Windows Fundamentals

The file system used in modern versions of Windows is the New Technology File System or simply NTFS.

Before NTFS, there was FAT16/FAT32 (File Allocation Table) and HPFS (High Performance File System).

You still see FAT partitions in use today. For example, you typically see FAT partitions in USB devices, MicroSD cards, etc. but traditionally not on personal Windows computers/laptops or Windows servers.

NTFS is known as a journaling file system. In case of a failure, the file system can automatically repair the folders/files on disk using information stored in a log file. This function is not possible with FAT.

NTFS addresses many of the limitations of the previous file systems; such as:

- Supports files larger than 4GB
- Set specific permissions on folders and files
- Folder and file compression
- Encryption (Encryption File System or EFS)

Let's speak briefly on some features that are specific to NTFS.

On NTFS volumes, you can set permissions that grant or deny access to files and folders.

The permissions are:

- Full control
- Modify
- Read & Execute
- List folder contents
- Read
- Write

The below image lists the meaning of each permission on how it applies to a file and a folder

Permission	Meaning for Folders	Meaning for Files
Read	Permits viewing and listing of files and subfolders	Permits viewing or accessing of the file's contents
Write	Permits adding of files and subfolders	Permits writing to a file
Read & Execute	Permits viewing and listing of files and subfolders as well as executing of files; inherited by files and folders	Permits viewing and accessing of the file's contents as well as executing of the file
List Folder Contents	Permits viewing and listing of files and subfolders as well as executing of files; inherited by folders only	N/A
Modify	Permits reading and writing of files and subfolders; allows deletion of the folder	Permits reading and writing of the file; allows deletion of the file
Full Control	Permits reading, writing, changing, and deleting of files and subfolders	Permits reading, writing, changing and deleting of the file

Refer to the Microsoft documentation to get a better understanding of the NTFS permissions for Special Permissions .

Another feature of NTFS is Alternate Data Streams (ADS).

Alternate Data Streams (ADS) is a file attribute specific to Windows NTFS (New Technology File System).

Every file has at least one data stream ($DATA), and ADS allows files to contain more than one stream of data. Natively Windows Explorer doesn't display ADS to the user. There are 3rd party executables that can be used to view this data, Powershell also gives you the ability to view ADS for files. We will cover how you can use PowerShell to view any ADS for any files in the Windows PowerShell room.

From a security perspective, malware writers have used ADS to hide data.

Not all its uses are malicious. For example, when you download a file from the Internet, there are identifiers written to ADS to identify that the file was downloaded from the Internet.

What are Alternate Data Streams?

Alternate Data Streams (ADS) are a file attribute only found on the NTFS file system.

In this system a file is built up from a couple of attributes, one of them is *$Data*, aka the data attribute. Looking at the regular data stream of a text file there is no mystery. It simply contains the text inside the text file. But that is only the primary data stream.

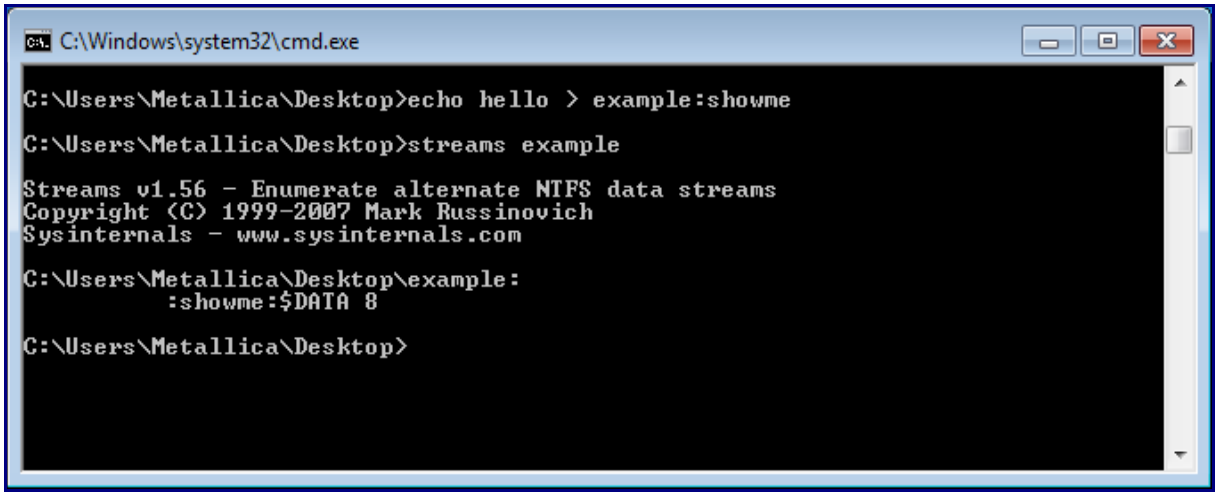
This one is sometimes referred to as the unnamed data stream since the name string of this attribute is empty (“”) . So any data stream that has a name is considered alternate.

These data streams suffer from a bad reputation since they have been used and abused to write hidden data. Varying from data about where a file came from to complete malware files.

If you are up for an experiment, we can easily create and read an alternate data stream.

Streams

The first tool you can use was developed by Sysinternals (later bought by Microsoft) and is called Streams *(nomen est omen)*.



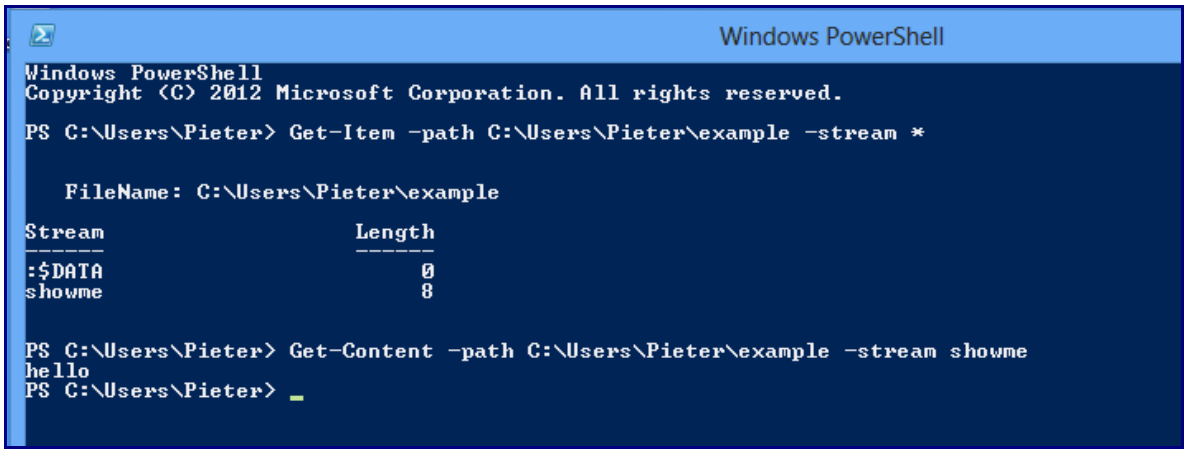
In the example above we used the echo command to create an empty file called example with an alternate data stream called showme.

By using streams we can check which files have alternate data-streams. In the results visible in the above command prompt, *$Data* is the name of the attribute (as discussed earlier) and the 8 tells us the size.

But since we are looking at it, we obviously would like to see what is inside the alternate data streams. Unfortunately, streams do not offer that option.

Get-Item

If you are using Windows 8 (or newer) there is a built-in option to read ADS. You can use PowerShell commands to achieve this. For those that have no experience with it, you can start it by typing PowerShell in the Run box (Windows key + R) and follow the lines in this screenshot.



Set-item

Another thing that you can do with Powershell is add streams to a file. The Powershell command syntax is:

set-content - path {path to the file} - stream {name of the stream}

Doing so will initiate a cmdlet where you can enter the content of the stream under Value[i]

Search for ADS

If you want to search a directory or drive for ADS you can use this command in the root of the target:

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

Be warned that if you include the Windows directory in your search you will likely receive an enormous list.

Remove ADS

A word of warning here. Removing ADS is not always advisable. Some of them are needed for the proper use of the software that created the streams. So make sure you have done your research before removing them. The syntax is:

remove-item –path {path to the file} –stream {name of the stream}

Malwarebytes Anti-Malware scans for and removes unwanted ADS (as Rootkit.ADS)

Summary

Alternate Data Streams (ADS) have been given a bad reputation because their capability to hide data from us on our own computer, has been abused by malware writers in the past. Hopefully this article will clear up some of the questions and mystique you had about ADS.