

Incident Response and Alternate Data Streams

Written by Tyler Weiss

Exercise 1

Create an Alternate Data Stream in command prompt.

Task 1

Create a simple text file using this echo command: echo Normal File > file_normal.txt

```
C:\Users\tyler>echo "hello this is my stream public" > stream_public.txt

C:\Users\tyler>dir stream_*
Volume in drive C is Windows
Volume Serial Number is A2B2-CE28

Directory of C:\Users\tyler

04/20/2024  11:00 AM                35 stream_public.txt
               1 File(s)                35 bytes
               0 Dir(s) 89,643,671,552 bytes free

C:\Users\tyler>type stream_public.txt
"hello this is my stream public"

C:\Users\tyler>
```

Task 2

Using the above method generate a hidden message as an alternate data stream.

```
echo Evil Malware > badfile.txt:hiddenfile.txt
```

```

C:\Users\tyler>echo "hello this is my hidden stream" > stream_public.txt:stream_private.txt

C:\Users\tyler>dir stream_*
Volume in drive C is Windows
Volume Serial Number is A2B2-CE28

Directory of C:\Users\tyler

04/20/2024  10:58 AM                0 stream_public.txt
             1 File(s)                0 bytes
             0 Dir(s)  89,644,830,720 bytes free

C:\Users\tyler>type stream_public.txt
C:\Users\tyler>

```

Task 3

While in the directory that you created the file with the ADS, run the following command:

```
dir /r *.txt
```

```

C:\Users\tyler>dir /r *.txt
Volume in drive C is Windows
Volume Serial Number is A2B2-CE28

Directory of C:\Users\tyler

01/17/2024  10:30 AM                7 hello.txt
02/09/2024  11:17 AM           69,162 recon-proc.txt
04/20/2024  11:02 AM           35 stream_public.txt
             35 stream_public.txt:stream_private.txt:$DATA
             3 File(s)           69,204 bytes
             0 Dir(s)  89,644,802,048 bytes free

C:\Users\tyler>

```

The '/r' command will list all alternate data streams attached to files and '*.txt*' will narrow the listing to only text documents. The '\$Data' tells us that this is a data type stream.

Task 4

What was the output of that dir command, what did that output mean?

```
C:\Users\tyler>notepad.exe stream_public.txt:stream_private.txt
```

```
C:\Users\tyler>
```



recon1.sh

recon2.sh

recon1-bonus.!

recon3a.sh

Victim

File Edit View

"hello this is my hidden stream"

```
C:\Users\tyler>notepad.exe stream_public.txt:stream_private.txt
```

```
C:\Users\tyler>more < stream_public.txt:stream_private.txt
```

```
"hello this is my hidden stream"
```

```
C:\Users\tyler>
```

The output shows that there is an alternate data stream attached to 'stream_public.txt' and that this stream is called 'stream_private.txt'.

Exercise 2

Prerequisite fciv must be in the system path. (protip, save it to your nmap folder, it is already in your path. C:\\Program Files (x86)\\Nmap)

Create an Alternate Data Stream and check the MD5 hash of the streams.

Task 1

Create a simple text file using this echo command:

```
echo Normal File > file2.txt
```

```
C:\Users\tyler>echo Normal File > file2.txt

C:\Users\tyler>dir file2.txt
Volume in drive C is Windows
Volume Serial Number is A2B2-CE28

Directory of C:\Users\tyler

04/20/2024  12:31 PM                14 file2.txt
               1 File(s)                14 bytes
               0 Dir(s)  89,512,378,368 bytes free

C:\Users\tyler>type file2.txt
Normal File

C:\Users\tyler>
```

The above screen shot show the file creation. Use 'dir' and 'type' to verify its existence and contents.

Task 2

Obtain a hash of file2.txt

```
C:\Users\tyler>fciv file2.txt -both
//
// File Checksum Integrity Verifier version 2.05.
//
               MD5                      SHA-1
-----
27d306fd5ac51bee8414d5d3ecbcc481 658b5eb20269b233cbba58986cddcac95fba397e file2.txt

C:\Users\tyler>
```

Should be: 27d306fd5ac51bee8414d5d3ecbcc481 (MD5)

I used 'fciv.exe' with the '-both' to obtain the hash for both MD5 and SHA1 algorithms.

Task 3

Add an alternate data stream to the file.

```
echo Evil Malware > file2.txt:evil.txt
```

```

C:\Users\tyler>echo Evil Malware > file2.txt:evil.txt

C:\Users\tyler>dir /r *txt
Volume in drive C is Windows
Volume Serial Number is A2B2-CE28

Directory of C:\Users\tyler

04/20/2024  12:35 PM                14 file2.txt
                                15 file2.txt:evil.txt:$DATA
01/17/2024  10:30 AM                 7 hello.txt
04/20/2024  12:26 PM                14 justafile.txt
02/09/2024  11:17 AM            69,162 recon-proc.txt
04/20/2024  11:02 AM                35 stream_public.txt
                                35 stream_public.txt:stream_private.txt:$DATA
               5 File(s)              69,232 bytes
               0 Dir(s)  89,506,131,968 bytes free

C:\Users\tyler>more < file2.txt:evil.txt
Evil Malware

C:\Users\tyler>

```

Use 'dir /r' to verify the ADS was created. Use 'more < <file>:<fileADS>' to print the contents of the ADS to the console.

Task 4

Where are your data integrity gods now?

RE- Obtain a hash of file2.txt

```

C:\Users\tyler>fciv file2.txt -both
//
// File Checksum Integrity Verifier version 2.05.
//
               MD5                               SHA-1
-----
27d306fd5ac51bee8414d5d3ecbcc481  658b5eb20269b233cbba58986cddcac95fba397e file2.txt

C:\Users\tyler>

```

It is STILL: 27d306fd5ac51bee8414d5d3ecbcc481 (MD5)

Alternate Data Streams (ADS) were originally created to be compatible with Mac HFS+ file systems. This was used to attach related data to a file. It also be used to hide files, attach executables, and check file integrity. There are two types of ADS, the associated and isolated ADS. Isolated ADS does not attach itself to an existing file stream (ex. `echo "put malware here" > :evil.txt`).

```

C:\Users\tyler>fciv file2.txt:evil.txt -both
//
// File Checksum Integrity Verifier version 2.05.
//
           MD5                               SHA-1
-----
63ce801629077d80d09c52e4552c45b0 ac8301a5d1f2e22a86cb1b4114fc0162ec7d8724 file2.txt:evil.txt
C:\Users\tyler>

```

Notice that the hash is different from the above example. This proves that the data streams are separate. ADS was created this way to not interfere with integrity checks of the original file stream.

Exercise 3

Use the 'Get-Item' cmdlet in powershell to retrieve the file information. We will display the information for the file we created previously 'file2.txt'. Specify the file using '-Path' parameter. If you do not specify the file exactly (ie `Get-Item -Path ./`) the cmdlet will produce information on the current directory instead. Use this cmdlet with the '-Stream *' parameter to print both Original Data Stream and ADS object information to the console.

```

PS C:\Users\tyler> Get-Item -Path .\file2.txt -Stream *

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

PSPath      : Microsoft.PowerShell.Core\FileSystem::C:\Users\tyler\file2.txt:evil.txt
PSParentPath : Microsoft.PowerShell.Core\FileSystem::C:\Users\tyler
PSChildName  : file2.txt:evil.txt
PSDrive      : C
PSProvider   : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName     : C:\Users\tyler\file2.txt
Stream       : evil.txt
Length       : 15

```

Notice that the two data stream objects are printed out separately with their own property values displayed.

You could also use the command in this way without explicitly using the '-Path' parameter. The following commands show the object output of all text files in the directory.

```

PS C:\Users\tyler> Get-Item *.txt -Stream *

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

PSPath      : Microsoft.PowerShell.Core\FileSystem::C:\Users\tyler\file2.txt:evil.txt
PSParentPath : Microsoft.PowerShell.Core\FileSystem::C:\Users\tyler
PSChildName  : file2.txt:evil.txt
PSDrive      : C
PSProvider   : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName     : C:\Users\tyler\file2.txt
Stream       : evil.txt
Length       : 15

PSPath      : Microsoft.PowerShell.Core\FileSystem::C:\Users\tyler\hello.txt::$DATA
PSParentPath : Microsoft.PowerShell.Core\FileSystem::C:\Users\tyler
PSChildName  : hello.txt::$DATA
PSDrive      : C
PSProvider   : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName     : C:\Users\tyler\hello.txt
Stream       : :$DATA
Length       : 7

```

The below command can be used to search this directory and all parent directories for a '.txt' objects and their ADS.

```

Get-ChildItem -Path "C:\path\to" -Recurse | ForEach-Object { Get-Item -
Path $_.FullName -Stream * }

```

```
PS C:\Users\tyler>
PS C:\Users\tyler> Get-ChildItem -Path ./*.txt -Recurse | ForEach-Object { Get-Item -Path $_.FullName -Stream * }
```

```
PSPath      : Microsoft.PowerShell.Core\FileSystem::C:\Users\tyler\.vscode\extensions\formulahendry.c
              ode-runner-0.12.1\node_modules\applicationinsights\License.txt::$DATA
PSParentPath : Microsoft.PowerShell.Core\FileSystem::C:\Users\tyler\.vscode\extensions\formulahendry.c
              ode-runner-0.12.1\node_modules\applicationinsights
PSChildName  : License.txt::$DATA
PSDrive      : C
PSProvider   : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName     : C:\Users\tyler\.vscode\extensions\formulahendry.code-runner-0.12.1\node_modules\applica
              tioninsights\License.txt
Stream       : :$DATA
Length       : 1101
```

```
PSPath      : Microsoft.PowerShell.Core\FileSystem::C:\Users\tyler\.vscode\extensions\formulahendry.c
              ode-runner-0.12.1\LICENSE.txt::$DATA
PSParentPath : Microsoft.PowerShell.Core\FileSystem::C:\Users\tyler\.vscode\extensions\formulahendry.c
              ode-runner-0.12.1
PSChildName  : LICENSE.txt::$DATA
PSDrive      : C
PSProvider   : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName     : C:\Users\tyler\.vscode\extensions\formulahendry.code-runner-0.12.1\LICENSE.txt
Stream       : :$DATA
Length       : 1064
```

```
PSPath      : Microsoft.PowerShell.Core\FileSystem::C:\Users\tyler\.vscode\extensions\ms-python.debug
              py-2024.0.0-win32-x64\bundled\libs\debugpy\_vendored\pydevd\pydevd_attach_to_process\RE
              ADME.txt::$DATA
PSParentPath : Microsoft.PowerShell.Core\FileSystem::C:\Users\tyler\.vscode\extensions\ms-python.debug
              py-2024.0.0-win32-x64\bundled\libs\debugpy\_vendored\pydevd\pydevd_attach_to_process
PSChildName  : README.txt::$DATA
PSDrive      : C
PSProvider   : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName     : C:\Users\tyler\.vscode\extensions\ms-python.debugpy-2024.0.0-win32-x64\bundled\libs\deb
              ugpy\_vendored\pydevd\pydevd_attach_to_process\README.txt
Stream       : :$DATA
Length       : 987
```

```
PSPath      : Microsoft.PowerShell.Core\FileSystem::C:\Users\tyler\.vscode\extensions\ms-python.debug
```