Filesystem Timestamps: What Makes Them Tick?

Tony Knutson

Filesystem Timestamps: What Makes Them Tick?

GIAC GCFA Gold Certification

Author: Tony Knutson, ark236@psu.edu

Advisor: Richard Carbone

Accepted: March 23, 2016

Abstract

The purpose of this paper is to delve into how file system timestamps work not only between NTFS, FAT32 and exFAT, but also between Windows Operating Systems. Currently, much disparaging information remains concerning file system analysis. The purpose of this research paper is to assist in putting together the work of the foremost experts in filesystem analysis concerning Created, Modified Changed, File Modified and Access dates and how they work across the spectrum of Microsoft Operating Systems. This information will be gathered from the three main file systems used by Microsoft. The functioning of these timestamps has a direct impact on both the findings and reporting conducted by forensicators in their day-to-day examinations. This paper hopes to serve as a centralized source of information in order to assist others with the necessary knowledge and understanding they need to correctly conduct digital forensic examinations.

1. Background

One of the most critical aspects of a forensic investigation is what and where a file has been. However, this only gives minor details as to the file and its location. So much information is not taken into consideration with basic fact gathering if forensic examiners are merely scratching the surface during a detailed examination, if they are merely looking at the data layer as defined by Lee (2015). Spencer-Thomas (2012) solicits journalists to provide as much detail as possible in their fact gathering, also known at the Five W's, in order to provide a clear picture of what has taken place. This is something that every forensic examiner should be doing with their own report writings as well. However, how this is done with the amount of data now seen in typical forensic examinations is reaching staggering proportions, with multiple devices and file systems being analyzed. As a result, the need to follow the Five W's has never been more needed.

While many digital artifacts exist to prove that a file was opened, the most essential piece of information needed is the file's timestamp information. This is resonated by Chow, Law & Kwan (2007) as they state time stamps shed light on the causes and correlations of the revealed digital data. Moreover, these timestamps house a threshold of information that a forensic examiner can use to conclude not only when a file was created, but when it was also modified and if it was copied from another location prior to the examination. However, the timestamps for files is often an overlooked area of standard investigations where Time of Incident (TOI) is not of much concern. This is largely due to the fact if the file exists then there is no use reinforcing how it got on the device. What has begun to be scene though is strengthening the "how" a file got to the device is becoming more important not only in TOI cases, but also with dead box forensic cases such as criminal or internal investigations.

Tony Knutson, ark236@psu.edu

2. Background

This paper assumes the reader already has a foundational understanding of how NTFS, FAT32 and exFAT file systems function. As a result, there is very little discussion about how the FAT allocation table or NTFS MFT work or the digital remnants that should be considered during a forensic examination or incident response. Standard definitions described in courses such as SANS FOR408 or FOR508 are not fully explained as a result of this knowledge set.

2.1. Disparity of Training

To better understand how timestamps can be used in an investigation, forensicators need to understand the relationship between the Operating System (OS) and the file system and how they handle the timestamps for files located on them. However, since there is such a wide variety of both OS's and file systems there has been much confusion as to what and how they interact with one another. This has been seen within SANS courses, to Carrier (2006) who is largely considered to be the foremost expert on NTFS and FAT fie systems and how they interact with data located on them. As a result of these kinds of disparity among experts in digital forensics, Chow, Law & Kwan (2007) assert many examiners would be reluctant to conclude their analysis of a digital device entirely on timestamps because of the potential risk of being wrong based on their own training.

This paper aims to provide forensicators with this knowledge base and how to better utilize timestamps within their own investigations. As with any report provided to attorneys or management, the ability to have a concise analysis on how particular files of interest not only reside on the device, but also how these files interact with the OS will only give better conclusive evidence pertaining to who may have done it. Moreover, by looking at anti-forensic areas where timestamp modifications have taken place can show files that may have been missed through timeline analysis due to being outside the scope of when an incident occurred. With all of this information, forensicators can then provide a "What" the file is, "Where" the file resides, "Why" it could be residing there, "How" it got on the device and most importantly narrow the "Who" put it there.

Tony Knutson, ark236@psu.edu

2.2. Definitions

Before going in depth for the knowledge and datasets, it is key that specific definitions of timestamp be understood. This is largely needed, as there is some disparity between experts across the digital forensic field. The goal is to hopefully bring a resolution to any discussions within the forensic community as to an agreed upon definition of these terms.

These are the definitions as defined by Carrier (2006):

- *Creation Time (C):* This is the time the file was created.
- *Modified Time (M):* This is the time that the contents attributes are modified.
- *MFT modified Time (B)*: Time that the metadata of the file was last modified (NTFS) and is not showing in Windows under Properties.
- *Accessed Time (A):* Last time that the content of a file is opened.

These are the definitions as defined by Lee (2015):

- *Metadata change Time (C):* Time the MFT Record was last modified.
- *Data Content Change Time (M):* Time Data content of a file was last modified.
- *Metadata Creation Time (B)*: File was created in the volume/directory.
- *Accessed Time (A):* Approximate time file data was last accessed.

This information is certainly more to chew on for the casual examiner and coming from two of the most well known individuals with respect to file system forensics, it is easy to see why there is a divide among those in the DFIR world as to what definitions are followed in their respective examinations. However, these definitions are not that far off from one another and it appears to be more of a play on words than the definitions themselves. For this paper, the following definitions are being used from both Carrier (2006) and Lee (2015), respectively:

- *Creation Time (C):* This is the time the file was created (Carrier, 2006).
- *Modified Time (M):* Time content of a file was last modified (Lee, 2015).
- *MFT modified Time (B)*: Time that the metadata of the file was last modified (NTFS) and is not showing in Windows under Properties. (Carrier, 2015).
- *Accessed Time (A):* Approximate time file data was last accessed (Lee, 2015).

3. FAT32 File System

One of the most universal file systems across all three OS platforms is the File Allocation Table, or FAT. Since FAT32 is now typically only seen with peripheral media (e.g. thumb drives, SD Cards, etc.), examination of these devices may not be seen as in-depth due to the lack of attributes a forensic examiner would use for attribution. Moreover, as Lee (2015) declares, FAT systems require the knowledge of the time zone information related to the device in order to be properly analyzed through timeline analysis. While many examiners can assume this information to be within their own time zone, they cannot testify under oath in court as to the exact time relating to the timestamps of data contained on the device.

3.1. MAC Timestamps

Carrier (2006) details the MAC timestamps for FAT as being a 16-bit value where 7 bits are related to the year, 4 bits for month and 5 bits for the day. Further analysis by Carrier (2006) describes how the year range for FAT32 file systems are between 1980 and 2107 as a result of the 7 bit limit for the year range. While this covers the date and year, the time follows the same concept with a 16-bit value being used for hour, minutes and seconds. Since there is only a finite amount of time in a day, this is an area that is not nearly as trivial as we see with the year portion of the 32-bit value when combined together.

Time Stored	Time Resolution	Date Modified	Date Accessed	Date Change	Birth
UTC	Jan 1, 1970 in local time	Updated	Updated	N/A	Creation

Table 1: FAT32 Modification times (Lee, 2015)

An example of this type of MAC timestamp would be if a user were to create a file on their C:\Users\%USERNAME%\Desktop that is named “myfile.doc.” When the file is created all three times in the *C, M and A* will coincide with one another. However,

since the FAT file system cannot update the *A* time at a "speedy time," as Lee (2015) explains, can take over a day to update. Moreover, since the timestamp will be in local time adjustments must be made when doing analysis to ensure the proper timestamps are even used. For example, if a computer had the file myfile.doc but it was put there by a thumb drive from an unknown computer who is to say that the time zones would match between the computers in order for the file to have the time? This is still a holdover for mainstream forensic suites such as FTK by AccessData that require a time zone field be populated from the registry hive, or best guess by the examiner, in order to process the information. Other tools like SleuthKit and Log2Timeline also can benefit from knowing this information when processing FAT file systems as well.

4. ExFAT File Systems

One of the more recent file systems created in 2006, exFAT is also known as FAT64 (Rusinovich et. al.). The creation of this file system was largely pushed by those in the film industry requesting a file system that could perform continuous recording in a single file that was not restricted to previous FAT file size restrictions. However, it took until 2009 for Shullich (2009) to write about the unknown components of how the file system actually works. This becomes a key area for forensic examiners alike as this file system is beginning to pick up speed with peripheral devices, such as thumb drives, as there is much more support between Apple and Microsoft products. What makes this unique is the inherent issue of moving from multiple files to a single file, there will only be one timestamp associated with potential evidence that could be several hours long.

4.1. MAC Timestamps

Since exFAT is just the 64-bit version of the FAT file system, it still is restricted by the same rules other versions of FAT are held to. According to Shullich (2009), the only exception to these rules is that exFAT provides support to store the timestamps in UTC time as opposed to local time. However, it is different from NTFS where the UTC timestamp is stored as a 64-bit number in 100ns intervals and in epoch time (Shullich, 2009). Additionally, the time resolution is set to Jan 1, 1601, as opposed to what is seen with other renditions of FAT (Lee, 2015). Shullich (2009) also claims the byte after the

timestamps of a file will tell if UTC was supported or not, and if not will have a different number other than zero for an examiner to resolve what time the files timestamps were generated, respectively. However, both Shullich and Lee do not test how exFAT works on Windows 7 and other systems in their examples. This will be conducted in Windows 7 and 8.1, and Shullich's hypothesis on the timestamps from XP will be tested and validated.

Time Stored	Time Resolution	Date Modified	Date Accessed	Date Change	Birth
UTC	Jan 1, 1601	Updated	Updated	N/A	Creation

Table 2: exFAT Modification times (Lee, 2015)

Key examples of exFAT file systems are going to be seen with thumb drives or other peripheral devices. Since NTFS houses much more information pertaining to the operating of the device to keep it in working order, there is little doubt Microsoft or other operating systems would transition to a more primitive allocation table. However, as external media gets larger in size and with many users utilizing cross platform OS's, exFAT is a terrific way to bridge file systems between HFS+ with OS X and NTFS with Windows. Furthermore, since NTFS is not supported with HFS natively exFAT becomes one of the very few options available for users who have files larger than what traditional FAT systems would allow.

5. NTFS File System

NTFS was introduced by Windows in 1993 and has been a part of every Windows oriented digital forensic investigation since Windows XP. What makes this file system so robust and useful for digital forensics is the journaling and documenting it does for all the data that resides on a NTFS-enabled storage device. Additionally, Carrier (2006) details the exact location of the $MFT from Windows XP forward as being located more within the middle of the file system as opposed to previous versions of Windows where it was located directly after the $BOOT location. This is important to note as the

$STANDARD_INFORMATION data structure that houses the metadata related to MACB times will be within the $MFT and could be located and parsed successfully to list information pertaining to a corrupt media device.

5.1. MACB Timestamps

Outside the standard difference between FAT and NTFS that most forensic examiners understand already, there is only one timestamp that NTFS carries which FAT does not: MFT Modified Time. Naturally since FAT file systems do not have a MFT to store metadata in, this is an area that makes NTFS forensics very unique and extremely beneficial to forensicators. Carrier (2006) defines the MFT Modified Time (*B date as defined earlier)* as the "the time that metadata of the file was last modified." Moreover, another key difference to note with examinations of NTFS is the Modified Time is updated as the $DATA and $INDEX attributes being modified. What this means is examiners are seeing two attributes being responsible for the Modified Time as opposed to being strictly updated as the file is changed within FAT.

Time Stored	Time Resolution	Date Modified	Date Accessed	Date Change	Birth
UTC	Jan 1, 1601	Updated	Creation	Updated	Creation

Table 3: NTFS Modification times (Lee, 2015)

5.2. $STANDARD_INFORMATION vs. $FILE_NAME

One very unique issue with NTFS is the location of two separate sets of metadata relating to file(s) timestamp information. Carrier (2006) recommends when looking for timestamp information the best location to analyze is the $STANDARD_INFORMATION attribute as it updates with the timestamps as they occur. A common misconception is that the $FILE_NAME attribute would update just as $STANDARD_INFORMATION does. However, as Carrier (2006), Lee (2015) and Chow et. al. (2007) describe these are merely temporal values and are not updated outside of when the file is created, moved or renamed only. This means $FILE_NAME would be more in correlation with the *B* timestamp as opposed to the *M* timestamp. Despite this

claim by many in the DFIR field, this may not be the case when copying and moving files to other partitions or file systems. During the research into this topic, there was very little published and peer reviewed information as to whether this is correct. Thus, the author will endeavor to substantiate the claims made by the aforementioned authors.

6. Enhancing Forensic Investigations with Timestamps

With a foundational understanding of how both file systems and operating systems function in relation to timestamps, the actual analysis of these timestamps becomes the next step for a forensic examiner to not only understand but also be able to put into context of how they can be affected by users and systems alike. Casey (2010) encourages his readers to understand what is happening behind the scenes of their analysis in order to articulate why the timestamps of files are being seen as he is opposed to guesswork based on the output of specific software tools.

6.1. How can the time be set?

Most traditional methods to creating timestamps are done based on the file system itself. As discussed previously, the FAT file system utilizes the local time from the computer in order to determine the timestamp information for the files. However, as Microsoft (2009) and Hagen (2015) describe, a registry key located in the SYSTEM\CurrentControlSet\Services\W32Time\Parameters and the value NtpServer will display where there OS is calling for the system to sync the system clock with UTC time.

Value name	Value type	Data	Value slack
NtpServer	RegSz	time.windows.com,0x9	00-00
ServiceDll	RegExpandSz	%systemroot%\system32\w32time.dll	
ServiceDllUnloadOnStop	RegDword	1	
ServiceMain	RegSz	SvchostEntry_W32Time	00-00
Type	RegSz	NTP	A8-1B-41-00

Figure 1: Registry Explorer Output of SYSTEM Registry Hive

This is confirmed to be within Windows XP and newer versions of the OS. Moreover, Microsoft (2009) does state the utilizing of the "Net Time" and "/querysntp" will display the NTP server as well from command line on Windows XP and older OS's. Moreover, looking at areas for NTP synchronization can be incredibly useful if files are being transferred between two computers over the network. Knowing how the file times may or be affected by the NTP server is often an overlooked area within digital forensics as most, but certainly not all, computers are utilizing the time.windows.com, 0x9 values to synchronize. As Hagen (2015) states, the NTP should be one of the first places to look before doing any type of forensic investigation as the touching and transferring of files over the network, including the Internet, inherently depend on this clock for timestamps.

6.2. Registry Keys

An overlooked area of difference between Windows XP and later renditions of Microsoft Windows is the Last Access Time (*A*). According to Microsoft (2016) this timestamp is determined to be on or off by the Registry and is located in SYSTEM/CurrentControlSet/Control/FileSystem/NtfsDisableLastAccessUpdate. For Windows XP this value was set to "0" by default, which means the A, value is updated as a folder or file are opened by the user. However, with Windows 7, Microsoft (2014) switched this value to "1" which means that the *A* timestamps are no longer updated on the files and folders when opened. This is a huge area of concern as many examinations in the past had the ability to use these times as a means of determining intent when viewing certain files or folders within a specific date range. However, this is no longer the case and should be an area that is analyzed prior to timestamp or timeline analysis to determine if these values are of evidentiary value. Microsoft (2014) also states this was done in order to provide greater performance within the file system as the number of writes needing to be made to files being accessed are greatly reduced.

Value name	Value t...	Data
DisableDeleteNotification	RegDword	0
NtfsAllowExtendedCharacter8dot3Rename	RegDword	0
NtfsBugcheckOnCorrupt	RegDword	0
NtfsDisable8dot3NameCreation	RegDword	2
NtfsDisableCompression	RegDword	0
NtfsDisableEncryption	RegDword	0
NtfsDisableLastAccessUpdate	RegDword	1
NtfsDisableVolsnapHints	RegDword	2
NtfsEncryptPagingFile	RegDword	0
NtfsMemoryUsage	RegDword	0
NtfsMftZoneReservation	RegDword	0
NtfsQuotaNotifyRate	RegDword	3600
SymlinkLocalToLocalEvaluation	RegDword	1
SymlinkLocalToRemoteEvaluation	RegDword	1
SymlinkRemoteToLocalEvaluation	RegDword	0
SymlinkRemoteToRemoteEvaluation	RegDword	0
UdfsCloseSessionOnEject	RegDword	3
UdfsSoftwareDefectManagement	RegDword	0
Win31FileSystem	RegDword	0
Win95TruncatedExtensions	RegDword	1

Figure 2: System Registry Hive from Windows 7 with default settings

Another key reason to not trust Last Access times within the NTFS is the reliability of the files and what this timestamp may represent. Casey (2006) mentions that many files are "touched" by other programs and files in order to run properly. For example, if a company employs anti-virus software this could change the *A* times to the time the anti-virus software was running in order to see if those specific files were illicit or not. Lee (2015) even states these times should never be trusted regardless if the timestamp is being updated or not because of these events.

6.3. Anti-Forensics

One area of growing concern among forensic examiners is the utilization of anti-forensic techniques being used. Afonin, Nikolaev and Gubanov (2015) define anti-forensics as "a set of precautionary measures a user can perform in order to hide traces of his activity, making investigations on digital media more complicated…potentially rendering evidence of illegal activities difficult or impossible to obtain." Why anti-forensics has gained ground among computer users has been debated in many circles, but many believe it is the accessibility and usability of the software in present day that has made it easier to perform. According to Berinato (2007), this is not because more sophisticated tools are being created, but because anti-forensic tools are sliding down the

learning curve for novice computer users to utilize. In fact, this has been a big business area for many companies who solicit these tools as privacy cleaners and computer performance boosters.

Another tool that has made anti-forensics much easier to conduct has been MetaSploit. Typically known more as a hacker's gateway to weaknesses in operating systems, it also has the ability to time stomp files as needed. According to Offensive Security (2016), this tool is extremely beneficial to attackers who wish to remove their footprints within the file system of the computer they are infiltrating. Moreover, Offensive Security (2016) advocates the use of MetaSploit as it is loaded into memory and does not directly touch the file system of the device they are infiltrating. This makes the framework a very formidable foe for DFIR examiners who may see these tools used against a compromised system. This technique is very similar to that of adjusting the computer's BIOS clock, if available, in order to change the times to files. Both do still leave behind a wealth of artifacts that examiners can analyze to track these date changes.

6.3.1. Privacy Cleaners

In today's society and culture in the United States, hiding one's tracks has become a booming trend to keep certain activities private and away from the watchful eye of law enforcement or employers. This has largely been seen in other traditional forensic avenues such as Google's Incognito Mode or even encryption. However, what would someone need to do in order to hide actual files? One of the more common ways of eliminating evidence from digital forensics has been to fully remove them. According to Afonin, Nikolaev and Gubanov (2015), these tools tend to remove documents and other remnants of file(s) that could be incriminating.

Berinato (2007) cites the example of CCleaner created by Piriform as one method of deleting file remnants and cleaning up registry files to speed up performance, but also to remove artifacts such as web browsing. One unique analyst note to mention about CCleaner is that Lee (2015) asserts the files are deleted by overwriting the file as opposed to a secure wipe of zeros. This typically is seen with a unique pattern of the letter "Z" being used to remove the filename and extension. While this is not substantial to timestamp analysis, there may be chances the timestamp would remain, but this has not

been tested through peer-reviewed research or in this paper. However, another technique to show timestamp manipulation using CCleaner, or many other privacy cleaners, is discussed in the USN Journal analysis.

6.3.2. Timestomper

Time stomping is a very unique method of changing the metadata of a file after its creation or modification. Berinato (2007) cites Liu, who uses timestomping during testing and training of personnel as a prime indicator as to why timeline analysis may not work as a file can be manipulated in such a way where an examiner may miss it because of the creation or modification date being changed to another date. This means that when performing a strict timeline analysis through a pivot point as taught by Lee (2015) in the SANS FOR508 course, may not be sufficient, depending on the complexity of the case. However, Lee (2015) does indicate through SANS FOR508 how to determine if timestomping has been utilized on files when conducting either incident response or dead box forensics. This is largely due to laziness on the individual executing timestomping not changing the time information which would display at 00:00:00 UTC within any given area of the MACB instead of a more relevant timestamp that could be glazed over during analysis. This continues to be an area of concern during incident response where files that were touched by intruders are changed and files of evidentiary value need to be closely analyzed.

6.4. NTFS MFT and USN Journal Analysis

Keen and savvy ways to defeat anti-forensics in the past was to try to catch individuals in the act of deleting the files of importance. But there is another way forensic examiners can show when activities like privacy cleaners and timestomping have taken place on a drive or network. In the case of privacy cleaners, parsing the MFT and USN Journal file can provide a wealth of information to what privacy cleaners, such as CCleaner, are doing to the files. Using tools such as Obsidian Forensics' (2015) USN Journal Visualizer, forensicators can see what exactly is happening to these files at the nanosecond. This means even if an individual were employing this method of hiding their tracks, timestamp information on NTFS volumes will still prove extremely beneficial to

the case as finding when a file was created, modified or deleted will show in this valuable area.

In the case of Timestomper, Cho (2013) provides examples of using tools such as using forensic suites to see what is taking place with files and looking for abnormalities. This is largely done by seeing a file timestamp that are greatly exaggerated or have other indicators such as a 00:00:00 time. The odds of any file having the exact midnight hour with no seconds would be extremely suspect and would require further analysis for other artifacts.

6.5. What if we used FAT32 or exFAT to NTFS or vice versa?

An area that can be forgotten while doing an investigation is where the file(s) resided before the device was being examined. What if the file was moved from a Windows 7 machine to a thumb drive that was formatted FAT32? Or exFAT? What if the file was moved from that thumb drive to a Windows 7 computer? These questions should always be considered to ensure the timestamp is being properly documented. The biggest area of consideration here is an examiner cannot look at files as just being on a peripheral device or not: it must be looked at closely to see if it was copied or moved.

According to Lee (2015), if we copy a file from a FAT system to a NTFS system it will keep the same modified date but will change the create date and time to the current time. Where Lee (2015) notes if the file is cut and pasted it will keep the same modified date and time and will keep the creation date the same as before.

7. Observations and Findings

Much has been done in the field of DFIR to demonstrate and prove timestamp analysis to be a worthy use of time and combined with other forensic techniques can assist an examiner in corroborating when a certain incident has taken place. However, one area that has not been analyzed in great detail is what happens with these files as they are moved and copied to and from different operating systems. According to Statcounter (2016), 55% of all users globally are using Windows 7 and shockingly 17% were using Windows XP.

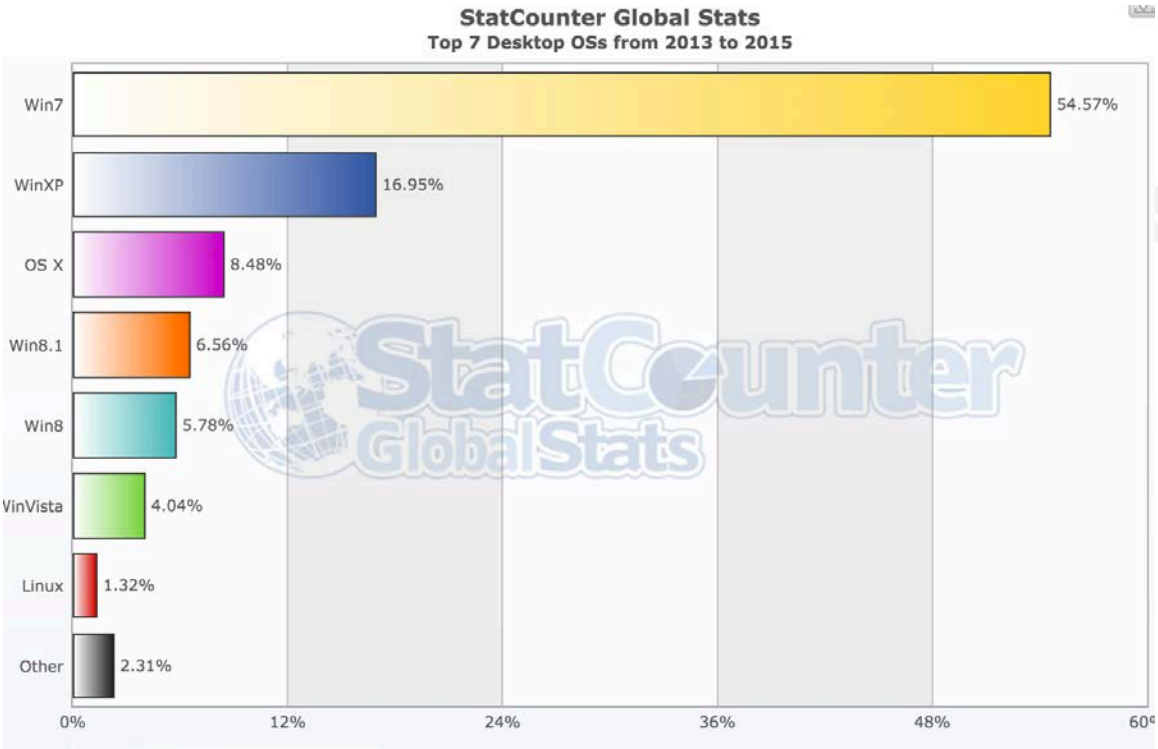

Figure 3: Operating System Usage (Statcounter, 2016)

This is important for examiner's to recognize and note while doing examinations as this graph does show that a very wide range of Windows OS's are still in use and could be seen in their own environment. Moreover, since Windows 7 is still one of the most used operating systems, it will be the center of the testing along with Windows XP and Windows 8. Windows Vista was not considered as it has very low consumer usage and is relatively close to Windows 7. Additionally, Windows 10 was not looked at as of this writing it is still very new and has not seen a dramatic uptick in consumer usage to warrant analysis at this time. Furthermore, the findings of these analyses aim to substantiate and expand upon the works of Chow et. al. (2007) as his group has already laid the groundwork of timestamp analysis on file systems.

7.1. Observations 1 – XP, Win7, Win 8 Standalone

The following analyses were conducted with the default settings of a newly installed Windows OS to assist in observing how file timestamps work on a standalone computer. The MACB attributes among all three prominent Windows OS's concluded the following information. File timestamps will be gathered using FTK Imager version

3.1.1.8 and the modifications to the files will be conducted on the cut-and-paste version of the file. This is important to note as the MFT sequence number would be the same between Creation and the Cut-and-Paste versions of the file.

7.1.1. File Creation

Creation of Test.txt on all three boxes yielded this information pertaining to *C, M, A, and B* dates respectively. This area serves to produce the expected results of a file being created on a Windows OS with no modifications to the entries. All times are in UTC time as they are being created on NTFS volumes.

Filename	**Windows Version**	***C Date***	***M Date***	***A Date***	***B Date***
Test.txt	Win XP	3/15/2016 12:01:22	3/15/2016 12:01:30	3/15/2016 12:01:30	3/15/2016 12:01:30
Test.txt	Win 7	3/14/2016 23:08:05	3/14/2016 23:08:12	3/14/2016 23:08:05	3/14/2016 23:08:12
Test.txt	Win 8.1	3/14/2016 23:12:50	3/14/2016 23:12:50	3/14/2016 23:12:50	3/14/2016 23:12:51

Table 4: File Creation on standard systems

7.1.2. Copy and Paste (i.e., copying the file)

Files were created on the C:\Users\%USERNAME%\Desktop. From there they were right-clicked, copied and pasted in the C:\Users\%USERNAME%\MyDocuments folder. The following information was recorded.

Filename	**Windows Version**	***C Date***	***M Date***	***A Date***	***B Date***
Test.txt	Win XP	3/16/2016 13:19:39	3/15/2016 12:01:30	3/16/2016 13:019:39	3/15/2016 12:01:30
Test.txt	Win 7	3/16/2016 13:34:31	3/14/2016 23:08:12	3/14/2016 23:08:05	3/16/2016 13:34:31
Test.txt	Win 8.1	3/16/2016 13:47:27	3/14/2016 23:12:50	3/16/2016 13:47:27	3/14/2016 23:12:51

Table 5: Copying of files on standalone machine

7.1.3. Cut and Paste (i.e., moving the file)

Files were created on the C:\Users\%USERNAME%\Desktop. They were then right-clicked, cut and pasted to C:\Users\%USERNAME%\MyDocuments\EXAMPLE folder. The following information was recorded.

Filename	**Windows Version**	***C Date***	***M Date***	***A Date***	***B Date***
Test.txt	Win XP	3/15/2016 12:01:22	3/15/2016 12:01:30	3/16/2016 13:19:26	3/16/2016 13:25:17
Test.txt	Win 7	3/14/2016 23:08:05	3/14/2016 23:08:12	3/14/2016 23:08:05	3/16/2016 13:39:57
Test.txt	Win 8.1	3/14/2016 23:12:50	3/14/2016 23:12:50	3/14/2016 23:12:50	3/16/2016 13:47:47

Table 6: Moving of file on standalone machine

7.1.4. Modifications

Files located in the EXAMPLE folder that were moved were modified by adding text to the document to invoke a change in modification to the files. The following was recorded.

Filename	**Windows Version**	***C Date***	***M Date***	***A Date***	***B Date***
Test.txt	Win XP	3/15/2016 12:01:22	3/16/2016 13:29:00	3/16/2016 13:29:00	3/16/2016 13:29:00
Test.txt	Win 7	3/14/2016 23:08:05	3/16/2016 13:42:37	3/14/2016 23:08:05	3/16/2016 13:42:37
Test.txt	Win 8.1	3/14/2016 23:12:50	3/16/2016 13:51:41	3/14/2016 23:12:50	3/16/2016 13:51:41

Table 7: Modification to file on standalone machine

7.1.5. Findings

All information recorded matches exactly with what has been noted by Carrier (2006), Lee (2015), Chow (2007) and Bang et. al. (2011). This also includes Microsoft (2009) and Lee's (2015) assessment of timing difference between updates to *B* and *A* attributes, respectively. The most interesting finding from this information is the MFT Entry Modified time, or *B* attribute. When the files were copied, the *B* attribute would remain the same, but the MFT entry did change. However, when the file was moved the *B* attribute was modified for show the new time. Moreover, when the file was modified, the time changed to the *M* attribute. This correlation should be noted for DFIR experts to ensure they are looking at the *B* attribute along with *C* and *M* attributes. As expected, the *A* attribute was only updated within Windows XP and showed no modification for the *C* attribute in Windows 7 and 8.1.

7.2. Observations 2 – Cross OS Win 8 to Win 7 via exFAT

The following analyses were conducted with default settings of a newly installed Windows OS to assist in observing how file timestamps work on computers as files were

moved from older to newer versions to a newer version via an exFAT formatted USB thumb drive. Windows 7 was selected as the initial creation of the file as it is one of the most common operating systems used today in both personal and business worlds.

7.2.1. File Creation

For this observation, the testing.txt file was created on a Windows 7 machines in the C:\Users\%USERNAME%\Desktop location. One sentence of text was added to the file prior to initial saving.

Filename	**Windows Version**	***C Date***	***M Date***	***A Date***	***B Date***
Testing.txt	Win 7	3/19/2016 15:06:31	3/19/2016 15:06:43	3/19/2016 15:06:31	3/19/2016 15:06:43

Table 8: Creation of file in Windows 7 environment

7.2.2. exFAT properties (copied)

The file was copied over to an exFAT formatted 4GB thumb drive in order to generate a new creation time for the file but maintain the Modification timestamp. There were no other files located on this thumb drive at the time of the copying of the Testing.txt file.

Filename	**Windows Version**	***C Date***	***M Date***	***A Date***	***B Date***
Testing.txt	exFAT	3/19/2016 10:15:09	3/19/2016 10:06:44	3/19/2016 20:17:10	N/A

Table 9: Copied file to exFAT thumb drive from NTFS windows machine

7.2.3. Copy and Paste (i.e., copying the file)

The thumb drive was then removed from the Windows 7 operating system and plugged into a Windows 8.1 virtual machine used in the other observations of this research. It was then copied and pasted to C:\Users\%USERNAME%\Desktop.

Filename	**Windows Version**	***C Date***	***M Date***	***A Date***	***B Date***
Testing.txt	Win 8.1	3/19/2016 16:06:45	3/19/2016 16:06:44	3/19/2016 16:06:45	3/19/2016 16:06:45

Table 10: Copying of file from exFAT thumb drive to Windows 8.1 machine

7.2.4. Cut and Paste (i.e., moving the file)

Once the times were recorded in previous observations, the Testing.txt file was then cut and pasted from the thumb drive to the Windows 8.1 in the User's "My Video" location in order to preserve the original Testing.txt that was copied to the Desktop.

Filename	**Windows Version**	***C Date***	***M Date***	***A Date***	***B Date***
Test.txt	Win 8.1	3/19/2016 15:15:09	3/19/2016 15:06:44	3/19/2016 16:08:34	3/19/2016 16:08:34

Table 11: Movement of file from exFAT thumb drive to Windows 8.1 machine

7.2.5. Findings

Quite easily the most interesting findings have been with exFAT and NTFS and their relationship with one another. Since exFAT supports UTC as Lee (2015) and Carrier (2006) suggest, the times should not be this different. The operating systems' clocks were not altered in any way to make these changes happen. However, going from NTFS to exFAT (which supports UTC) shows the files changed in file time by 5 hours. Consequently at the time of this analysis Central Standard Time is -5 hours from UTC. Looking at the file offset, it was noted to be in Central Standard Time as opposed to UTC. Moreover, the *A* attribute is all messed up and reported back a false positive for access time of over 20:00 hours. This is yet another reason to not trust Access Times when doing forensic analysis as the files may present issues that would require further analysis to determine their true times. Moreover, this file was not opened in any way on

the exFAT thumb drive and was merely copied from the drive to the OS's desktop and later moved to the OS's desktop.

7.3. Observations 4 – Win 7 to NTFS and to FAT32 Partitions

The following analysis was conducted with default settings of a newly installed Windows OS to assist in observing how file timestamps work on computers as files were moved from Windows 7 to a newly created partition formatted in NTFS and FAT32 respectively.

7.3.1. File Creation

For this observation, the Test.txt file was created on a Windows 7 machines in the C:\Users\%USERNAME%\Desktop location. This was the same file as seen with Observation 1 with all times recorded as seen in FTK Imager prior to advancing.

Filename	**Windows Version**	***C Date***	***M Date***	***A Date***	***B Date***
Test.txt	Win 7	3/14/2016 23:08:05	3/16/2016 13:42:37	3/14/2016 23:08:05	3/16/2016 13:42:37

Table 12: Creation of file on Windows 7 machine for analysis

7.3.2. Copy and Paste (i.e., copying the file)

The Test.txt file was then copied from the NTFS volume it was created on and placed on the FAT32 partition that was created. No other files were located in this partition. Once times were recorded, the original file located on the NTFS volume was again selected for copying and pasted into the newly created NTFS volume. No other files resided on the NTFS volume.

Filename	Windows Version	*C Date*	*M Date*	*A Date*	*B Date*
Test.txt	FAT32	3/16/2016 11:54:38 CST	3/16/16 8:42:38 CST	3/16/2016	N/A
Test.txt	NTFS	3/16/2016 16:54:43	3/16/2016 13:42:37	3/16/2016 16:54:43	3/16/2016 16:54:43

Table 13: Copying of file to FAT32 and NTFS created partitions

7.3.3. Cut and Paste (i.e., moving the file)

The original file Test.txt located on both the FAT32 and NTFS volumes was deleted prior to the start of moving the original file. The virtual machine was reverted back to its snapshot in order to bring the file back to its original state once the file had been moved and times were recorded.

Filename	Windows Version	*C Date*	*M Date*	*A Date*	*B Date*
Test.txt	FAT32	3/14/2016 18:08:05 CST	3/16/2016 08:42:38 CST	3/16/2016	N/A
Test.txt	NTFS	3/14/2016 23:08:05	3/16/2016 13:42:37	3/16/2016 16:59:20	3/16/2016 16:59:20

Table 14: Moving of file to FAT32 and NTFS partitions

7.3.4. Findings

Interestingly enough, the *C and M* attributes resolved as expected according to Carrier (2006) and Lee (2015). The most peculiar timestamps came from both the copying and moving of the test.txt to FAT32 where the *A* attribute had no resolution to the timestamp outside of the current date. When the files were opened, however, they did update to the most current local time. Another interesting result was the *A* attribute which

when the file was moved to NTFS resolved to the same time as *B*, which is the MFT modified timestamp. This means the *A* attribute will take the time from the MFT as opposed to the creation timestamp as seen when the file is simply moved from one partition to another with the same file system. This is an interesting finding as no peer reviewed papers closely looked at the *B* value and instead merely associated their research to the MAC timestamps as opposed to what the MFT was doing while updating.

Additional analysis using AnalyzeMFT.py to parse the MFT for both $STANDARD_INFORMATION and $FILE_NAME provided similar results in relation to the file's creation date when copied and moved. Further analysis may be need in order to provide a more accurate finding in relation to the $FILE_NAME attribute and whether it will always show this trait.

7.4. Observations 5 -Timestomping

For timestomping, the Windows 7virtual machine was reverted back to its snapshot in order to use the Test.txt file located in the C:\Users\%USERNAME%\Desktop location.

7.4.1. Creation

The following file was used in order to observe the timestamps of a file being timestomped using Timestomper-GUI.exe. The following information about the Test.txt file was recorded prior to the timestomping tool being used.

Filename	**Windows Version**	***C Date***	***M Date***	***A Date***	***B Date***
Test.txt	Win 7	3/14/2016 23:08:05	3/16/2016 13:42:37	3/14/2016 23:08:05	3/16/2016 13:42:37

Table 15: Creation of file to be analyzed on Windows 7 machine

7.4.2. Modifications

Once the original times were noted, the timestomping executable was used in order to change the attributes of the file. These included all attributes as would be seen in

typical intrusion cases. Random times were used along with random dates to make it appear as if the file could blend in with other files located on the device.

Filename	**Windows Version**	***C Date***	***M Date***	***A Date***	***B Date***
Test.txt	Win 7	3/19/2017 15:06:20	3/19/2017 15:06:20	3/19/2017 15:06:20	3/19/2017 15:06:43

Table 16: Implementation of timestomping to artificially change timestamps

7.4.3. Findings

Using the timestomper.exe program to change all the times of the Test.txt file, there was conclusive evidence that a forensic examiner would not be able to tell if the file was changed at the MAC level as opposed to looking at the *B* attribute. Moreover, as Lee (2015) cites, since tools like timestomper merely alter the $STANDARD_INFORMATION attribute the $FILE_NAME attribute will still hold the original dates of creation for the file. While this may not be of importance in many IR cases, in forensic examinations where timeline analysis is needed in order to show when certain files were created, this becomes a crucial area to analyze if timestomping is suspected.

8. Conclusion

Based on the research from many of the foremost experts in the field of file system analysis and the author's own findings, the area of timestamp analysis should never be overlooked in a digital forensic examination. What was discovered is how important NTFS timestamps are regardless of the type of examination as there is a wealth of metadata available for the examiner to review that could indicate what happened with a particular file at those given times. Having the capability of knowing when the file was created and what else was created around it or during the modification time or MFT content change time could open an entirely new window that would have been missed through timeline analysis or more traditional forensic examinations. Moreover, with other

areas to examine, such as the USN Journal, examiners can deep dive into the weeds of the file system and OS to see what exactly was going on during the timeframe of a particular file of interest. While this is labor inductive, this may be the primary area of interest in order to find the smoking gun to solve the case.

As forensic examiners, there is no shortage of techniques to prove that something occurred and when it occurred. However, being able to prove the Why, How and most importantly, When a specific file(s) was created or used goes further to prove who was behind the keyboard during the time of the incident than merely finding the file(s) and determining that the case is solved. Utilization of proper timestamp analysis can be used to solidify the capability and knowledge of the examiner during testimony in court or to their organization.

Tony Knutson, ark236@psu.edu

9. References

Afonin, O., Nikolaev, D., & Gubanov, Y. (2015). Countering anti-forensic efforts – part 1. *DFI News*.

Bang, J. Byeongyeong, Y. & Sangjin, L. (2011). *Analysis of changes in file attributes with file manipulation*. Science Direct: Digital Investigation (7), pp: 125-144.

Berinato, S. (2007). How Online Criminals Make Themselves Tough to Find, Near Impossible to Nab. *CIO.* Retrieved from: http://www.cio.com/article/2438867/intrusion/how-online-criminals-make-themselves-tough-to-find--near-impossible-to-nab.html

Carrier, B. (2006). *File System Forensic Analysis.* Boston, Mass.: Addison-Wesley.

Casey, E. (2010). *Handbook of Digital Forensics and Investigation*. London: Academic.

Cho, G. (2013). *A Computer forensic method for detecting timestamp forgery in NTFS.* Science Direct: Computers & Security (34), pp: 36-46

Chow, F., Law, M,. & Kwan, P. (2007). The Rules of Time on NTFS File System. IEEE.

Hagen, P. (2015). *Advanced Network Forensics* (SANS FOR572). Bethesda, MD: SANS Institute.

Marcella, A. & Guillossou, F. (2012). *Cyber Forensics: From Data to Digital Evidence*. John Wiley & Sons.

Lee, R. (2015). *Advanced Digital Forensics and Incident Response* (SANS FOR508). Bethesda, MD: SANS Institute.

LSoft Technologies (2016). exFAT Timestamp Format. LSoft Technologies. Retrieved from: http://ntfs.com/exfat-time-stamp.htm

Microsoft (2016). NtfsDisableLastAccessUpdate. Microsoft Company. Retrieved from: https://technet.microsoft.com/en-us/library/cc959914.aspx

Microsoft (2014). Disable Last Access Time Stamps (Standard 7 SP1). Microsoft Company. Retrieved from: https://msdn.microsoft.com/en-us/library/ff794679(v=winembedded.60).aspx

Tony Knutson, ark236@psu.edu

Microsoft (2009). Windows Time Service Tools and Settings. Microsoft Company. Retrieved from: https://technet.microsoft.com/en-us/library/cc773263(v=ws.10).aspx

Obsidian Forensics (2015). Visualizing USN Journal Activity. Obsidian Forensics. Retrieved from: http://www.obsidianforensics.com/blog/visualizing-usn-journal-Activity

Offensive Security (2016). TimeStomp. Offensive Security. Retrieved from: https://www.offensive-security.com/metasploit-unleashed/timestomp/

Russinovich, M.E., Solomon, D.A., & Ionescu, A. (2012). *Windows Internals*. WA: Microsoft Press.

Shullich, R. (2009). *Reverse Engineering the Microsoft Extended FAT File System (exFAT)*. SANS Institute. Retrieved February 29, 2016 from: https://www.sans.org/readingroom/whitepapers/forensics/reverse-engineering-microsoft-exfat-file-system-33274

Spencer-Thomas. (2012). Press Release: Getting the facts straight." Retrieved from owenspener –thomas.com

Statcounter (2016). Top 7 Desktop OS's from 2013 to 2015. Retrieved from: http://gs.statcounter.com/#desktop-os-ww-yearly-2013-2015-bar

Willassen, S. (2008). Timestamp evidence correlation by model based clock hypothesis testing. *Department of Telematics, Norwegian University of Science and Technology*.

Tony Knutson, ark236@psu.edu

Last Updated: July 2nd, 2020

Upcoming SANS Training

Click here to view a list of all SANS Courses

SANS Amsterdam August 2020 Part 1	**Amsterdam, NL**	**Aug 03, 2020 - Aug 08, 2020**	**Live Event**
SANS Reboot - NOVA 2020	**Arlington, VAUS**	**Aug 10, 2020 - Aug 15, 2020**	**Live Event**
SANS FOR508 Canberra August 2020	**Canberra, AU**	**Aug 17, 2020 - Aug 22, 2020**	**Live Event**
SANS Amsterdam August 2020 Part 2	**Amsterdam, NL**	**Aug 17, 2020 - Aug 22, 2020**	**Live Event**
SANS Virginia Beach 2020	**Virginia Beach, VAUS**	**Aug 30, 2020 - Sep 04, 2020**	**Live Event**
SANS Philippines 2020	**Manila, PH**	**Sep 07, 2020 - Sep 19, 2020**	**Live Event**
SANS London September 2020	**London, GB**	**Sep 07, 2020 - Sep 12, 2020**	**Live Event**
SANS Baltimore Fall 2020	**Baltimore, MDUS**	**Sep 08, 2020 - Sep 13, 2020**	**Live Event**
SANS Munich September 2020	**Munich, DE**	**Sep 14, 2020 - Sep 19, 2020**	**Live Event**
SANS Network Security 2020	**Las Vegas, NVUS**	**Sep 20, 2020 - Sep 25, 2020**	**Live Event**
SANS Northern VA - Reston Fall 2020	**Reston, VAUS**	**Sep 28, 2020 - Oct 03, 2020**	**Live Event**
SANS San Antonio Fall 2020	**San Antonio, TXUS**	**Sep 28, 2020 - Oct 03, 2020**	**Live Event**
SANS OnDemand	**OnlineUS**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s OnlyUS**	**Anytime**	**Self Paced**