Digital Tool Marks (DTMs): a forensic analysis of file wiping software

Journal:	*Australian Journal of Forensic Sciences*
Manuscript ID	TAJF-2019-0076
Manuscript Type:	Original Paper
Keywords:	File Wiping, Digital Forensics, Digital Tool Marks, Forensics

Digital Tool Marks (DTMs): a forensic analysis of file wiping software

Abstract
Whilst difficult to ascertain the full extent to which so called anti-forensic software applications are in use by the public, their threat to an investigation of digital content is tangible, where of particular interest is the use of file wiping tools, which remains the focus of this work. This work presents the examination of eight freely available wiping tools in order to identify the existence of 'digital tool marks' (DMTs) left on a system following their use. Further attempts are made to ascertain whether such DTMs can be attributable to a particular wiping tool. Analysis is focused on the impact each tool has on system at a file system level, where in this work both FAT32 and NTFS are the subject of investigation. DMTs relating to each wiping tool are provided and recoverable file system metadata post-wipe is described.

Keywords: File Wiping; Digital Forensics; Deletion; Recovery; Digital Tool Marks; Forensics

1 Introduction

Software applications capable of wiping digital data from a device (termed 'wiping tools (WTs)), placing content beyond current forensic powers of recovery are now available online for download and have been for some time. Many solutions offered are available free of charge and easily locatable via an online search engine query, making them an accessible option for those seeking to remove specific digital content from their system. For a WT to be effective it must completely overwrite a targeted set of digital data rendering this original content non-recoverable. Anecdotally, historic WTs are considered to have implemented a standard delete function as their primary way of removing data from a system, mistakenly under the guise of terminology which describes 'wiping'. These processes may leave data retrievable using typical forensic file recovery methods. Arguably now, as a greater understanding exists regarding how file allocation and recovery is performed with such knowledge largely available online and to some degree publicised in the media, many WTs are likely to implement algorithms which effectively overwrite data. As a result, when a WT is now discovered as part of a digital forensic (DF) investigation, the chance of file content being recoverable providing a WT has been used effectively is diminished. The 'effective use' of a WT requires emphasis (as wiped data is generally considered not recoverable[1] and implies that such applications can be used in a way which compromises their goal of data removal. Whilst possible, WTs may be pre-configured on install to perform their actions effectively meaning default settings may lead to a complete data removal. As such, these tools are accessible for those with limited computing knowledge where it is suggested they have been previously commonly used[2]. As a result, it is argued that the DF practitioner can no longer rely on the misuse of a WT in order to recover file content subject to such an application or expect to recover the original file content through carving processes.

Given the aforementioned statements, it becomes easy to be of the mindset of 'why bother' in cases of suspected WT usage given that the data retrieval post-wipe is unlikely. Yet evidence of WT usage may in itself be an inference of a criminal act and in some circumstances be prohibited (for example, in cases where periodic computer checks are required under various supervision orders to enable an offenders usage history to be vetted). Although finding original (and potentially offending) file content following the attempted use of a WT may be considered the 'holy grail', any evidence denoting tool usage should also be considered an asset in terms of establish offender behavior and in some cases may be a primary source of evidence in its self.

Despite being historic in reference, Kessler[3] highlights that in relation to WTs, 'most of the programs leave identifiable traces of the wiping', a statement which is explored as part of this work. Evidence of usage left behind on a system can be classified as a 'digital tool mark' (DTM) drawing direct reference to the traditional forms of tool mark analysis carried out by traditional forensic science disciplines [4,5]. DTMs which can be attributed to WT usage may support the identification of how a WT was used, when it was used and with what frequency. In addition, following the removal of a WT, any remaining DTMs may support a practitioner's understanding of any resulting digital data and in some cases allow any digital trace evidence to be attributed to the use of a specific WT. All of these assertions require forensic exploration and remain an area under researched by the DF community.

This work forensically examines the impact of eight freely available WTs when used to wipe test file content on both FAT32 and NTFS file systems. File system metadata is discussed demonstrating the DTMs left by each tool following their use. This work opts to focus on file system metadata as it remains an under researched area in regards to WT DTMs where often, focus is given to operating system artefacts (for example, Prefetch, Link Files, Registry information). The impact of the chosen WTs on both file systems are documented in order to demonstrate any apparent DTMs attributable to each tool. Finally results are discussed and conclusions drawn.

2 Methodology

Generally WTs are placed under the umbrella term of anti-forensics (AF) and although some historic work considering and exploring the forensic examination of these tools exists (see [6,7,8]), there are limited current works. This work explores WT usage at the file system level examining the impact of such tools on file system metadata. Figure 1 documents the methodology deployed within this work. All file system time analysis was carried out using 'FTKi' (v4.2.0.13) to query file system metadata and 'NTFS $Logfile Parser' (v2.0.0.46) to query NTFS $Logfile content. The same test data set (consisting of six standard files (four *.jpg and tw0*.pdf)) was utilised in all tests for consistency and to support the identification of any recoverable content following wiping actions.

Figure 1: An overview of the methodology deployed in this article.

This work examines the following eight wiping tools, brought to public attention by Digital Citizen[9] in their article – '*8 best free data erasing apps that permanently delete your files and folders*'.

Table 1:- List of examined WTs and their download links.

2.1 Configuration and Terminology

It is necessary to consider the 'out of the box' configuration of downloaded tools and where relevant, this work highlights the relevance of any preconfigured settings in operation for each of the eight chosen tools. Identifying default settings allow an evaluation of each tool's potential DTMs to be assessed when an application is utilised by the hypothetical 'lay-person' who possesses limited technical knowledge. In addition, terminology bespoke to each tool will be highlighted where necessary. To provide an example, a comparison between CCleaner's utilisation of well-known wiping algorithm terminology such as Gutmann against Freeraser's use of the term 'fast' is drawn. Whilst without access to source code information, neither process can be 100% validated, the use of the term 'fast' to describe a tool's chosen wiping algorithm is non-descript and in need of investigation.

2.2 CCleaner (v5.51.6939)

CCleaner is a facility to 'clean' a computer system and whilst paid options are available, it can also be downloaded for free. The term clean encompasses a wide range of actions from redundant file removal to privacy enhancing features such as Internet browser history removal. CCleaner reports over 5 billion downloads worldwide and over 35 million GB of data 'cleaned every month[10] and is one of the most popular tools of this type available online. The popularity and mainstream publicity of this tool increases the likelihood of its usage in device maintenance and therefore the potential of it impacting upon digital forensic investigations.

2.2.1 Setup

Analysis of CCleaner commences with its 'out the box' setup (its default configuration once installed). It is necessary to consider 'out the box' set up due to the potential diversity of users which inherently come with varying degrees of technical knowledge. This becomes important as CCleaner has the ability to remove digital data with varying levels of success, depending on its set up. Following testing, CCleaner is configured by default to carry out a 'normal' file deletion as opposed to 'secure' deletion. Testing confirmed that after a 'normal' file deletion, file content was still recoverable using forensic methods. A user must configure CCleaner to operate a secure wipe (with a default setting of one pass selected for this option, with three, seven and 35 passes available). Following testing, all secure options resulted in the non-recovery of test files. Whilst such configuration changes are minor, it does require an 'opt-in' from a user, where the act of doing so potentially infers an acknowledgement of understanding of this process. Further, the need to make this change means that potentially, those seeking an 'out the box' solution for file wiping may mistakenly believe CCleaner achieves this by default and as a result, recoverability may be more likely.

2.2.2 CCleaner on FAT32

The first consideration of CCleaner usage is on the FAT32 file system (see [11] for an in depth discussion of FAT file system structures which are applicable to work covered in this article). Settings were configured to ensure secure wiping was enabled. Figure 2 demonstrates the file system metadata changes which occur following a CCleaner wipe of test file content.

Figure 2. A comparison of file system metadata for a FAT32 formatted device when CCleaner 1 pass wipe is enabled.

Following the use of CCleaner to wipe test target files on a FAT32 file system, the following key metadata changes occur:

File Name: Following tests, both the Long and Short filenames for the wiped file are consistently changed to 'Z' characters. The length of the filename remains the same (a ten character filename is replaced with 10 'Z' characters). A file's extension is also replaced with 'Z' characters. The presence of 'Z'-ed file names is a consistent DTM of CCleaner use.

Time & Date: Whilst the modified time following wiping reflects the 'time of wipe', the created date for the file remains uncompromised and reflects the time and date when the file was created on the target media (all intricacies surrounding created date allocation still apply, i.e. copying of files to a new volume etc.). The accessed date is also updated to reflect the date of wipe.

Wiped Area: CCleaner's wiping process wipes the entire physical disk space allocated to the file (both logical file size and slack space).

CCleaner's securing wiping processes are effective in preventing recovery of any file content.

2.2.3 CCleaner on NTFS

The second consideration of CCleaner usage is on the NTFS file system (see [11] for an in depth discussion of NTFS file system structures). As with above, settings were configured to ensure secure file deletion occurs. Figure 3 demonstrates the file system metadata changes which occur following a CCleaner wipe.

Figure 3: An example of file system metadata changes following a CCleaner wipe.

Following the use of CCleaner to wipe test target files on a NTFS file system, the following metadata changes occur:

File Name: Following tests, as with FAT32, the filename for the wiped file are consistently changed to 'Z' characters. The length of the filename remains the same (a ten character filename is replaced with 10 'Z' characters). A file's extension is also replaced with 'Z' characters. Following a wipe, the $MFT maintains no original file name information.

Time & Date: In comparison to FAT32, the NTFS file system maintains a more complex structure with multiple time and date recording attributes. As a starting point, the $MFT entry for the target file records the created and accessed timestamps for the target file post-wipe as unchanged. Modified and changed timestamps post-wipe reflect the time of wiping. INDX information is no longer available.

Wiped Area: CCleaner's wiping process wipes the entire physical disk space allocated to the file (both logical file size and slack space).

CCleaner's secure wiping process is effective in preventing recovery of any file content. CCleaner wipes file content with hex characters '0x00' on both FAT32 and NTFS and there is no option to amend this.

2.2.4 The $Logfile

One of the features of NTFS is its Log File Service (LFS) which provides potential recoverability in the case of a crash[12]. LFS information on an NTFS formatted disk is contained within the $Logfile. To understand the importance of the $Logfile in terms of establishing file system metadata, Microsoft[13] provides the following overview.

> "*NTFS views each I/O operation that modifies a system file on the NTFS volume as a transaction, and manages each one as an integral unit. Once started, the transaction is either completed or, in the event of a disk failure, rolled back (such as when the NTFS volume is returned to the state it was in before the transaction was initiated).*
>
> *To ensure that a transaction can be completed or rolled back, NTFS records the suboperations of a transaction in a log file before they are written to the disk. When a complete transaction is recorded in the log file, NTFS performs the suboperations of the transaction on the volume cache. After NTFS updates the cache, it commits the transaction by recording in the log file that the entire transaction is complete.*
>
> *Once a transaction is committed, NTFS ensures that the entire transaction appears on the volume, even if the disk fails*." (Microsoft, 2008)

CCleaner does not wipe NTFS $Logfile content and as a result, the $Logfile captures file transactions on the file system including file interactions carried out by CCleaner (such as those noted in Figure 2). Whilst the $Logfile does not keep an indefinite record of transactions due to size limitations, a timely acquisition of this content can support the identification of file metadata pre-wipe as CCleaner does not wipe appropriate records from this transaction log. Using Joakim Schicht's 'NTFS $Logfile Parser' tool (https://github.com/jschicht/LogFileParser) $Logfile content parsing is possible. Referring back to Figure 2, it can be seen that the test file 4n6_Content.docx has its metadata residing in $MFT record 41. Using NTFS Logfile Parser it is possible to establish 'all the various filenames a given $MFT Record has had during the timespan that the $Logfile covered'[14]. Therefore an examination of MFT records which have been 'Z'-ed can potentially reveal the original name assigned to the file before the wipe took place.

2.2.5 Usage Limitation for Forensic Exploitation

CCleaner has a number of preconfigured 'cleaning facilities (browser targeted artefact removal, registry key removal), but of focus in this work is the ability for a user to customise the software to target user created content (for example, imagery in a user defined folder on a profile Desktop which a user wishes to remove from their system). This is achievable in CCleaner in 2 ways; a single file selector or directory/drive selector. If the user opts

to select individual files, the file selector dialog box in CCleaner is subject to the same restrictions as Windows Explorer. If the user has opted not to show 'protected operating system files', then such content cannot be individually selected and wiped as CCleaner cannot see it. However, where a folder has been wiped, all content is captured within the wipe process. Both individual file, and folder select cannot wipe previously deleted content as technically it cannot path to it as deleted content is not viewable via CCleaner's file/folder selector dialog box. Therefore where the user has interacted with file types (for example, the Microsoft Office Suite) which may generate temporary content which is then removed, such traces remain intact and may be of value when identifying what content was wiped (see Figure 4).

Figure 4: An example of temporary content remaining intact following an initiated wipe.

2.3 File Shredder (v2.5)

File Shredder offers a basic file and folder secure delete option with no comprehensive system cleaning features available like those seen with CCleaner. Unlike CCleaner, there is no option to standard delete, all options result in file content being placed beyond forensic powers of recovery. File Shredder by default offers a 'Simple One Pass' wipe and as a result this algorithm is the focus of this work. To commence, File Shredder usage on a FAT32 system is considered.

2.3.1 File Shredder on FAT32

Figure 5 documents the result of targeted file wiping via File Shredder. When files are selected for wiping, the original file container is maintained and a secondary numerically named file is created. The numeric naming of the file appears random following testing. The original file container maintains the original file's name and its created time stamp reflects the time and date when the file was created on the target media (all intricacies surrounding created date allocation still apply, i.e. copying of files etc.). The modified time stamp represents the time of wipe.

Figure 5: Simple one pass wipe by File Shredder on FAT32

Regardless of which of the secure wipe algorithms chosen (five in total, 'Simple one pass', 'Simple two pass', 'DoD 5220-22.M', 'Secure erasing algorithm with 7 passes' and 'Gutman'), Figure 4 documents the consistent DTMs left by File Shredder, including a file's extension being replaced with 'Z' characters. Following a file wipe, allocated file size if always zero.

2.3.2 File Shredder on NTFS

On NTFS, File Shredder functions differently, where the original file's container and metadata is gone (unlike on FAT32), replaced only by a numerically named wipe container as shown in Figure 6.

Figure 6: Simple one pass wipe by File Shredder on NTFS

Each of the numeric wipe files contains date and time information where the created time stamp denotes the time and date of the file's creation on the volume (all previously noted caveats apply) and the modified time stamp denotes the time of wipe.

File Shredder, as with CCleaner, does not wipe $Logfile content. Therefore as a result, it is possible to identify specific name changes for an $MFT record entry assigned to each numerically named wipe file. Similarly with FAT32, on NTFS File Shredder leaves DTMs in the form of numerically named files with 'Z' character extensions of file size zero.

2.4 Eraser (v 6.2.0.2982)

Eraser offers 13 wiping algorithms ranging from one pass to 35 (with known-named algorithms listed – i.e. US DoD 5220.22M). By default, Eraser is configured to deploy the Gutmann 35 pass wipe algorithm, which effectively removes file content following testing.

2.4.1 Eraser on FAT32

Following testing, wiped files are assigned an alphanumeric file name, minus a file extension. All available file system time stamps are replaced with the value 01-01-1980 and are of file size zero (see Figure 7).

Figure 7: Eraser use on FAT32

2.4.2 Eraser on NTFS
Similar to FAT32, wiped files on an NTFS file system are assigned an alphanumeric file name, minus a file extension. All available file system time stamps are replaced with the value 01-01-1601 except for the Date Changed (MFT) value which reports the time of wipe. $Logfile original file name information remains available as described in previous test cases.

2.5 Freeraser (v1.0.0.23)
Freeraser offers three modes of deletion, 'fast', 'forced' and 'ultimate', where all result in a successful file wipe. By default, 'fast' is selected, but no information is available as to what algorithm is utilised.

2.5.1 Freeraser on FAT32
Files wiped by Freeraser on FAT32 maintain their file name and size (see Figure 8). Whilst the modified time following wiping reflects the 'time of wipe', the created date for the file remains uncompromised and reflects the time and date when the file was created on the target media (all intricacies surrounding created date allocation still apply, i.e. copying of files etc.). The accessed date is also updated to reflect the date of wipe. File size metadata is still available

Figure 8: Freeraser on FAT32

2.5.2 Freeraser on NTFS
Files wiped by Freeraser on NTFS maintain their file name and size. Further, file system time stamp metadata remains available. File created time and date remain and reflected file created time and date, with the file modified timestamp reflecting the time of wipe. $Logfile metadata remains as a source of information as previously noted in past cases.

2.6 WipeFile (v2.4.1.0)
WipeFile offers 14 wiping algorithms for the user to choose from. By default, a one pass wipe is selected.

2.6.1 WipeFile on FAT32
Following a file wipe, created time and modified timestamps are replaced with 01/01/1980. File names are replaced with an alphanumeric string which in length and extension is equal to the length of original final name (see Figure 9).

Figure 9: WipeFile on FAT32

2.6.2 WipeFile (v2.4.1.0) on NTFS
Following a file wipe, the same file name issues exist as with FAT32 and all timestamps are set to 01/01/1980. $Logfile information remains in operation.

2.7 DPWipe (v1.1)
DPWipe offers six different wiping algorithms, where by default 'US DoD 5220.22-M 3x' is selected.

2.7.1 DPWipe on FAT32
Following a file wipe, three file containers are created as shown in Figure 10. All containers have identical file system metadata, where one file maintains the original file name, one has a file name of all 'a' characters and the final is a FAT short file named file with a randomised naming convention. The created time stamp remains intact with the modified timestamp reflecting the time of wipe. It remains difficult to establish a definitive DPWipe DTM with regards to the wiped file area are dependent upon the algorithm utilised.

Figure 10: DPWipe on FAT32

2.7.2 DPWipe on NTFS
A DPWipe on NTFS removes the files original name replacing it with a single 'a' character. File modified, accessed and created times are set to 01/01/1980. File size metadata is removed. Post wipe, MFT Filename timestamps remain where the 'created time' reflects the created time of the file on the volume and the 'changed' time reflects the time of wipe (see Figure 11). $Logfile information remains available.

Figure 11: DPWipe on NTFS

2.8 Moo0 Shredder (v1.21)

Moo0 Shredder offers 4 wipe types where by default it is set to 'Shred Once (Normally Unrecoverable)'. There is no discussion of what each algorithm does as bespoke informal names have been assigned to each (Vaporize (vanish it!), Into Ashes (Extremely Secure), Extra Careful (Even More Secure) and Shred Once (Normally Unrecoverable)).

2.8.1 Moo0 Shredder on FAT32

Following a wipe on FAT32, files names are replaced with an alphanumeric randomised string which is the same length in characters as the original file name. Randomly generated modified, accessed and created date and times are assigned to the wiped file (see Figure 12).

Figure 12: Moo0 Shredder on FAT32

2.8.2 Moo0 Shredder on NTFS

On NTFS, the same file naming principles apply post wipe as with FAT32. Randomly generated modified, accessed and created date and times are assigned to the wiped file. However, MFT Filename timestamps remain where the created time reflects the created time of the file on the volume and the 'changed' time reflects the time of wipe. $Logfile information remains available.

2.9 BitKiller (v2.0) on FAT32

BitKiller offers five wiping algorithms, where by default, it is set to wipe with 'Random data'.

2.9.1 BitKiller on FAT32

Following a wipe, BitKiller leaves both the original file metadata in place and creates a second file container of nine characters long (random character naming) which the same time stamp information. The created timestamp reflects the time and date the file was created on that volume and the modified time reflects the time of wipe (see Figure 13).

Figure 13: BitKiller on FAT32

2.9.2 BitKiller on NTFS

Unlike with FAT32, on NTFS, post wipe, no file name information remains (except in the $Logfile). MFT Filename timestamps remain where the 'created time' reflects the created time of the file on the volume and the 'changed' time reflects the time of wipe (see Figure 14).

Figure 14: BitKiller on NTFS

2.10 Wipe Summary

To provide a collated summary of the performance and available file system metadata post-wipe from all eight tools, Tables 2 and 3 offer a breakdown of actions on both FAT32 and NTFS.

Table 2: Wipe summary on FAT32

Table 3: Wipe summary on NTFS

3 Conclusions

The volatility of digital data and ease of its destruction now mean that DF practitioners must be prepared to face more questions regarding why a system may be absent of a specific evidence type. Whilst it may not be possible to answer this question in some cases, the presences of DTMs may offer some insight into a user's behaviour with

regards to their system. DTMs remain an under researched area in DF and as a result, the field maintains limited documented evidence of the remnants left behind on a device by WTs. Increasing a practitioner's knowledge of DTMs in this area provides an opportunity to support the practitioner during an investigation to address the following points:-

1. To identify what tool has potentially been used to remove content from a system.
2. To understand what volume and type of evidence is likely to be left following an assumed tools' usage.
3. To identify any potentially limits imposed on the investigation given the likelihood of a specific tool being used.

This work provides the DTMs associated with eight WTs when used on FAT32 and NTFS file systems. Results indicate that regardless of the wiping algorithm chosen by any of the eight tools, wiping of target test files was shown to be effective. In regards to FAT32, five tools left file system metadata capable of being used to establish the original created date and time of a wiped file and the 'date and time of a file wipe event'. Seven tools demonstrated notable post-wipe DTMs which can be used to identify that wiping had taken place, possibly linking such behaviours to a specific tool. The original file name of a wiped file was retrievable following a file wipe from four tools.

On NTFS, seven tools left file system metadata capable of being used to establish the original created date of a wiped file, and six tools left information describing the 'date and time of a file wipe event'. Seven tools demonstrated notable post-wipe DTMs which can be used to identify that wiping had taken place, possibly linking such behaviours to a specific tool. Seven tools demonstrated notable post-wipe DTMs which can be used to identify that wiping had taken place, possibly linking such behaviours to a specific tool. The original file name of a wiped file was retrievable following a file wipe from all tools following the use of the NTFS $Logfile (and in only one case without the need for the $Logfile). It should be noted that all algorithms available on each of the eight tools tested results in an effective file wipe.

This work aims to provide a starting point for DTM research in DF where further sustained work is required in order to fully identify digital trace evidence which can be used to determine system events, even after typical evidential file content may have since been removed.

References

1. Wright, C., Kleiman, D. and RS, S.S., 2008, December. Overwriting hard drive data: The great wiping controversy. In *International Conference on Information Systems Security* (pp. 243-257). Springer, Berlin, Heidelberg.
2. Hilley, S., 2007. Anti-forensics with a small army of exploits. *digital investigation*, *4*(1), pp.13-15.
3. Kessler, G.C., 2007, March. Anti-forensics and the digital investigator. In *Australian Digital Forensics Conference* (p. 1).
4. Bonte, W., 1975. Tool marks in bones and cartilage. *Journal of Forensic Science*, *20*(2), pp.315-325.
5. Du Pasquier, E., Hebrard, J., Margot, P. and Ineichen, M., 1996. Evaluation and comparison of casting materials in forensic sciences Applications to tool marks and foot/shoe impressions. *Forensic Science International*, *82*(1), pp.33-43.
6. Forte, D. and Power, R., 2007. A tour through the realm of anti-forensics. *Computer Fraud & Security*, *2007*(6), pp.18-20.
7. Geiger, M., 2005, August. Evaluating Commercial Counter-Forensic Tools. In *DFRWS*.
8. Harris, R., 2006. Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem. *Digital investigation*, *3*, pp.44-49.
9. Digital Citizen (2018) '8 best free data erasing apps that permanently delete your files and folders' Available at: https://www.digitalcitizen.life/which-are-best-file-erasers-comparing-5-most-popular-file-shredders (Accessed 26 February 2019)
10. Piriform (2019) 'Our Mission' Available at: https://www.ccleaner.com/about (Accessed 15 January 2019)
11. Carrier, B., 2005. *File system forensic analysis*. Addison-Wesley Professional.
12. LSoft (2018) 'NTFS Journaling' Available at: http://www.ntfs.com/transaction.htm (Accessed 15 January 2019)
13. Microsoft (2008) 'Recovering Data with NTFS' Available at: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/cc976815(v=technet.10) (Accessed 15 January 2019)
14. Schicht, Joakim (2019) 'LogfileParser tool' Available at: https://github.com/jschicht/LogFileParser (Accessed 15 January 2019)

Figures:-

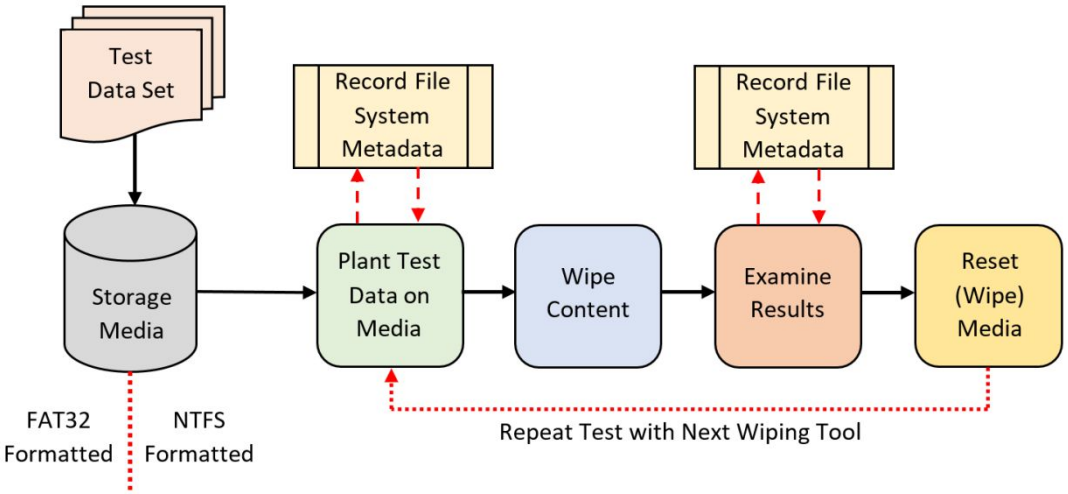


Figure 1: An overview of the methodology deployed in this article.

Name	Forensic_Tester__Doc.docx	Name	ZZZZZZZZZZZZZZZZZZZZ.ZZZZ
File Class	Regular File	File Class	Regular File
File Size	12,304	File Size	13,312
Physical Size	13,312	Physical Size	13,312
Start Cluster	7	Start Cluster	7
Date Created	15/01/2019 10:51:18	Date Created	15/01/2019 10:51:18
Date Modified	14/01/2019 20:40:56	Date Modified	15/01/2019 10:56:08
Actual File	True	Actual File	True
Start Sector	16,522	Start Sector	16,522
Date Accessed	2019-01-15	Date Accessed	2019-01-15
DOS Attributes		**DOS Attributes**	
8.3 Short Filename	FORENS~1.DOC	8.3 Short Filename	!ZZZZZ~1.ZZZ

Figure 2. A comparison of file system metadata for a FAT32 formatted device when CCleaner 1 pass wipe is enabled.

Name	4n6_Content.docx	Name	ZZZZZZZZZZZ.ZZZZ
File Class	Regular File	File Class	Regular File
File Size	12,304	File Size	13,312
Physical Size	13,312	Physical Size	13,312
Start Cluster	5,796	Start Cluster	5,796
Date Accessed	15/01/2019 12:30:16	Date Accessed	15/01/2019 12:30:16
Date Created	15/01/2019 12:30:16	Date Created	15/01/2019 12:30:16
Date Modified	14/01/2019 20:40:55	Date Modified	15/01/2019 12:35:44
Encrypted	False	Encrypted	False
Compressed	False	Compressed	False
Actual File	True	Actual File	True
Start Sector	11,720	Start Sector	11,720
NTFS Information		**NTFS Information**	
MFT Record Number	41 (41984)	MFT Record Number	41 (41984)
Date Changed (MFT)	15/01/2019 12:30:15	Date Changed (MFT)	15/01/2019 12:35:44
Resident	False	Resident	False
Offline	False	Offline	False
Sparse	False	Sparse	False
Temporary	False	Temporary	False
Owner SID	S-1-5-21-1004336348-	Owner SID	S-1-5-21-1004336348-1
Group SID	S-1-5-21-1004336348-	Group SID	S-1-5-21-1004336348-1
Filename Date Created (MFT)	15/01/2019 12:30:16	Filename Date Created (MFT)	15/01/2019 12:30:16
Filename Date Modified (MFT)	15/01/2019 12:30:16	Filename Date Modified (MFT)	15/01/2019 12:35:44
Filename Date Accessed (MFT)	15/01/2019 12:30:16	Filename Date Accessed (MFT)	15/01/2019 12:30:16
Filename Date Changed (MFT)	15/01/2019 12:30:16	Filename Date Changed (MFT)	15/01/2019 12:35:44
Filename File Size (MFT)	0	Filename File Size (MFT)	13,312
Filename Physical Size (MFT)	13,312	Filename Physical Size (MFT)	13,312
INDX Entry Filename	4n6_Content.docx		
INDX Entry File Size	12,304		
INDX Entry Physical Size	13,312		
INDX Entry Date Created	15/01/2019 12:30:16		
INDX Entry Date Modified	14/01/2019 20:40:55		
INDX Entry Date Accessed	15/01/2019 12:30:16		
INDX Entry Date Changed	15/01/2019 12:30:15		

Figure 3: An example of file system metadata changes following a CCleaner wipe.

Name	Size	Type	Date Modified
!WRD1596.TMP	12	Regular F...	15/01/2019 12:14:06
!WRL1608.TMP	13	Regular F...	14/01/2019 20:40:56
!ZZZZZZ.ZZZ	1	Regular F...	15/01/2019 12:19:52

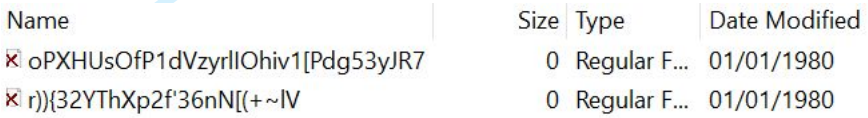
Figure 4: An example of temporary content remaining intact following an initiated wipe.

Name	Size	Type	Date Modified
!936058.ZZZ	0	Regular F...	01/02/2019 10:11:20
!938123.ZZZ	0	Regular F...	01/02/2019 10:11:20
Butterfly-in-front-of-flower.jpg	0	Regular F...	01/02/2019 10:11:20
pexels-photo-248797.jpeg	0	Regular F...	01/02/2019 10:11:20

Figure 5: Simple one pass wipe by File Shredder on FAT32

8242613.ZZZ	0	Regular F...	01/02/2019 11:06:29
8243629.ZZZ	0	Regular F...	01/02/2019 11:06:28
8244561.ZZZ	0	Regular F...	01/02/2019 11:06:28
8244699.ZZZ	0	Regular F...	01/02/2019 11:06:28

Figure 6: Simple one pass wipe by File Shredder on NTFS



Name	Size	Type	Date Modified
oPXHUsOfP1dVzyrllOhiv1[Pdg53yJR7	0	Regular F...	01/01/1980
r)){32YThXp2f'36nN[(+~lV	0	Regular F...	01/01/1980

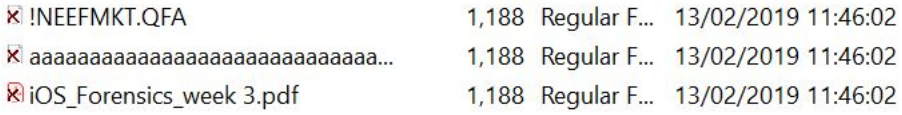
Figure 7: Eraser use on FAT32

Name	Size	Type	Date Modified
Butterfly-in-front-of-flower.jpg	1,763	Regular F...	01/02/2019 20:36:40

Figure 8: Freeraser on FAT32

Name	Size	Type	Date Modified
bxjbxp9rw4YCjcfcLQBsvxjwujß1.xNp	0	Regular File	01/01/1980
jAV6GHJEUUM8YipyRfD.uQiw	0	Regular File	01/01/1980

Figure 9: WipeFile on FAT32



!NEEFMKT.QFA	1,188	Regular F...	13/02/2019 11:46:02
aaaaaaaaaaaaaaaaaaaaaaaaaaaa...	1,188	Regular F...	13/02/2019 11:46:02
iOS_Forensics_week 3.pdf	1,188	Regular F...	13/02/2019 11:46:02

Figure 10: DPWipe on FAT32

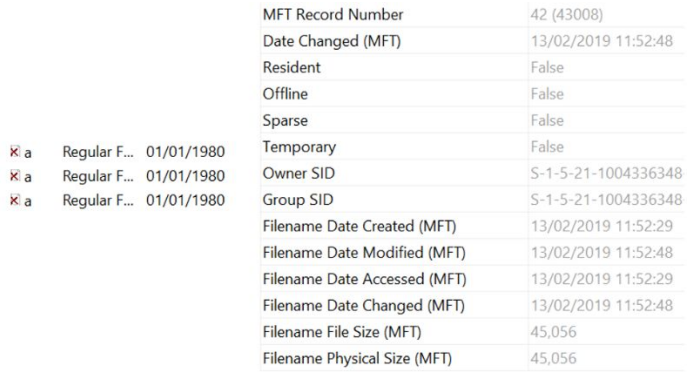


Figure 11: DPWipe on NTFS

Figure 12: Moo0 Shredder on FAT32

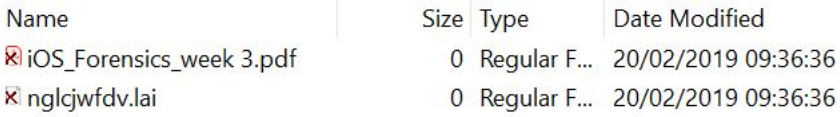


Figure 13: BitKiller on FAT32

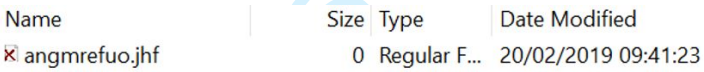


Figure 14: BitKiller on NTFS

Tables:-

Table 1:- List of examined WTs and their download links.

Name	Link
CCleaner (v5.51.6939)	https://www.ccleaner.com/
File Shredder (v2.5)	http://www.fileshredder.org/
Eraser (v6.2.0.2982)	https://eraser.heidi.ie/download/
Freeraser (v1.0.0.23)	https://freeraser.en.uptodown.com/windows/download
WipeFile (v2.4.1.0)	https://www.gaijin.at/en/software/wipefile
DP Wipe (v1.1)	https://www.softpedia.com/get/Security/Security-Related/DP-WIPER.shtml
Moo0 Shredder (v1.21)	https://www.softpedia.com/get/Security/Secure-cleaning/Moo0-FileShredder.shtml
BitKiller (v2.0)	https://sourceforge.net/projects/bitkiller/

Table 2: Wipe summary on FAT32

Tool	File Recoverable*	Time of Wipe	Created Time	File Name	DTMs

CCleaner (v5.51.6939)	No	Yes	Yes	No	Wiped files have names comprising of 'Z' characters. Wiped space is hex character 0x00.
File Shredder (v2.5)	No	Yes	Yes	Yes	Both an original file container and paired wipe container which has a file extension of three 'Z' characters. All files have allocated file size of zero.
Eraser (v6.2.0.2982)	No	No	No	No	Alphanumeric file name, minus a file extension. All available file system time stamps are replaced with the value 01-01-1980 and are of file size zero.
Freeraser (v1.0.0.23)	No	Yes	Yes	Yes	No unique traits.
WipeFile (v2.4.1.0)	No	No	No	No	File names are replaced with an alphanumeric string which in length and extension is equal to the length of original final name. Created time and modified timestamps are replaced with 01/01/1980.
DP Wipe (v1.1)	No	Yes	Yes	Yes	A file's original name is replaced with a single 'a' character. File modified, accessed and created times are set to 01/01/1980. File size metadata is removed. Post wipe, MFT Filename timestamps remain where the created time reflects the created time of the file on the volume and the 'changed' time reflects the time of wipe
Moo0 Shredder (v1.21)	No	No	No	No	Files names are replaced with an alphanumeric randomised string which is the same length in characters as the original file name. Randomly generated modified, accessed and created date and times.
BitKiller (v2.0)	No	Yes	Yes	Yes	The original file metadata remains in place and a second file container of nine characters long (random character naming) which the same time stamp information is created.

Table 3: Wipe summary on NTFS

Tool	File Recoverable*	Time of Wipe	Created Time	File Name	DTMs

CCleaner (v5.51.6939)	No	Yes	Yes	Yes ($Logfile only)	Wiped files have names comprising of 'Z' characters. Wiped space is hex character 0x00.
File Shredder (v2.5)	No	Yes	Yes	Yes ($Logfile only)	Wipe file containers are created with numeric file names seven characters in length with file extension of three 'Z' characters. All files have allocated file size of zero.
Eraser (v6.2.0.2982)	No	Yes	No	Yes ($Logfile only)	Alphanumeric file name, minus a file extension. All available file system time stamps are replaced with the value 01-01-1601 except for the Date Changed (MFT).
Freeraser (v1.0.0.23)	No	Yes	Yes	Yes	No unique traits.
WipeFile (v2.4.1.0)	No	No	No	Yes ($Logfile only)	File modified, accessed and created times are set to 01/01/1980. File size metadata is removed. Post wipe, MFT Filename timestamps remain where the created time reflects the created time of the file on the volume and the 'changed' time reflects the time of wipe.
DP Wipe (v1.1)	No	Yes	Yes	Yes ($Logfile only)	The presence of three file containers which all have identical file system metadata, where one file maintains the original file name, one has a file name of all 'a' characters and the final is a FAT short file named file with a randomised naming convention.
.Moo0 Shredder (v1.21)	No	Yes	Yes	Yes ($Logfile only)	Files names are replaced with an alphanumeric randomised string which is the same length in characters as the original file name. Randomly generated modified, accessed and created date and times. MFT Filename timestamps remain where the created time reflects the created time of the file on the volume and the 'changed' time reflects the time of wipe.
BitKiller (v2.0)	No	Yes	Yes	Yes ($Logfile only)	The original file metadata remains in place and a second file container of nine characters long (random character naming) which the same time stamp

					information is created. MFT Filename timestamps remain where the created time reflects the created time of the file on the volume and the 'changed' time reflects the time of wipe.