
CSC153: Lab 3 - Examining NTFS Disks

Ryan Kozak



2019-11-11

Objectives

- Become familiar with the WinHex forensics tool.
- Use WinHex to explore the MFT and be able to analyze both resident and non-resident files.

Part 1: Explore MFT of a file

To begin, we create a text file named `forensicsclass.txt` and put it on our Desktop.

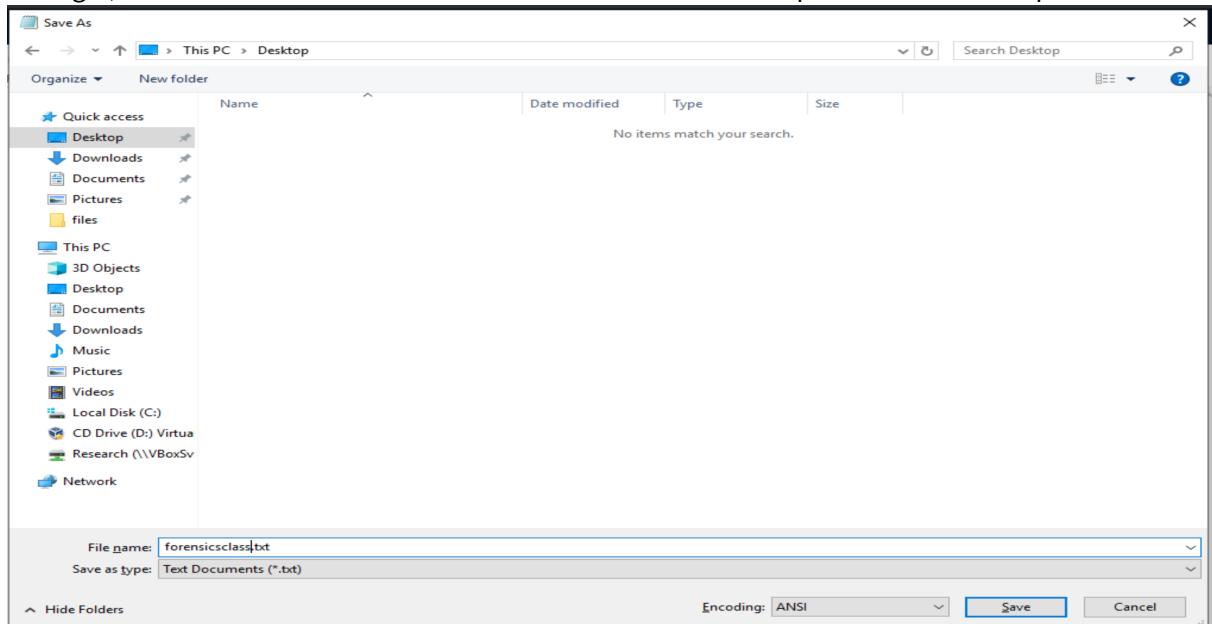


Figure 1: Create text file `forensicsclass.txt`.

In the file, we type `We will have a forensics class on Monday.`

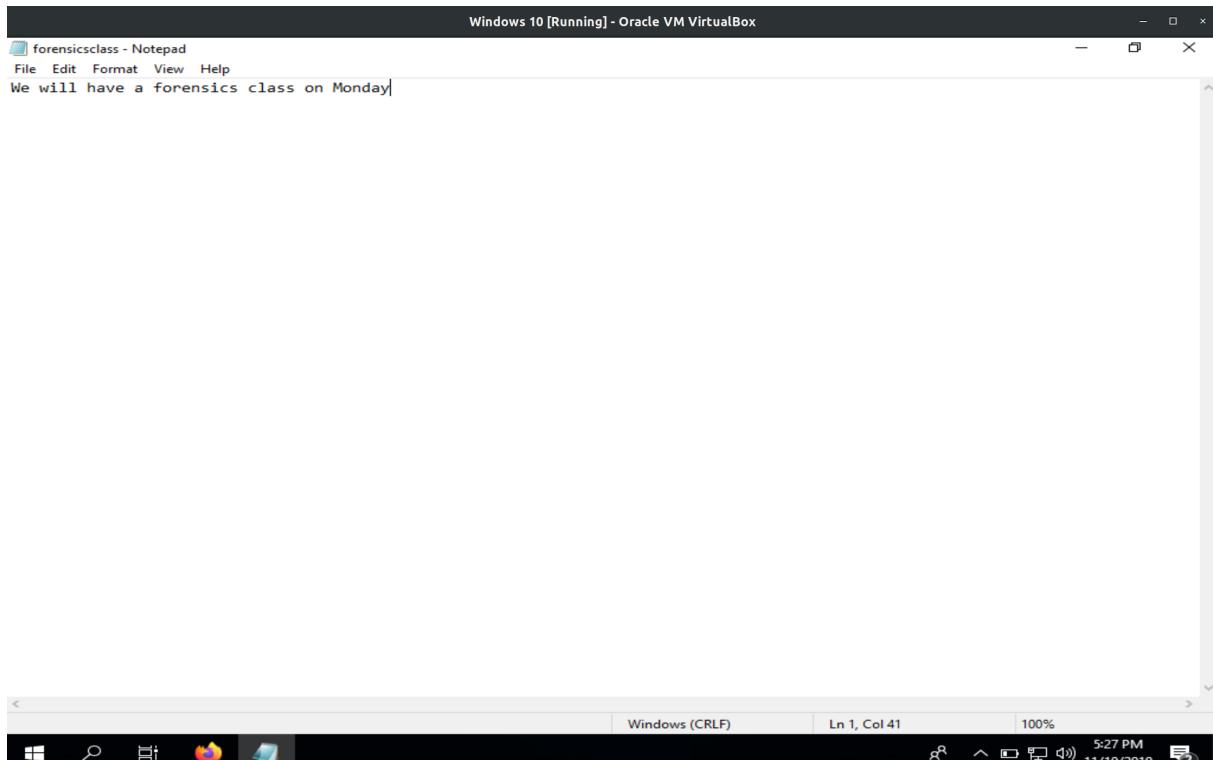


Figure 2: We will have a forensics class on Monday.

Next we append an alternate data stream containing `If you study hard, then you are likely to succeed` to the file via the command below.

```
1 echo If you study hard, then you are likely to succeed > forensicsclass.txt:secret
```

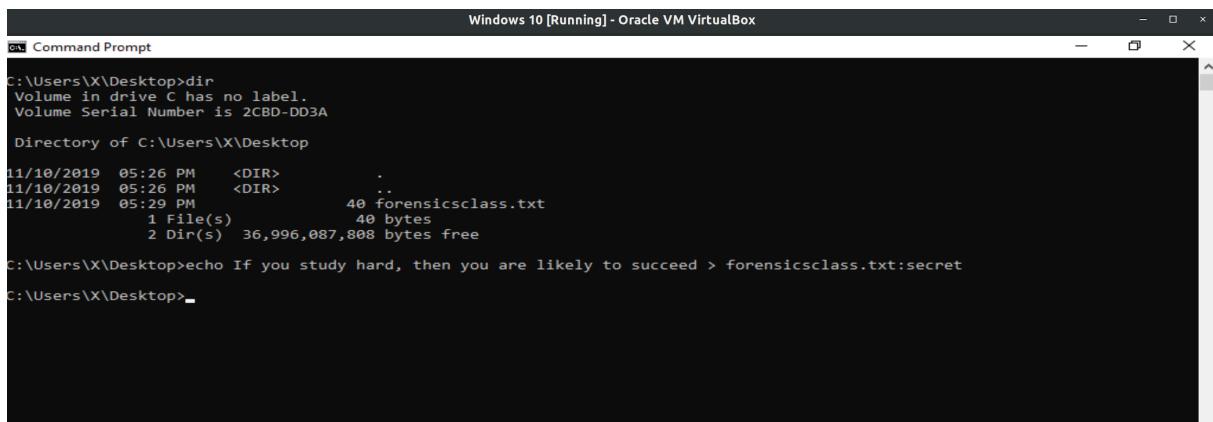
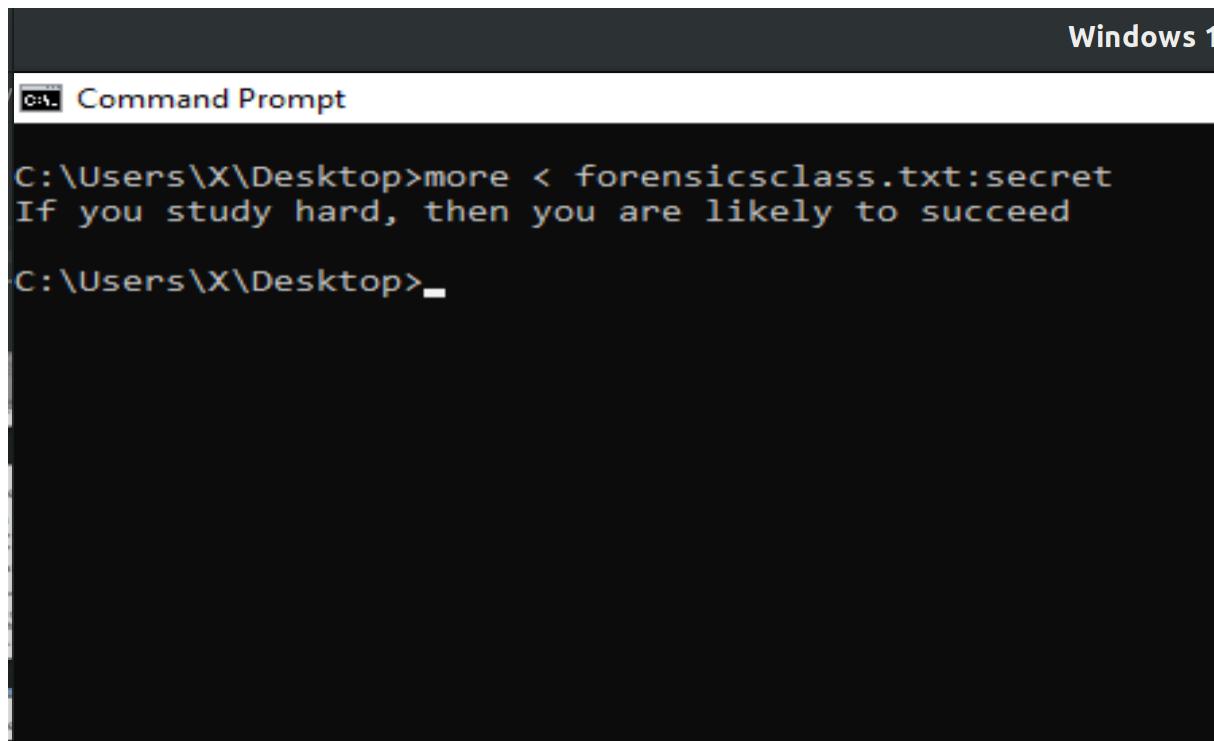


Figure 3: Appending alternate data stream `secret` to `forensicsclass.txt`.

We can now display that data stream via the command `more < forensicsclass.txt:secret`.



The image shows a Windows Command Prompt window titled "Windows 1". The title bar also includes "Command Prompt". The command entered is "more < forensicsclass.txt:secret". The output displayed is "If you study hard, then you are likely to succeed". The prompt "C:\Users\X\Desktop>" is visible at the bottom.

Figure 4: Displaying alternate data stream `secret` for `forensicsclass.txt`.

Next, we examine the metadata of the `forensicsclass.txt` file stored in the `$MFT` file. First we run WinHex as an administrator.

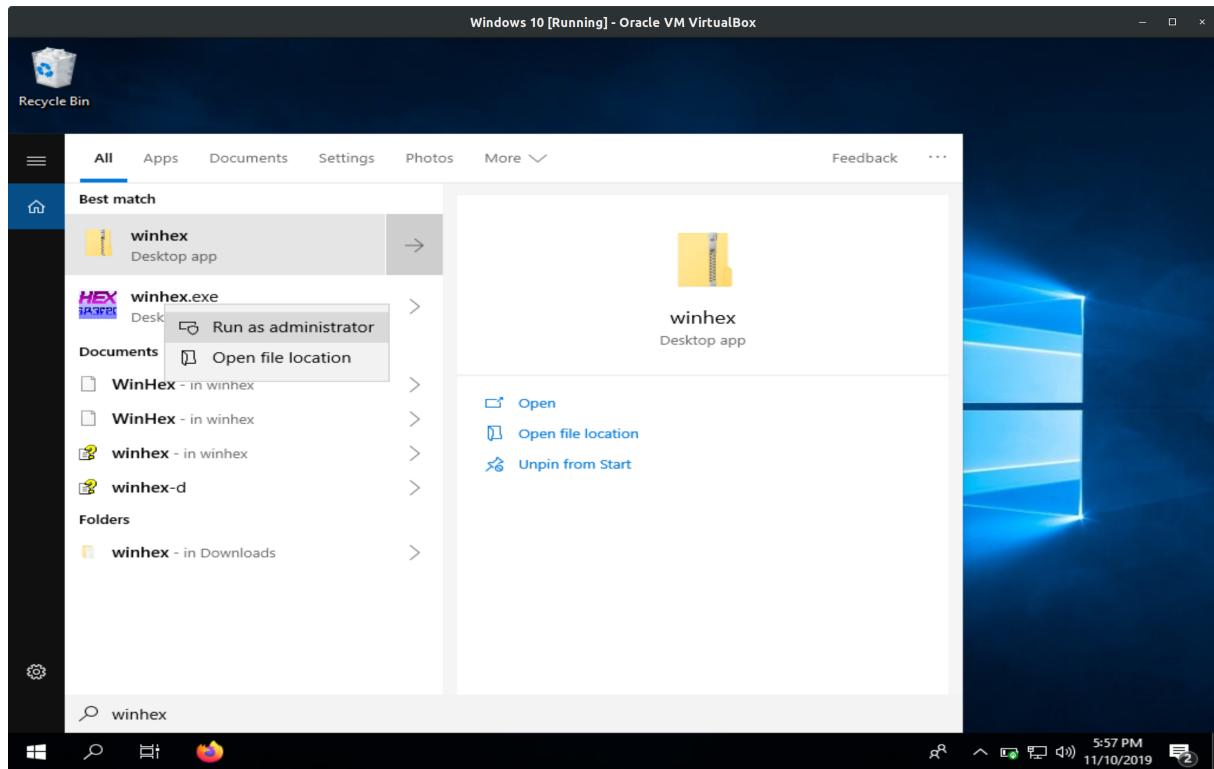


Figure 5: Launching WinHex in administrator mode..

As safety precaution, click `Options -> Edit Mode` and select `Read-Only Mode (=write protected)` from the `Select Mode` dialog box.

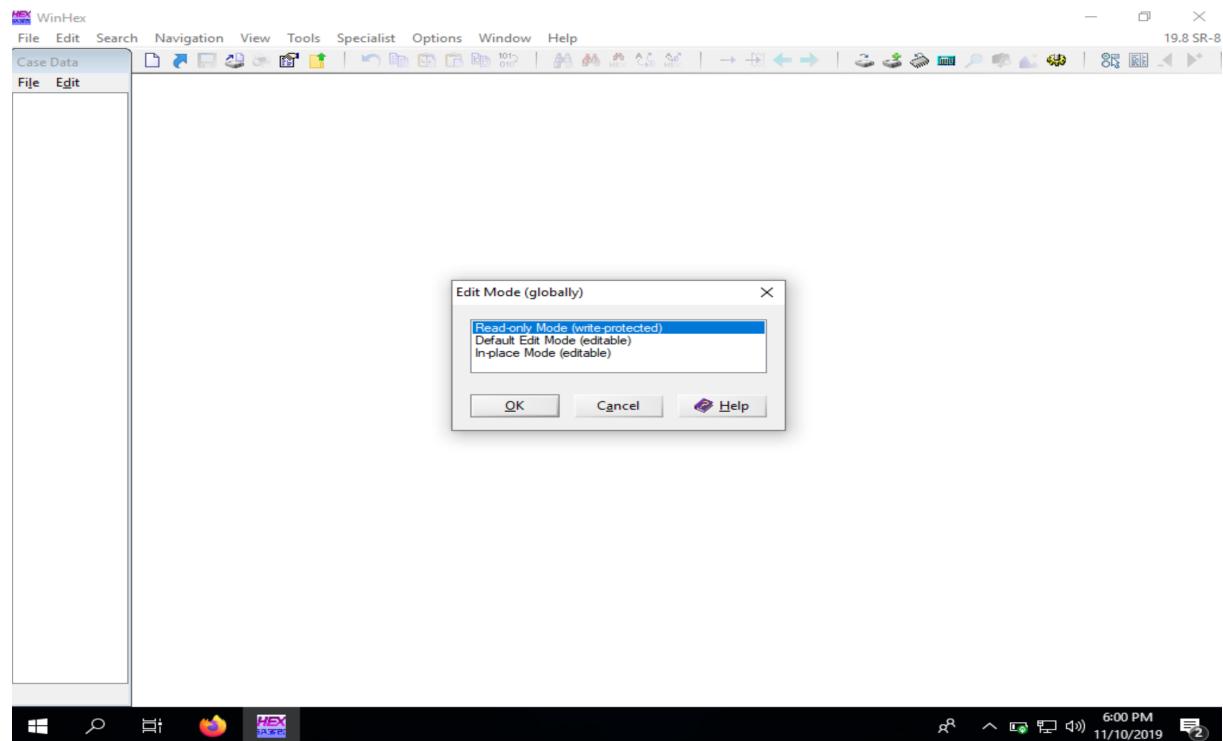


Figure 6: Set Edit Mode to Read-Only.

To examine our disk, we click [Tools](#) → [Open Disk](#) from the menu. In the View Disk dialog box we select the [C:](#) drive and then click OK.

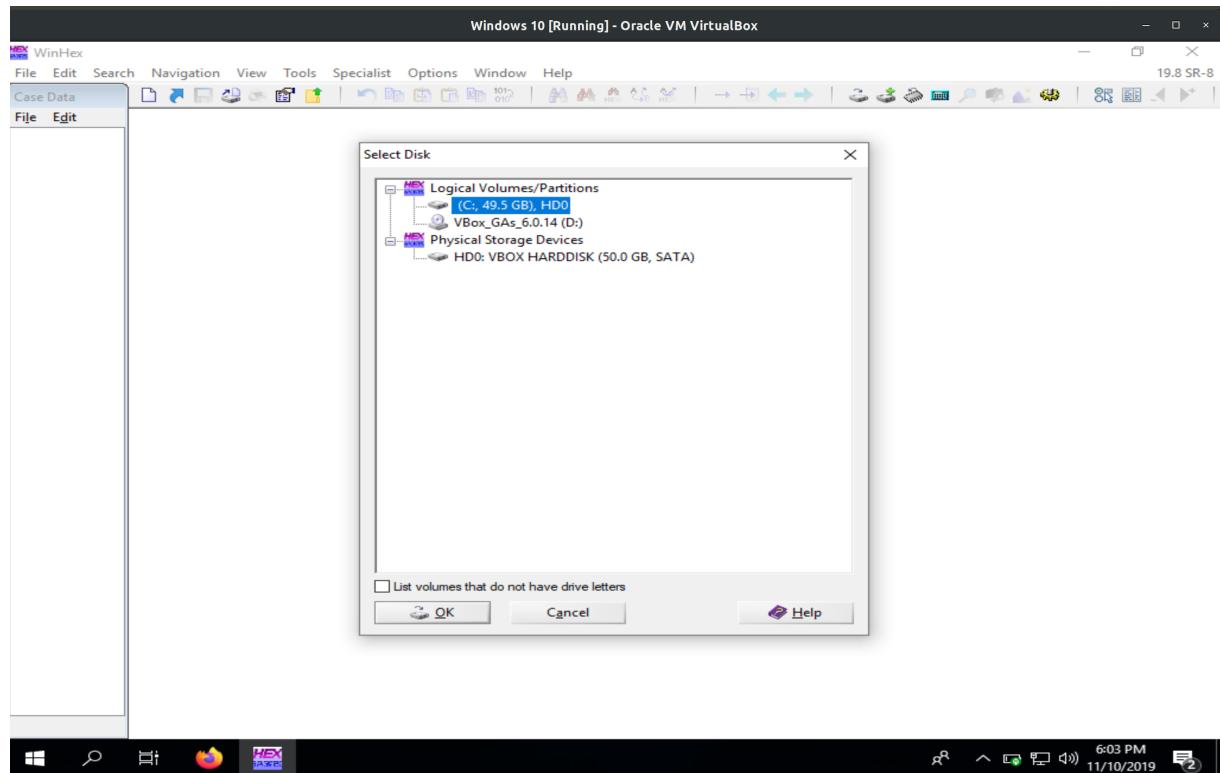


Figure 7: Selecting the C: drive as our disk to open.

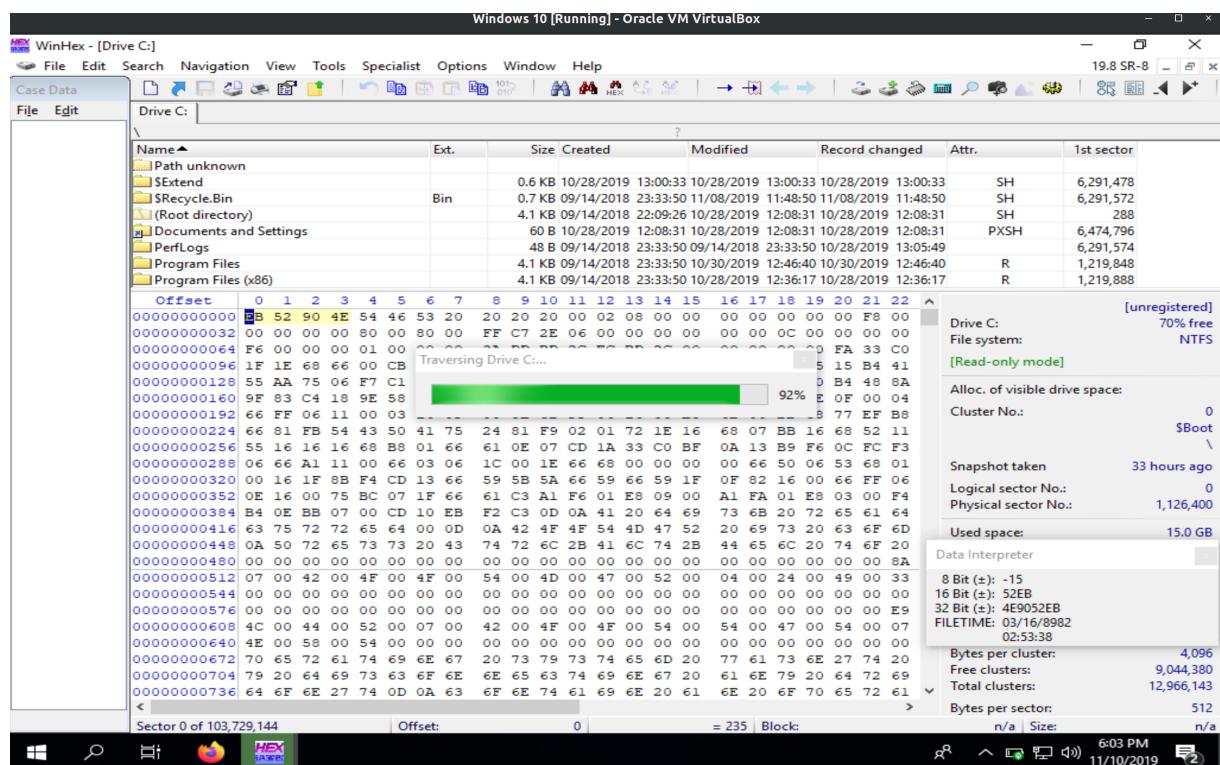


Figure 8: WinHex traversing the C: drive.

Before we examine the disk we need to navigate to [Options](#) → [Data Interpreter](#) from the menu. In the [Data Interpreter Options](#) dialog box, we click the [Win32 FILETIME \(64 bit\)](#) check box, shown in Figure 9, and then click OK. The Data Interpreter should then have FILETIME as an addition display item.

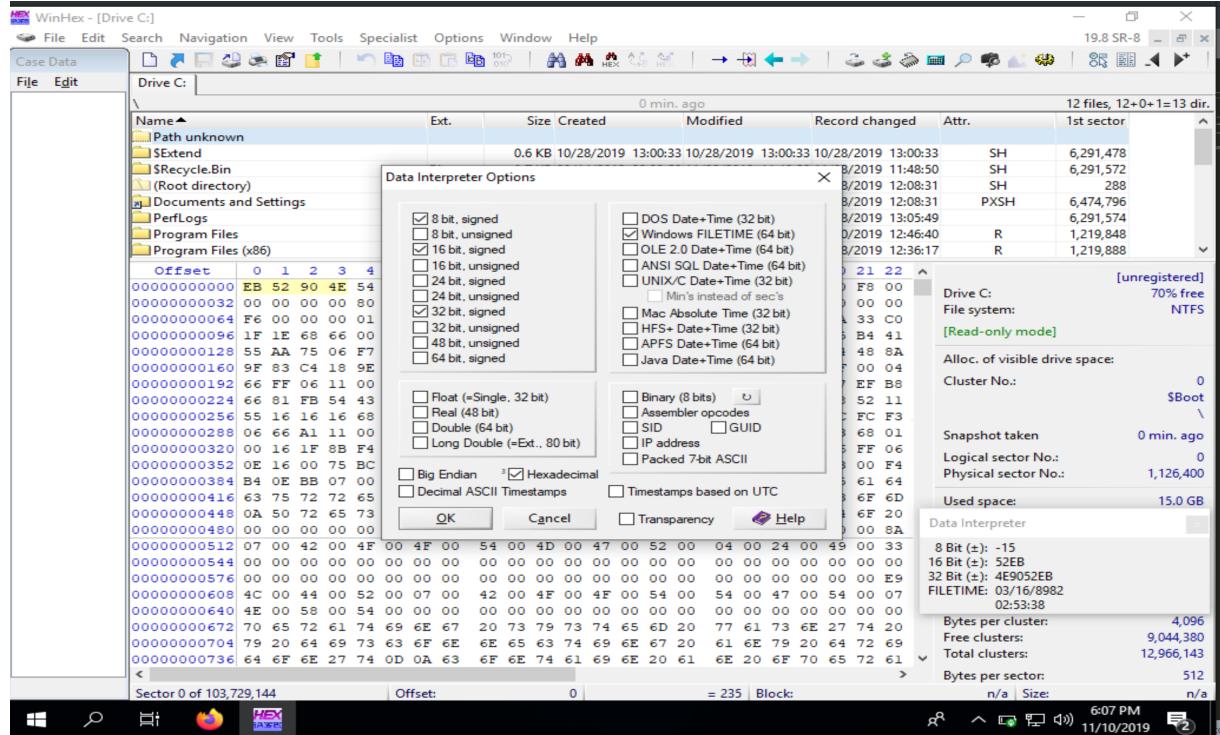


Figure 9: Data Interpreter Option include [Win32 FILETIME \(64 bit\)](#).

Now in WinHex we need to navigate to where we saved [forensicsclass.txt](#) and click it.

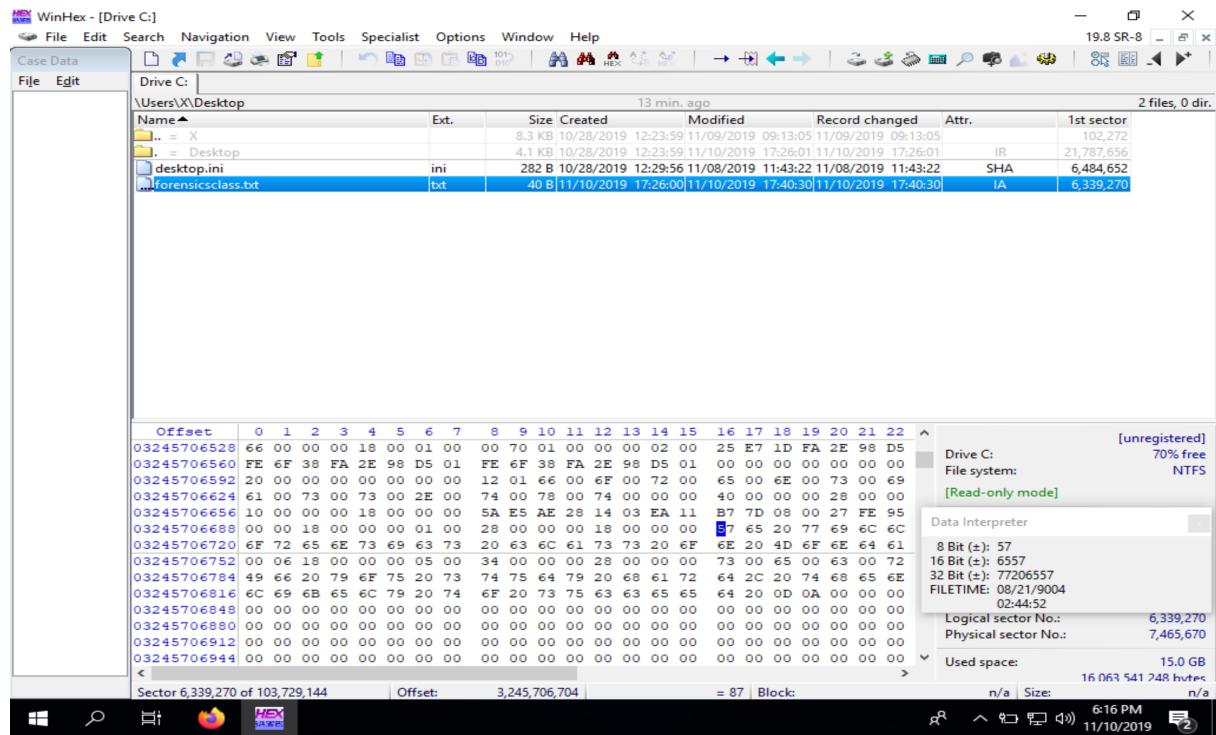


Figure 10: Selecting `forensicsclass.txt`.

Next we click at the beginning of the record, on the letter `F` in `FILE`, and then drag down and to the right while monitoring the hexadecimal counter in the lower-right corner. At offset `0x38` from the beginning of the MTF record we find the start of the attribute `0x10`.

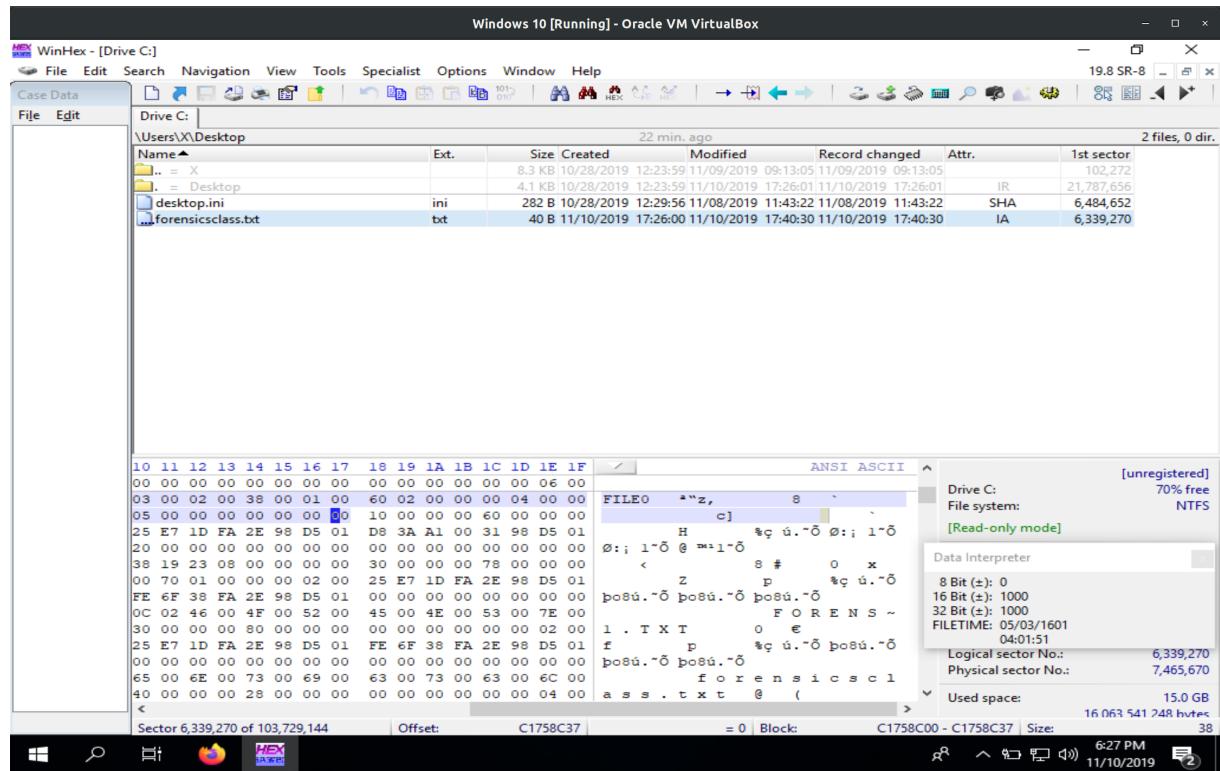


Figure 11: Attribute 0x01 at offset 0x38 from the start of the MFT record.

The file's created date and time can be found from offset 0x18 to 0x1F from the beginning of attribute 0x10. In the same manner we used above, we can determine the files created date and time to be 11/10/19 17:26:00.

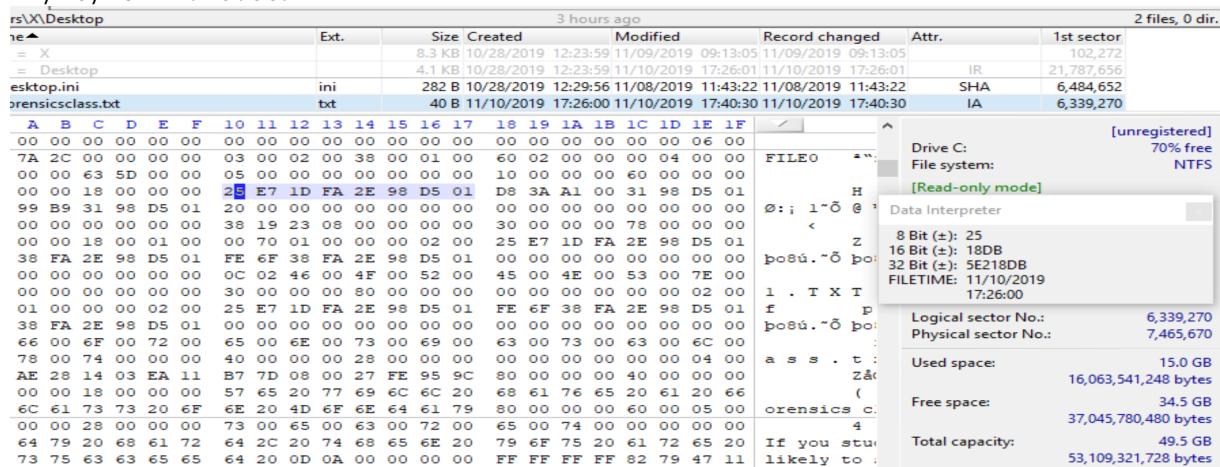


Figure 12: File created date and time for forensicsclass.txt.

Questions for Part 1

1. According to the data interpreter, what is the created date and time for the file `forensicsclass.txt`?

- The created date and time for `forensicsclass.txt` is **11/10/19 17:26:00**. It's found from offset `0x18` to `0x1F` from the beginning of the attribute `0x10`. It should be noted that this is with the box for `Timestamps based on UTC` selected.

3 hours ago																		2 files, 0 dir.					
File		Ext.	Size	Created	Modified	Record changed		Attr.	1st sector														
=	X		8.3 KB	10/28/2019 12:23:59	11/09/2019 09:13:05	11/09/2019 09:13:05			102,272														
= Desktop			4.1 KB	10/28/2019 12:23:59	11/10/2019 17:26:01	11/10/2019 17:26:01		IR	21,787,656														
esktop.ini	ini		282 B	10/28/2019 12:29:56	11/08/2019 11:43:22	11/08/2019 11:43:22		SHA	6,484,652														
orensicsclass.txt	txt		40 B	10/11/2019 17:26:00	11/10/2019 17:40:30	11/10/2019 17:40:30		IA	6,339,270														
A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F		
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
7A	2C	00	00	00	00	00	03	00	02	00	38	00	01	00	60	02	00	00	00	04	00	00	
00	00	63	5D	00	00	05	00	00	00	00	00	00	00	00	10	00	00	00	60	00	00	00	
00	00	18	00	00	00	02	2E	7E	1D	FA	2E	98	DS	01	D8	3A	AI	01	31	98	DS	01	
99	B9	31	98	DS	01	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00	00	00	00	00	00	00	38	19	23	08	00	00	00	00	30	00	00	00	78	00	00	00	
00	00	18	00	01	00	00	70	01	00	00	00	02	00	25	E7	1D	FA	2E	98	DS	01	00	
38	FA	2E	98	DS	01	FE	6F	38	FA	2E	98	DS	01	00	00	00	00	00	00	00	00	00	
00	00	00	00	00	00	00	02	0C	02	46	00	4F	00	52	00	45	00	4E	00	53	00	7E	00
00	00	00	00	00	00	00	30	00	00	00	80	00	00	00	00	00	00	00	02	00	00	00	
01	00	00	00	02	00	00	25	7E	1D	FA	2E	98	DS	01	FE	6F	38	FA	2E	98	DS	01	
38	FA	2E	98	DS	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
66	00	6F	00	72	00	65	00	6E	00	73	00	69	00	63	00	73	00	63	00	6E	00	00	
78	00	74	00	00	00	40	00	00	00	28	00	00	00	00	00	00	00	00	00	04	00	00	
AE	28	14	03	EA	11	B7	7D	08	00	27	FE	95	9C	80	00	00	00	40	00	00	00	00	
00	00	18	00	00	00	57	65	20	77	69	6C	6C	20	68	61	76	65	20	61	20	66	00	
6C	61	73	73	20	6E	6E	20	4D	6E	64	61	79	80	00	00	00	60	00	05	00	00	00	
00	00	28	00	00	00	73	00	65	00	63	00	72	00	65	00	74	00	00	00	00	00	00	
64	79	20	68	61	72	64	2C	20	74	68	65	6E	20	79	6F	75	20	61	72	65	20	60	
73	75	63	63	65	65	64	20	0D	0A	00	00	00	00	FF	FF	FF	FF	82	79	47	11	00	

Figure 13: File created date and time for `forensicsclass.txt`.

2. What is the size of the MFT record?

- The size of the MTF record is, in big endian, 00 00 04 00 . We can find this information at from offset 0x1C to 0x1F from attribute 0x00.

The screenshot shows the WinHex application interface. The menu bar includes File, Edit, Search, Navigation, View, Tools, Specialist, Options, Window, Help, and a language setting of 19.8 SR-8. The toolbar contains icons for opening files, saving, zooming, and navigating. The status bar at the bottom right shows Sector 6,339,270 of 103,729,144, Offset C1758C1F, Block = 0, and Mode hexadecimA1.

File List:

Name	Ext.	Size	Created	Modified	Record changed	Attr.	1st sector
desktop.ini	ini	8.3 KB	10/28/2019 12:23:59	11/10/2019 09:13:05	11/10/2019 09:13:05		102,272
forensicsclass.txt	txt	4.1 KB	10/28/2019 12:23:59	11/10/2019 17:26:01	11/10/2019 17:26:01	SHA	21,787,656
							6,484,652
							6,339,270

File Content:

ANSI ASCII view of desktop.ini:

```
FILE0 ^z, 0
c]
H %q u.-.0 :; i-0
:; i-0 @ mii-0
< 8 # 0 x
z p %q u.-.0
bosu.-.0 bosu.-.0 bosu.-.0
F O R E N S ~
1 . T X T 0 €
f p %q u.-.0 bosu.-.0
bosu.-.0 bosu.-.0
f o r e n s i c s c l
a s s . t x t @ (
Z@( ) 'p@e @
( We will have a
forensics class on Monday€
```

ANSI ASCII view of forensicsclass.txt:

```
4 ( secret
If you study hard, then you are
likely to succeed
yyyy, yG
```

File System Information:

- Drive C: 70% free NTFS
- File system: [Read-only mode]
- Data Interpreter
- Logical sector No.: 6,339,270
- Physical sector No.: 7,465,670
- Used space: 16,063,541,248 bytes (15.0 GB)
- Free space: 37,045,780,480 bytes (34.5 GB)
- Total capacity: 53,109,321,728 bytes (49.5 GB)
- Bytes per cluster: 4,096
- Free clusters: 9,044,380
- Total clusters: 12,966,143
- Bytes per sector: 512
- Sector count: 103,729,144
- Physical disk: 0

Figure 14: The size of the MTF record for `forensicsclass.txt` is `0x0400`.

3. What is the length of the header?

- The header length for the MFT record is **0x38**. This can be found at offset **0x14** from attribute **0x00**.

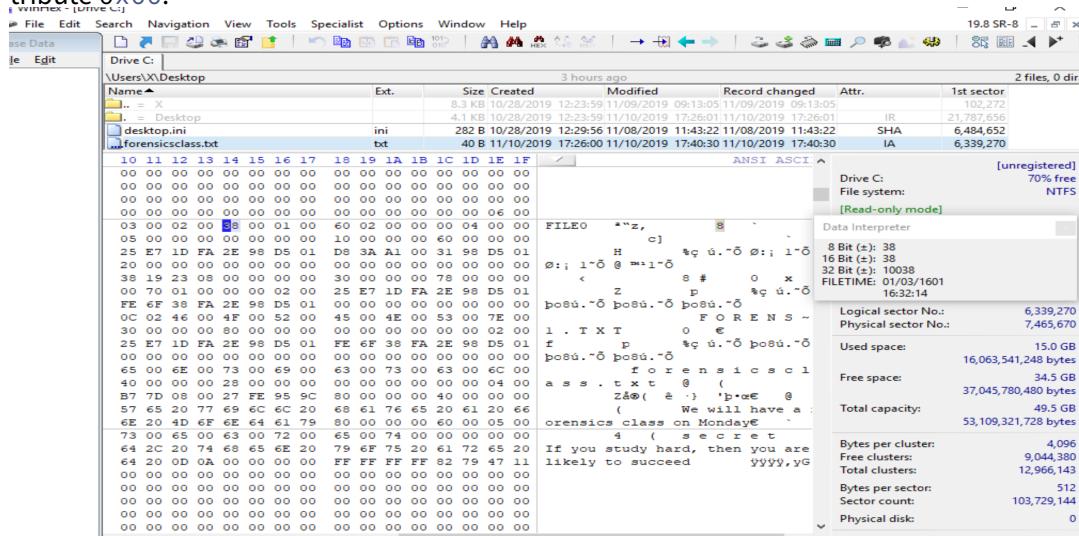


Figure 15: The length of the MFT record header for `forensicsclass.txt` is **0x38**.

4. What is the file's last modified date and time?

- The file's last modified date and time is **11/10/19 17:26:00**. We can find that information from offset **0x20** to **0x27** of attribute **0x10**. It should be noted that this is with the box for Timestamps based on UTC selected.

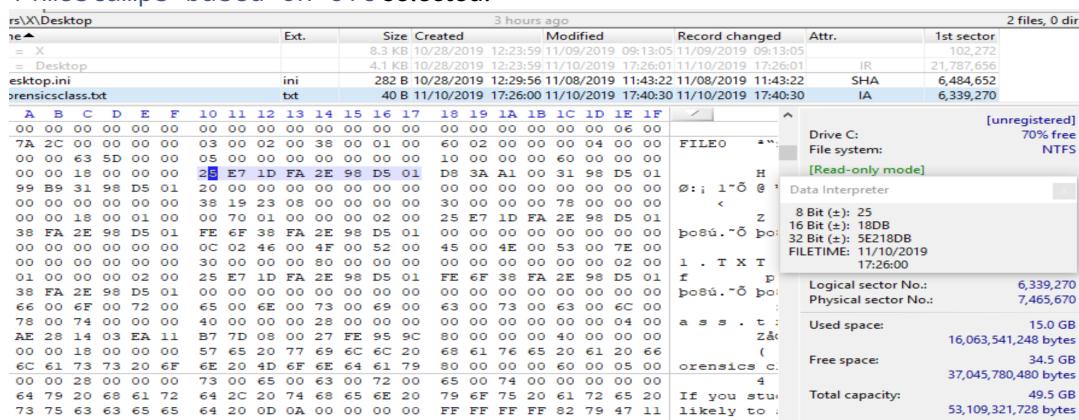


Figure 16: File last modified date and time for `forensicsclass.txt`.

5. How many **0x30** attributes does this file have? Why?

- There are **two** attribute **0x30**'s. This is because our file name is longer than 8 characters, so we have a **short file name**, and a **long file name**.

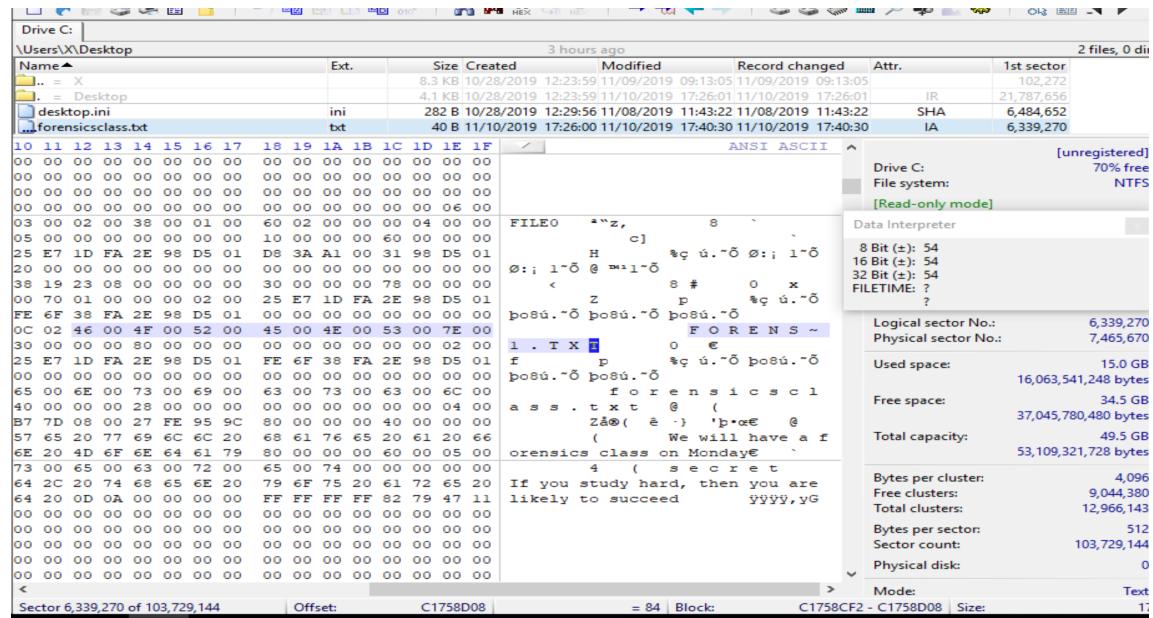


Figure 17: The short file name at **0x5A** from the first **0x30** attribute.

- Long file names are found at offset **0x5A** from the **second** **0x30** attribute. Our long file name is **forensicsclass.txt**.

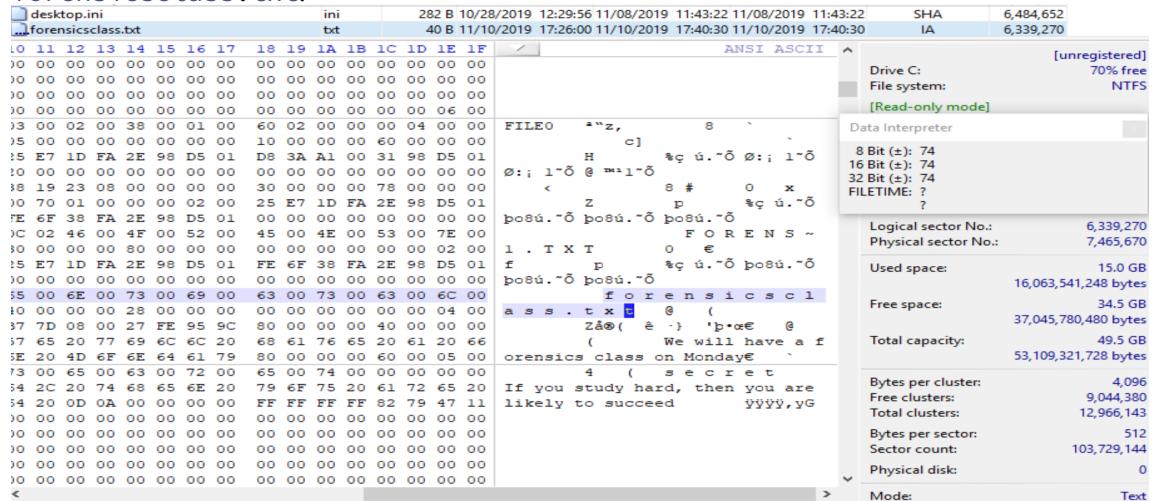


Figure 18: The long file name at **0x5A** from the second **0x30** attribute.

6. What is the name of this file?

- As stated above there are two file names, a short file name and long file name.
- Short file names are found at offset **0x5A** from the **first** **0x30** attribute. Our short file name is **FORENS~1.TXT**.

The screenshot shows a hex editor window for a file named 'forensicsclass.txt' located at offset 0x5A. The ASCII view displays the file's content as 'forensicsclass'. The status bar at the bottom right shows the file size as 40 B. The status bar also indicates that the file is 40 Bytes in size.

Figure 19: The short file name at $0x5A$ from the first $0x30$ attribute.

- Long file names are found at offset $0x5A$ from the **second** $0x30$ attribute. Our long file name is `forensicsclass.txt`.

The screenshot shows a hex editor window for a file named 'forensicsclass.txt' located at offset 0x5A. The ASCII view displays the file's content as 'forensicsclass'. The status bar at the bottom right shows the file size as 40 B. The status bar also indicates that the file is 40 Bytes in size.

Figure 20: The long file name at $0x5A$ from the second $0x30$ attribute.

- Is this file a resident file or nonresident file? Where can you find the evidence?

- The resident/nonresident flag exists at offset $0x08$ from attribute $0x80$. In this case we can see it is a **resident file**. This makes sense because it is only 40 Bytes in size.

Figure 21: The resident/non-resident flag set to `0x00`, meaning resident.

8. Did you find the hidden message in the file when you check the MFT record?

- Yes, it was not difficult to find the hidden message. It lies inside of a second `0x80` attribute, and is easily found by looking at the `ascii` screen on WinHex.

desktop.ini

10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
03	00	02	00	38	00	01	00	60	02	00	00	04	00	00	00
05	00	00	00	00	00	00	00	10	00	00	60	00	00	00	00
25	E7	1D	FA	2E	98	D5	01	D8	3A	A1	00	31	98	D5	01
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
38	19	23	08	00	00	00	00	30	00	00	78	00	00	00	00
70	01	00	00	00	02	00	00	25	E7	1D	FA	2E	98	D5	01
FE	6F	38	FA	2E	98	D5	01	00	00	00	00	00	00	00	00
0C	02	46	00	4F	00	52	00	45	00	4E	00	53	00	7E	00
30	00	00	00	80	00	00	00	00	00	00	00	02	00	00	00
25	E7	1D	FA	2E	98	D5	01	FE	6F	38	FA	2E	98	D5	01
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
65	00	EE	00	73	00	69	00	63	00	73	00	63	00	6C	00
40	00	00	00	28	00	00	00	00	00	00	00	00	04	00	00
B7	7D	08	00	27	FE	95	9C	80	00	00	40	00	00	00	00
57	65	20	77	69	6C	6C	20	68	61	76	65	20	61	20	66
6E	20	4D	6F	6E	64	61	79	80	00	00	00	60	00	05	00
73	00	65	00	63	00	72	00	65	00	74	00	00	00	00	00
64	2C	20	74	68	65	6E	20	79	6F	75	20	61	72	65	20
64	20	0D	AA	00	00	00	00	FF	FF	FF	FF	82	79	47	11
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
<															

ANSI ASCII

FILE0 *^z, 8 -`
H %ç ú.-ö Ø; i-ö
Ø; i-ö @ m-i-ö
< Ø # o x
z p %ç ú.-ö
bosú.-ö bosú.-ö bosú.-ö
F O R E N S ~
1 . T X T o €
f p %ç ú.-ö bosú.-ö
bosú.-ö bosú.-ö
for e n s i c s c l
a s s . t x t @ (zä®(è-) 'b@e' @
(We will have a f
orensics class on Monday@`
 | (secret
If you study hard, then you are
likely to succeed yyyyy,yyG

Drive C:
File system:
[Read-only mode]

Data Interpreter

Logical sector No.: 6,339,271
Physical sector No.: 7,465,671

Used space: 15,0 GiB
Free space: 34,5 GiB
Total capacity: 49,5 GiB
Bytes per cluster: 4,096
Free clusters: 9,044,384
Total clusters: 12,966,144
Bytes per sector: 512
Sector count: 103,729,144
Physical disk:
Mode:
Sector offset: C1758E08 - C1758E50 | Size: 4

Figure 22: The secret message contained in the second `0x80` attribute.

- More specifically, it's located in the data run for the second `0x80` attribute at offset `0x18`.

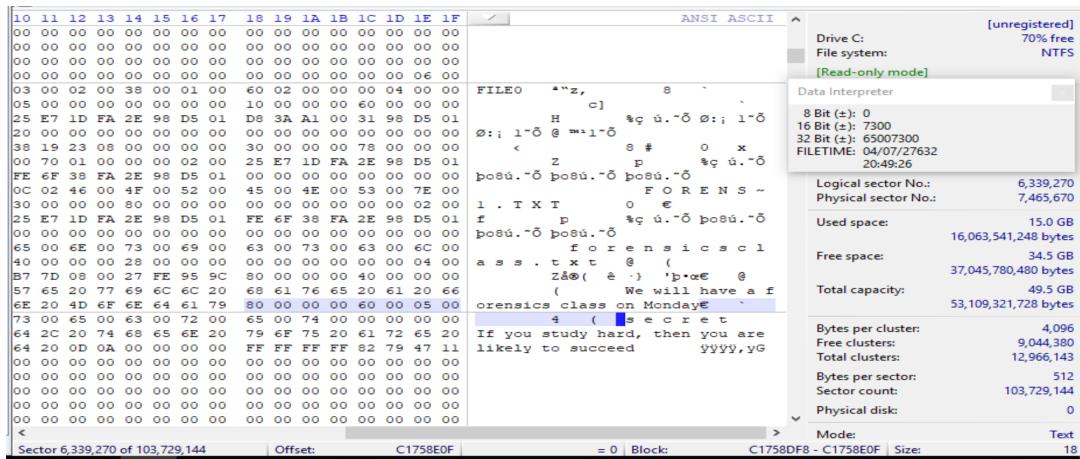


Figure 23: Secret message contained inside of the data run for the second `0x80` attribute.

9. How many `0x80` attributes does this file have? What is the possible reason?

- The reason for this would be the **hidden data stream**. This creates an additional `0x80` attribute for the stream. We can verify this by going to offset `0x18` for the second `0x80` attribute. This is where the data run is for resident files. This contains the secret message.

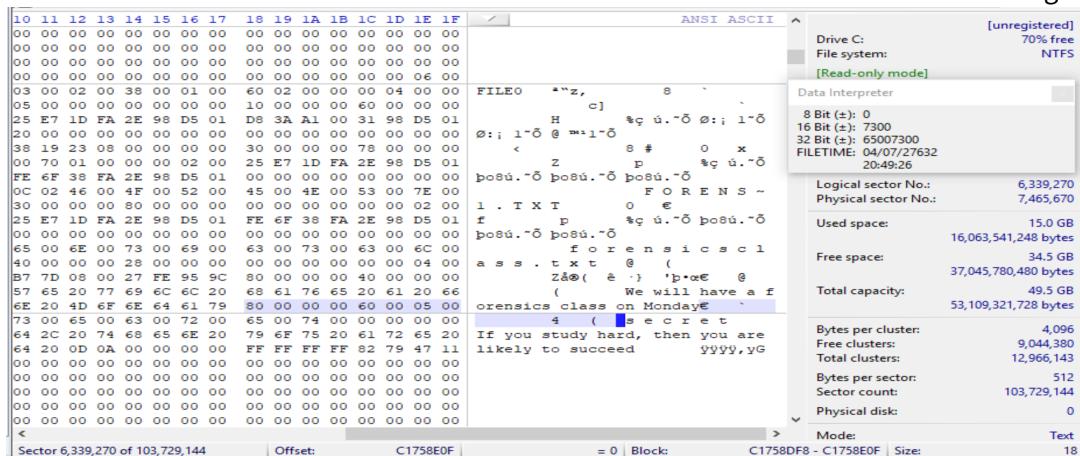


Figure 24: Secret message contained inside of the data run for the second `0x80` attribute.

Part 2: Analyze a given MFT record

Given the MFT record below, please answer the questions from 10-15.

Figure 25: Answer questions 10-15.

Questions for Part 2

10. Is this file a resident file or nonresident file? Where can you find the evidence?

- The resident/nonresident flag exists at offset `0x08` from attribute `0x80`. In this case we can see it is a **nonresident file**, as the flag is `0x01`.

Figure 26: Nonresident file flag.

11. How many data runs does this file have?

- This file has two data runs.

Figure 27: Bytes underlining start of data runs in red, and remainders in black.

12. What is the starting cluster address value for the first data run (LCN)?

- The starting cluster address value is `0x0C0000`. We multiply this by the cluster size, which is 4096 in decimal or `0x1000` in hexadecimal. So the cluster address value for the first data run is $0x0C0000 * 0x1000 = 0xC00000000$.

$$0x0C0000 * 0x1000 = 0xC00000000.$$

Figure 28: Starting LCN address for first data run.

13. How many clusters are assigned to the first data run?

- The number of clusters assigned to the first data run is `0x00C820`.

00	C0	00	00	00	00	00	06	49	4C	45	30	00	03	00	C8	3D	45	B1	00	00	00	00	00	00
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Figure 29: Number of clusters assigned to the first data run.

14. Does this file have other data runs? If yes, what is the starting cluster address value for the second data run (LCN)?

- Yes there is a second data run. The starting cluster address value is `0x00C16AAE`. We multiply this by the cluster size, which is 4096 in decimal or `0x1000` in hexadecimal. So the cluster address value for the second data run is `0x00C16AAE * 0x1000 = 0xC16AAE0000`.

00C0000000	46	49	4C	45	30	00	03	00	C8	3D	45	B1	00	00	00	00
00C0000020	00	00	00	00	00	00	00	00	07	00	00	00	00	00	00	00

Figure 30: Starting LCN address for second data run.

15. How many clusters are assigned to the second data run?

- The number of clusters assigned to the second data run is **0x3C20**.

	FILEO	È=È±	8	,
00C000000000	16 49 4C 45 30 00 03 00	C8 3D 45 B1 00 00 00 00	A0 01 00 00 00 04 00 00	
00C000002000	00 00 00 00 00 00 00 00	07 00 00 00 00 00 00 00	DC 00 00 00 00 00 00 00	H 0
00C000004000	00 18 00 00 00 00 00 00	48 00 00 00 18 00 00 00	CD 05 B3 AF 55 1D D4 01	íô»-u ò íô»-u ò íô»-u ò
00C000006000	CD 05 B3 AF 55 1D D4 01	CD 05 B3 AF 55 1D D4 01	CD 05 B3 AF 55 1D D4 01	íô»-u ò íô»-u ò íô»-u ò
00C000008000	00 00 00 01 00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	30 00 00 68 00 00 00
00C00000A000	00 18 00 00 03 00 00 00	4A 00 00 00 18 00 01 00	05 00 00 00 00 05 00 00	CD 05 B3 AF 55 1D D4 01
00C00000C000	CD 05 B3 AF 55 1D D4 01	CD 05 B3 AF 55 1D D4 01	CD 05 B3 AF 55 1D D4 01	íô»-u ò íô»-u ò íô»-u ò
00C00000E000	40 00 00 00 00 00 00 00	06 00 00 00 00 00 00 00	04 03 24 00 4D 00 46 00	íô»-u ò íô»-u ò íô»-u ò
00C000010000	00 00 00 50 00 00 00 00	01 00 40 00 00 06 00 00	00 00 00 00 00 00 00 00	3F 04 01 00 00 00 00 00
00C000012000	00 00 00 00 00 00 00 00	00 00 44 10 00 00 00 00	00 00 44 10 00 00 00 00	é P ?
00C000014033	3D C8 00 00 00 0C 42	íô»-u 3C AB 6A C1 00 00 00	B0 00 00 00 48 00 00 00	00 00 00 00 00 00 00 00
00C000016000	00 00 00 00 00 00 00 00	09 00 00 00 00 00 00 00	40 00 00 00 00 00 00 00	3 È B <ójÁ D D D
00C000018008	00 00 00 00 00 00 00 00	08 90 00 00 00 00 00 00	21 0A 66 51 00 00 00 00	é ! fQ yyy
00C00001A0FF	FF FF FF 00 00 00 00 00	FF FF FF FF 00 00 00 00	FF FF FF FF 00 00 00 00	FF FF FF FF 00 00 00 00
00C00001C000	00 00 00 00 00 00 00 00	01 00 00 00 00 00 00 00	40 00 00 00 00 00 00 00	FF FF FF FF 00 00 00 00
00C00001E008	10 00 00 00 00 00 00 00	08 10 00 00 00 00 00 00	31 01 FF FF 0B 11 01 FF	00 00 00 00 00 00 00 00
00C0000200FF	FF FF FF 00 00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	DC 00 00 00 00 00 00 00
				yyyy

Figure 31: Number of clusters assigned to the second data run.