Forensic Techniques to Detect Hidden Data in Alternate Data Streams in NTFS

Rahul Hermon
Computer Science and Engineering Department
Defense Institute of Advanced Technology
Pune, India
rahulhart@gmail.com,

Upasna Singh
Computer Science and Engineering Department
Defense Institute of Advanced Technology
Pune, India
upasnasingh@diat.ac.in,

Bhupendra Singh
Computer Science and Engineering Department
Indian Institute of Information Technology
Pune, India
bhupendra@iiitp.ac.in

***Abstract*— Alternate Data Streams (ADS) have been a feature of the New Technology File System (NTFS) since its introduction in 1993. Alternate Data Streams (ADS) were introduced to address compatibility within the existing Operating Systems. Lately Hackers/Cyber Criminals have used Alternate Data Streams (ADS) as a means for launching Cyber- Attacks. Alternate Data Streams (ADS) allow data hiding, same being difficult to detect without adequate knowledge. In this paper we shall bring out the various Forensic techniques in which hidden data in Alternate Data Streams (ADS) can be detected. Finally, we compared the Forensic techniques to detect data hidden in Alternate Data Streams (ADS) in both Windows 10 and 11 Operating System.**

***Index Terms*—Forensic Techniques, Data Hiding, New Technology File System (NTFS), Alternate Data Streams (ADS).**

I. INTRODUCTION

Alternate Data Streams (ADS) were introduced along with NTFS in Window NT 3.1. The same was a joint collaboration between Microsoft and IBM in 1993. NTFS is a file system in which every data is treated as a file. The heart of NTFS is the Master File Table (MFT). Master File Table (MFT) is index of every file on the entire volume, including itself. They contain record for every file and folder on the NTFS volume. MFT including NTFS system files are considered Metadata files and will begin with $ sign[2][5].

A. NTFS File Structure

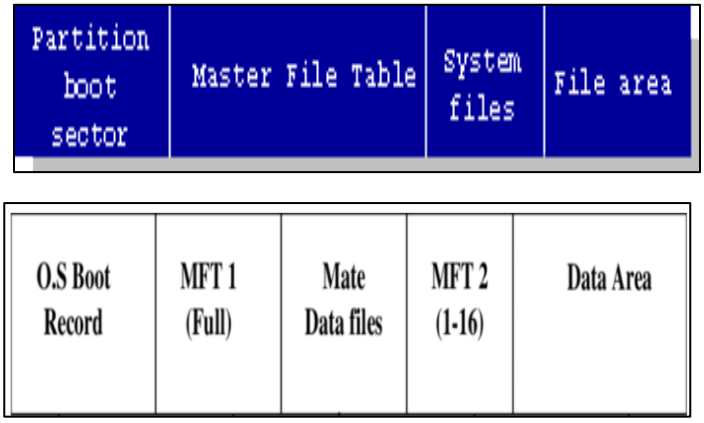

Fig. 1. NTFS File Structure

B. MFT File Structure

Each MFT record is 1 KB or 1024 bytes in size by default. MFT will create extent to fit in files of big size. The first 26 records are reserved for metadata. 12-23 are reserved for suture use and 24-26 are for transaction logs and error recovery. The file attributes, size, date/timestamps and permissions are saved in MFT entries. Each file has a header which are the first 56 bytes. Every file also has a signature. They are sequentially numbered. New records are created on a first available basis, overwriting is allowed[2].

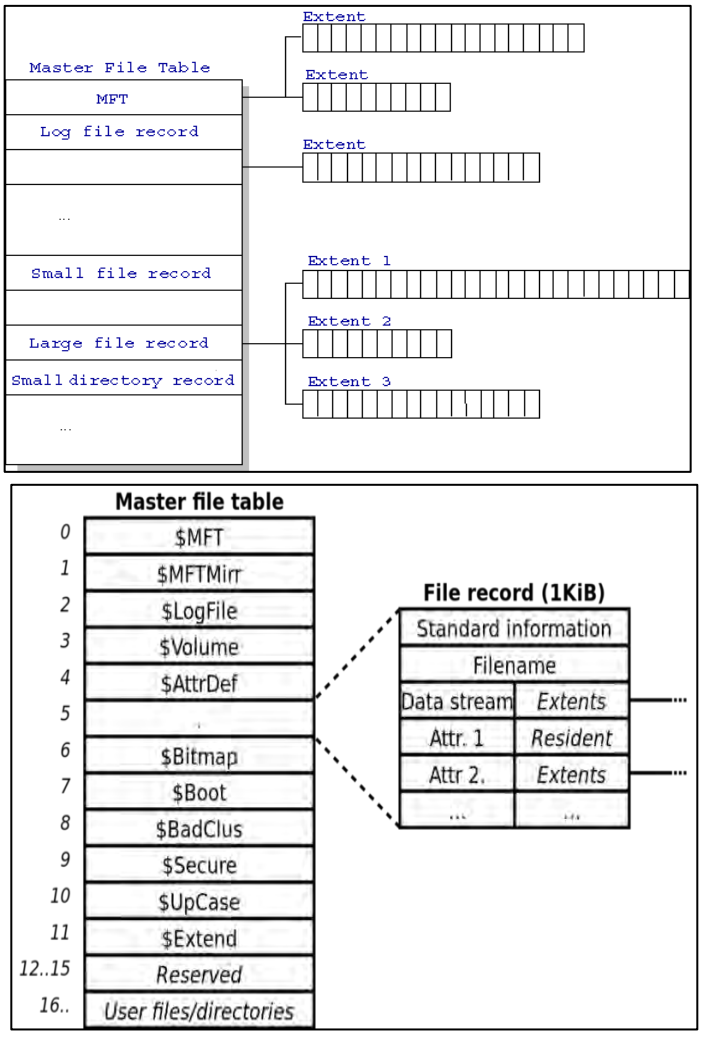

Fig. 2. MFT File Structure

This paper is organized as follows: Section II describes the concept of Alternate Data Streams (ADS) in NTFS. Section III brings out the methods to detect hidden data in Alternate Data Streams (ADS) using forensic techniques. Section IV compares method used in Section III for Windows 10 and 11. Section V talks about Conclusion and future work.

II. ALTERNATE DATA STREAMS (ADS)

Alternate Data Streams (ADS) are a Microsoft Substructure of NTFS. The main aim to launch Alternate Data Streams (ADS) was to take care of the inter-operability issues between various Operating Systems, especially Macintosh. They are not visible to maximum of the existing applications and remain hidden[1][4].

Alternate Data Streams (ADS) can be used to hide sensitive data by a user, to not be accessible to undesired users. Similarly Alternate Data Streams (ADS) can be used by malicious users to hide malicious scripts, files, worm etc. Lately they have been used to hide data for malicious purposes.

Data Hidden in Alternate Data Streams (ADS) have been used to launch Ransomware attacks in the USA and also launching attacks in several hospitals in Scotland. Maximum users are unaware of existence of Alternate Data Streams (ADS) in NTFS in Windows OS. This often is used as an advantage by Hackers/Cyber Criminals to launch Cyber-attacks on targeted systems[3].

BitPaymer was the Ransomware which exploited the Alternate data Streams (ADS) to launch Ransomware attacks. BitPaymer Ransomware used two Alternate Data Streams (ADS) to which the malware copied itself.

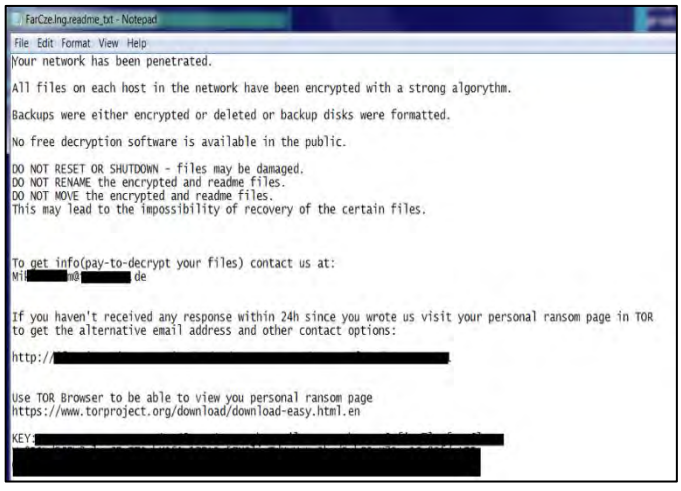

Fig. 3. BitPaymer Ransomware attack

Alternate Data Streams (ADS) are also used to launch Denial of Service Attack (DoS). They can be used to hide files the size of the existing space available on the volume. For example, if 300 GB space is available on a certain drive then around the same amount data can be hidden in the Alternate data Stream (ADS) in some particular folder in that drive. This would lead to inefficient functioning of the drive due to lack of space available. The drive will face lot of performance issues. In this way Alternate Data Streams (ADS) can also be used to launch Denial of Service Attacks (DoS)[1][3].

The interesting information about Alternate Data Stream (ADS) is that the ADS file size is not included in the default file size. The size of the file in whose ADS data is hidden will not change and will remain the same. The Alternate Data Streams (ADS) does not change a file's checksum. Hence due to all these characteristics of ADS, it is very difficult to detect it[4].

Data can be hidden in the Alternate Data Streams using Command Prompt, Windows PowerShell or Python language. In the above Command Prompt is the easiest way to hide data in ADS. The following command will be used once Command Prompt has been launched: -

Type VIDEO.mp4>>TEXT.txt: VIDEO.mp4

Using the above command, a video file can be hidden in the Alternate Data Streams (ADS) of TEXT.txt file. The name of the video file can be changed from video to anything desired by the attacker. Similarly different types of data files can be hidden in the ADS of an existing or newly created folder/file.

The Alternate Data Stream (ADS) in NTFS is by default called :$Data. Alternate Data Streams (ADS) are part of the latest Windows 11 OS. The Alternate Data Streams (ADS) are integral to NTFS only and do not exist in Linux or any other file systems. The only way to tackle Anti-Forensic element of ADS is to have adequate awareness and knowledge of ADS[2][4][5].

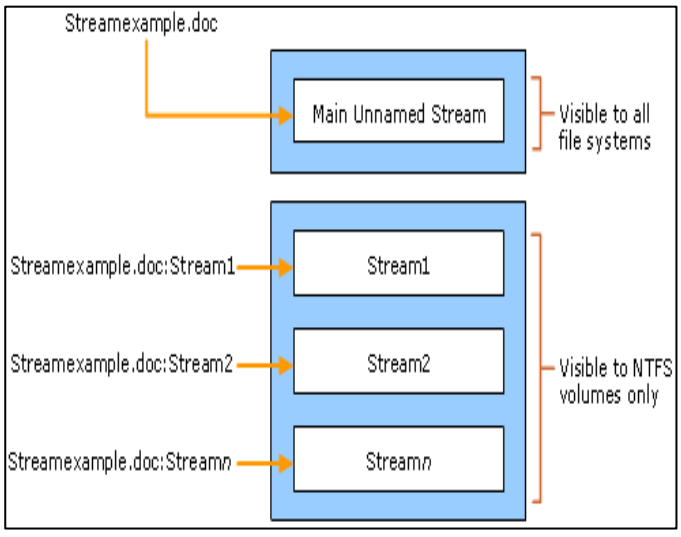

Fig. 4. File Structure of Alternate Data Streams (ADS)

III. FORENSIC TECHNIQUES TO DETECT HIDDEN DATA IN ALTERNATE DATA STREAMS (ADS)

Alternate Data Streams (ADS) as seen from Section II can very well be used to hide data and subsequently launch Cyber-attacks. To detect these hidden data in Alternate Data Streams (ADS) we require knowledge, awareness about same. In this section we will see the various Forensic techniques which can be used to detect the hidden data in ADS. Some of these methods are manual while some are tool based. Some of these methods not only detect the Alternate Data Streams (ADS) but also extract them. In this section we will carry out experiments for the same[4].

A. *Folder Properties*

In this method the user must know where to look for the Alternate Data Streams (ADS). If the user does not know where to look then this would become a tedious task, as this method involves looking at properties of every existing folder in the volume. Therefore, it would be of great advantage if the user already has some prior information about the folders/files to look into. In this method we will compare the size and size on disk, if there is a significant difference then that is a sign of Alternate Data Streams (ADS). The disparity between the actual size and the size on disk is a clear indicator of existence of Alternate Data Streams (ADS)[2][3][8][14][21].

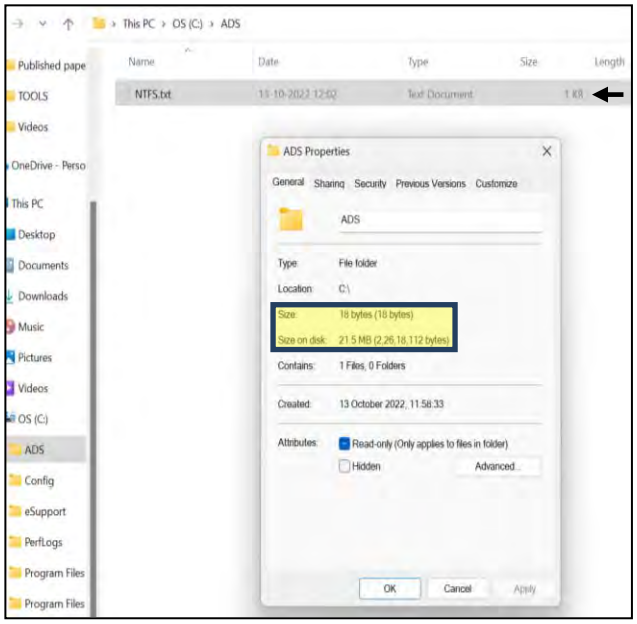

Fig. 5. Folder Properties

As seen from figure 5 there is a clear discrepancy between Size (18 bytes) and size on disk (21.5 MB). This is a clear indicator of presence of ADS. This method can only find out the folder or file containing ADS, it cannot ascertain the type of data file (text, video or image) or extract the same.

B. *Command Prompt*

In this method we will be using Command prompt to first detect the Alternate Data Streams (ADS) and then extract the same. This method also requires the user to have prior information about the folders/files to look into. In this method once the folder/file has been identified the Alternate Data Stream (ADS) will first be detected and its identity revealed and then the file will be extracted to a location as desired by the user. We will run Command Prompt as Administrator for implementation of this method.

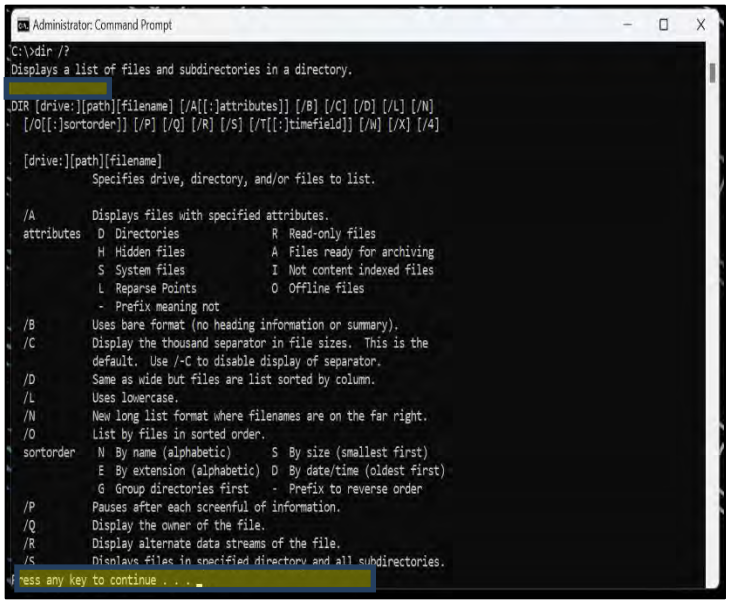

Fig. 6. Dir /? Command

As seen from figure 6, the Dir /? Command helps us in knowing that /R command will display the Alternate Data Streams (ADS) of any folder/file. The command Dir /R can be used on any folder/file which is suspicious of having an ADS.

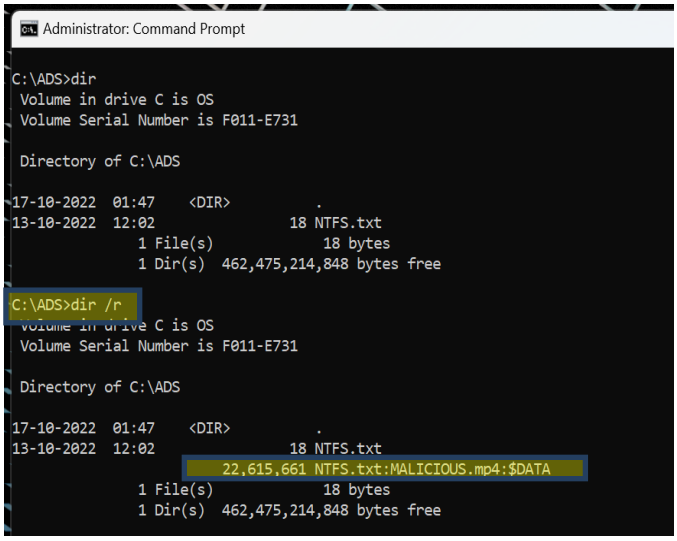

Fig. 7. Dir /R Command

As seen from figure 7, the simple Dir command does not display the ADS but the Dir /R command clearly displays the ADS in form of NTFS.txt: MALICIOUS.mp4 $DATA (21.5 MB) which is a video file and was being hidden in the Alternate Data Streams (ADS) of NTFS.txt.

The video file being hidden in the Alternate Data Streams (ADS) has been detected and also been identified. To extract the same, we will use the following command in command prompt: -

expand NTFS.txt: MALICIOUS.mp4 VIDEO.MP4

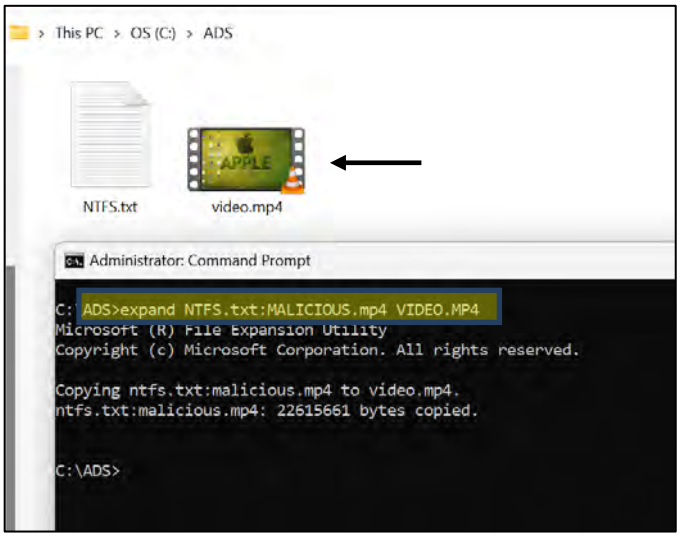

Fig. 8. Extracting File from ADS

The extracted file has been renamed as VIDEO.mp4, although the existing file name can also be used. As seen from figure 8 the video file has been extracted successfully in the ADS folder itself. Hence this is how a hidden file in ADS is detected, identified and extracted using command prompt. The only drawback being pinpoint information is required about the suspicious folders/files otherwise this method would also be time consuming.

C. PowerShell

PowerShell is an internal feature of Windows designed by Microsoft. It was designed for system administration. PowerShell can also be used to detect Alternate Data Streams (ADS). Certain commands can be used to detect the hidden files in the ADS. When the suspected folder/file is not known then we will use the following command: -

gci -recurse | % { gi $_.FullName -stream * } | where stream -ne ':$Data'

The above command when used for a particular volume will display all existing Alternate Data Streams (ADS) on the particular volume. GCI is the Get-ChildItem, it gets child items inside a container (directory). This particular command can only be used to detect the Alternate Data Streams (ADS). If we know the particular folder/file and details of the hidden data, then we use another command to bring out the details of the hidden data in the Alternate Data Streams (ADS). The following command will be used

Get-Item -path .\NTFS.txt:MALICIOUS.mp4

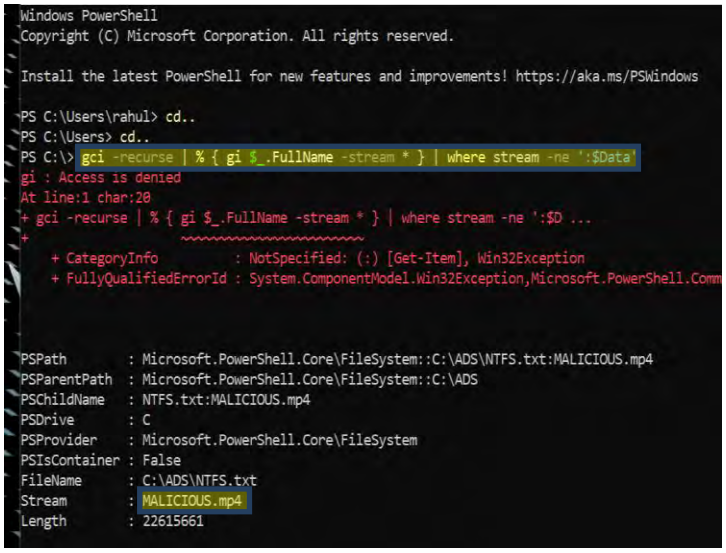

Fig. 9. Get-ChildItem(GCI) Command

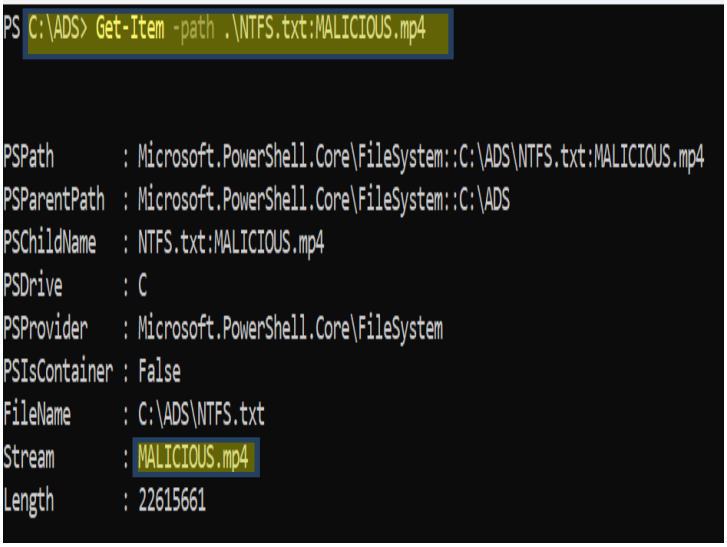

Fig. 10. Get-Item Command

As seen from figure 9 and figure 10 both commands are able to detect and bring out the details of the hidden data in Alternate Data Streams (ADS).

D. Streams

Streams was developed by Sysinternals and was eventually purchased by Microsoft and is available for download in the Microsoft website. Streams was developed to detect Alternate Data Streams (ADS). Streams uses simple commands to detect the Alternate Data Streams (ADS) in the volume as desired by the user. Streams can only detect the hidden data in the ADS but cannot extract it. Streams can be run using Command Prompt. To use Streams, we have to enter the folder where streams is installed and then run the commands in the Command Prompt. Streams can also be used to delete the Alternate Data Streams (ADS) of a particular folder/file.

The following commands will be used: -

streams.exe -s C:\ADS (if the folder/file is known)

streams.exe -s D:\ (To search entire volume if folder/file not known).

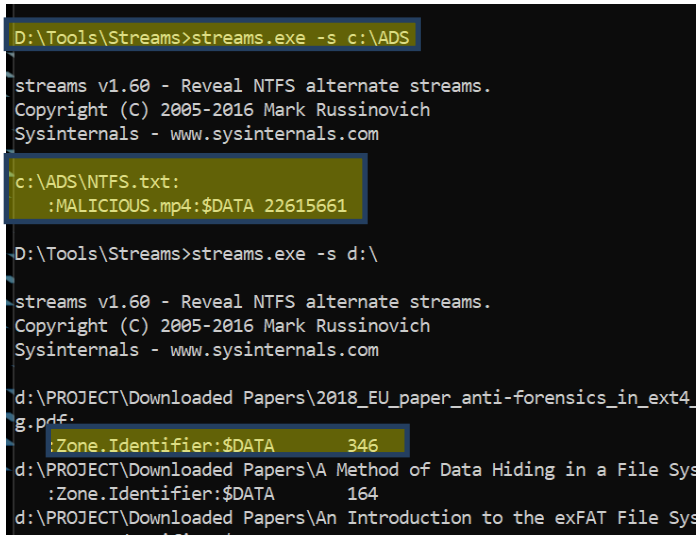

Fig. 11. Streams command when ADS is known/unknown

E. Freeware tools

1. ADS Manager – It is a freeware available on the internet. It was developed to detect, extract, create, rename, modify and delete the Alternate Data Streams (ADS) on existing volumes. It is a simple and user-friendly tool which provides results in very less time.

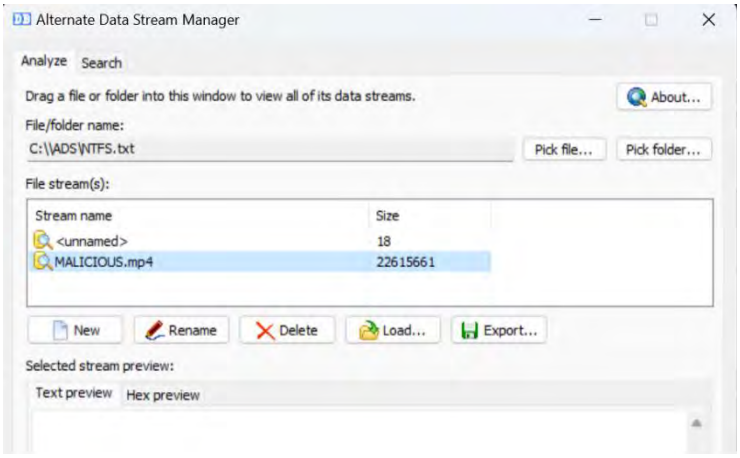

Fig. 12. Alternate Data Stream Manager

2. ADS Spy v1.11 – It is a freeware available on the internet. It was developed to detect and delete the Alternate Data Streams (ADS) on existing volumes. The extraction process is complicated as the freeware converts the detected ADS into binary form and only then the user can save the particular file. Hence for a large video file hidden in the ADS, the same will be very time consuming. This freeware is most efficient for detecting and deleting the Alternate Data Streams (ADS).

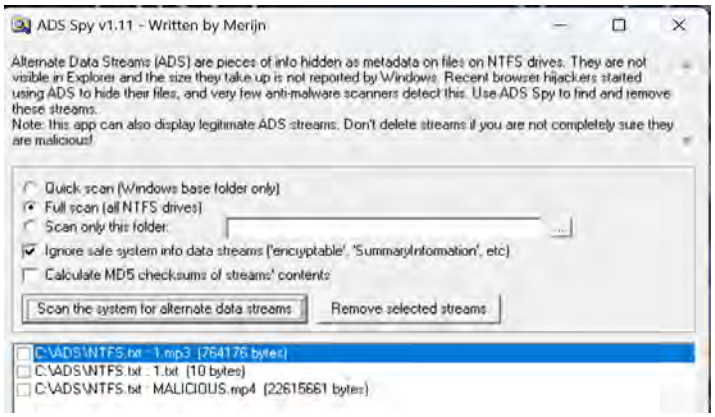

Fig. 13. ADS Spy v1.11

3. AlternateStreamView – It is a freeware available on the internet. It was developed to detect, extract, and delete the Alternate Data Streams (ADS) on existing volumes. It can be run without being installed. It is a user-friendly tool which utilizes very minimal CPU memory.

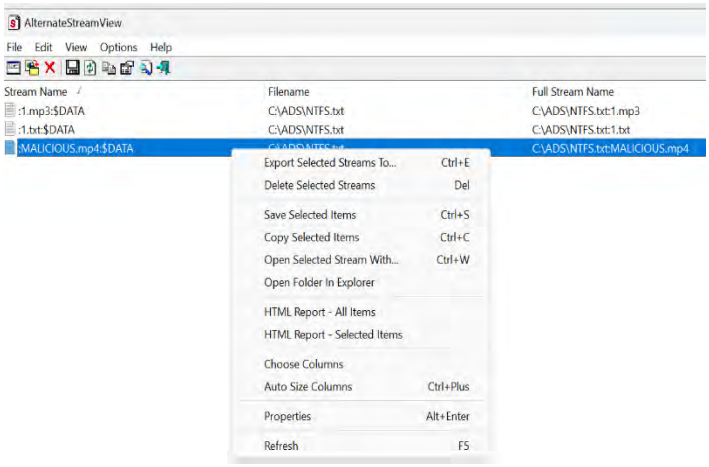

Fig. 14. AlternateStreamView

4. NoVirusThanks Stream Detector – It is a freeware available on the internet. It was developed to detect, extract, and delete the Alternate Data Streams (ADS) on existing volumes. It can also export the scan report onto a log file. It is very light in memory and CPU usage.

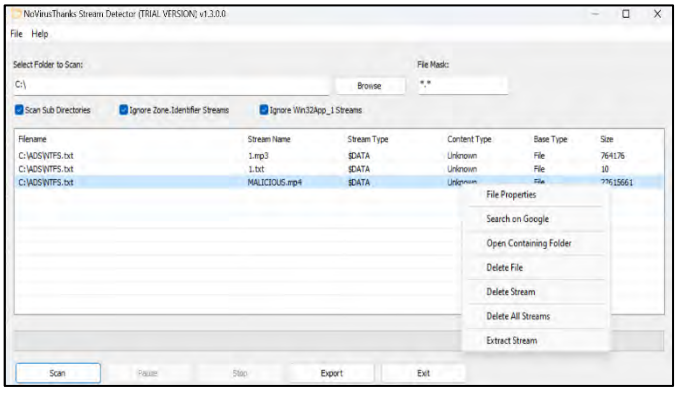

Fig. 15. NoVirusThanks Stream Detector

IV. WINDOWS 10 VS WINDOWS 11

All the methods seen in Section III were executed and implemented on both Windows 10 and Windows 11. The following was observed when the experiments were carried out on both OS: -

1) All Command/Syntax produced the same results for both Windows 10 and Windows 11 without any change in Command/Syntax.

2) All the freeware tools were run on both Windows 10 and Windows 11. All tools were able to execute on both the OS. All tools produced the same results in both the OS.

Hence it is safe to conclude that Forensic Techniques for both Windows 10 and Windows 11 remain the same when it comes to detect hidden data in the Alternate Data Streams (ADS) in NTFS.

V. CONCLUSION AND FUTURE WORK

As seen from the above experiments, there are many ways to detect hidden data in Alternate Data Streams (ADS). This paper brings out methods both manual and tool based. Implementing these methods might not seem complicated as most of the Commands/Syntax and tools are easy to use and execute. The major problem lies in the lack of awareness and knowledge of users when it comes to Alternate Data Streams (ADS).

Alternate Data Streams (ADS) in the past have been used for Cyber-attacks and also Denial-of-Service (DoS) attack. They will continue to be used to launch these types of Cyber-attacks in future also. It is the individual user who has become aware about the Alternate Data Streams (ADS) and its capability. Once the user is aware then he can use both Commands/Syntax and the freeware tools to easily detect the malicious hidden data.

There are many commercial tools also available which can be used to detect the Alternate Data Streams (ADS) as brought out by the paper. Although that is not the primary task of these tools, only an added feature. The tools we have used in Section III have been developed with the sole purpose of detecting/extracting/deleting Alternate Data Streams (ADS). Only four tools have been demonstrated and other such tool do exist which can detect the Alternate Data Streams (ADS) but not as their primary and only task.

This paper clearly brings out how Alternate Data Streams (ADS) can be exploited for Cyber-attacks and how the same can be avoided by using Commands/Syntax and tools which help detect the same. In future there might be a version of Windows OS in which Alternate Data Streams are detectable by default. The same will then ensure there is no requirement of using any of the above methods. But till such time it important that these methods be known to users for prevention against Cyber-attacks.

VI. REFERENCES

[1] N. A. Hassan and R. Hijazi, “Data Hiding Techniques in Windows OS”, Elsevier, 2017.

[2] B.M. Shafiei, Farshid Iranmanesh and Fariborz Iranmanesh,” Review NTFS Basics,” Australian Journal of Basic and Applied Sciences, 6(7): 325-338, 2012.

[3] M. Broomfield, "NTFS Alternate Data Streams: focused hacking," in Network Security, Elsevier, vol. 2006, pp. 7-9, August 2006.

[4] E. Huebner, D. Bem and C.K. Wee, “Data hiding in the NTFS file system,” Digital Investigation, Elsevier, vol. 3, pp. 211-226, 2006.

[5] B. Carrier, “File System Forensic Analysis,” Addison-Wesley, Upper Saddle River, NJ, 2005.

[6] C. Altheide, H. Carvey, “Digital Forensics with Open Source Tools,” Syngress, 2011.

[7] E. Casey, “Handbook of Digital Forensics and Investigation,” Academic Press, 2010.

[8] A.I. Martini, A. Zaharis and C. Ilioudis, "Detecting and Manipulating Compressed Alternate Data Streams in a Forensics Investigation," *2008* Third International Annual Workshop on Digital Forensics and Incident Analysis, 2008, pp. 53-59, doi: 10.1109/WDFIA.2008.9.1

[9] H. Carvey, “Windows Forensic Analysis Toolkit,” Syngress, 2014.

[10] Tariq Bin Azad,” Securing Citrix Presentation Server in the Enterprise,” Syngress 2008.

[11] B. Wotring and B. Potter, “Host Integrity Monitoring Using Osiris and Samhain,” Syngress, 2005.

[12] R.L. Means, “Alternate Data Streams: Out of the Shadows and into the Light,”SANS Institute, 2003:http:??www.sans.org/reading room/whitepapers /honors/1503.php

[13] R. Mahajan, S. Miglani, M. Singh “ADS: Protecting NTFS From Hacking,” Recent Advances and Innovations in Engineering, IEEE, pp. 1-4, 2014.

[14] Sameer H. Mahant, B.B. Meshram, 2012, ADS Examiner: Tool for NTFS Alternate Data Streams Forensics Analysis, International Journal of Engineering Research & Technology (IJERT) Volume 01, Issue 04 (June 2012).

[15] Da-Yu Kao, Yuan-Pei Chen, Neng-Hsin Shih, “Reconstructing ADS data hiding in windows NTFS: A temporal analysis” in Digital Investigation, Volume 26, Supplement,2018, Page S137.

[16]A. Castiglione, A. De Santis and C. Soriente, “Taking advantages of a disadvantage: Digital forensics and steganography using document metadata,” Journal of Systems and Software, vol. 80, no. 5, pp. 750-764, May 2007.

[17]Michael T. Raggo and Chet Hosmer, Data Hiding: Exposing Concealed Data in Multimedia, Operating Systems, Mobile Devices and Network Protocols, Elsevier, 2013.

[18]K. Eckstein and M. Jahnke, ―Data Hiding in Journaling File Systems‖, Proceedings of Digital Forensic Research Workshop (DFRWS), pp. 1-8, 2005.

[19]G.-S. Cho, ―A New NTFS Anti-Forensic Technique for NTFS Index Entry,‖ The Journal of Korea Institute of Information, Electronics, and Communication Technology (ISSN 2005-081X), vol. 8, no. 4, 2015.

[20]F. A. P. Petitcolas, R. J. Anderson, AND M. G. Kuhn, ―Information Hiding—A Survey,‖ Proceedings of the IEEE, Vol. 87, No. 7, pp. 1062-1078, July 1999.

[21]H. Berghel, and N. Brajkovska. "Wading into alternate data streams." ACM, vol. 4, no. 4, April 2004.

[22]I. Thompson, and M. Monroe, “FragFS: an advanced data hiding technique”, BlackHat Federal. Jan. 2006.

[23] A. Srinivasan, S. Kolli, and J. Wu, "Steganographic information hiding that exploits a novel file system vulnerability," Int. J. Security and Networks, Vol. 8, No. 2, Aug. 2013.

[24] Gyu-Sang Cho, “A New NTFS Anti-Forensic Technique for NTFS Index Entry,” The Journal of Korea Institute of Information, Electronics, and Communication Technology, Vol. 8, No. 4, pp. 327-337, Aug. 2015.

[25] Liang Jinqian;Zhang Yue, “The Main Data Structure of NTFS File System,”Computer Engineering and Applications[J],2003, pp.116-130.