Alternate Data Streams (ADS)

Did you know that the New Technology File System (NTFS) includes support for ADS? This is not a well known feature and was designed to provide compatibility with the old Hierarchical File System (HFS) from Mac.

Every file has at least one data stream and ADS allows files to contain more than one stream. The default data stream in Windows is “$DATA”. Additional streams will be tied to various metadata such as, timestamp, security descriptors, or file name.

However, Bill Gates may have had the best of intentions but in the end ADS is more commonly associated with malware than mac. This is because when you use ADS the data becomes fairly hidden unless you know what or how to look for it

Ok, now that we know a bit more on ADS, lets learn how to find it and maybe a bit more.

First lets go ahead and create make a file with the text “Normal File” inside.

```
C:\Users\Chris\Downloads\ADS>echo Normal File > file_normal.txt

C:\Users\Chris\Downloads\ADS>
```

Ok, now lets put the text “Evil Malware” in the ADS of badfile.txt

```
C:\Users\Chris\Downloads\ADS>echo Evil Malware > badfile.txt:hiddenfile.txt

C:\Users\Chris\Downloads\ADS>
```

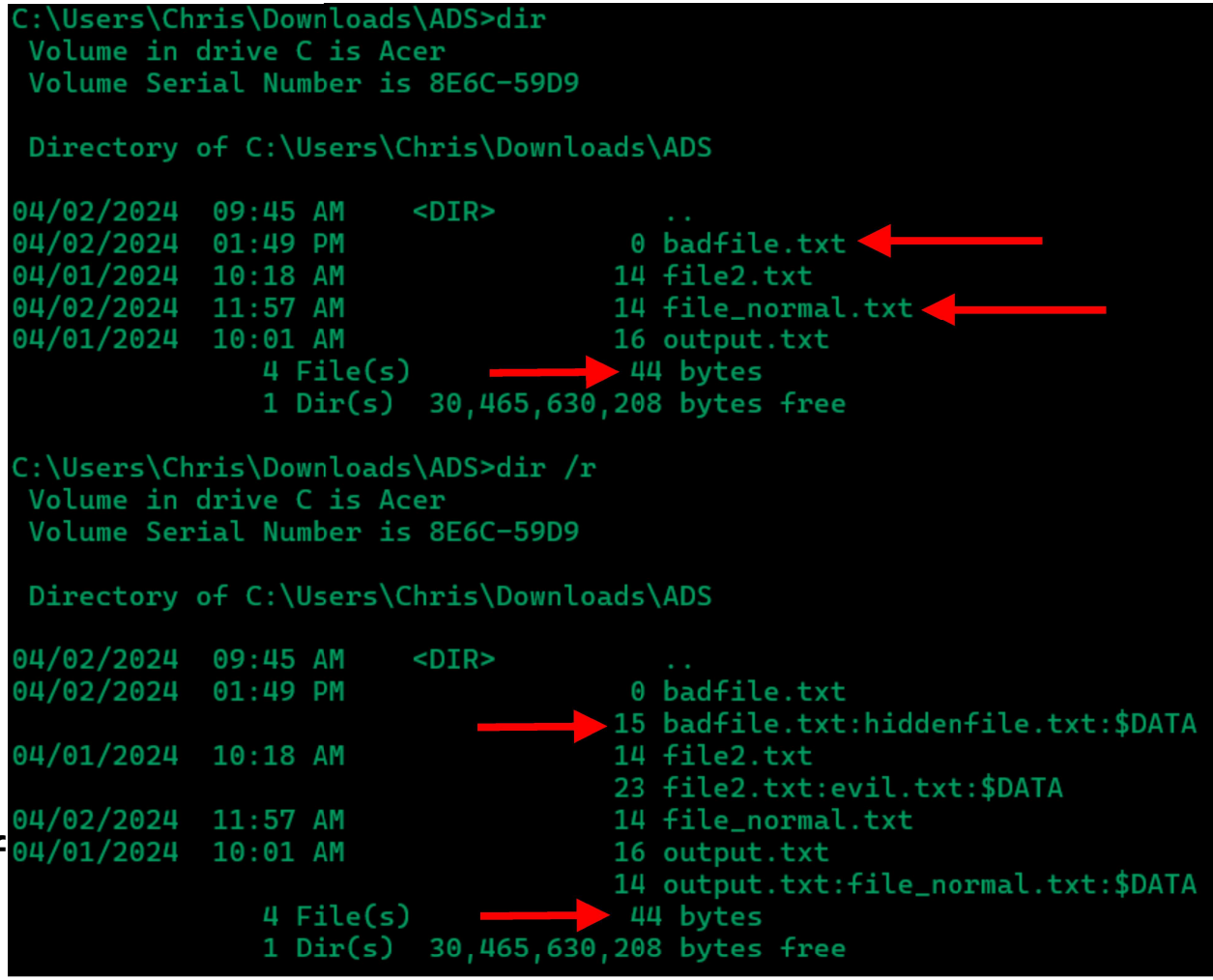
Yes, it is that easy. Lets look more into this.

file_normal 14 bits
Badfile 0 bits

We know text was saved to Badfile but Windows says the size is 0. The power of ADS!

Ok, lets prove its there by adding /r to the dir command.

Aha, we now see the ADS for badfile and the true file size of 15bytes



We now have seen data in the ADS hides itself fairly well. Just how well does it hide itself though? What about if we hash the file...

```
C:\Users\Chris\Downloads\ADS>echo Normal File > file2.txt
```

Lets make file2 and inside put “Normal File”. Then lets grab the has of the file.

```
C:\Users\Chris\Downloads\ADS>echo Normal File > file2.txt

C:\Users\Chris\Downloads\ADS>fciv file2.txt
//
// File Checksum Integrity Verifier version 2.05.
//
27d306fd5ac51bee8414d5d3ecbcc481 file2.txt
```

Now, lets hide some data and inside the same file (file2.txt) add the text “Evil Naughty Malware” to the ADS evil.txt

```
C:\Users\Chris\Downloads\ADS>echo Evil Naughty Malware > file2.tx
t:evil.txt
```

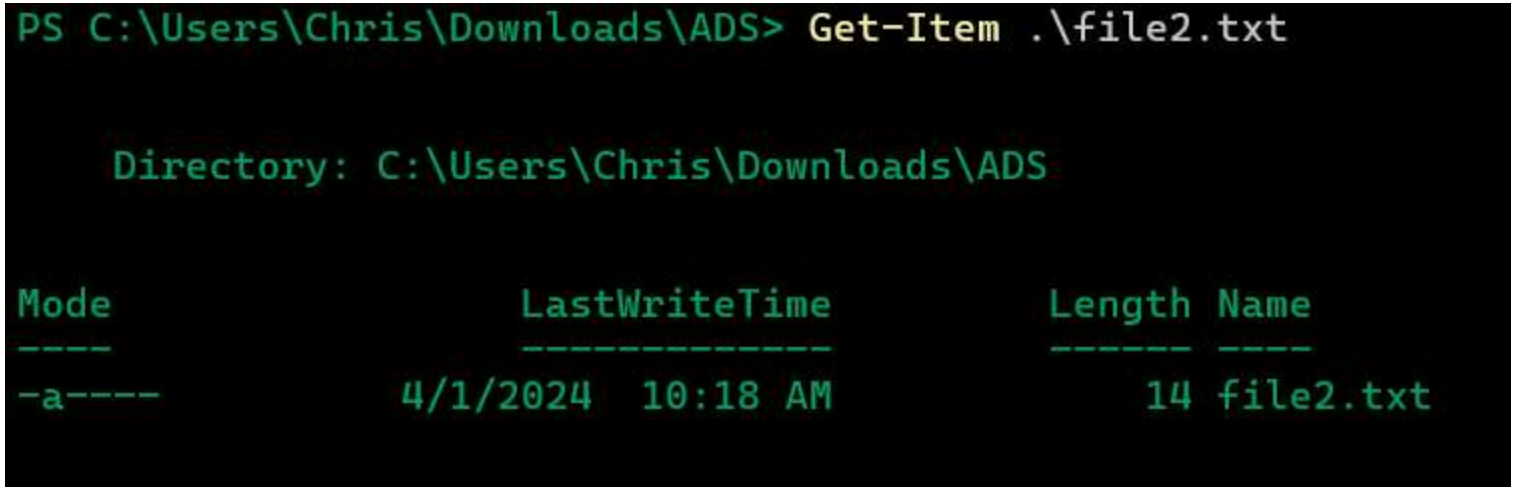
Now, check the hash and...wow same hash but we know there is additional data in there.

```
C:\Users\Chris\Downloads\ADS>fciv file2.txt
//
// File Checksum Integrity Verifier version 2.05.
//
27d306fd5ac51bee8414d5d3ecbcc481 file2.txt
```

I think we all are starting to better understand how ADS can be dangerous.

Lets quick check if Powershell can see into ADS.

We can “Get-Item .\file2.txt” and see what it displays.



Again, ADS remains hidden from of us.

Lets try one more time but this time add “-Stream *”

Ooh found it! Good show!

```
PS C:\Users\Chris\Downloads\ADS> Get-Item .\file2.txt -Stream *


PSPath        : Microsoft.PowerShell.Core\FileSystem::C:\Users\Chris\Downlo
                ads\ADS\file2.txt::$DATA
PSParentPath  : Microsoft.PowerShell.Core\FileSystem::C:\Users\Chris\Downlo
                ads\ADS
PSChildName   : file2.txt::$DATA
PSDrive       : C
PSProvider    : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName      : C:\Users\Chris\Downloads\ADS\file2.txt
Stream        : :$DATA
Length        : 14

PSPath        : Microsoft.PowerShell.Core\FileSystem::C:\Users\Chris\Downlo
                ads\ADS\file2.txt:evil.txt
PSParentPath  : Microsoft.PowerShell.Core\FileSystem::C:\Users\Chris\Downlo
                ads\ADS
PSChildName   : file2.txt:evil.txt
PSDrive       : C
PSProvider    : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName      : C:\Users\Chris\Downloads\ADS\file2.txt
Stream        : evil.txt
Length        : 23
```