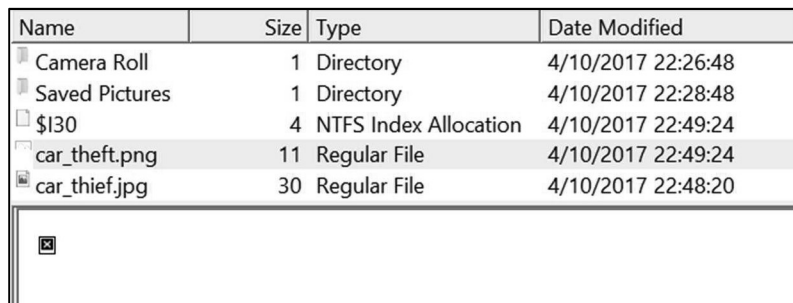


Figure 14. PNG File Preview After SDelete Implementation- FTK Imager

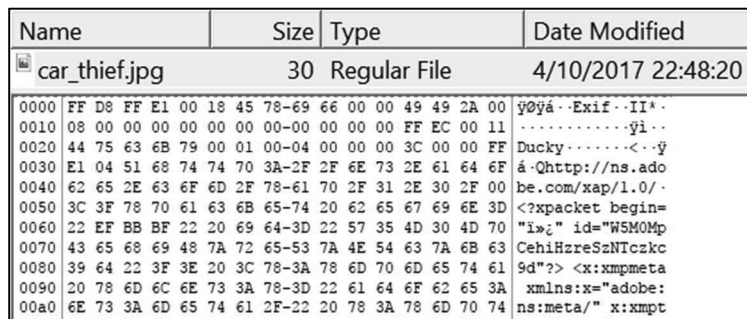


Figure 15. JPG Hexadecimal Representation Before SDelete Implementation- FTK Imager

Name	Size	Type	Date Modified
car_thief.jpg	30	Regular File	4/10/2017 22:48:20
0000	00 00 00 00 00 00 00 00	00-00	00 00 00 00 00 00 00 00
0010	00 00 00 00 00 00 00 00	00-00	00 00 00 00 00 00 00 00
0020	00 00 00 00 00 00 00 00	00-00	00 00 00 00 00 00 00 00
0030	00 00 00 00 00 00 00 00	00-00	00 00 00 00 00 00 00 00
0040	00 00 00 00 00 00 00 00	00-00	00 00 00 00 00 00 00 00
0050	00 00 00 00 00 00 00 00	00-00	00 00 00 00 00 00 00 00
0060	00 00 00 00 00 00 00 00	00-00	00 00 00 00 00 00 00 00
0070	00 00 00 00 00 00 00 00	00-00	00 00 00 00 00 00 00 00
0080	00 00 00 00 00 00 00 00	00-00	00 00 00 00 00 00 00 00
0090	00 00 00 00 00 00 00 00	00-00	00 00 00 00 00 00 00 00
00a0	00 00 00 00 00 00 00 00	00-00	00 00 00 00 00 00 00 00

Figure 16. JPG Hexadecimal Representation After SDelete Implementation- FTK Imager

A hash search for the two erased picture files returned Microsoft Edge artifacts created when the pictures were viewed in the Edge browser. The cache⁵ in which these artifacts were found contained a viewable copy of thousands of pictures that were viewed through Microsoft Edge including photos on visited webpages and image search results that weren't specifically clicked on by the user.



Figure 17. Artifact Found for Erased PNG- Autopsy

⁵ The Microsoft Edge cache is located at `\Users\\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\#!001\MicrosoftEdge\Cache\` (DataForensics, 2015).

A thumbnail can remain in the Windows thumbnail cache⁶ if an original image or file has been deleted. A manual examination of the contents of the Test2 machine's thumbnail cache resulted in the following traces of the erased files.

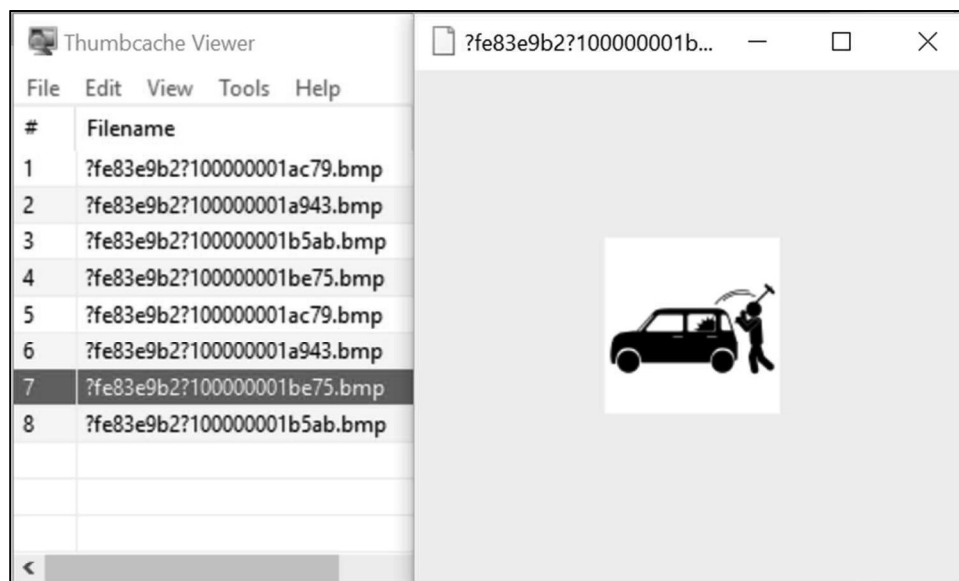


Figure 18. Thumbnail Found for Erased PNG File- Thumbcache Viewer

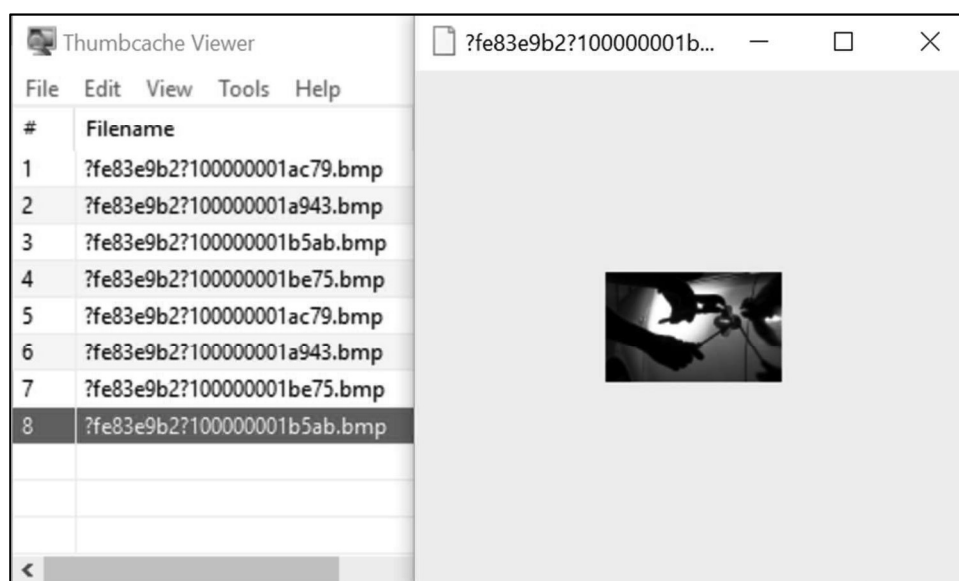


Figure 19. Thumbnail Found for Erased JPG File- Thumbcache Viewer

⁶ The Windows 10 thumbnail cache is located at \Users\\AppData\Local\Microsoft\Windows\Explorer\.

An examination of the MFT, LogFile and USN Journal with ANJP all confirm the existence and original path of each erased picture file. Accurate timestamp and LNK file information is also available in the parsed ANJP database. Ten additional entries existed in the LogFile for each picture file erased with SDelete, some of which disclosed the date and time SDelete was run.

Date	Time	Timezone	Source	Sourcetype	Filename
4/10/2017	23:56:38	UTC	\$LogFile	UpdateResidentValue - UpdateResidentValue	\Users\John\Pictures\car_theft.png
4/10/2017	22:49:24	UTC	\$LogFile	UpdateResidentValue - UpdateResidentValue	\Users\John\Pictures\car_theft.png
4/10/2017	22:49:25	UTC	\$LogFile	UpdateResidentValue - UpdateResidentValue	\Users\John\Pictures\car_theft.png
4/10/2017	22:49:24	UTC	\$LogFile	UpdateResidentValue - UpdateResidentValue	\Users\John\Pictures\car_theft.png

Figure 20. \$LogFile contents for PNG before SDelete Implementation

Date	Time	Timezone	Source	Sourcetype	Filename
4/10/2017	22:49:24	UTC	\$LogFile	DeleteIndexEntryAllocation - AddIndexEntryAllocation	\Users\John\Pictures\car_theft.png
4/20/2017	20:38:18	UTC	\$LogFile	DeleteIndexEntryAllocation - AddIndexEntryAllocation	\Users\John\Pictures\car_theft.png
4/20/2017	20:38:18	UTC	\$LogFile	DeleteIndexEntryAllocation - AddIndexEntryAllocation	\Users\John\Pictures\car_theft.png
4/10/2017	22:49:24	UTC	\$LogFile	DeleteIndexEntryAllocation - AddIndexEntryAllocation	\Users\John\Pictures\car_theft.png
4/20/2017	20:38:18	UTC	\$LogFile	UpdateResidentValue - UpdateResidentValue	\Users\John\Pictures\car_theft.png
4/20/2017	20:38:18	UTC	\$LogFile	UpdateResidentValue - UpdateResidentValue	\Users\John\Pictures\car_theft.png
4/10/2017	22:49:24	UTC	\$LogFile	UpdateResidentValue - UpdateResidentValue	\Users\John\Pictures\car_theft.png
4/10/2017	22:49:25	UTC	\$LogFile	UpdateResidentValue - UpdateResidentValue	\Users\John\Pictures\car_theft.png
4/10/2017	23:56:38	UTC	\$LogFile	UpdateResidentValue - UpdateResidentValue	\Users\John\Pictures\car_theft.png
4/10/2017	22:49:24	UTC	\$LogFile	UpdateResidentValue - UpdateResidentValue	\Users\John\Pictures\car_theft.png
4/10/2017	23:56:38	UTC	\$LogFile	UpdateResidentValue - UpdateResidentValue	\Users\John\Pictures\car_theft.png
4/10/2017	22:49:24	UTC	\$LogFile	UpdateResidentValue - UpdateResidentValue	\Users\John\Pictures\car_theft.png
4/10/2017	22:49:25	UTC	\$LogFile	UpdateResidentValue - UpdateResidentValue	\Users\John\Pictures\car_theft.png
4/10/2017	22:49:24	UTC	\$LogFile	UpdateResidentValue - UpdateResidentValue	\Users\John\Pictures\car_theft.png

Figure 21. \$LogFile contents for PNG after SDelete Implementation

LNK files remained for the two deleted files. The Automatic Jump List for File Explorer as well as Microsoft Photos App verified the files “car_thief.jpg” and “car_theft.png” were browsed to and viewed by those applications at specific dates and times.

Name	Modified Time	Change Time	Access Time	Created Time	Size
 car_theft.lnk	2017-04-10 19:56:38 EDT	2017-04-10 19:56:38 EDT	2017-04-10 19:56:38 EDT	2017-04-10 18:49:24 EDT	636
 car_thief.lnk	2017-04-10 19:56:41 EDT	2017-04-10 19:56:41 EDT	2017-04-10 19:56:41 EDT	2017-04-10 18:48:20 EDT	636

Figure 22. LNK Files After SDelete Implementation- Autopsy

A prefetch file existed for the SDelete executable file but the MACE times and size displayed all zeros. A hash search for the executable file yielded no results but a hash search for

the downloaded zip folder “SDelete.zip” returned its location in the Downloads folder. The Microsoft Edge history database shows evidence of the Bing search for “sdelete” along with the download link. These artifacts can be used to show the program existed and was run on the system.

Name	Modified Time	Change Time	Access Time	Created Time	Size
SDELETE64.EXE-6BAABFB1.pf	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0

Figure 23. SDelete Prefetch File- Autopsy

Source File	MD5 Hash
SDelete.zip	965b6f8064f3559902c2bf68c9d5c066

Figure 24. SDelete ZIP File Downloaded- Autopsy

Container_11 [Table ID = 29, 25 Columns]
Url
Visited: John@https://www.bing.com/search?q=sdelete&FORM=EDGEND
Visited: John@https://technet.microsoft.com/en-us/sysinternals/sdelete.aspx

Figure 25. SDelete Internet History- ESEDatabaseView

Summary

SDelete successfully overwrote the contents of the specified files with zeros, rendering FTK Imager and Autopsy unable to view the files at their original locations. SDelete did not, however, change or remove the metadata of the erased files at their original locations. Hash analysis for the erased pictures returned viewable versions of the securely erased pictures in a Microsoft Edge cache, a forensic artifact that could be tremendously valuable in a forensic investigation in which a known bad hash database exists. The erased picture files also remained as thumbnails in the Windows thumbnail cache. LNK files and Jump Lists show the files were viewed by a specific user on the system. Additional entries were created for the erased files in the system’s LogFile, some of which possessed a different date and time than most other entries.

This raises suspicion that something uncommon has occurred and in this case, the entries indicate the time SDelete modified the desired files. SDelete did an advanced job at hiding its execution. A prefetch file existed for SDelete's executable file indicating it was run but the prefetch file's metadata displayed blanks in FTK Imager, zeros in Autopsy and included no content. A hash search for the executable file returned no results but a hash search for the downloaded SDelete.zip file successfully located the zipped folder in the user's Downloads folder.

Eraser

Tool Description

Eraser is another free data removal tool for Windows which allows a user to securely remove files and folders by overwriting the data several times. It also allows a user to securely wipe free space to remove data of previously deleted files (Eraser, 2016). The Eraser software offers the option to wipe files on demand or to schedule wiping for specific times and dates. It also offers target type and erasure method options, ranging from 1 pass to 35 passes.

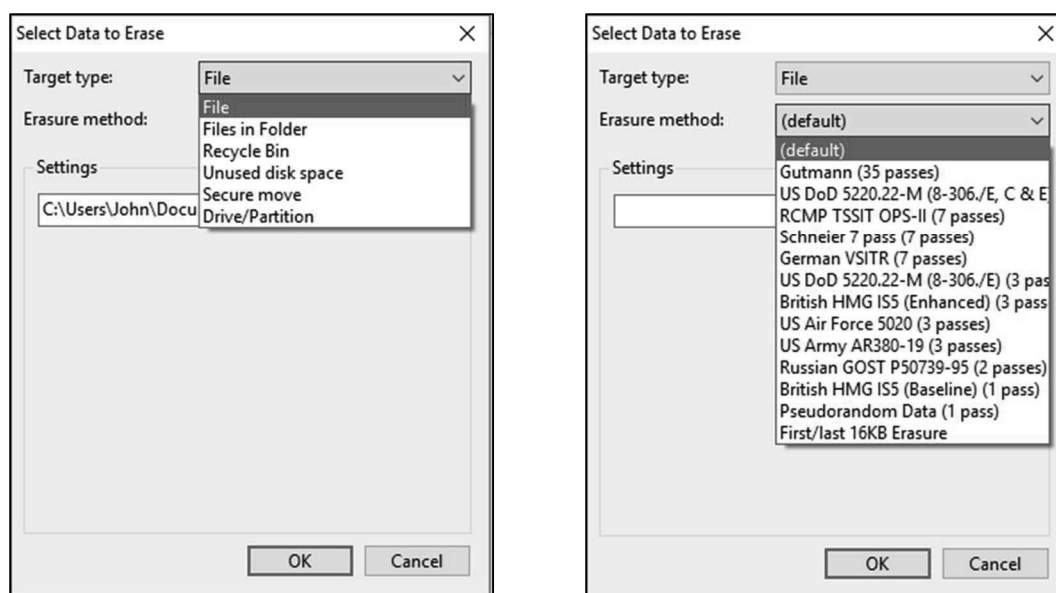
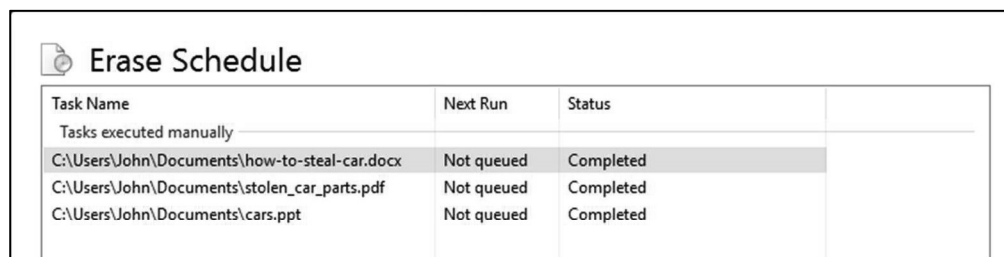


Figure 26. Target Type and Erasure Options- Eraser

Tool Implementation and Evaluation

Eraser 6.2.0.2979 was downloaded by following a link on <https://eraser.heidi.ie/download/> using Microsoft Edge. The link connected to the site sourceforge.net and Eraser was downloaded to the Downloads folder on test machine “Test3”. The application file was run from the Downloads folder which began the Eraser Setup Wizard. The End-User License Agreement was accepted. There were multiple options for Setup Type including “Typical”, “Custom” or “Complete”. “Complete” was chosen and Eraser was installed. Three tasks were added to the Eraser schedule and completed, the first one implementing a 1-pass method (Pseudorandom Data), the second implementing a 3-pass method (US Air Force 5020) and the third implementing a 7-pass method (Schneier 7 pass) on files “cars.ppt”, “stolen_car_parts.pdf” and “how-to-steal-car.docx”, respectively.



Task Name	Next Run	Status
Tasks executed manually		
C:\Users\John\Documents\how-to-steal-car.docx	Not queued	Completed
C:\Users\John\Documents\stolen_car_parts.pdf	Not queued	Completed
C:\Users\John\Documents\cars.ppt	Not queued	Completed

Figure 27. Eraser Tasks

The base machine and the Test3 machine were viewed in FTK Imager and Autopsy. On the base machine, before Eraser implementation, the three erased files were visible in the user’s Documents folder.

Name	Modified Time	Change Time	Access Time	Created Time	Size
how-to-steal-car.docx	2017-04-10 19:04:28 EDT	2017-04-10 19:04:28 EDT	2017-04-10 19:04:28 EDT	2017-04-10 18:56:02 EDT	12910
best_selling_classics.docx	2017-04-10 19:10:38 EDT	2017-04-10 19:10:38 EDT	2017-04-10 19:10:38 EDT	2017-04-10 19:06:53 EDT	13188
classic_books.pptx	2017-04-10 19:45:53 EDT	2017-04-10 19:45:53 EDT	2017-04-10 19:45:53 EDT	2017-04-10 19:42:58 EDT	89029
cars.ppt	2017-04-10 19:36:11 EDT	2017-04-10 19:37:37 EDT	2017-04-10 19:36:11 EDT	2017-04-10 19:36:11 EDT	259072
John_Doe_Wiki.pdf	2017-04-10 18:55:11 EDT	2017-04-10 19:56:09 EDT	2017-04-10 18:55:09 EDT	2017-04-10 18:55:09 EDT	393974
stolen_car_parts.pdf	2017-04-10 19:52:58 EDT	2017-04-10 19:56:20 EDT	2017-04-10 19:52:57 EDT	2017-04-10 19:52:57 EDT	611327

Figure 28. Metadata Before Eraser Implementation- Autopsy

Name	Modified Time	Change Time	Access Time	Created Time	Size
how-to-steal-car.docx	2017-04-10 19:04:28 EDT	2017-04-10 19:04:28 EDT	2017-04-10 19:04:28 EDT	2017-04-10 18:56:02 EDT	12910
best_selling_classics.docx	2017-04-10 19:10:38 EDT	2017-04-10 19:10:38 EDT	2017-04-10 19:10:38 EDT	2017-04-10 19:06:53 EDT	13188
classic_books.pptx	2017-04-10 19:45:53 EDT	2017-04-10 19:45:53 EDT	2017-04-10 19:45:53 EDT	2017-04-10 19:42:58 EDT	89029
cars.ppt	2017-04-10 19:36:11 EDT	2017-04-10 19:37:37 EDT	2017-04-10 19:36:11 EDT	2017-04-10 19:36:11 EDT	259072
John_Doe_Wiki.pdf	2017-04-10 18:55:11 EDT	2017-04-10 19:56:09 EDT	2017-04-10 18:55:09 EDT	2017-04-10 18:55:09 EDT	393974
stolen_car_parts.pdf	2017-04-10 19:52:58 EDT	2017-04-10 19:56:20 EDT	2017-04-10 19:52:57 EDT	2017-04-10 19:52:57 EDT	611327

Figure 29. Metadata Unchanged After Eraser Implementation- Autopsy

After the execution of Eraser, all three file names were still visible in the Documents folder along with the correct metadata for each file. Before implementation, the files could be previewed, exported and viewed. After implementation, although the file sizes and metadata remained the same, the exported files were not able to be opened and the hexadecimal representation of the data was completely different.

A hash search for the three erased files returned no results. A keyword search in Autopsy for the names of the erased files each returned the file “Eraser (x86).msi”. This implicates the Eraser tool is connected to these files in some way. A search for “online car industry”, a phrase in the file “cars.ppt”, returned the file in Figures 30 and 31, which includes the content of the PPT file. This PPT file was originally downloaded from the internet and the file that resulted from the phrase search was created when the download was automatically scanned by Windows Defender. A search for “Stolen Vehicle Parts”, a phrase from “stolen_car_parts.pdf”, returned the file and contents in Figures 32 and 33. The erased PDF had been viewed in Microsoft Edge which stored its details in this database that was detected by Autopsy.

`/ProgramData/Microsoft/Windows Defender/Scans/FilesStash`

Figure 30. Location of Artifact Found for Erased PPT- Autopsy

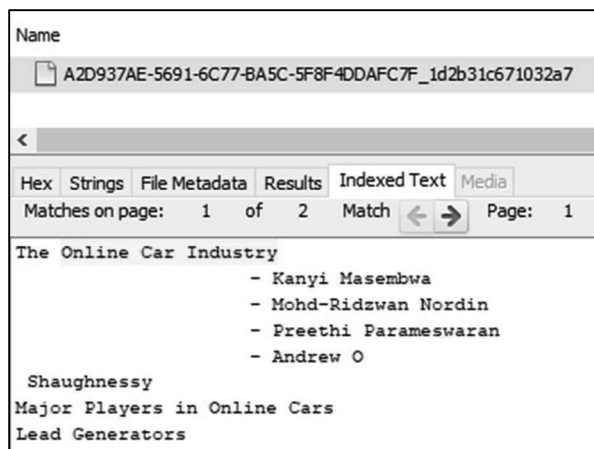


Figure 31. Part of Artifact Found for Erased PPT- Autopsy

/Users/John/AppData/Local/Packages/Microsoft.MicrosoftEdge_8wekyb3d8bbwe/AC/#!001/MicrosoftEdge/Cache/UBYWSHOP

Figure 32. Location of Artifact "tandi215[1].dat" Found for Erased PDF- Autopsy

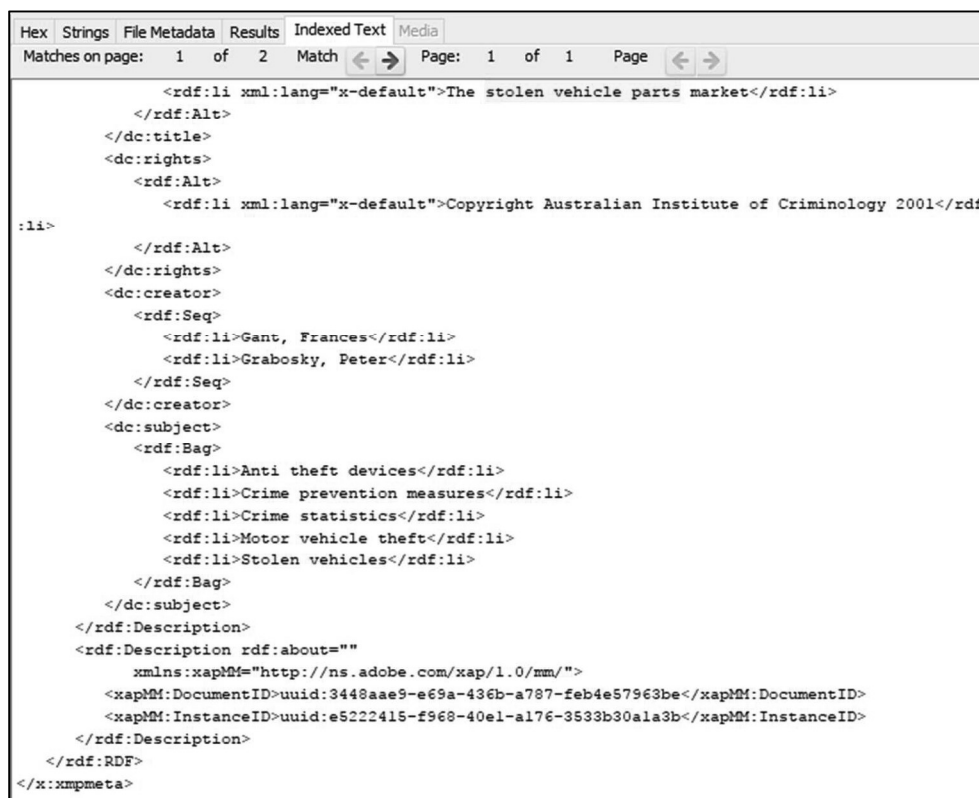


Figure 33. Contents of Artifact "tandi215[1].dat"- Autopsy

An examination of the MFT, LogFile and USN Journal with ANJP confirmed the existence and original path of each of the three erased files. Accurate timestamp information, LNK file and Microsoft Office and Windows Recent Documents information is also available in the parsed ANJP database. There were significantly less entries in the LogFile for the files deleted with Eraser and certain LogFile entries had time and dates of 00:00:00 01/01/1601 after Eraser implementation.

LNK files existed for all three of the deleted files. The Automatic Jump List for Microsoft Office Power Point verified the file “cars.ppt” was accessed with that application. The Jump List for Edge Browser verified the file “stolen_car_parts.pdf” was accessed with that application at a certain time and the Jump List for Microsoft Office Word 365 verified the file “how-to-steal-car.docx” was accessed with that application. The Jump List for File Explorer contained each of the three erased files.

Name	Modified Time	Change Time	Access Time	Created Time	Size
cars.lnk	2017-04-10 19:55:52 EDT	2017-04-10 19:55:52 EDT	2017-04-10 19:55:52 EDT	2017-04-10 19:37:36 EDT	574
classic_books.lnk	2017-04-10 19:56:01 EDT	2017-04-10 19:56:01 EDT	2017-04-10 19:56:01 EDT	2017-04-10 19:43:08 EDT	628
how-to-steal-car.lnk	2017-04-10 19:56:05 EDT	2017-04-10 19:56:05 EDT	2017-04-10 19:56:05 EDT	2017-04-10 18:56:43 EDT	643
John_Doe_Wiki.lnk	2017-04-10 19:56:09 EDT	2017-04-10 19:56:09 EDT	2017-04-10 19:56:09 EDT	2017-04-10 18:55:09 EDT	623
more_best_sellers.lnk	2017-04-10 19:56:17 EDT	2017-04-10 19:56:17 EDT	2017-04-10 19:56:17 EDT	2017-04-10 19:11:03 EDT	643
stolen_car_parts.lnk	2017-04-10 19:56:20 EDT	2017-04-10 19:56:20 EDT	2017-04-10 19:56:20 EDT	2017-04-10 19:52:57 EDT	638

Figure 34. LNK Files After Eraser Implementation- Autopsy

A prefetch file existed for the Eraser executable file and a hash search for the Eraser executable returned the file in the Downloads folder. Evidence of the Bing search for “eraser” and the URLs followed to download Eraser existed in the Microsoft Edge history database. These artifacts can be used to show the program existed as well as the times and dates it was run on the system under analysis.

Name	Modified Time	Change Time	Access Time	Created Time	Size
ERASER 6.2.0.2979.EXE-D5953862.pf	2017-04-19 20:34:00 EDT	2017-04-19 20:34:00 EDT	2017-04-19 20:34:00 EDT	2017-04-19 20:34:00 EDT	27933

Figure 35. Eraser Prefetch File- Autopsy

Source File	MD5 Hash
Eraser 6.2.0.2979.exe	56d30b6c54259910e9ac4642f43957de

Figure 36. Eraser Executable File- Autopsy

Container_11 [Table ID = 29, 25 Columns]
Uri
Visited: John@https://www.bing.com/search?q=eraser+tool&form=EDGHPC&q=PF&cvid=2e21014d505c4eb497f86bd48ee3a07d&pp=eraser+tool&cc=US&setlang=en-US
Visited: John@ms-appx-web://microsoft.microsoftedge/assets/errorpages/dnserror.html
Visited: John@https://eraser.heidi.ie/
Visited: John@https://eraser.heidi.ie/download/
Visited: John@https://sourceforge.net/projects/eraser/files/Eraser%206/6.2/Eraser%206.2.0.2979.exe/download
Visited: John@https://sourceforge.net/projects/eraser/postdownload?source=dlp

Figure 37. Eraser Internet History- ESEDatabaseView

Summary

Eraser offers a user-friendly interface to overwrite the content of desired files and folders. All three erasure methods, 1-pass, 3-pass and 7-pass rendered the file content inaccessible at its original location. Files with the original erased file names and metadata remained in their original locations. LNK files, Jump Lists, and MFT, LogFile and USN Journal entries confirm the erased files existed and were accessed on the system by a specific user. Phrase searches for contents of the erased files recovered the full contents of two of the three files from artifacts created by Microsoft Edge and Windows Defender. Although time-consuming, searching for known keywords or phrases may prove extremely beneficial by turning up entire documents in automatically created Windows files. A phrase search for the names of the erased files returned a file associated with the Eraser program, providing strong indication the tool had been used on those files. Prefetch files, a hash search for the executable file and internet history artifacts can also be used to confirm the Eraser tool existed and was executed on this system.

CCleaner

Tool Description

Piriform Ltd. (2017) describes CCleaner as a “small, effective utility” that cleans out the ‘junk’ that accumulates over time and protects your privacy by cleaning your browsing history and temporary internet files. CCleaner’s webpage also states, “if you run CCleaner with its default settings, you’ll thwart most attempts at recovery”. CCleaner offers multiple versions including free, professional and professional plus. The professional version allows a 14-day free trial that offers complete cleaning, automatic history cleaning and real-time junk monitoring.

Tool Implementation and Evaluation

CCleaner Professional v5.29.6033 was downloaded from piriform.com/ccleaner/download/professional using Microsoft Edge. The setup executable file was run from the Downloads folder and CCleaner Professional was installed on test machine “Test4”. The application was run from the shortcut created on the Desktop and the free trial was initiated. Three files were added to “files and folders to be deleted” including “cars.ppt”, “stolen_car_parts.pdf” and “how-to-steal-car.docx”. All other Windows and Application artifact options that could possibly contain a trace of those files were selected to be erased and the option to “Wipe Free Space” was selected with all default settings. The cleaner was run.

Cleaning Complete - (251.464 secs)			
272 MB removed.			
Details of files deleted			
	Microsoft Edge - Internet Cache	188,407 KB	3,554 files
	Microsoft Edge - Internet History	0 KB	5 files
	Microsoft Edge - Cookies	1,071 KB	252 files
	Microsoft Edge - Download History	0 KB	1 files
	Microsoft Edge - Session	167 KB	7 files
	Windows Explorer - Recent Documents	12 KB	17 files
	Windows Explorer - Thumbnail Cache	4,111 KB	15 files
	System - Empty Recycle Bin	0 KB	2 files
	System - Temporary Files	76,799 KB	71 files
	System - Windows Log Files	5,627 KB	25 files
	System - Windows Error Reporting	30 KB	11 files
	System - Font Cache	328 KB	1 files
	Advanced - Old Prefetch data	1,341 KB	105 files
	Advanced - Custom Files and Folders	863 KB	3 files
	Applications - Office 2016	128 KB	10 files
	Utilities - Windows Defender	40 KB	5 files

Figure 38. CCleaner Files Deleted

The base machine and the Test4 machine were viewed in FTK Imager and Autopsy. On the base machine, before CCleaner implementation, the three erased files were visible in the user's Documents folder.

Name	Modified Time	Change Time	Access Time	Created Time	Size
best_selling_classics.docx	2017-04-10 19:10:38 EDT	2017-04-10 19:10:38 EDT	2017-04-10 19:10:38 EDT	2017-04-10 19:06:53 EDT	13188
book_revenue.xlsx	2017-04-10 19:51:37 EDT	2017-04-10 19:51:37 EDT	2017-04-10 19:51:37 EDT	2017-04-10 19:46:03 EDT	8037
cars.ppt	2017-04-10 19:36:11 EDT	2017-04-10 19:37:37 EDT	2017-04-10 19:36:11 EDT	2017-04-10 19:36:11 EDT	259072
cars.ppt:Zone.Identifier	2017-04-10 19:36:11 EDT	2017-04-10 19:37:37 EDT	2017-04-10 19:36:11 EDT	2017-04-10 19:36:11 EDT	26
classic_books.pptx	2017-04-10 19:45:53 EDT	2017-04-10 19:45:53 EDT	2017-04-10 19:45:53 EDT	2017-04-10 19:42:58 EDT	89029
desktop.ini	2017-04-10 18:24:30 EDT	2017-04-10 18:24:30 EDT	2017-04-10 18:24:30 EDT	2017-04-10 18:24:30 EDT	402
how-to-steal-car.docx	2017-04-10 19:04:28 EDT	2017-04-10 19:04:28 EDT	2017-04-10 19:04:28 EDT	2017-04-10 18:56:02 EDT	12910
John_Doe_Wiki.pdf	2017-04-10 18:55:11 EDT	2017-04-10 19:56:09 EDT	2017-04-10 18:55:09 EDT	2017-04-10 18:55:09 EDT	393974
more_best_sellers.txt	2017-04-10 19:12:23 EDT	2017-04-10 19:12:23 EDT	2017-04-10 19:10:58 EDT	2017-04-10 19:10:58 EDT	220
stolen_car_parts.pdf	2017-04-10 19:52:58 EDT	2017-04-10 19:56:20 EDT	2017-04-10 19:52:57 EDT	2017-04-10 19:52:57 EDT	611327

Figure 39. Documents Folder Before CCleaner Implementation- Autopsy

Name	Modified Time	Change Time	Access Time	Created Time	Size
best_selling_classics.docx	2017-04-10 19:10:38 EDT	2017-04-10 19:10:38 EDT	2017-04-10 19:10:38 EDT	2017-04-10 19:06:53 EDT	13188
book_revenue.xlsx	2017-04-10 19:51:37 EDT	2017-04-10 19:51:37 EDT	2017-04-10 19:51:37 EDT	2017-04-10 19:46:03 EDT	8037
classic_books.pptx	2017-04-10 19:45:53 EDT	2017-04-10 19:45:53 EDT	2017-04-10 19:45:53 EDT	2017-04-10 19:42:58 EDT	89029
desktop.ini	2017-04-10 18:24:30 EDT	2017-04-10 18:24:30 EDT	2017-04-10 18:24:30 EDT	2017-04-10 18:24:30 EDT	402
John_Doe_Wiki.pdf	2017-04-10 18:55:11 EDT	2017-04-10 19:56:09 EDT	2017-04-10 18:55:09 EDT	2017-04-10 18:55:09 EDT	393974
more_best_sellers.txt	2017-04-10 19:12:23 EDT	2017-04-10 19:12:23 EDT	2017-04-10 19:10:58 EDT	2017-04-10 19:10:58 EDT	220

Figure 40. Documents Folder After CCleaner Implementation- Autopsy

After the execution of CCleaner, the three erased files were no longer visible in the Documents folder; this was the first of the three file deletion tools in this research to successfully remove the files from the Documents folder display in FTK Imager and Autopsy.

A hash search for the three erased files returned no results. A keyword search in Autopsy for erased file name “cars.ppt” disclosed a file created by Windows Defender that contained the URL this PPT was downloaded from. A search for substring “stolen_car_parts.pdf” returned a few Windows artifacts indicating the file existed and was visited under a specific user profile. Results for the substring search “how-to-steal-car.docx” also confirmed the file existed and was viewed on this system.

A search for the phrase “online car industry” from the file “cars.ppt” and the phrase “Stolen Vehicle Parts” from the erased file “stolen_car_parts.pdf”, both of which returned useful results in Test 3, returned no matches after CCleaner implementation. Phrase searches for substrings from “how-to-steal-car.docx” returned no matches.

CCleaner successfully removed traces of the selected files from the MFT, LogFile and USN Journal however, USN Journal entries were created for the erased files and other related files when CCleaner was run. Once this time and date are known, the MFT, LogFile and USN Journal can be searched for other activity in that timeframe. A search for this date revealed multiple entries in the MFT with file names “[unknown]\ZZZ..Z...ZZ..Z.Z\Z.Z.ZZ..Z...Z..Z”. The existence of unusual file names like this is an indication of anti-forensic use.

LNK files no longer existed for any files in the user's account, however Automatic Jump Lists still existed for this user and a Custom Jump List was created for the CCleaner application. The Automatic Jump List for File Explorer contained evidence that the three files had been associated with that application. The Automatic Jump List for Microsoft Office Power Point verified the file "cars.ppt" was accessed with that application. The Jump List for Edge Browser verified the file "stolen_car_parts.pdf" was accessed with that application at a certain time and the Jump List for Microsoft Office Word 365 verified the file "how-to-steal-car.docx" was accessed with that application.

A prefetch file existed for "CCLEANER.EXE", "CCLEANER64.EXE" and "CCSETUP529PRO.EXE" on the Test4 machine. These reveal the time, date and run count of each executable file. CCleaner prefetch file run times match the time of the "ZZZZ" file entries in the USN Journal which connects the CCleaner application with specific erased files. A hash search for the executables returned "CCleaner.exe" and "CCleaner64.exe" located in a "Program Files" folder named "CCleaner". The hash search located "ccsetup529pro.exe" in the user's Downloads folder. Evidence of the Bing search for "ccleaner" could not be located in Microsoft Edge history however a cookie existed from the Piriform website. In addition, locations for common internet artifacts such as the Microsoft Edge cache and last browse session were empty or nonexistent.

Name	Modified Time	Change Time	Access Time	Created Time	Size
CCLEANER.EXE-D4D76A60.pf	2017-05-08 02:28:07 EDT	2017-05-08 02:28:07 EDT	2017-05-08 02:19:19 EDT	2017-05-08 02:19:19 EDT	5030
CCLEANER64.EXE-779BD542.pf	2017-05-08 02:28:15 EDT	2017-05-08 02:28:15 EDT	2017-05-08 02:15:41 EDT	2017-05-08 02:15:41 EDT	17239
CCSETUP529PRO.EXE-E65E5907.pf	2017-05-08 02:14:27 EDT	2017-05-08 02:14:27 EDT	2017-05-08 02:14:27 EDT	2017-05-08 02:14:27 EDT	35357

Figure 41. CCleaner Prefetch Files- Autopsy



Source File	MD5 Hash
 CCleaner.exe	ffe2d028d996bc6279a2e4894f9fcbfd
 CCleaner64.exe	638ae77dc319958727fbaa403d37b2d6
 ccsetup529pro.exe	e32244fff9a4bf7f291562186484d950

Figure 42. CCleaner Executables- Autopsy

Summary

CCleaner offers a graphical user interface displaying multiple categories of artifacts to be “cleaned” from a system. The cleansing of Microsoft Edge artifacts and certain Windows, System and application files removed a few common forensic artifacts but left others untouched. There was no indication of the three erased files in their original location. MFT, LogFile and USN Journal entries for the erased files had been removed, however USN Journal entries were created with the use of CCleaner, pinpointing the time and date the tool was run. Keyword searches for the erased file names and contents returned Jump Lists and other Windows artifacts that indicate these files existed and were accessed at a certain point in time. The content of the files erased with CCleaner were unrecoverable in this test. Finally, a hash search for the CCleaner executable files and setup file successfully detected their existence and prefetch files confirmed their execution on this system.

Anti-Anti-Forensics

The good news for computer forensic investigators is that there are unavoidable flaws in anti-forensic tool and technique use. Common anti-forensic approaches such as data destruction and data transformation may destroy commonly evaluated evidence, however, as demonstrated in this paper and in previous research, operating systems harbor a myriad of automatically created artifacts that can prove valuable in a forensic investigation.

Certain forensic software have implemented mitigation efforts to detect and combat anti-forensic efforts. The EnCase tool suite can retrieve hidden data in host protected areas (HPA) and device configuration overlays (DCO) and many current versions of forensic tools recognize and alert investigators to file extension mismatches when they suspect file extension alterations have occurred. Packages have been designed for detection of specific anti-forensic techniques; at the time of this research, Guidance Software offers an Enscript that displays the eight NTFS timestamps in EnCase including those in the FNA that are typically overlooked by forensic tools (Guidance Software, 2017).

Analysts can mitigate data destruction techniques by seeking parts of files left by tools to prove the data was previously on the device under analysis. Hash and keyword searches can uncover pieces and, in some cases, the full content of files that have been securely erased. Some data destruction tools rename files with odd naming conventions such as multiple z's in a row. File names that look out of place can be worth looking into as they may have been intentionally destroyed. LNK files and Jump Lists are quick ways to show a file existed and was accessed on a system and can contain metadata useful for timeline formation. Cleaning of the thumbnails cache is often left out by disk cleaning tools and therefore may contain items that are no longer present elsewhere. An examination of volume shadow copies can uncover erased or encrypted files if the shadow copy was created before the file was destroyed or transformed.

Data transformation techniques such as transmutation and file signature masking can be fought with the use of "fuzzy hashing" which identifies similar but not identical files. An examination into Jump Lists can help identify transformed files; a DLL recently opened with Power Point would be an indication that alterations have occurred. Repetitive, missing or unlikely timestamps can indicate alterations of metadata that may affect the creation of an

accurate forensic timeline. Michael Perklin (2012) suggests mitigating timestamp alterations by ignoring metadata altogether and, instead, examining log files that record dates as strings (as these are more difficult to zero out or overwrite). The use of Triforce's ANJP tool in this research revealed original timestamps for altered files in each, the MFT, LogFile and USN Journal. Cross-referencing time-based data residing in log files, registry entries and the MFT can be used to detect metadata tampering by analyzing discrepancies or similarities among various evidence sources (Jain & Chhabra, 2014).

To detect anti-forensic tool use, hash analysis can be used once a hash database of known anti-forensic tool downloadable and executable files is compiled. The presence of anti-forensic software or applications alone is not necessarily incriminating but, depending on the circumstances, this may play an integral role in showing intent.

Conclusion

The goal of this research was to test at what level current anti-forensic tools and techniques confound traditional computer forensic tools as well as to provide recommendations for detecting, mitigating and combatting effects of anti-forensic techniques.

The tests conducted in this research show that many freely available anti-forensic tools do not render evidence completely inaccessible or difficult to uncover. Like Pajek and Pimenidis' 2009 conclusion, the findings of this research suggest it is likely the detection efficiency achieved by computer forensic tools depends on the sophistication of the anti-forensic techniques applied. Although the time-altering tool in this research successfully altered timestamps of files and directories as they were displayed in FTK Imager and Autopsy, many other artifacts revealed accurate timestamps for the altered files. In many cases, the tools applied for erasing data did not successfully delete all traces of the data; filenames and metadata of the

deleted files were visible in FTK Imager and Autopsy at their original locations in two of the three erasing tool implementations and the deleted filenames were recoverable in all tests by examining LogFile and USN Journal entries. The contents of many erased files were visible through FTK Imager and Autopsy by examining Microsoft artifacts including the Microsoft Edge cache, Windows thumbnails and Windows Defender scans. Each of the four tools left evidence that they were run in the form of prefetch files and the executable and/or setup files for each anti-forensic tool were located through a hash search in Autopsy.

This research had multiple limitations that future studies should address. The short time-frame of activity on the test machines may or may not be representative of a computer in the field. This contributed to a lack of volume shadow copies and allowed fairly quick keyword and hash searching in forensic tool Autopsy. In addition, this research was done under relaxed time constraints. Examining every potentially useful artifact in a large file system may not be feasible due to time and budget constraints in the forensics field.

While the anti-forensic tools used in this study were all freely and effortlessly accessible, many more anti-forensic tools are available that may implement additional anti-forensic techniques if an individual is willing to pay for that service. Piriform's CCleaner Pro, offered for \$24.95 per year, allows a user to set scheduled cleanings of temporary files, internet history, cookies, downloads, and autocomplete forms for the five most popular browsers along with allowing the removal (overwriting) of recycle bin items, recent documents, Windows log files, old registry entries and other third-party files (Piriform Ltd., 2017). Tracks Eraser Pro, available for \$29.95 per computer, offers many of the same features along with application-specific plugins for more thorough erasing (Acesoft, 2015). Invisible Secrets, available for \$39.95 per year, offers the ability to hide files in places "that appear innocent", i.e. picture files, sound files

or web pages (NeoByte Solutions, 2014). BatchPurifier, available for \$19 per computer, offers removal of hidden data and metadata from multiple file types (Digital Confidence, 2017).

The forensic examination tools used in this research were also openly available. Paid, commercial forensic tools may automatically uncover attempts at anti-forensic techniques and future studies should test their ability to detect and mitigate these effects. Future studies may also summarize best practices given certain factors of a case such as topic of the investigation, priorities of the investigation, total size of evidence files and time constraints. Al Fahdi, Clarke & Furnell (2013) predict the ability to locate relevant evidence will become increasingly challenging as people become more mindful of information security and as the promotion of anti-forensic technology for personal, legitimate use continues to grow. Garfinkel (2010) suggests it may be beneficial for forensic tools to migrate from identifying criminal evidence to identifying artifacts that may highlight misuse in order to inform an examiner on how to best proceed in the investigation. Continued, collaborative research into current and anticipated anti-forensic trends by the technical community, law enforcement professionals and forensic tool vendors will prove beneficial to the future of digital forensic investigations.

Locard's Exchange Principle (Gale, 2005) is often referenced in forensics work. This principle holds that with contact between two items, there will be an exchange. In traditional forensics, this involves the cross-exchange of physical evidence such as fingerprints, hairs, fibers and soil between a criminal and a crime scene. In the digital world, a perpetrator doesn't necessarily make physical contact with a crime scene but makes virtual contact leaving traces of digital evidence in the exchange. In the words of Blunden (2009), "there's no such thing as a foolproof anti-forensic tactic. With the right tools and know-how, it's just a matter of time before a savvy forensic investigator will overcome the fortifications that you've established."

References

- Acesoft. (2015). Buy Tracks Eraser. Retrieved from <http://www.acesoft.net/buy.htm>
- Afonin, O., Nikolaev, D., & Gubanov, Y. (2015). Countering Anti-Forensic Efforts - Part 1. Retrieved from <http://www.forensicmag.com/article/2015/09/countering-anti-forensic-efforts-part-1>
- Al Fahdi, M., Clarke, N. L., & Furnell, S. M. (2013, August). Challenges to digital forensics: A survey of researchers & practitioners attitudes and opinions. In Information Security for South Africa, 2013 (pp. 1-8). IEEE.
- Antonovich, Chris (2014). Jump List Forensics. Leahy Center for Digital Investigation. Champlain College. Retrieved from www.champlain.edu/Documents/LCDI/Jump%20List%20Forensics.pdf
- Bilby, D. (2006). Low Down and Dirty: Anti-Forensic Rootkits. Retrieved from Backhat Japan: <https://www.blackhat.com/presentations/bh-jp-06/BH-JP-06-Bilby-up.pdf>
- Bishop Fox. (2006). Metasploit Anti-Forensics Project. Retrieved from <https://www.bishopfox.com/resources/tools/other-free-tools/mafia/>
- Blunden, B. (2009). Anti-Forensics: The Rootkit Connection. Black Hat USA Conference.
- DataForensics. (2015). Microsoft Edge Forensics – Where to Find Artifacts? Retrieved from <http://www.dataforensics.org/microsoft-edge-browser-forensics/>
- Digital Confidence. (2017). BatchPurifier 7.2. Retrieved from <http://www.digitalconfidence.com/BatchPurifier.html>.
- Erasani, S. (2010). Implementation of Anti-Forensic Mechanisms and Testing with Forensic Methods. Department of Computing Sciences, Texas A&M University-Corpus Christi.
- Eraser – Erase Files from Hard Drives. (2016). Retrieved from <https://eraser.heidi.ie/>

- Gale, Thomson. (2005). "Locard's Exchange Principle." World of Forensic Science. Retrieved from Encyclopedia.com: <http://www.encyclopedia.com/science/encyclopedias-almanacs-transcripts-and-maps/locards-exchange-principle>
- Garfinkel, S. (2007). Anti-Forensics: Techniques, Detection and Countermeasures. 2nd International Conference on i-Warfare and Security, 77.
- Garfinkel, S. (2010). Digital forensics research: The next 10 years. *Digital Investigation*, 7, S64-S73.
- Guidance Software. (2017). EnCase App Central. Retrieved from <https://www.guidancesoftware.com/app/MFT-Date-Comparator>
- Harris, R. (2006). Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem. *Digital Investigation*, 3, 44-49. doi:10.1016/j.diin.2006.06.005
- Jain, A. & Chhabra, G. S. (2014). Anti-Forensics Techniques: An Analytical Review. IEEE, doi:10.1109/IC3.2014.6897209
- McQuaid, J. (2014). Forensic Analysis of LNK files. Retrieved from <https://www.magnetforensics.com/computer-forensics/forensic-analysis-of-lnk-files/>
- Mueller, L. (2009, February). Detecting timestamp changing utilities. Retrieved from <http://www.forensickb.com/2009/02/detecting-timestamp-changing-utilities.html>
- Neobyte Solutions. (2014). Invisible Secrets 4. Retrieved from <http://www.invisiblesecrets.com/order/html>
- Offensive Security. (2017). TimeStomp. Retrieved from <https://www.offensive-security.com/metasploit-unleashed/timestomp/>
- Oracle Corporation. (2017). Oracle VM VirtualBox User Manual. Retrieved from <https://www.virtualbox.org/manual/ch01.html>

- Pajek, P., & Pimenidis, E. (2009). Computer Anti-forensics Methods and Their Impact on Computer Forensic Investigation. *Global Security, Safety, and Sustainability Communications in Computer and Information Science*, 145-155. doi:10.1007/978-3-642-04062-7_16
- Patzakis, J. (2002). Encase Legal Journal Second Edition. Retrieved from <http://www.cosgrovecomputer.com/documents/EnCase%20Legal%20Journal.pdf>
- Perklin, M. (2012). Anti-Forensics and Anti-Anti-Forensics: Attacks and Mitigating Techniques for Digital-Forensic Investigations. Retrieved from <https://www.youtube.com/watch?v=BCnjKEFOH1M>
- Piriform Ltd. (2017). Introducing CCleaner. Retrieved from <http://www.piriform.com/docs/ccleaner/introducing-ccleaner>
- Reith, M., Carr, C., & Gunsch, G. (2002). An Examination of Digital Forensic Models. *International Journal of Digital Evidence*, 1(3), 1-12.
- Rekhis, S. & Boudriga, N. (2012). A Hierarchical Visibility Theory for Formal Digital Investigation of Anti-Forensic Attacks. *Computers & Security*, 31, 967-982.
- SDelete v2.0. (2017). Retrieved from <https://technet.microsoft.com/en-us/sysinternals/sdelete.aspx>
- SleuthKitWiki. (2017). SQLite Database v6 Schema. Retrieved from http://wiki.sleuthkit.org/index.php?title=SQLite_Database_v6_Schema
- USCERT. (2006). Computer Forensics. Retrieved from http://www.uscert.gov/reading_room/forensics.pdf
- WPATHULIN. (2013). Interpretation of NTFS Timestamps. Retrieved from <https://articles.forensicfocus.com/2013/04/06/interpretation-of-ntfs-timestamps/>

Appendix A

Forensic Tool	Identifier	Description
AccessData® FTK® Imager 3.4.2.6	FTK Imager	Opens vmdk image to collect data
Autopsy® 4.3.0	Autopsy	Opens dd image to collect data
Triforce ANJP 3.11.07	ANJP	Parses Windows log files
Nir Sofer ESEDatabaseView v1.50	ESEDatabaseView	Opens .dat files
PECmd version 0.9.0.0	PECmd	Windows prefetch parser
Thumbcache Viewer 1.0.3.4	Thumbcache Viewer	Windows thumbnail viewer