# How NTFS Alternate Data Streams Introduce Security Vulnerability

blog.netwrix.com/2022/12/16/alternate_data_stream

Joe Dibley

You may not be familiar with NTFS file streams, but you use them every day when you access files on any modern Windows system. This blog post explains this feature of NTFS ADS, shows how hackers can exploit file stream functionality in cyberattacks, and offers strategies for defending your organization.

Handpicked related content:

NTFS Permissions Management Best Practices

## Default Data Streams and Alternate Data Streams

A file stream is a sequence of bytes that contains data about a file, such as keywords or the identity of the user who created the file. Think of a data stream as a file within a file — a hidden file residing within a legitimate one. Each stream has its own disk space allocation, its own actual size (bytes in use) and its own file locks.

Every file in your NTFS file structure has at least one stream, its default stream. The **default data stream** is the normal, viewable file content — for example, the text in a .txt file or the executable code in a .exe file. This information is stored in the $Data attribute. Because the name of this default attribute is empty (set to “”), the default data stream is also often referred to as the “**unnamed data stream**”.

Files can also contain one or more **alternate data streams** (ADSs). An ADS must be named. Note that the default data stream remains unchanged with the addition of alternate data streams.

## How to Create Alternate Data Streams

It is quite easy to create alternate data streams for a file: Simply append a colon (“:”) to the file name or path, followed by the stream name. Since the colon is a reserved character not allowed in a filename, it doesn’t conflict with existing file names.

You can add multiple ADSs to a file. For example, here is how we can create two alternate data streams for a text file:

Myfile.txt:stream2

Myfile.txt:secretstuff

## Benefits of NTFS File Streams

While older Windows file systems such as FAT16 and FAT32 have no support for multiple data streams, ADS is not a new technology; it has been present in all versions of Microsoft’s NTFS file system since Windows NT.

Early on, the use of multiple data streams helped to enable a Windows server to also serve as a file server for Apple Macintosh computers. Mac files make use of two streams per file — one for data and one for resource information. With NTFS supporting multiple streams, a Mac user could copy files to a Windows server and then back to a Mac without losing the resource stream. In other words, the ADS was able to provide compatibility for both systems and their applications.

There are also legitimate reasons to utilize alternate data streams within Windows. For example, some archive management and backup software use ADS to store file revision information, and many web browsers add a stream to files downloaded from the internet that includes security information about where the file came from.

## The Sinister Side of ADS

While ADS has many legitimate purposes, hackers can misuse it for malicious purposes such as malware attacks. Like a secret compartment inside a suitcase used by a smuggler to hide contraband from an inspector, ADS can be used by threat actors to hide malicious code and execute future attacks while skirting basic security detection. An

ADS can store any type of file, including audio, video, images or malicious code such as viruses, trojans and ransomware. And because alternate data streams are hidden, users cannot detect them using directory listing commands.

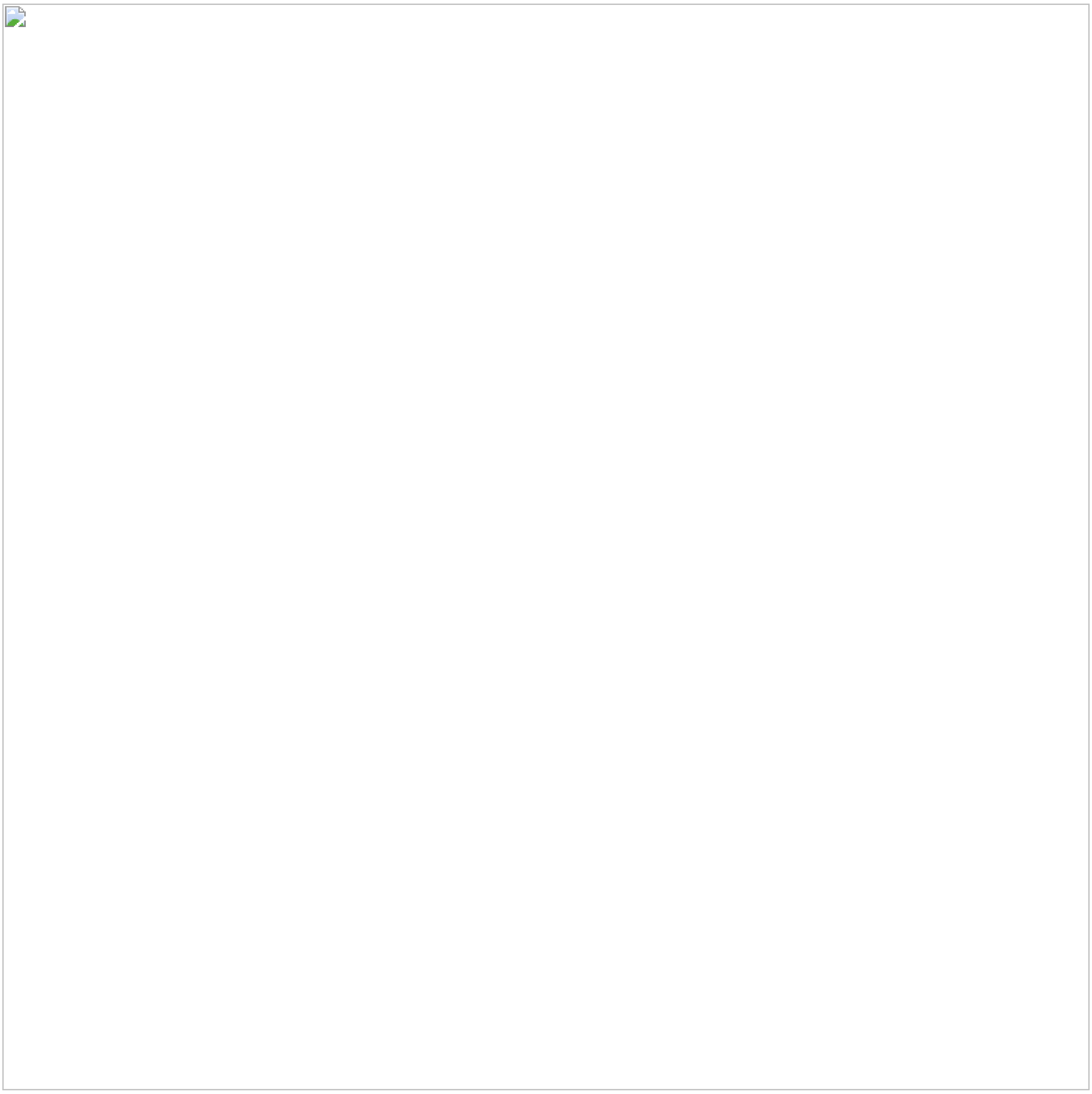
## Tools to Work with NTFS Streams

There are a few native tools you can use to gain more visibility into ADS. These include:

- Echo and More
- The Sysinternals Streams utility
- The /R option of the Dir command.
- PowerShell 3.0, which includes six cmdlets to directly manipulate content for ADS

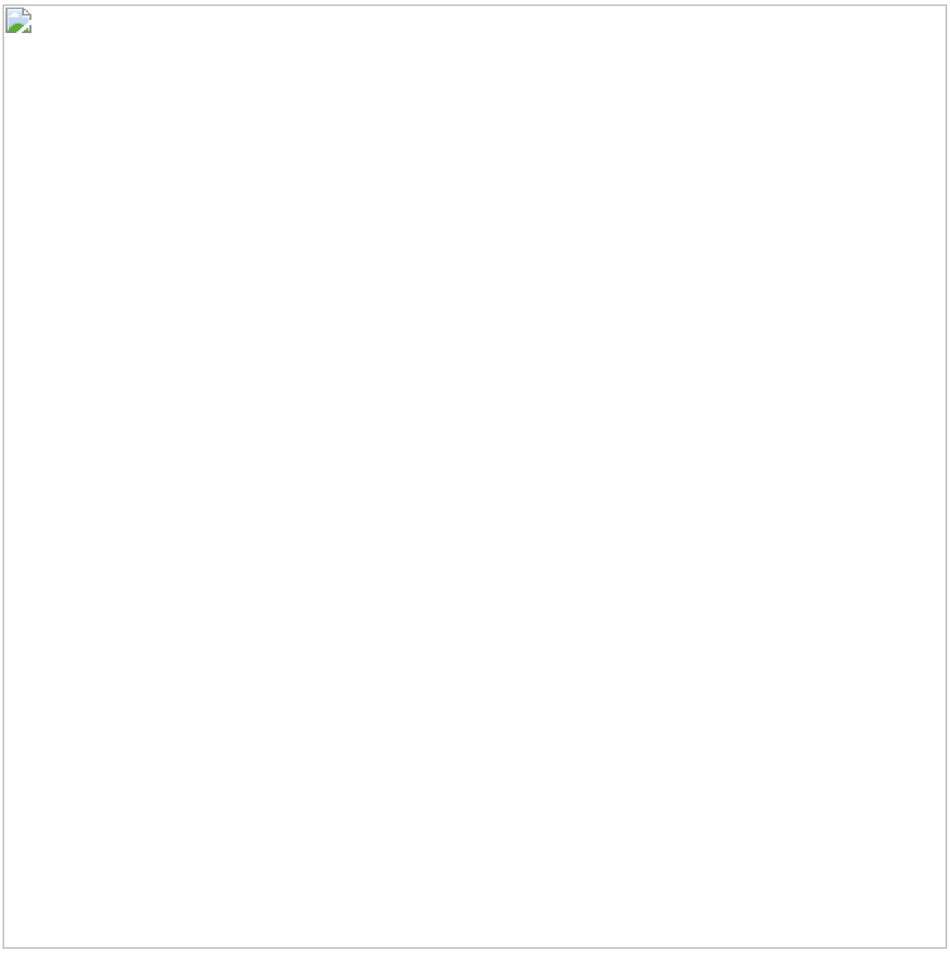
### Echo and More

Let’s start by using Echo and More. In the example below, the More command is used to append “:secret” to a text file called test.txt, and the Echo command is used to write a secret message that cannot be viewed by default. Notice that the Dir command doesn’t see the secret NTFS stream, either.



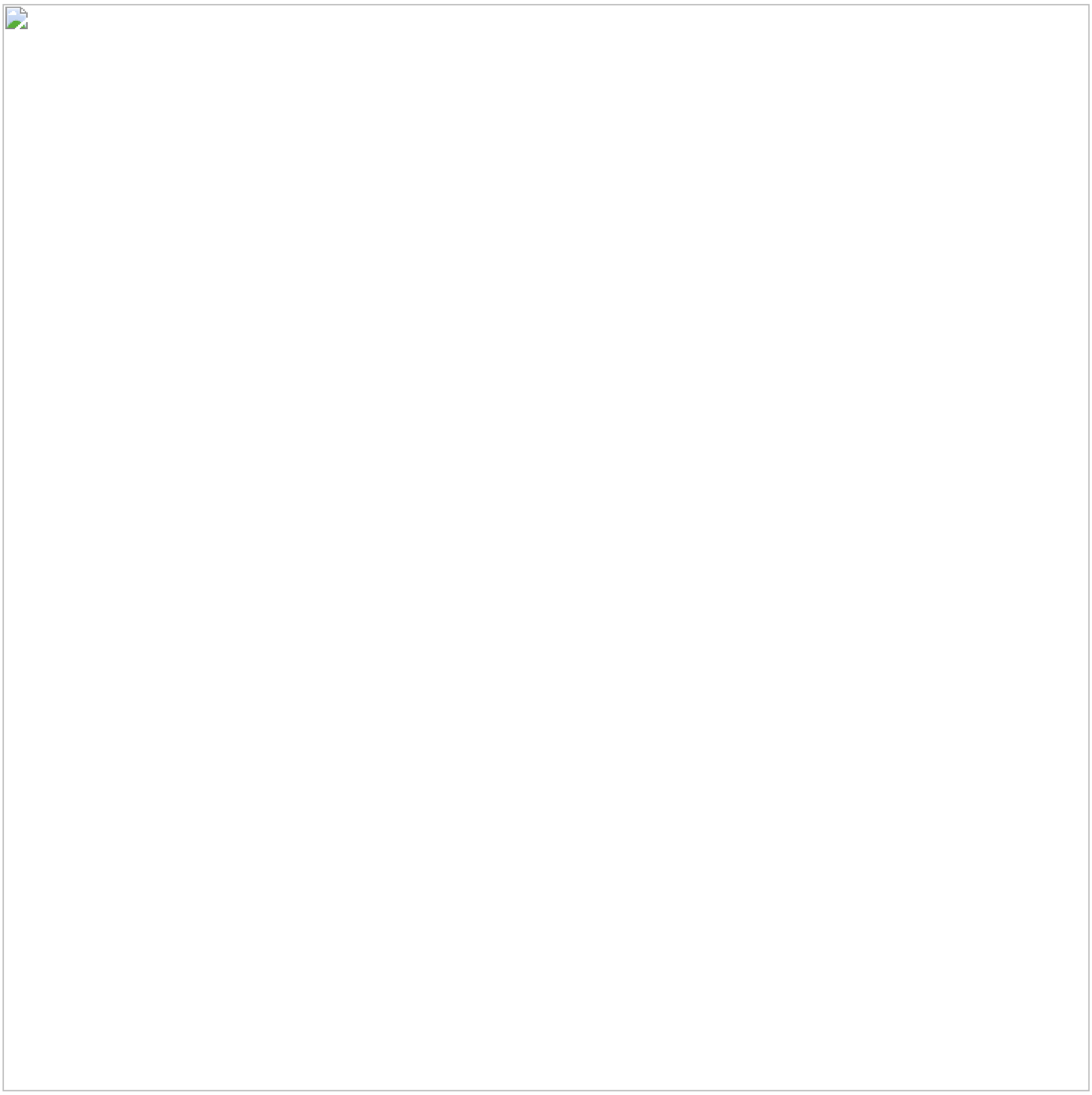
### Streams

Streams is a command-line tool available from Sysinternals. It is used to show which files in a folder use streams beyond the default data streams. The screenshot below shows that the file test.txt has an alternate stream named “secret” which has a file size of 86 bytes. Note that this is far more than the 26 bytes shown by the Dir command in the preceding example.



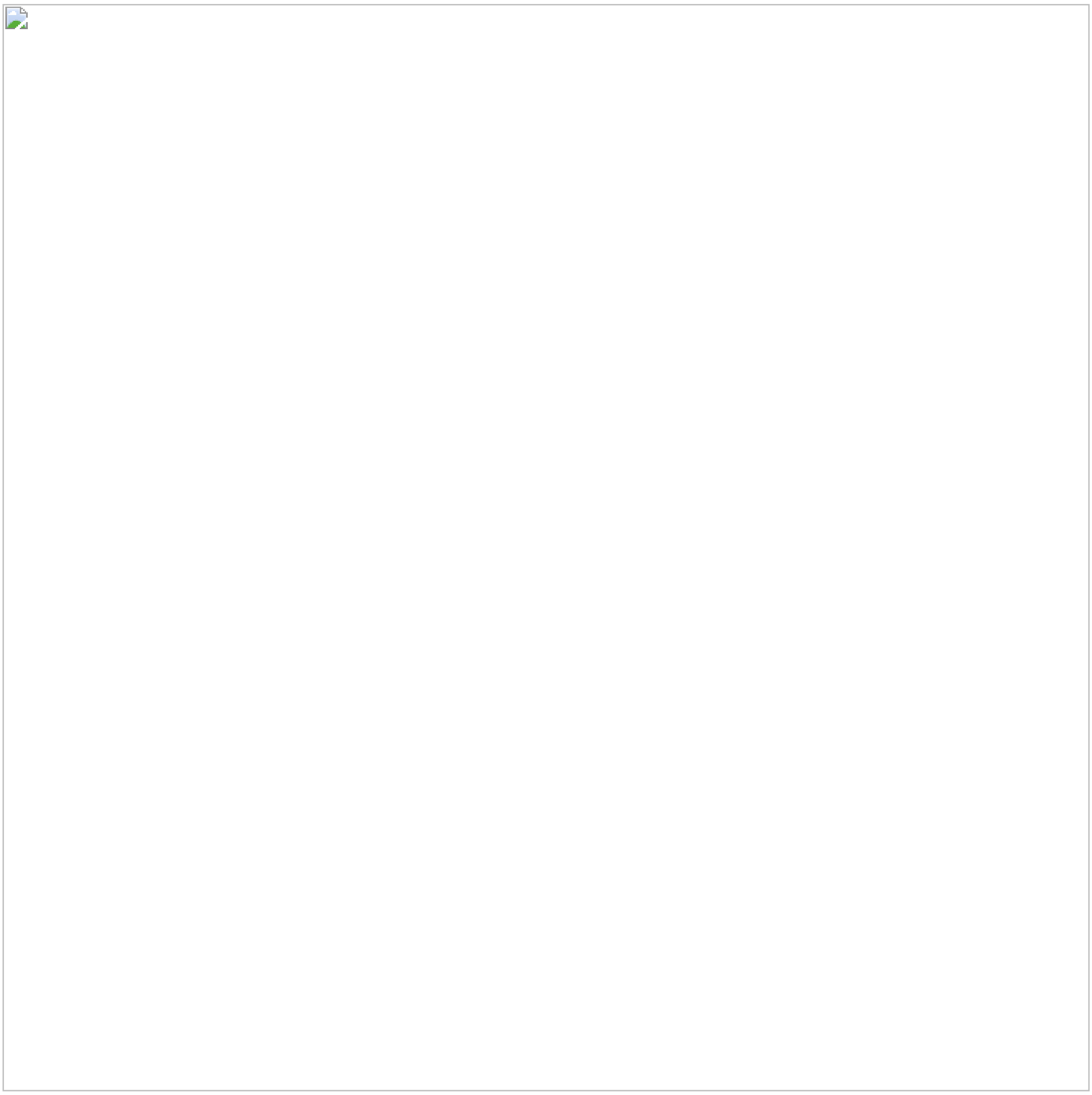
## Dir /R

The Dir /R option has been available since Windows Server 2003. As shown below, our file ‘test.txt’ appears twice when using the Dir /R option. It also shows the correct file size for both the default and secret file streams.



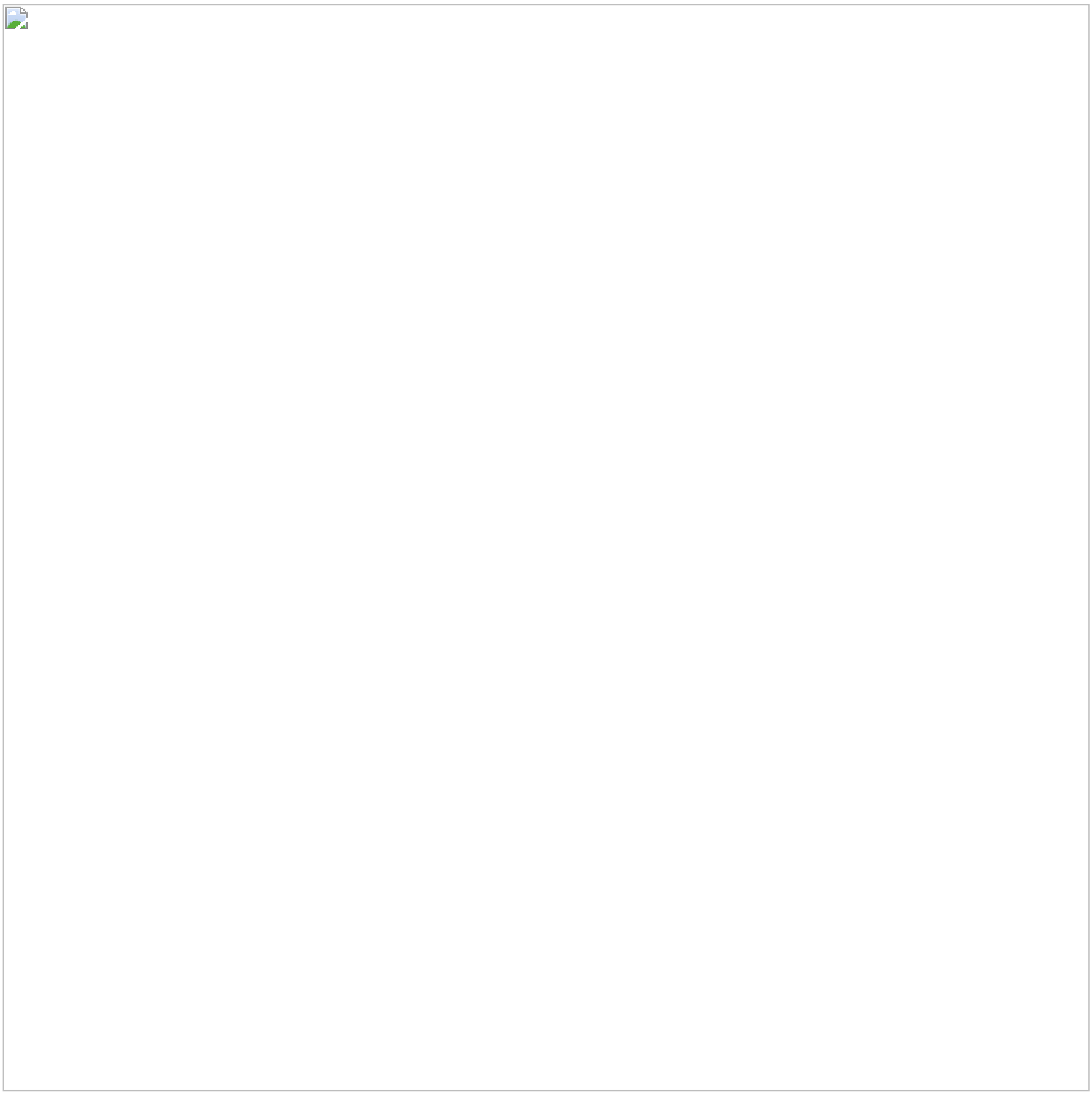
## PowerShell

You can also use PowerShell to identify the alternate data streams in a file. In the example below, we have used the command Get-Item with the Stream option and the wildcard parameter. The output shows both the streams for our file; the arrow highlights the alternate stream view.



You can also use PowerShell to clear an NTFS stream. In the screenshot below, we have used the clear-content command to delete the data associated with the secret data stream. Running the get-item command immediately afterwards confirms that the data in the stream was deleted, since the file size is now zero.

We can even do one better and delete the stream completely by using the remove-item command, as shown below. The get-item command confirms that the stream was deleted.



## How to Defend Against the ADS Threat

Adversaries can and do use ADS to hide malicious content, including ransomware and other malware, in your hierarchical file structure. Unfortunately, Windows File Explorer, the Dir command and related tools report information about a file’s default data stream only. The truth is that a simple .txt file or Word doc reporting 1k of data could actually encase megabytes of hidden data or executable code in an ADS.

Accordingly, when building your security management strategy, you need to improve visibility into ADS. In particular, consider investing in anti-virus products, data discovery tools, and data exfiltration detectors that can detect the existence of alternate data streams and scan for unauthorized content so you can proactively remove it. After all, incident prevention is always better than incident response.

Joe Dibley
Security Researcher at Netwrix and member of the Netwrix Security Research Team. Joe is an expert in Active Directory, Windows, and a wide variety of enterprise software platforms and technologies, Joe researches new security risks, complex attack techniques, and associated mitigations and detections.