International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 12 | Dec 2025 www.irjet.net p-ISSN: 2395-0072

An Artifact-Driven Framework for Detecting File Wiping and Timestamp Manipulation in NTFS Systems

Gopika Gopalakrishnan[1], Sreeja S C[2], Lloyd K L[3]

[1]*Researcher, ER & DCIIT, CDAC, Trivandrum, Kerala, India*
[2]*Scientist E, Cyber Forensics Section, CDAC, Trivandrum, Kerala, India*
[3]*Scientist E, Cyber Forensics Section, CDAC, Trivandrum, Kerala, India*

Abstract - *The rise of sophisticated anti-forensic techniques, particularly file wiping tools and timestamp manipulation, poses significant challenges to modern digital investigations by hindering the recovery and validation of critical evidence. These techniques are deliberately employed to obscure user activity, erase digital traces, and disrupt forensic timelines. This paper presents the development of an advanced anti-forensic detection methodology aimed at identifying and monitoring suspicious activities such as File wiping and Timestamp Manipulation. The paper presents multiple forensic analysis modules capable of parsing core NTFS artifacts such as $MFT, $Log File, and $UsnJrnl extracted from disk images. In addition, it incorporates Prefetch file analysis to detect the execution of known anti- forensic utilities. By correlating evidence across these sources, the tool enhances the capability of investigators to detect and interpret attempts to tamper with or eliminate digital evidence, thereby strengthening the integrity of forensic examinations. Experimental results highlight the importance of artifact-driven approaches in countering evolving anti-forensic threats.*

Key Words: **Digital Forensics, File Wiping, Timestamp Ma- Manipulation, NTFS Analysis, Anti-Forensics, Prefetch,$UsnJrnl, $Log File**

1. INTRODUCTION

Nowadays, cyber-attacks are becoming more frequent and sophisticated, so it is necessary to understand the techniques used by hackers to be able to carry out a correct forensic analysis leading to the identification of the perpetrators [1]. Anti-forensic techniques have evolved significantly over the past decade, posing major challenges to digital forensic practitioners [2]. Among these techniques, file wiping has emerged as one of the most effective methods for permanently destroying digital evidence. Unlike standard file deletion, file wiping overwrites the physical disk sectors that once contained the file's contents. This process may involve one or multiple overwrite passes using predictable patterns, making the recovery of residual magnetic traces extremely difficult, even with advanced hardware-based recovery technologies. As anti-forensic techniques or specific tool sari developed over time, the resulting associated traces will also change. While there normally appears to be a core group of relatively generic traces, signatures for anti- forensic techniques and programs will need to be maintained over time. While signature-based detection methods are like if a signature does not exist for each specific anti-forensic technique, then the technique cannot be detected. More data sources, such as Windows Restore Points and log files, should be included in the analysis [3]. Timestamp manipulation is another deliberate and strategic anti-forensic tactic employed to alter the recorded time-related metadata of files or folders on a digital storage system. Most modern file systems—such as NTFS (Windows), ext4 (Linux), or APFS (macOS)—maintain multiple types of timestamps for every file, including creation time, last modification time, last access time, and, in some systems, change time (i.e., when file metadata was last altered). These timestamps are critical for forensic investigators, as they help reconstruct timelines of user activity, file usage, system events, or potentially malicious behavior.

2. BACKGROUND

Anti-forensic methods have a great impact on the reliability and integrity of digital forensic processes, posing major challenges to evidence recovery. File wiping is a highly secure and intentional method of data deletion, designed to render files completely unrecoverable—even by advanced forensic recovery techniques. Unlike standard file deletion, which merely removes the pointer or reference to the file in the operating system's file system (such as the Master File Table or $MFT in NTFS), file wiping targets the actual physical location of the data on the storage medium. In a typical deletion, the data still resides on the disk and can often be recovered using forensic tools because the system only marks the space as" available" without immediately erasing the contents. Wiping, on the other hand, goes far beyond this superficial deletion. It directly overwrites the sectors where the file's data was stored [4]. This can be done using single or multiple passes of

binary patterns (such as 0x00, 0xFF, or random values). The more overwrite passes that are performed, the harder it becomes—even theoretically—for any residual data to be recovered using sophisticated hardware analysis techniques. Some of the common File Wiping tools used are Eraser, CCleaner, File Shredder, Delete (Sysinternals), Bleach Bit, Wipe File, and Filenuke.

Timestamp manipulation is a deliberate and strategic anti- forensic tactic employed to alter the recorded time-related metadata of files or folders on a digital storage system. Timestamp tampering is an act of anti-forensics in which a suspect manipulates the timestamps of data related to a crime to conceal traces, making it difficult for analysts to reconstruct the situation of the incident. This can delay investigations or lead to the failure to obtain crucial digital evidence. Therefore, various techniques have been developed to detect timestamp tampering [5]. Most modern file systems—such as NTFS (Windows), ext4 (Linux), or APFS (macOS)—maintain multiple types of timestamps for every file, including creation time, last modification time, last access time, and, in some systems, change time (i.e., when file metadata was last altered). These timestamps are critical for forensic investigators, as they help reconstruct timelines of user activity, file usage, system events, or potentially malicious behavior.

When a user or attacker modifies one or more of these timestamps, the goal is often to obscure the true nature or timing of an action. For example, a malicious actor may backdate a malware file's creation time to make it appear as though it was present on the system long before the actual intrusion occurred. Similarly, someone attempting to conceal the deletion of sensitive files may adjust the timestamps of remaining files to confuse or disrupt a forensic timeline analysis. Types of Timestamps: In most modern file systems (like NTFS, ext4, HFS+, etc.), files have multiple timestamps:

- Creation Time – When the file was first created.
- Modification Time (Last Write) – When the file's content was last changed.
- Access Time – When the file was last read or opened.
- Metadata Change Time (Change Time) – When the file's metadata (like name or permissions) was altered. These timestamps are crucial for building timelines during digital investigations.

Different Methods of Timestamp Manipulation:

- Using Anti-Forensics Tools like touch.exe, Time stomp (part of Meta sploit), Set MACE, or third-party programs allows attackers to change timestamps selectively.
- Indirect Manipulation via File System Behavior: Copying a file to another location may change or reset some timestamps depending on the system.

3. METHODOLOGY

3.1 Detection of Wiped files

Step 1: Identifying File Wiping Tools from the Prefetch files

Prefect files in Windows are an essential part of the operating system's performance optimization. These files store all the necessary information regarding the executable application [6]. Therefore, by parsing the prefetch files, we obtain the following attributes as shown in Fig. 1.

- **Executable Name**: The executable name refers to the name of the application that was run on the system, and it is a key part of the Prefetch file. Each Prefetch file is named using a specific pattern that includes the executable's name followed by a hash value. For example, if a user launches the tool named File Shredder, the system will create a Prefetch file named something like SHREDDER.EXE/-63FA1305.pf. This filename clearly identifies that the Prefetch file corresponds to the execution of SHREDDER.EXE. For forensic investigators, the executable name provides direct evidence of which application was executed, and it serves as the starting point for deeper analysis, such as determining the time of execution, run count, and associated file access.

- **Run count**: The Run Count is a numeric value stored within a Prefetch file that indicates how many times a specific application has been executed from its current location on the system. This counter increases each time the program is launched and it is maintained by the Windows operating system to help optimize future launches of the application.

- **Execution times**: The Last Execution Time refers to the exact date and time when an application was most recently executed on the system from its current file path. This timestamp is stored in the Prefetch file in Windows FILETIME format, which records the number of 100-nanosecond intervals since January 1, 1601 (UTC). It can be converted into a

standard human- readable format (e.g., YYYY-MM-DD HH:MM: SS) using forensic tools or scripts.

From a forensic standpoint, the Last Execution Time is critical for establishing temporal evidence—it shows when a particular application was used. This is especially useful when investigating malware activity, unauthorized access, or the use of anti-forensic tools. For instance, if a wiping tool was last executed just before key files disappeared, it strongly suggests intentional data destruction. Additionally, this timestamp can be cross-referenced with other artifacts such as event logs, $MFT entries, or user login sessions to corroborate timelines and attribute activity to specific users. In Windows 8 and above, prefetch files also store multiple past execution times.

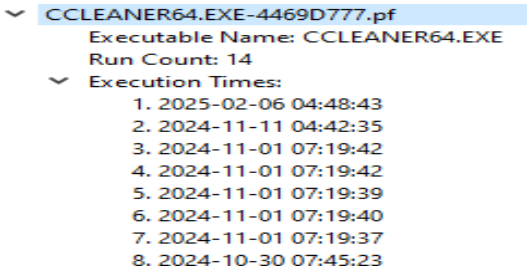

Fig. -1: Prefetch File Data

Step 2: Identifying wiped files from $UsnJrnl Files

An analysis of the $UsnJrnl revealed that wiped files could be identified through distinctive file-rename patterns. The USN Journal ($UsnJrnl) is a system file used by the NTFS file system in Windows to log changes made to files and directories on a volume. The $UsnJrnl file consists mainly of two data streams, i.e., $Max and $J. Each record in the $J stream logs actions such as file creation, modification, deletion, and renaming, along with associated metadata like timestamps, reason codes, and file names. With the help of the NTFS log tracker v1.8 tool, we were able to identify that the files wiped using the File shredder tool were renamed to a format such as 1278654.ZZZ.Also, by checking the $UsnJrnl files, we could see the original filenames as well as their renamed filenames. Therefore, we were able to identify a pattern for file wiping done by the file shredder tool. From the $UsnJrnl structure, we can identify the reason flags, which indicate the renaming of the files.0x1000 = RENAME_OLD_NAME: Indicates the file was renamed from this name. 0x2000 = RENAME_NEW_NAME: Indicates the file was renamed to this name.

Hence, by checking the 1278654.ZZZ format in the RENAME_NEW_NAME and checking their RE- NAME_OLD NAME, we can identify that the file is wiped by the file shredder tool. Here we take an example of a file whose old name is doc1.pdf and new name is 3282899.ZZZ after the file is wiped by the file shredder tool, which is shown in fig 2.

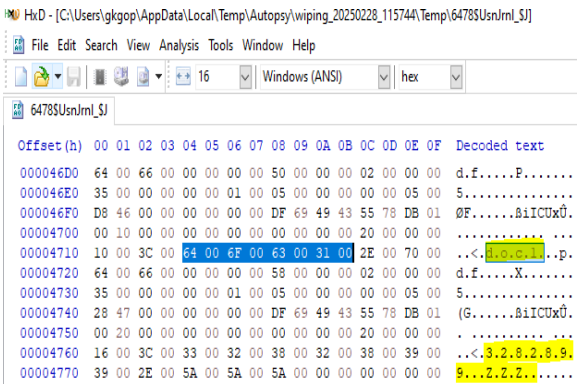

Fig. -2: $UsnJrnl structure loaded in HXD editor, which shows both the old and new file name highlighted

3.2 Detection of Time stomped files

Similar to that of Step 1 mentioned in the section on detecting wiped files, here also we can use prefetch files for checking the Time stomping Tools.

Method 1: Correlating $Log File and $MFT file

In NTFS, the $Log File maintains a transaction log of changes made to metadata within the file system, including updates to the $MFT (Master File Table)[7]. Each log record in $Log- File contains operation codes (opcodes) that describe specific actions performed on file metadata.

0x07 – Update Resident Value This opcode indicates that a resident value within an $MFT record has been updated. Specifically, it often refers to modifications in the $FILE NAME (0x30) attribute, which contains temporal information such as creation, modification, access, and entry change timestamps.

In normal file operations, when a legitimate timestamp update occurs (e.g., due to accessing or modifying a file), the 0x07 op code (described in fig 3) typically appears once in the associated $Log File transaction. In time stomping scenarios, where a tool is deliberately used to alter timestamps (e.g., to evade detection), the same operation is carried out repeatedly or across multiple attributes. As a result, the 0x07 opcode appears multiple times in a single or closely related set of $Log File entries. By correlating entries in the Log File with MFT records and observing the frequency of the 0x07 opcode, forensic investigators can identify potential time stomping attempts. A normal file will show a single instance of this operation during timestamp changes, whereas a time stomped file will typically generate multiple 0x07 operations, indicating possible manipulation by anti-forensic tools. After identifying files in the $Log File that have the 0x07 – Update Resident Value operation code flagged multiple times, the next step in confirming timestamp manipulation (time stomping) involves analyzing their metadata within the $MFT (Master File Table).

- $STANDARD INFORMATION ($SI): Contains

original timestamps such as file creation, last modification, last access, and entry modification times. These are the system-maintained timestamps.

- $FILE NAME ($FN): Also contains timestamps, but these are sometimes copied from $SI and not always updated the same way. They reflect metadata as seen from the parent directory's perspective.

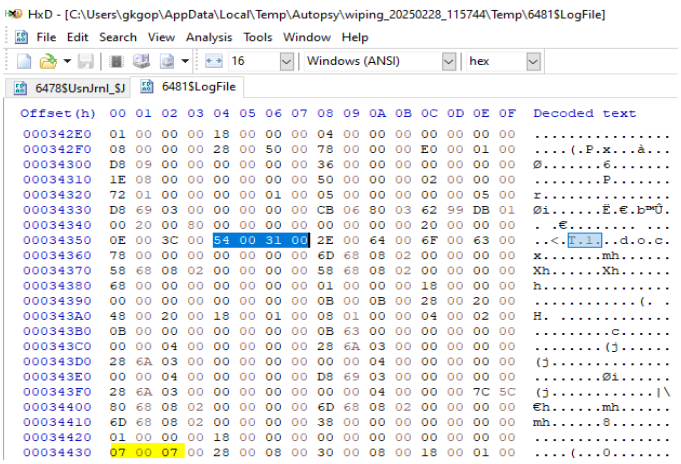

Fig. -3: Operation code 07 identified for a times tomped file T1.docx

Under normal conditions, the timestamps in the $SI and $FN attributes are usually closely aligned, if not identical. However, when a file has been time stomped using an anti- forensic tool, only the $SI timestamps are typically altered, while the $FN timestamps remain unchanged or reflect the original values. By comparing the $SI and $FN timestamps of files that have multiple 0x07 operations in the $Log File, investigators can detect inconsistencies. A clear discrepancy between these timestamp sets strongly indicates that the file was subjected to manual timestamp manipulation (time stomping). This technique helps validate and strengthen the evidence of anti-forensic activity. The timestamp value compared from the HXD editor is shown in Fig. 4, and their converted time, i.e, human-readable timestamps, are shown in Fig. 5.

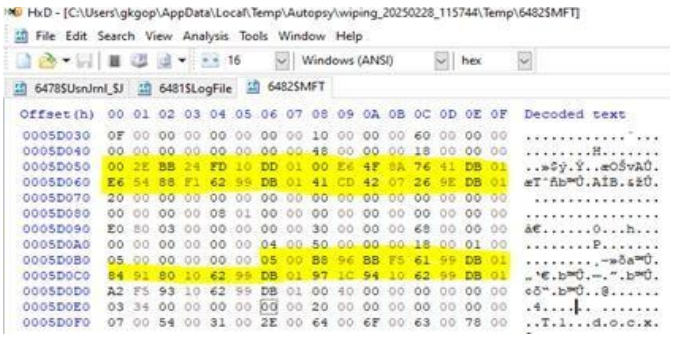

Fig -4: Timestamps compared from $MFT file

$SI

11-07-2026 6.18.52 AM	28-11-2024 9.18.52 AM
20-03-2025 6.40.17 AM	26-03-2025 8.06.50 AM

$FN

20-03-2025 6.33.15 AM	20-03-2025 6.34.00 AM
20-03-2025 6.34.00 AM	20-03-2025 6.34.00 AM

Fig. -5: Human-readable timestamps taken from HXD Editor

Method 2: Correlating $UsnJrnl and $MFT file

The USN Journal ($UsnJrnl) is a system file used by the NTFS file system to log changes to files and directories on a volume. It serves as a chronological record of file system activities, including creation, modification, renaming, and deletion. From a forensic perspective, $UsnJrnl is highly valuable because it retains evidence of file operations even if the files have been deleted or renamed and can reveal suspicious or anti-forensic behavior, such as timestamp manipulation or file wiping.
One key field within each USN record is the "reason flag", which identifies the type of change that occurred to the file. Of particular interest in time stomping investigations is the flag:

- 0x8000 – BASIC INFO CHANGE this flag indicates that a file's attributes or timestamps have been modified, potentially pointing to deliberate tampering. The reason flag is highlighted for a file in $UsnJrnl is shown in Fig. 6.

For a normal file, this 0x8000 flag typically appears once during regular attribute or timestamp updates. However, for time stomped files, this flag tends to appear multiple times in the journal records for that file. This repeated presence of the 0x8000 flag for a single file is an anomaly and raises suspicion of time stomping. If a timestamp mismatch is found between the SI and FN attributes for those files that have multiple 0x8000 flags in $UsnJrnl, it provides strong evidence of timestamp manipulation. By identifying files that have the 0x8000 reason flag occurring multiple times in $UsnJrnl and then comparing their SI and FN timestamps from the $MFT, investigators can detect time stomping with high confidence. This method is effective even if other forensic artifacts (like Prefetch or event logs) have been cleared, making it a powerful anti- anti-forensics technique.

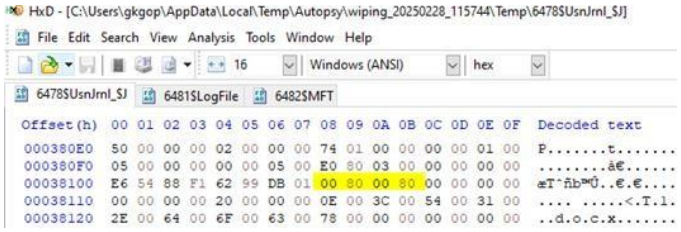

Fig -6: Reason flag 0x8000 appeared in $UsnJrnl file

4. EXPERIMENTS AND FINDINGS

4.1 Experiments performed for File Wiping

In this study, five files representing different data formats were selected to systematically examine the effects of anti-forensic wreaking techniques. The selected artifacts included a Microsoft Word document (Ant forensics), a PDF file (doc 1), a PowerPoint presentation (Internship ppt), a plain text document (P1) and a JPEG image (Pi7 tool photo). Each file was subjected to deliberate data destruction using different anti-forensic tools. Specifically, the Word document was wiped using the Eraser utility, the PDF file was processed through File Shredder, and the PowerPoint presentation was sanitized using Wipe File. Additionally, both the text document and the JPEG image were targeted and erased using File Nuke. The selection of various file types was designed to replicate real-world scenarios in which multiple data structures coexist within storage media. By applying different anti-forensic tools across varied file formats, the experiment sought to evaluate tool- specific wiping patterns, residual traces, and their impact on subsequent forensic recovery efforts.

- An image of the folder that consisted of those five files was taken and loaded into the autopsy tool, and the following was obtained.

- We identified that the contents are not wiped off completely by the tools instead, they were filled with random values. Henc,e it was understood that chances of detecting the files are possible and there might be a chance of recovery.

- During the analysis of the deleted files list, only the ”Internship PPT.pptx” file was identified - previously wiped using the WipeFile tool. This observation suggests that files erased with WipeFile can still be recovered, at least partially, using forensic tools such as Autopsy.The list of deleted files taken from Autopsy software is shown in Fig 7. However, it is important to note that Autopsy was only able to recognize the file as permanently deleted; no specific metadata or indicators were found to suggest that the file had undergone a deliberate wiping process.

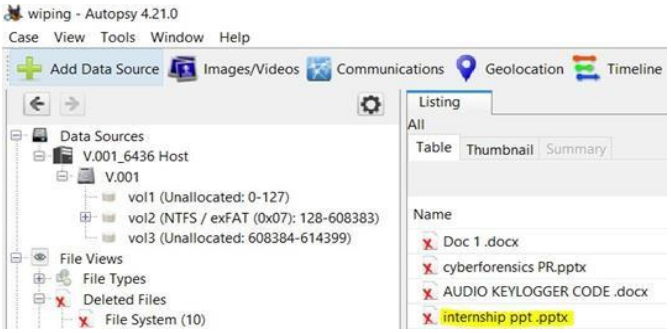

Fig -7: List of deleted files shown from the image file loaded inside the autopsy

- To further investigate the presence and activity related to other files, the $LogFile extracted through the Autopsy forensic tool was analyzed using the NTFS LogFile Tracker v1.8 utility. The $LogFile loaded inside the NTFS LogFile Tracker v1.8 is shown in Fig 8.This analysis aimed to examine transactional records maintained by the NTFS journaling system, enabling the identification of file creation, modi- fication, deletion, and renaming operations that might not be apparent from traditional file system views.

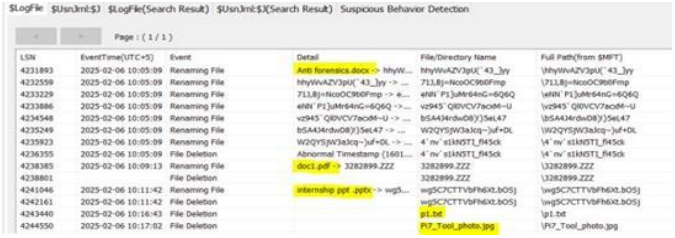

Fig -8: $logFile loaded inside the NTFS LogFile Tracker v1.8

Findings obtained are given as follows:

1) Anti forensics.docx:

- The file "Anti Forensics.docx" which is deleted using the eraser tool has undergone renaming 7 times and hence we can consider this as one of the features of the eraser tool. (Note : Eraser renames the wiping file multiple times before deleting it .Here I have used the erasure method as Gutman 35 passes. Therefore, Eraser might rename files multiple times as part of its overwriting process, especially when using complex methods like Gutmann's 35-pass overwrite)

- While checking the $Log File in HXD editor we could see that the renaming files appears after the anti-forensic file name and so while analyzing the $Log File, if we find any file having multiple renaming it could be identified that some sort of file wiping has been undergone for that file. The structure of $Logfile analyzed through the HXD editor has been shown in Fig 9.

Fig -9: Structure of $logfile which displays the renamed names in HXD editor

- Apart from the $LogFile records the copy of MFT records are also displayed in the $LogFile.
- Also, we were able to identify the last renamed file name in the deleted files list which was displayed in the Autopsy but identifying that name alone would not help the forensic analyst to jump into a conclusion regarding the wiped files.

2) doc1.pdf:

- The analysis of the $LogFile using NTFS LogFile Tracker clearly showed that the file was renamed to "3282899.ZZZ" before it was deleted.
- This file had been deleted after being wiped using the File Shredder tool.
- Further experiments were conducted by wiping additional files with File Shredder, and in every case, the files were renamed into a "number.ZZZ" format prior to deletion.
- This consistent renaming behavior indicates that the "number.ZZZ" naming pattern is a default feature of the File Shredder tool, rather than an isolated occurrence.
- Based on these observations, it can be concluded that files found with a numeric name and ".ZZZ" extension are strong indicators of having been renamed and wiped using File Shredder.

3) Internship PPT.pptx:

In this case, the file was renamed to a random new name consisting of arbitrary alphabetic char- acters, a behavior like that observed with the Eraser tool. However, unlike the File Shredder tool—which consistently renames files following a distinct "number.ZZZ" pattern—no consistent or recognizable naming pattern could be identified with Eraser. The renaming process appeared to be random, making it more challenging to associate the renamed files directly with a specific anti forensic tool based solely on the filename structure.

4) p1.txt & pi7 Tool photo.jpg:

It was observed that these two files did not undergo any renaming prior to deletion by the file nuke tool compared to other file wiping tools. Instead, they were directly deleted without any modification to their original filenames. The original file names could be retrieved from the $Log File by analyzing the transactional records, however no specific metadata was found within the $Log File entries to suggest that these files had been subjected to a file wiping process. As a result, while the deleted files were identifiable, their deletion appeared similar to normal file deletion rather than anti forensic wiping activities.

4.2 Experiments performed for Timestamp Manipulation

For studying timestamp manipulation, three files of different formats were selected and processed using three distinct timestamp modification tools. The selected artifacts were as follows: a Microsoft Word document (T1) was manipulated using the Bulk Extractor tool; a PowerPoint presentation (T2) was processed with New File Time; and a plain text document (T3), created using Notepad, was modified using the Touch utility. Each tool was applied individually to alter the metadata timestamps associated with the respective files. This approach was designed to assess how different timestamping utilities modify file system records and to evaluate the potential forensic challenges posed by such anti forensic techniques.

- During the analysis of the $MFT file extracted using the Autopsy tool, it was observed that each file entry contains three main attributes: the Standard Information (SI) attribute, the File Name (FN) attribute, and the Data attribute.
- Among these, both the Standard Information and File Name attributes each store four timestamps (Creation Time, Modification Time, MFT Modification Time, and Access Time) as shown in Fig 10.
- Upon examining the manipulated files, it was identified that the timestamping tools altered only the timestamps present in the Standard Information attribute.
- In contrast, the timestamps recorded in the File Name attribute remained intact, preserving the original creation and modification times.
- This discrepancy between the Standard Information and File Name attributes can be effectively used as a method to detect potential timestamp manipulation (time stomp- ing) during forensic investigations.
- However, relying solely on the comparison between these two attributes may not be sufficient for comprehensive detection of time stomped files, as sophisticated tools may alter both attributes and cause inconsistencies that are harder to detect.

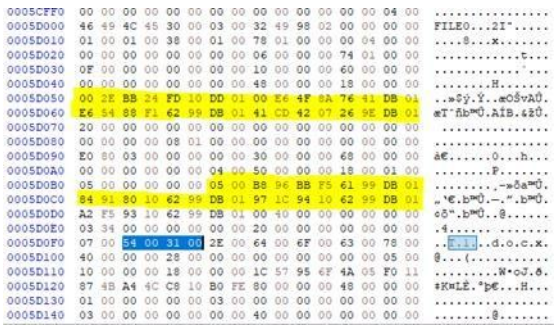

Fig -10: Timestamp obtained from HXD editor

3. CONCLUSIONS

As the use of anti-forensic techniques by cybercriminals continues to increase, it is important for forensic investigators to have a range of countermeasures to mitigate their impact[8]. The increasing sophistication of Anti forensic techniques continues to undermine the reliability of digital investigations by obscuring user actions and corrupting the evidentiary timeline. Anti-forensics techniques and tools are ever-changing and therefore, a solid set of forensic tools is required combating them [9] . The data wiping tools can completely delete OS artifacts. To overcome this limitation, many researchers have tried to use file system artifacts to identify data wiping traces.[10]. This work has proposed a method for generally identifying whether anti-forensic tools exist, and – in some cases – to what extent those tools have been used [11.].By integrating forensic analysis modules capable of parsing key NTFS artifacts—including $MFT, $Log File, and $UsnJrnl—with Prefect-based execution tracing, the proposed methodology provides a comprehensive mechanism for identifying suspicious deletion patterns and temporal inconsistencies. The correlation of findings across these diverse sources enables a more accurate reconstruction of events and exposes attempts to tamper with or eliminate digital traces. Experimental evaluations confirm that artifact-driven analysis remains a highly effective strategy in combating modern anti-forensic threats. Overall, this work contributes a robust and scalable framework that enhances the integrity, reliability, and depth of digital forensic examinations in the presence of deliberate evidence-manipulation techniques.

REFERENCES

[1] R. González Arias, J. Bermejo Higuera, J. J. Rainer Granados, J. R. Bermejo Higuera, and J. A. Sicilia Montalvo, "Systematic Review: Anti-Forensic Computer Techniques,"

[2] Justin Jacob1, Dr. Sandhya R2, "A Study of Anti Forensic Techniques And Their Mitigation Strategies"

[3] Jong-Hwa Song1, Hyun-Seob Lee2*" A Design of Timestamp Manipulation Detection Method using Storage Performance in NTFS"

[4] Antonio Savoldi*, Mario Piccinelli, Paolo Gubian, "A statistical method for detecting on disk wiped areas"

[5] E. J. Lee, S. Y. Lee, H. Kwon, S. J. Lee and G. B. Kim, "Identification of data wiping tools based on deletion patterns in ReFS $Logfile," in Proceedings of the Digital Forensic Research Conference (DFRWS APAC) 2023.

[6] Yeonsoo Kim†,Jewan Bang††,Jinkook Kim†,Sangjin Lee†††, "A Study of Trace for Data Wiping Tools"

[7] J. Oh, S. Lee, and H. Hwang, "Forensic Recovery of File System Metadata for Digital Forensic Investigation," School of Cybersecurity, Korea University, Seoul, South Korea, and The Affiliated Institute of ETRI, Daejeon, South Korea.

[8] Satvik Gurjar1, Dhaval Naik2, Aarti Sardhara3 , "Anti- Forensic Techniques and Its Impact on Digital Forensic"

[9] Emre Hosgor , "Detection and Mitigation of Anti- Forensics"

[10] Dong Bin Oh, Kyung Ho Park, Huy Kang Kim , "De- Wipimization: Detection of data wiping traces for investigating NTFS file system"

[11] Kyoung Jea Park,Jung-Min Park,Eun-jin Kim,Chang Geun Cheon,Joshua I. James "AntiForensic Trace Detection in Digital Forensic Triage Investigations"