I. J. Computer Network and Information Security, **2021, 4, 62-69**
Published Online August 2021 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2021.04.06

Detection of Suspicious Timestamps in NTFS using Volume Shadow Copies

ALJI Mohamed
Engineering Science Laboratory, National School for Applied Sciences, Ibn Tofail University, Kenitra, Morocco
E-mail: mohamed.alji@uit.ac.ma

CHOUGDALI Khalid
Engineering Science Laboratory, National School for Applied Sciences, Ibn Tofail University, Kenitra, Morocco
E-mail: khalid.chougdali@uit.ac.ma

Received: 28 February 2021; Revised: 17 April 2021; Accepted: 23 May 2021; Published: 08 August 2021

Abstract: When a computer gets involved in a crime, it is the mission of the digital forensic experts to extract the left binary artifacts on that device. Among those artifacts, there may be some volume shadow copy files left on the Windows operating system. Those files are snapshots of the volume recorded by the system in case of a needed restore to a specific past date. Before this study, we did not know if the valuable forensic information hold within those snapshot files can be exploited to locate suspicious timestamps in an NTFS formatted partition. This study provides the reader with an inter-snapshot time analysis for detecting file system timestamp manipulation. In other words, we will leverage the presence of the time information within multiples volume shadow copies to detect any suspicious tampering of the file system timestamps. A detection algorithm of the suspicious timestamps is contributed. Its main role is to assist the digital investigator to spot the manipulation if it has occurred. In addition, a virtual environment has been set up to validate the use of the proposed algorithm for the detection.

Index Terms: Timestamp Manipulation, MACB Tampering, Time Forgery, Inter-snapshot Analysis, Volume Shadow Copy.

1. Introduction

When a crime involving a digital device happens, the police department requires that the digital forensic expert team take part in the investigation by extracting and interpreting the artifacts left on the device. Thus, the digital forensic team ensures providing the help needed in solving the crime through digital evidence. Other crimes, such as those classified as cybercrimes, require that the digital forensic and the incident response teams take the lead in the digital investigation to interpret the digital evidences. The latter can be in different shapes including a recovered file fragments from browser history, email or application document [1] and can be retrieved from unusual components such as images, cookies, log files, time and frequency of user visiting the page [2]. In both cases, leading or helping to solve the crime, the digital investigator emits an expert opinion and hypotheses to answer the Five-W questions (Who, What, When, Where, Why) that may identify the perpetrator and link him/her to the victim in a digital manner using Digital Forensic tools [3]. When the crime happened is of crucial importance since the most relevant digital artifacts are gathered around that specific time, for instance, the "most-recently-used" (MRU) artifact can shed light on the applications and the files used in last by the suspect.

The time information has forensic value for a digital investigation. It can be extracted from different artifacts in the digital source, for instance, the date and time stored within the application log files, or those stored in the file system special files, etc. In the context of this research, the file system timestamps are of interest. They are vital attributes for the reconstruction of the suspect activity on the digital device. However, a malicious user may tamper with the file system time information using tools such as "Timestomp" or "SetMace". Knowing this, the way the digital examiner apprehends the file system timestamps has a direct impact on the investigation [4] and may lead or mislead the investigation.

Detecting the manipulation of the time information in a digital evidence is a challenging task. We have categorized at least six methods to help spot the suspicious timestamps in an NTFS formatted partition as show in Fig. 1. Each method has its own specificities and rely on a specific artifact for the detection, for instance: checking the truncated micro-seconds relies on the time metadata associated with a suspicious file. On the other hand, detecting the use of an anti-forensic tool relies on the Windows Prefetch files. Thus, making it hard to compare the methods with a common ground-truth.

The major focus is on the following research questions. Can we detect the file system time forgery somehow differently from the usual known methods ? for instance, by using volume shadow copy files (VSCs) ? Can we suggest

some time rules that govern the normal user behavior versus the malicious one of a tampered file ? In sum, the present study contribution falls within the following aspects: detecting the tampering of timestamps in the NTFS file system using a new method involving available volume shadow copies. Furthermore, we designed an experiment to answer those research questions.

In this paper, new time rules are presented allowing the detection of timestamp forgery using volume shadow copies. The paper is organized as follows. Section I is the Introduction to the subject. Section II presents the involved concepts. Section III summarizes the related works. Section IV describes the proposed methodology. Section V comprises the experimental results and proposes a new detection algorithm. Section VI discusses the conclusions and the limitations of the approach.

2. Background and involved concepts

The research objectives involve the detection of user tampered file system timestamps using volume shadow copies. In this section, we will introduce the NTFS file system and we will present how it stores the timestamps within the MFT file. From there, we will talk in general words about the volume shadow copies and the anti-forensic technique of timestamp tampering.

2.1. File system timestamps

The New Technology File System (NTFS) is a proprietary file system designed by Microsoft. It is the default file system for the most popular operating system Microsoft Windows NT+ and Windows Server. It provides the user with a mechanism to store data hierarchically and the system with an accessible structure of the user data. The NTFS relays on multiple system files. The most important one is the Master File Table (named $MFT on the hard disk). The MFT file is a metadata file that contains information about all files and directories in the current NTFS-formatted partition.

The MFT file contains multiple entries representing files on the system. Each entry comes with a different set of attributes such as STANDARD_INFORMATION ($SIA) and FILENAME ($FNA) attributes [5,6]. The attributes are nothing more than data structures that hold details such as filename, ownership, date and time related information, and sometimes the file's content (resident files) [7]. Fig. 1, illustrates an example of the structure of a single entry within the MFT file.

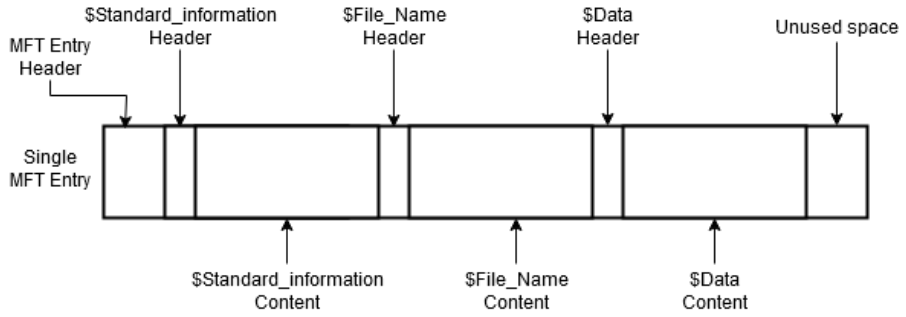

Fig.1. An example of one MFT entry with three attributes: $STANDARD_INFORMATION, $FILENAME and $DATA.

The timestamps of files on the system are stored within $SIA and $FNA attributes. Each attribute holds 04 timestamps abbreviated as "MACB". "MACB" times stands for: "last Modified", "last Accessed", "last MFT entry Changed", and "Birth" times. Those timestamps are the temporal information of interest in this study. They come in a format of a 64-bit value that represents the number of 100-nanoseconds (0.1 microseconds) intervals that have elapsed since 12:00 A.M. of January 1, 1601, Coordinated Universal Time [8]. It seems to be a high-precision time resolution until the point that this research [9] explored the applicability of embedding information on the low-precision part of the file timestamp and using it as a steganographic channel.

2.2. Volume shadow copies and timestamp manipulation

When the Windows XP release comes, a restore system point was possible each time an application is installed or a system is updated. It allows the back-up and the restoration of the operating system files and not the user files. Then comes the volume shadow service (VSS) which allows a system administrator to manually backup (for instance, through a scheduled backup) the user files and the system-critical files to revert the system to its previous state in case of malware infection. With Microsoft Windows 7 and Windows Vista, the VSS can create and maintain snapshots (volume shadow copies) of the entire main volume [10]. The VSS works at the block level (16 kb on Windows 7). Whenever a block changes, the block is copied to the volume shadow copy file creating a differential backup [11]. Throughout this study, we will be mainly focusing on the timestamps available within the volume shadow copy files on the Windows 10 operating system.

According to the research study [12], timestamp manipulation (or timestomping) is an anti-forensic method, that consists of modifying timestamp values to make a deceptively false representation of files. It is an intentional act of falsifying timestamps associated with content [5]. The aim of timestomping differs. It may be to make content appear to have been edited, accessed, modified, copied, etc, at a time that is not the real one, thus, serving the tamperer purposes to fabricate an alibi. It may also be to evade detection. Malware may use a common date and time to alter the integrity and the validity of a forensic timeline. In an attempt to hide its digital footprints, an intruder may also set the file system timestamps of some malicious files to some old date and time, to break away from the time interval investigated by the examiner. For illustration purposes, the Win32/USBStealer, a malware used by the APT28 hacking group, sets the timestamps of its dropper files to the "last-access" and "last-write" timestamps of a standard Windows library chosen on the system [13]. Another aim of timestomping may be to hide information in plain sight. NTFS Timestamps may be used as a mean, or as a steganographic channel, to hide information in the sub-second timestamp granularity. The feasibility of such a steganographic system was proposed in the research study [9].

3. Related Work

Detecting the use of timestamps manipulators is like finding a needle in a haystack. Small details and some left artifacts may expose such anti-forensic technique. Fig. 2, summarizes the existing detection techniques of timestamps manipulation found by reviewing the literature. According to Arman Gungor [14], some timestamp manipulation tools truncate the micro-seconds value with zeros. The truncated 7 digits of the low-precision part of the file time transform the precision of an NTFS timestamp to seconds precision only. From a digital forensic perspective, the 7 digits beyond the seconds provide 10 million possibilities, and the probability to have a randomly set timestamp ending with .0000000 is extremely unlikely. Thus, a microsecond set to zeros could be considered an indicator of a date and time forgery. Such observation is relevant in the time forgery and the proposed algorithm in this study can be extended to incorporate the possibility of the truncated micro-seconds check. Nevertheless, the full precision of NTFS timestamps could be lowered by copying a file from a file system with less precision, such as the FAT file system.

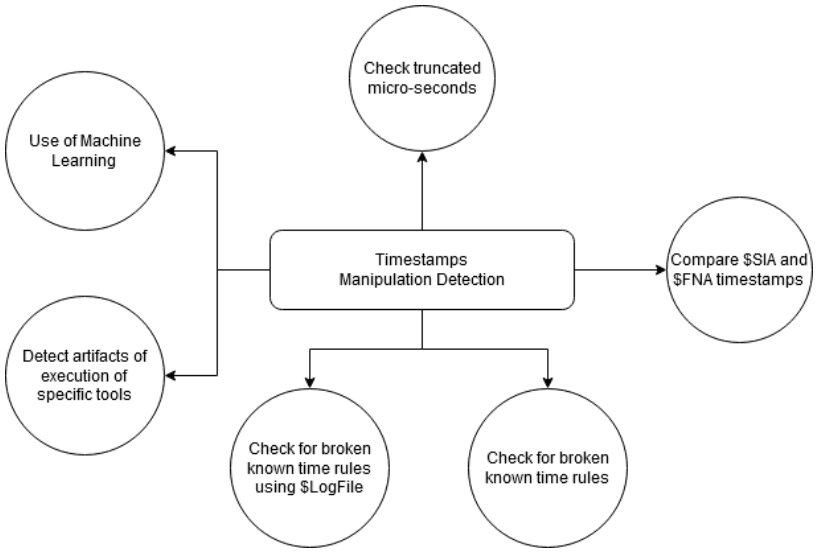

Fig.2. Techniques of the detection of timestamp manipulation. $SIA and $FNA stand for standard information attribute and filename attribute (respectively).

According to Carrier [7], "Windows does not typically update this ($FNA) set of temporal values like it does with those in the $SIA attribute, and they frequently correspond to when the file was created, moved, or renamed". From such observation, the digital forensic community compares the $SIA timestamps to its counterpart $FNA timestamps to detect the time tampering. This method is simple and widely adopted. Other studies [4,12,17] focus on the usually known time rules of the common user behaviors on the file system. K. P. Chow and al [4] provides the digital forensic community with a set of seven time rules that characterize the behavior of MACB times on the NTFS file system. Breaking such time rules may suggest a time anomaly requiring further investigation for a possible timestamp manipulation. The idea behinds such research is to leverage the normal user behavior as the gold-standard, and any irregularities in the time metadata have to be further investigated. However, there is no research that processes and lists the complete list of the normal user actions between different operating systems and different filesystem formatted partitions. In a different approach [15] that showcases the possible use of supervised machine learning algorithm to detect massive timestamp manipulation. The experiment extracts features from the MFT file on a Windows 10 virtual machine. However, the approach is based on a limited and a controlled virtual environment. The last category of methods suggest that the timestamp manipulation tool may leave digital footprints on the Windows OS itself, such as evidence of its execution among the Windows Prefetch

files. Such artifacts should not be neglected. Nevertheless, the whole feature of Windows Prefetching can be easily disabled. In this section, we presented a couple of methods for detecting the tampering of the file system timestamps. As it can be seen, the methods are very different and use different approaches to tackle the same problem.

4. Proposed Methodology

In this part, we describe the current set-up and the methodology followed in order to re-create a near-real environment. Using the data generated, we will check when it is possible to detect, or not, the timestamps alteration. Based on the same starting virtual machine, we generated three scenarios (I, II and III) as shown in table 1. The scenario I represents the timestomping before the recording of a VSC. The scenario II represents the timestomping after the recording after a VSC. The scenario III is similar to the previous one and aims to validate with more files. By doing so, we are considering all possible cases.

We initiate this section by detailing our software environment and the tools used to perform the experiment. We used a Debian GNU/Linux 10 as a host OS. Using Oracle VirtualBox v6.1.12 manager, we set up a virtual machine with a Windows 10 Pro version 2004 x64 as a guest OS. To perform the extraction of the MFT file, we used 7zip v16.02. However, to extract the MFT file from the volume shadow copies, we used Volume shadow tools v20191221 combined with 7zip. The parsing of the MFT file was performed using "DFIR_NTFS" python module v1.0.6. Finally, for CSV reading, writing and diffing, we used "Pandas" python library v1.0.5 and "csvdiff" v0.3.3 (respectively).

The methodology consists of processing a disk image and outputting the files susceptible to have been timestamp manipulated. In order to do so, first, we mounts each volume shadow copy file into a mounting point and then extract each MFT file. Same steps are performed for the main volume of the disk image. The results are N+1 MFT file where N stands for the number of VSCs. After that, a dismantle and a clean up is performed of the mounted volumes (main and shadow copies). "DFIR_NTFS" module parses the N+1 MFT files to CSV format. Then, based on the below-named algorithm 1 we establish the list of the files susceptibles to have been timestamp manipulated. Fig. 3, illustrates the steps of the proposed methodology.

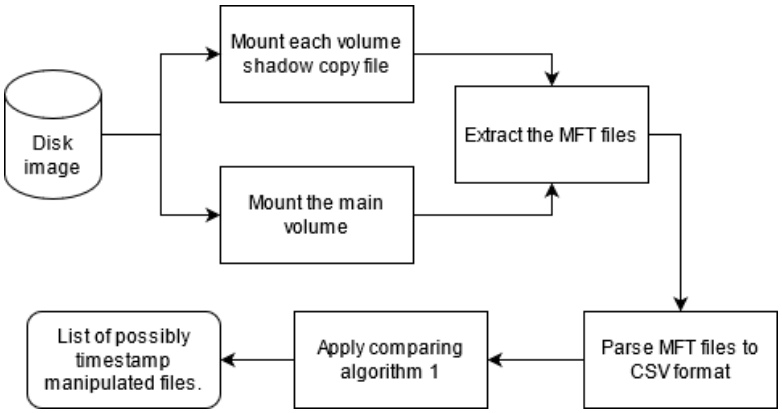

Fig.3. A flow chart describing the steps of the proposed methodology.

Next, we will be describing the designed scenarios. We set up a virtual machine (VM) with a dynamically allocated virtual disk. We install on the VM a none-activated Windows OS with the installation wizard default settings. After an update of the OS, we pause the Windows Update for a couple of days and disable networking on the virtual machine. The purpose of such action is to minimize the "touched" or accessed files by the update manager. We turn-on the system protection for the local disk "C:" in the advanced computer settings. We adjust the space usage allocated for the maximum size of restore points to 40% of the disk space (around 20 Gb). Finally, we activate the volume shadow service (VSS). In sum, we have at our disposal a starting virtual machine where we can perform the three aforementioned scenarios based on a single VM.

The scenarios held can be summarized as shown in table 1. We manipulate the $SIA MACB timestamps of a single file before snapshotting a volume shadow copy and after that. Then, we manipulate the $SIA MACB of 991 files from Govdocs1 dataset [16] after the recording of the VSC. The timestamps of conern are the "last-Modified", "last-Access", "last entry Changed", and "Birth" timestamps of the "Standard_information" attribute. We timestomped a single file on two first scenarios. However, in the third scenario we validate the case with 991 files from Govdocs dataset.

Table 1. Summary of all the performed scenarios with or without the use of the dataset "Govdocs1" [16].

Scenario	Number of VSCs	Number of timestomped files	Timestamps of concern	Timestomp before VSC or after
I	01	01	$SIA MACB	Before the VSC
II	01	01	$SIA MACB	Between VSC and the main volume
III	01	991 files from "Govdocs1" dataset	$SIA MACB	Between VSC and the main volume

In other words, in the 1st scenario, we create a single text file with "Lorem Ipsum" content. We ensure it is not an MFT resident file. Then, using timestomp utility, we manipulate the $SIA MACB times of the text file before proceeding to create a manual volume shadow copy (using the following command "wmic shadowcopy call create Volume=C:" with administrator privileges). In the 2nd scenario, we reproduce the same steps on a duplicate copy of the starting VM with one single detail. We manipulate the times of the text file after recording the VSC. The 3rd scenario is similar to the previous one at one single exception. We are timestomping 991 files from "Govdocs1" dataset [16] on a duplicate copy of the starting VM and then performing the VSC recording as usual.

The results of the three scenarios are held within three virtual disks. We convert each virtual disk image to a RAW format and then extract the VSC with the Volume shadow tool. In sum, we have 3 main volumes and 3 VSCs stored in RAW format disk images. DFIR_NTFS allows the extraction of the 6 MFT files and generates the CSV format files for further analysis and comparisons.

5. Result Analysis and Discussion

The volume shadow copy file is nothing more than a differential backup and a snapshot of the system at a certain moment in the past time. According to that, we have established the following three observations in case of normal user behavior. Usually, the recording of the VSC is triggered automatically once a day on Windows Vista, every 7 days on Windows 7, etc but in the circumstances of a controlled environment such as the three Windows 10 scenarios, the aim of manually creating VSCs is to validate the subsequent observations and rules.

All the MACB timestamps stored on an MFT of a certain volume shadow copy file are less or equal to the date and time of the creation of that same volume shadow copy file (OBSERVATION 1). Indeed, if we have a working system and we take a snapshot of its files and their metadata, each file has a "last-Modified", "last-Access", "last entry Changed", and "Birth" timestamps less or equal the time and date of the creation of that volume shadow copy file.

If we suppose we have more than one volume shadow copy, the timestamps associated with a file that exists in the eldest volume shadow copy are less or equal to the timestamps of the same file on the less old volume shadow copy and so on. For illustration purposes, suppose the "last-Access" and "last-Modified" times of a file in the oldest volume shadow copy VSC1 are 01/01/2020 01:01:01, if there is newest volume shadow copy VSC2 and it contains the same file, that file will either be accessed or modified in between the time of the creation of the VSC1 and VSC2. Thus, the "Last Access" and "Last-Modified" times of the file at VSC1 are less or equal to "last Access" and "last-Modified" times of the same file at VSC2. We conclude by the following observation: If multiple volume shadow copies are captured and available, the MACB of any file of the first created VSC is less or equal to the MACB timestamp of the same file on the second created VSC and so on (OBSERVATION 2).

We can generalize the second observation into a third one by extending the last volume shadow copy to the forensic image copy of the analyzed media. Indeed, each MACB timestamp associated with a file in the main volume of the forensic image is greater or equal to the MACB timestamp of any captured volume shadow copies in the same forensic image (OBSERVATION 3). The third observation is established because no more events had happened on the system after the forensic imaging process was done.

In the next paragraph, some definitions are considered for establishing three equations which are the findings related to the timestamp rules. In the following equations, VSC stands for volume shadow copy available within a forensic image. All the VSC available within a forensic image are mentioned as VSC_s. The $n^{ieme}\ VSC$ is the n^{ieme} VSC in the forensic image based on the volume shadow copy date and time of creation. $DTCreation(VSC)$ returns the date and time of creation of the VSC in input. In parallel, $MFT(VSC)$ returns the MFT file of the VSC in a list of MACB timestamps. The timestamps $MACB_i$ stands for the i^{eme} MACB timestamps in the list of the MFT file.

Based on the previous observations and the definitions established, we present the following findings, which are mainly three rules of time that describe the normal behaviors of timestamps in an inter-snapshot analysis:

$$\forall VSC \in VSC_s, \forall MACB_i \in MFT(VSC),\ \ MACB_i \leq DTCreation(VSC) \tag{1}$$

$$\forall MACB_i \in MFT(VSC_1) \leq\ MACB_i \in MFT(VSC_2) \leq \cdots \leq MACB_i \in MFT(VSC_n) \tag{2}$$

$$\forall VSC\ \in VSC_s, \forall MACB_i \in MFT(VSC), MACB_i \in VSC \leq MACB_i \in MFT(MainVolume) \tag{3}$$

After establishing the rules of time, we propose a new algorithm (ALGORITHM 1) which allows us to spot and reduce the number of the files possibly timestamp manipulated by analyzing and comparing the MACB timestamps between all the recorded volume shadow copies and the main volume of the forensic image. The algorithm is based on the idea that on normal behavior, the MACB timestamps available do not break the time rules (2) and (3). Thus, any MACB timestamp that does not respect the time rules has possibly been timestamp-manipulated.

Algorithm 1: A new algorithm to detect and reduce the number of files susceptible to be timestamp manipulated.

```
Data:All MFT of the volume shadow copies and the main volume
Result:A list of file records of possibly timestamp manipulated files

/* Appending MFT files to a list by creation datetime */
MFT_list=[MFT ∀ MFT∈VSCs sorted by creation datetime]
MFT_list <- MFT(MV)

/* Browsing each timestamp of each file record */
i <- 0
ln <- length(MFT_list)
while i ≤ ln−1 do
        MFT_before <- MFT_list[i]
        MFT_after <- MFT_list[i+1]
        for i in MFT_before do
            MACB_before <- MACBi
            if i in MFT_after then
                MACB_after <- MACBi
                if MACB_before > MACB_after then
                    possibly_altered_MACB_record <- i
                end
            end
        end
        i <- i+1
end
```

We run the implementation of the algorithm on the resulting CSV files of the three scenarios of the experiment. As expected, the timestomped files have been detected when a VSC is recorded and available before the time of timestomping. We observed that there is a criterion that ensures whether the timestamps will be detected or not. The timing of the creation of the volume shadow copy in relation to the event of timestomping. In case, the previously available files are timestomped after a snapshot of VSC, the algorithm can detect the timestamp manipulation. Otherwise, in case of the timestomping is performed out of the scope of any VSC, it won't be detected (scenario I).

Table 2. Summary of all the performed scenarios with or without the use of the dataset "Govdocs1" [16] and the output result of the algorithm.

Scenario	Details of the previous scenarios	Results
I	A single file timestomped ($SIA MACB) before the recording of the VSC.	Not detected
II	A single file timestomped ($SIA MACB) between the recording of the VSC and the main volume.	Detected
III	991 files from "Govdocs1" dataset timestomped ($SIA MACB) between the recording of the VSC and the main volume.	Detected

In order to calculate the complexity in time and space, we consider that sorting the MFT files in the list by creation date is O(1) in time since we know their order of creation. Thus, the algorithm is just appending them to a list. And if we suppose that we have N volume shadow copies and the main volume, and the dimension of the biggest VSC is M∗K, K is of a fixed value and M is of big magnitude, the complexity in time is (N+1)∗M∗K, which totalize O(N∗K∗M).

For N=2 with N the number of VSCs on a freshly installed Windows 10 v2004 64-bits as guest OS, we get the execution time of 16min using Python 3.8 on a processor of type Intel Core i3-3110M @ 2.40GHz x 4.

6. Conclusion

The volume shadow copies contain a significant amount of historical data, thus, allowing not only access to the previous content but also provide valuable time-related information. We went through three scenarios and proposed an approach in order to fight the anti-forensic method of timestamp manipulation using volume shadow copy. The method was capable of spotting the time manipulation in a controlled virtual environment. The experiment held reproduces as near as possible a real-life scenario where a digital forensic expert gets hold of a computer, or a server, where the

protection with volume shadow service is activated. In such a case, our approach will help the forensic expert to detect the timestamp manipulated files and frame “when” some events happened during the breach.

The inter-snapshot analysis and timestomping are still open research topics. Our contribution meant to explore the possibility to detect timestomping using recorded shadow copies and to fill that gap. For future work, and knowing that different file systems exhibit different time behaviors, a similar research study could be conducted for other proprietary file systems such as ReFS, we could also consider expanding the timestamp manipulation to other new timestamps such as Window Subsystem Linux timestamps.

References

[1] Shuaibur Rahman, M. N. A. Khan,"Digital Forensics through Application Behavior Analysis", International Journal of Modern Education and Computer Science, Vol.8, No.6, pp.50-56, 2016.

[2] T. Raja Sree, S. Mary Saira Bhanu," Investigation of Application Layer DDoS Attacks Using Clustering Techniques", International Journal of Wireless and Microwave Technologies, Vol.8, No.3, pp.1-13, 2018.

[3] Dhwaniket Ramesh Kamble, Nilakshi Jain, Swati Deshpande,"Cybercrimes Solutions using Digital Forensic Tools", International Journal of Wireless and Microwave Technologies, vol.5, no.6, pp.11-18, 2015.

[4] Chow, K.-P., Law, F. Y., Kwan, M. Y., & Lai, P. K. (2007). The rules of time on ntfs file system. In Second International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE’07), pages 71–85. IEEE, DOI: 10.1109/SADFE.2007.22.

[5] Minnaard, W., de Laat, C., & van Loosen MSc, M. (2014). Timestomping ntfs https://delaat.net/rp/2013-2014/p48/report.pdf last accessed: 20/02/2021

[6] Documentation online, Microsoft (2018a). Master file table (online) https://docs.microsoft.com/en-us/windows/win32/fileio/master-file-table last accessed: 20/02/2021.

[7] Carrier, B. (2005).File system forensic analysis. Addison-Wesley Professional.

[8] Documentation online, Microsoft (2018b). File times (online) https://docs.microsoft.com/en-us/windows/win32/sysinfo/file-times Last accessed: 20/02/2021.

[9] Neuner, S. & all (2016). Time is on my side: Steganography in filesystem metadata.Digital Investigation, 18:S76 – S86, DOI: 10.1016/j.diin.2016.04.010.

[10] Carvey, H. (2014). In Carvey, H., editor,Windows Forensic Analysis Toolkit (Fourth Edition). Syngress, Boston, fourth edition edition.

[11] Sreeja, S. C. & Balan, C. (2016). Forensic analysis of volume shadow copy in windows 7. In2016 International Conference on EmergingTechnological Trends (ICETT), pages 1–6, DOI:10.1109/ICETT.2016.7873670.

[12] Jang, D.-i., Hwang, G.-J. A. H., & Kim, K. (2016). Understanding anti-forensic techniques with timestamp manipulation. In2016 IEEE17th International Conference on Information Reuse and Integration (IRI), pages 609–614. IEEE, DOI: 10.1109/IRI.2016.94.

[13] MITRE A. (2017). Win32/usb stealer, https://attack.mitre.org/software/S0136/ (online) last accessed: 20/02/2021.

[14] Gungor, A. (2014). Date forgery analysis and timestamp resolution. (online) https://www.meridiandiscovery.com/articles/date-forgery-analysis-timestamp-resolution/ last accessed: 20/02/2021.

[15] Alji, M., & Chougdali, K. (2019). Detection of Timestamps Tampering in NTFS using Machine Learning. *Procedia Computer Science*, *160*, 778-784, DOI: 10.1016/j.procs.2019.11.011.

[16] Garfinkel, S., Farrell, P., Roussev, V., & Dinolt, G. (2009). Bringing science to digital forensics with standardized forensic corpora.digital investigation, 6:S2–S11, DOI: 10.1016/j.diin.2009.06.016.

[17] Cho, G.-S. (2013). A computer forensic method for detecting timestamp forgery in ntfs.Comput. Secur., 34:36–46. DOI:10.1016/j.cose.2012.11.003.

Authors’ Profiles

Eng. ALJI Mohamed is a PhD student at the Engineering Science Laboratory, National School for Applied Sciences, Ibn Tofail University, Kenitra, Morocco and a computer science engineer from the ENSEIRB-MATMECA engineering school, Bordeaux, France.

Dr. CHOUGDALI Khalid is an associate professor in computer science at the National School for Applied Science, Ibn Tofail University, Kenitra Morocco. Its main research areas are information security, digital forensics and data analysis.

How to cite this paper: ALJI Mohamed, CHOUGDALI Khalid, "Detection of Suspicious Timestamps in NTFS using Volume Shadow Copies", International Journal of Computer Network and Information Security(IJCNIS), Vol.13, No.4, pp.62-69, 2021. DOI: 10.5815/ijcnis.2021.04.06