Alternate Data Streams

References:

Nelson, Chapter 5, Section “NTFS Alternate Data Streams”

Introduction to ADS

NTFS can have the contents of a file forked into different or alternate "streams" of data

One stream holds the contents of the data that you expect to see

If the file is a shortcut, an alternate stream could contain link information

While metadata (i.e., access rights, ownership, dates of creation, modification, etc.) are usually in the inodes ($MFT file)

Some claim that this represents a stream.

There can be multiple data streams in a single file

Including one that you can explicitly put there -- **ADS**

Introduction to ADS

All versions of NTFS support ADS

ADS allows the ability to fork file data into existing files

Doesn't affect their "visible" functionality or size

ADS is completely hidden

*Is not visible using traditional file browsing utilities like **File Explorer***

Size doesn't reflect the existence of ADS

ADS was originally conceived to allow for compatibility with the Macintosh Hierarchical File System (HFS)

In HFS, file information is sometimes forked into separate resources

Legitimate Uses of ADS

Tiny ADS are added within browsers to indicate that files have been downloaded from external sites

Google Chrome and Opera:

- Zone identifier
- Referrer URL
- Host URL

Microsoft Edge:

- Zone identifier
- Browser name

Firefox and Tor:

- Zone identifier

Legitimate Uses of ADS

Alternate Data Streams have been used by a few media players to hold proprietary metadata such as

Image thumbnails

Author information

...

Archive/backup metadata

Malicious Uses of ADS

- Alternate Data Streams have also been used for malicious purposes
  - *Some browser helper objects (BHOs) store malicious files inside ADS*
    - Very few anti-spyware/malware tools detect it
  - *ADS has been used to remotely exploit a web server*
    - More on this later

NTFS Attributes

Attributes are data structures that store a specific type of data

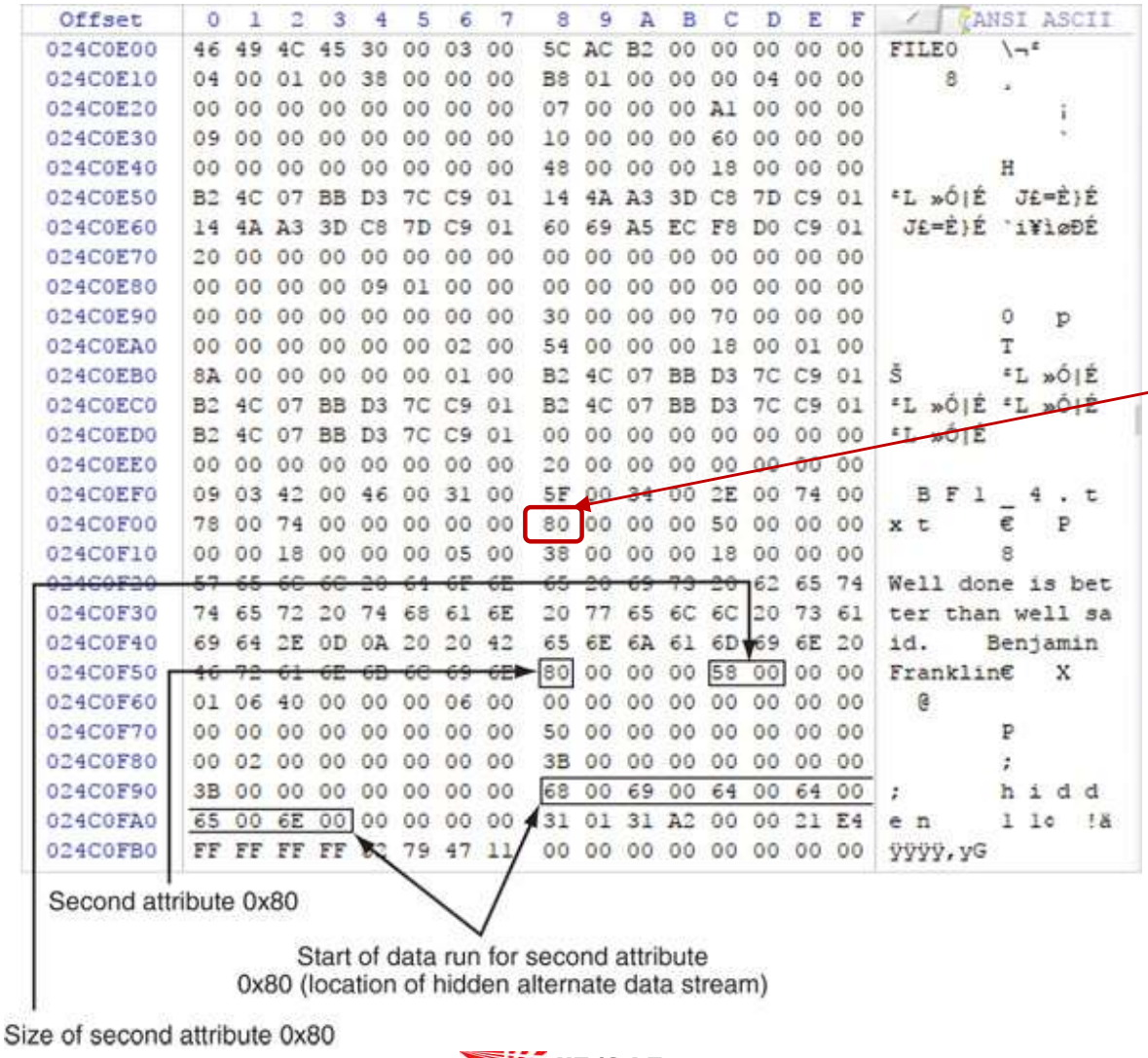
There are many types of attributes, and each has its own internal structure

Every file has a *$DATA* attribute, which contains the file content

If the content is over roughly ~800 bytes in size, it becomes non-resident (stored outside the MFT) and is saved in external clusters

When a file has more than one *$DATA* attribute, the additional attributes are called Alternate Data Streams(ADS)

MFT Record of File with ADS



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
024C0E00	46	49	4C	45	30	00	03	00	5C	AC	B2	00	00	00	00	00	FILE0 \¬²
024C0E10	04	00	01	00	38	00	00	00	B8	01	00	00	00	04	00	00	8 ¸
024C0E20	00	00	00	00	00	00	00	00	07	00	00	00	A1	00	00	00	¡
024C0E30	09	00	00	00	00	00	00	00	10	00	00	00	60	00	00	00	`
024C0E40	00	00	00	00	00	00	00	00	48	00	00	00	18	00	00	00	H
024C0E50	B2	4C	07	BB	D3	7C	C9	01	14	4A	A3	3D	C8	7D	C9	01	²L »Ó|É J£=È}É
024C0E60	14	4A	A3	3D	C8	7D	C9	01	60	69	A5	EC	F8	D0	C9	01	J£=È}É `i¥ìøÐÉ
024C0E70	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
024C0E80	00	00	00	00	09	01	00	00	00	00	00	00	00	00	00	00	
024C0E90	00	00	00	00	00	00	00	00	30	00	00	00	70	00	00	00	0 p
024C0EA0	00	00	00	00	00	00	02	00	54	00	00	00	18	00	01	00	T
024C0EB0	8A	00	00	00	00	00	01	00	B2	4C	07	BB	D3	7C	C9	01	Š ²L »Ó|É
024C0EC0	B2	4C	07	BB	D3	7C	C9	01	B2	4C	07	BB	D3	7C	C9	01	²L »Ó|É ²L »Ó|É
024C0ED0	B2	4C	07	BB	D3	7C	C9	01	00	00	00	00	00	00	00	00	²L »Ó|É
024C0EE0	00	00	00	00	00	00	00	00	20	00	00	00	00	00	00	00	
024C0EF0	09	03	42	00	46	00	31	00	5F	00	34	00	2E	00	74	00	B F 1 _ 4 . t
024C0F00	78	00	74	00	00	00	00	00	80	00	00	00	50	00	00	00	x t € P
024C0F10	00	00	18	00	00	00	05	00	38	00	00	00	18	00	00	00	8
024C0F20	57	65	6C	6C	20	64	6F	6E	65	20	69	73	20	62	65	74	Well done is bet
024C0F30	74	65	72	20	74	68	61	6E	20	77	65	6C	6C	20	73	61	ter than well sa
024C0F40	69	64	2E	0D	0A	20	20	42	65	6E	6A	61	6D	69	6E	20	id. Benjamin
024C0F50	46	72	61	6E	6B	6C	69	6E	80	00	00	00	58	00	00	00	Franklin€ X
024C0F60	01	06	40	00	00	00	06	00	00	00	00	00	00	00	00	00	@
024C0F70	00	00	00	00	00	00	00	00	50	00	00	00	00	00	00	00	P
024C0F80	00	02	00	00	00	00	00	00	3B	00	00	00	00	00	00	00	;
024C0F90	3B	00	00	00	00	00	00	00	68	00	69	00	64	00	64	00	; h i d d
024C0FA0	65	00	6E	00	00	00	00	00	31	01	31	A2	00	00	21	E4	e n 1 1¢ !ä
024C0FB0	FF	FF	FF	FF	82	79	47	11	00	00	00	00	00	00	00	00	ÿÿÿÿ‚yG

Things You Should Know About ADS

There is no limit on the size of streams and there can be more than one stream linked to a normal file

ADS are not visible in Explorer or via command prompt*

ADS size is not reported by Windows

Stream can be attached not only to files but also to folders and drives

The content of an ADS is not limited to simply text data

Any stream of binary information can constitute a file, which includes executables, Mpeg files, Jpeg files, etc.

** Not exactly true. To be discussed later.*

Things You Should Know About ADS

ADSs have no attributes of their own other than its own separate *$DATA*

The access rights assigned to the named stream are the rights that control any operation on the associated ADSs

Examples: creation, deletion, or modification

This means if a user cannot write to a file, that user cannot add an ADS to that file

A user with guest privileges can also create such streams in every file where she has write access

Things You Should Know About ADS

- Windows File Protection prevents the replacement of protected system files
  - *It does not prevent a user with the appropriate permissions from adding ADS to those system files*
  - *The System File Checker (*sfc.exe*) will verify that protected system files have not been overwritten, but will not detect ADS*
- Until Vista, Windows provided no tools or utilities either within the operating system software distribution or the Resource Kits for detecting the presence of ADS
  - *More on this later*

Things You Should Know About ADS

The stream can only be executed if called directly by a program with the full path to the file given. It is very difficult and maybe impossible to accidentally execute an ADS.

None of the Internet protocols enabling file transfer such as SMTP, FTP, etc., support streams.

This means that ADS can't be sent via Internet.

However, files containing ADS can be sent across a local LAN provided the target drive is in the NTFS format

Things You Should Know About ADS

In certain cases, streams have been used to remotely exploit a web server

Some web servers are susceptible to having their file source read via the *$DATA* stream

Suppose a server-side script such as PHP or ASP is running on a web server which is not patched properly

Instead of getting output as a result of processing a script, the source code of the ASP/PHP file could be viewed by using a URL like this:

http://www.abcde.com/index.asp::$DATA

This is a critical vulnerability

The server-side source code could reveal sensitive information including

How the site has been coded

How the information is flowing

This information could be used by the attacker to launch a specific attack on the server

Questions

What happens to a file with Alternate Data Streams when you copy it over to a different file system?

The Alternate Data Streams are lost
-- only the primary stream is copied over

A hash value is calculated on files to determine if they have been modified by Intrusion Detection Systems. Does this value change if you add an Alternate Data Stream?

No, it doesn't. (Hashing tools typically only calculate the hash over the primary data stream.)

ADS Lab

Follow this lab exactly.

The `blue fixed width font` shows exactly what you should type.

Create a Folder

Create the folder **`ADSLab`** on your RADISH Windows 10 desktop

Open a ***cmd.exe*** window **<u>as Administrator</u>**

Navigate to your new ***ADSLab*** folder

Open ***File Explorer*** and navigate to the ***ADSLab*** folder

Arrange the two windows so that they don't overlap

Keep both ***cmd.exe*** and ***File Explorer*** open so you can watch what is happening

File System Metadata

In the ***cmd.exe*** window, create a visible text file as follows:

```
echo Visible Text File 1 > vis1.txt
```

Verify that it is there. How?

`type vis1.txt` *or*

`notepad vis1.txt` *or*

Double click on **`vis1.txt`** *in your* ***File Explorer***

Close notepad if you have it open

Create a Text ADS Stream

Create a secret ADS file

```
echo You can't see me > vis1.txt:hid1.txt
```

Verify:

Type: `dir`

What do you see?

Only the **`vis1.txt`** file in the dir listing

Type: `notepad vis1.txt`

What do you see?

Only the contents of the vis1.txt file

Close notepad

Type: `notepad hid1.txt`

What do you see?

Notepad is empty. Also, do not create a new file

Close notepad

Type: `notepad vis1.txt:hid1.txt`

What do you see?

The contents of the ADS text file **`hid1.txt`**

Close notepad

Create a WordPad ADS Stream

Create a second hidden text file using ***cmd.exe***:

```
echo Hidden text file 2 > hid2.txt
```

Open ***WordPad***. You can easily do this by typing

```
write.exe
```

*Type in the following text in **WordPad**:*

```
We will use this WordPad file as a
  cover file to hide hid2.txt by making
  it an ADS
```

*Save it in your **ADSLab** folder as the file* ***wp.rtf***

Close ***WordPad***

Put **hid2.txt** into **wp.rtf** as an ADS

```
type hid2.txt  > wp.rtf:hid2.txt
```

Verify What I Just Did

Now, ***using File Explorer***, open **`wp.rtf`** with ***WordPad***

What do you see?

Only the Wordpad file. No text file.

How can you see the hidden text file?

Must extract it to a normal file

Extraction

```
more < wp.rtf:hid2.txt > foo.txt
```

Verify your extraction. How?

Type **`foo.txt`**

Or double-click on **`foo.txt`**