

Activity 6: Recovering Graphics Files

Objectives:

- Split and combine files in Linux.
- Use WinHex to recover graphics files.

Part 1: Split and combine files in Linux.

Since the Winhex evaluation version cannot process a file that is bigger than 200KB, you need to split a file into pieces in order to edit and save a big file. So the first thing we have to do is download 'funpicture' on canvas onto CAINE. The firefox on CAINE wouldn't let me access the files on canvas so I had to download it another way.

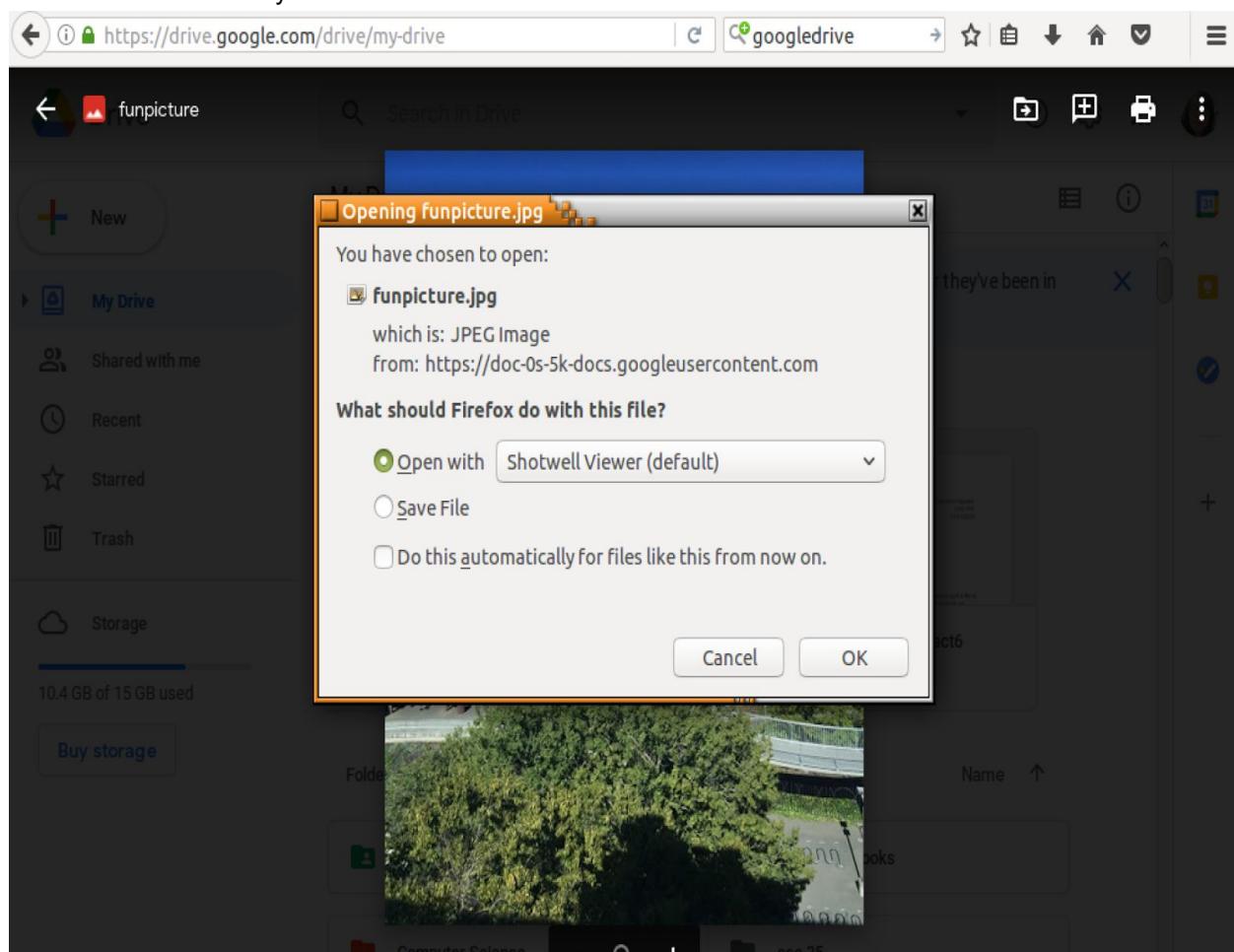


Figure 1. Downloading funpicture.jpg from google drive.

Once the download is complete it is time to open up the terminal and locate the directory in which that file is stored. This will allow us to start our splitting.

```
caineacf@caineacf:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
caineacf@caineacf:~$ cd Downloads
caineacf@caineacf:~/Downloads$ ls
4Basic Forensic Analysis using Autopsy.pdf    image_zero.dd
case.aut                                     investigators.txt
case.log                                      vol1-C..boring.xlsx
funpicture.jpg                                vol1-C..where.were.you.docx
image_zero(1).dd.part
caineacf@caineacf:~/Downloads$
```

Figure 2. Changing the directory to where funpicture.jpg is downloaded.

Now that we are in the designated directory, we must split the file via command ‘split -b 19000 funpicture’. Now because I have to download this file another way I need to add the full name of the file which is funpicture.jpg. Note: I changed the location of the file because I didn’t want the splits to get mixed up.

```
root@caineacf:/home/caineacf/Downloads/fp# ls
funpicture.jpg
root@caineacf:/home/caineacf/Downloads/fp# split -b 19000 funpicture.jpg
root@caineacf:/home/caineacf/Downloads/fp# ls
funpicture.jpg xai xar xba xbj xbs xcb xck xct xdc xdl xdu xed xem xev xfe
xaa           xaj xas xbb xbk xbt xcc xcl xcu xdd xdm xdv xee xen xew xff
xab           xak xat xbc xbl xbu xcd xcm xcv xde xdn xdw xef xeo xex
xac           xal xau xbd xbm xbv xce xcn xcw xdf xdo xdx xeg xep xey
xad           xam xav xbe xbn xbw xcf xco xcx xdg xdp xdy xeh xeq xez
xae           xan xaw xbf xbo xbx xcg xcp xcy xdh xdq xdz xei xer xfa
xaf           xao xax xbg xbp xby xch xcq xcz xdi xdr xea xej xes xfb
xag           xap xay xbh xbp xbz xci xcr xda xdj xds xeb xek xet xfc
xah           xaq xaz xbi xbr xca xcj xcs xdb xdk xdt xec xel xeu xfd
root@caineacf:/home/caineacf/Downloads/fp#
```

Figure 3. Splitting funpicture.jpg file, and listing the results from the split.

After splitting the image we need to combine the file pieces to a new file by typing the command ‘cat x*>newfun’.

```
root@caineacf:/home/caineacf/Downloads/fp# cat x*>newfun
root@caineacf:/home/caineacf/Downloads/fp# ls
funpicture.jpg xah xaq xaz xbi xbr xca xcj xcs xdb xdk xdt xec xel xeu xfd
newfun        xai xar xba xbj xbs xcb xck xct xdc xdl xdu xed xem xev xfe
xaa           xaj xas xbb xbk xbt xcc xcl xcu xdd xdm xdv xee xen xew xff
xab           xak xat xbc xbl xbu xcd xcm xcv xde xdn xdw xef xeo xex
xac           xal xau xbd xbm xbv xce xcn xcw xdf xdo xdx xeg xep xey
xad           xam xav xbe xbn xbw xcf xco xcx xdg xdp xdy xeh xeq xez
xae           xan xaw xbf xbo xbx xcg xcp xcy xdh xdq xdz xei xer xfa
xaf           xao xax xbg xbp xby xch xcq xcz xdi xdr xea xej xes xfb
xag           xap xay xbh xbp xbz xci xcr xda xdj xds xeb xek xet xfc
root@caineacf:/home/caineacf/Downloads/fp#
```

Figure 4. Combining the image with the cat command.

Finally, we need to rename the combined file to include the .jpg extension, by running the command ‘mv newfile newfun.jpg’

```
root@cainecf:/home/cainecf/Downloads/fp# mv newfun newfun.jpg
root@cainecf:/home/cainecf/Downloads/fp# ls
funpicture.jpg  xah  xaq  xaz  xbi  xbr  xca  xcj  xcs  xdb  xdk  xdt  xec  xel  xeu  xfd
newfun.jpg      xai  xar  xba  xbj  xbs  xcb  xck  xct  xdc  xdl  xdu  xed  xem  xev  xfe
xaa            xaj  xas  xbb  xbk  xbt  xcc  xcl  xcu  xdd  xdm  xdv  xee  xen  xew  xff
xab            xak  xat  xbc  xbl  xbu  xcd  xcm  xcv  xde  xdn  xdw  xef  xeo  xex
xac            xal  xau  xbd  xbm  xbv  xce  xcn  xcw  xdf  xdo  xdx  xeg  xep  xey
xad            xam  xav  xbe  xbn  xbw  xcf  xco  xcx  xdg  xdp  xdy  xeh  xeq  xez
xae            xan  xaw  xbf  xbo  xbx  xcg  xcp  xcy  xdh  xdq  xdz  xei  xer  xfa
xaf            xao  xax  xbg  xbp  xby  xch  xcq  xcz  xdi  xdr  xea  xej  xes  xfb
xag            xap  xay  xbh  xbz  xci  xcr  xda  xdj  xds  xeb  xek  xet  xfc
root@cainecf:/home/cainecf/Downloads/fp#
```

Figure 5. Renamed newfun to newfun.jpg.

Part 2: Practice recovering graphics files using Winhex.

We start by downloading ‘smallsmallsmall’ from canvas and save it into a folder. The header is a .png file, but its been modified by the suspect.

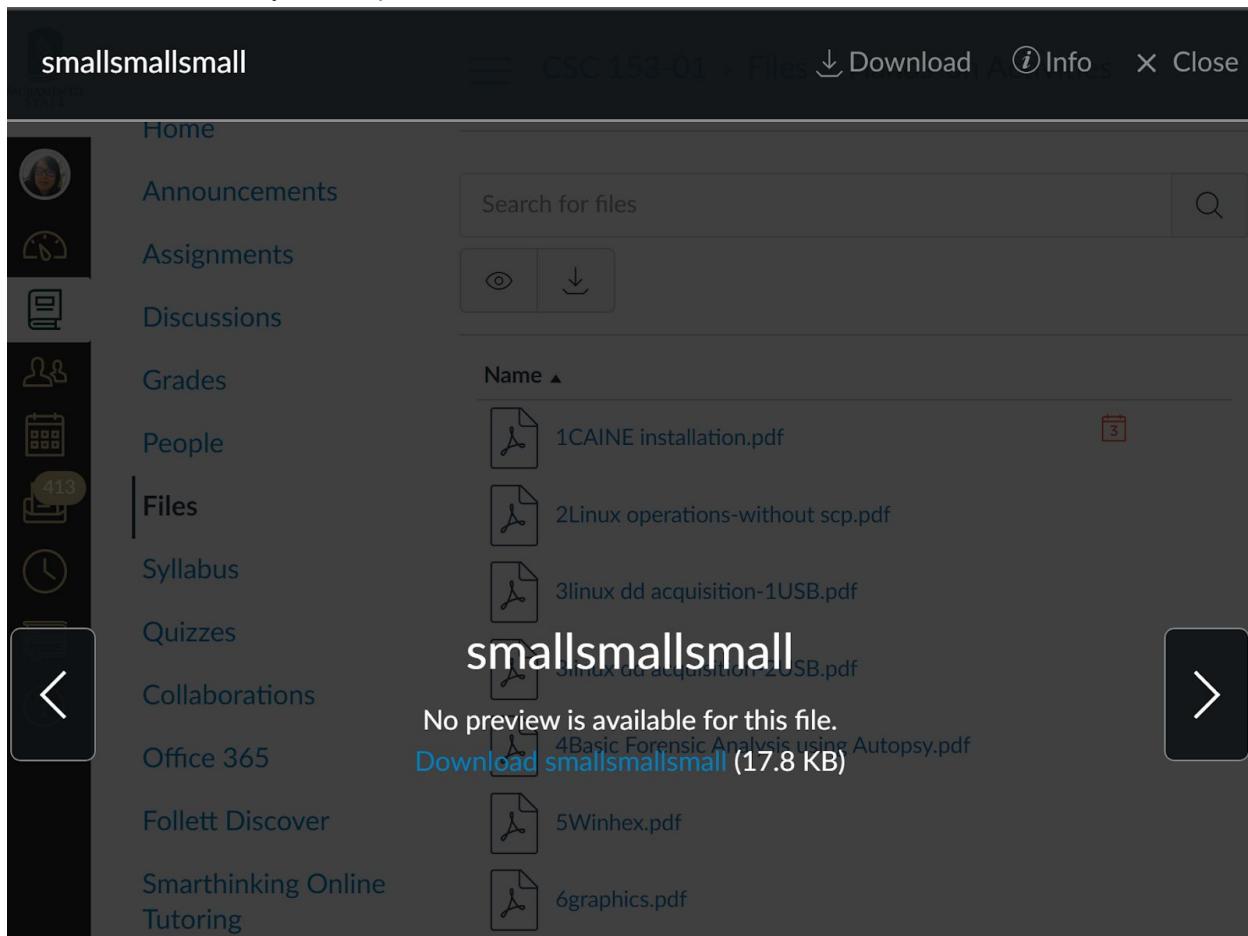


Figure 6. Downloading smallsmallsmall to our local machine.

Now we need to open up WinHex and open it as administrator. As a safety precaution, we can click on options and edit mode. Selecting the Read-Only Mode and click ok.

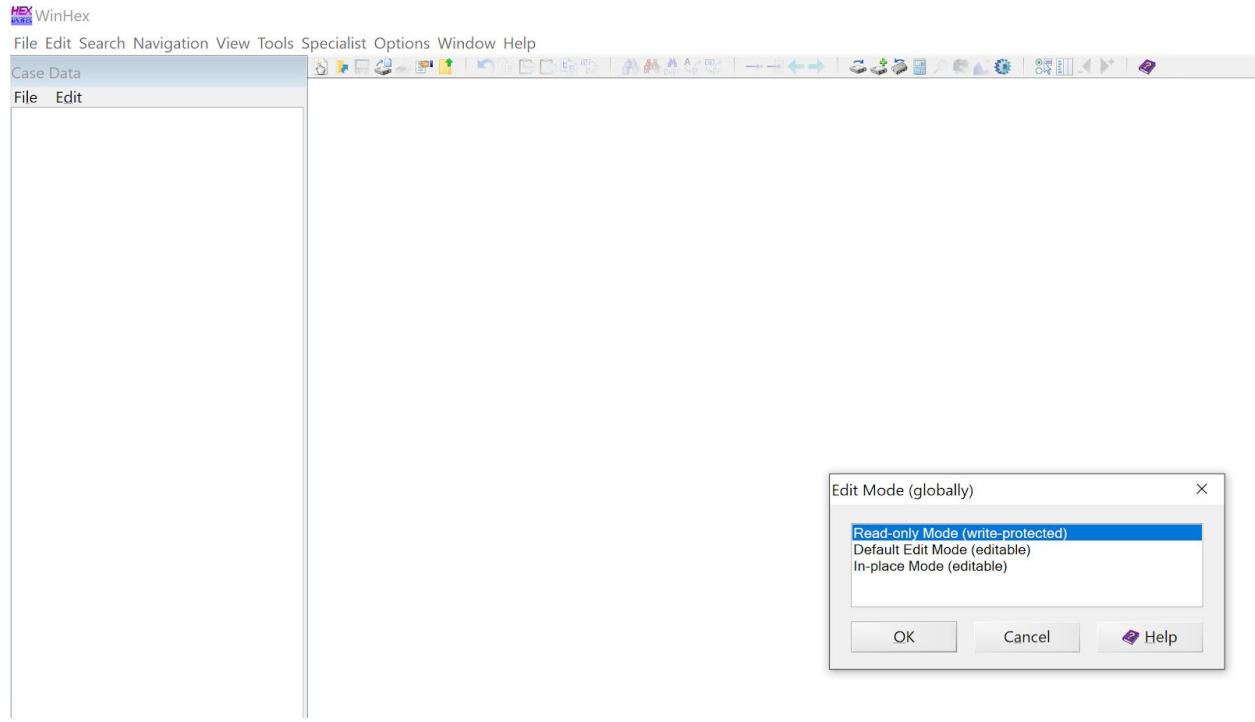


Figure 7. Selecting the read-only mode for safety.

Next, we need to click on Tools and Open Disk. Then we can select the drive in which the file 'smallsmallsmall' is located so we can examine it. In this case, the drive will be C:. After clicking ok WinHex will then begin traversing the drive. This may take a while depending on the size of the drive.

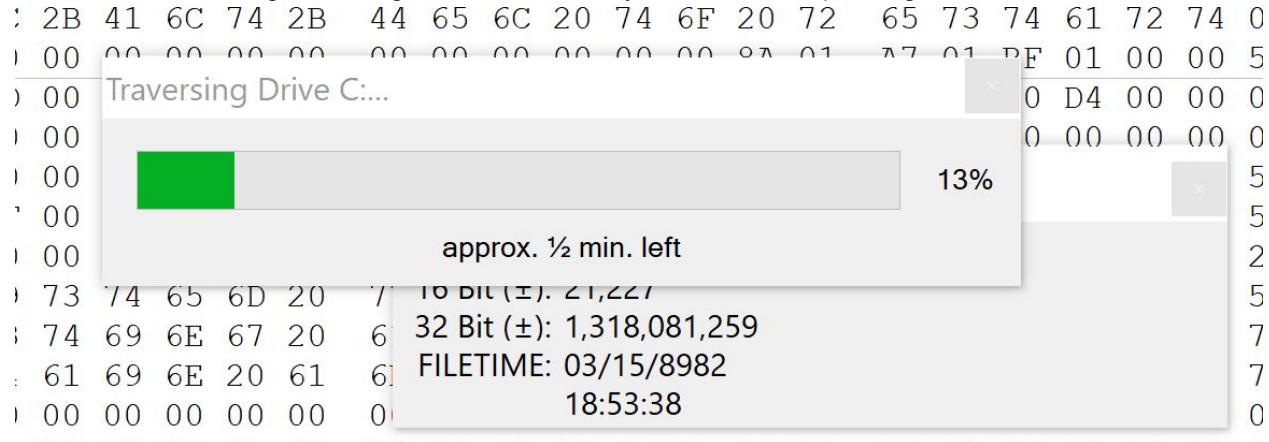


Figure 8. Winhex is traversing the C: drive.

After the traversing is complete we can begin scrolling down to find the \$MFT file, right-click it and choose open. Then open the file in a separate window to examine. This simulates a real investigation because we already know where the file is located so doing this method is preferred for practice.

Drive C: \$MFT	Offset	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	FILE
000000000000	46 49 4C 45 30 00 03 00 36 B3 55 93 0C 00 00 00 01 00 01 00 38 00 01 00 D0 01 00 00 00 04 00 00	FILE	
00000000032	00 00 00 00 00 00 00 00 00 00 0D 00 00 00 00 00 00 67 03 00 00 00 00 00 10 00 00 00 60 00 00 00		
00000000064	00 00 18 00 00 00 00 00 48 00 00 00 18 00 00 00 4E B4 D2 7F 80 2A D5 01 4E B4 D2 7F 80 2A D5 01		
00000000096	4E B4 D2 7F 80 2A D5 01		
00000000128	00 00 00 00 00 01 00 N'Ò		
00000000160	00 00 18 00 00 00 03 00 4A 00 00 00 18 00		
00000000192	4E B4 D2 7F 80 2A D5 01		
00000000224	00 00 08 3B 00 00 00 00 00 06 00	N'Ò ;	
00000000256	80 00 00 00 78 00 00 00 01 00 40 00		
00000000288	40 00		
00000000320	33 00 96 00 00 00 00 0C 33 0F C8 00 51 55 4 n/a n/a		
00000000352	2A 42 2C 02 32 00 59 DE C7 E7 42 40 25 3D BD 5E FE 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	B0 00 00 00 50 00 *B,	
00000000384	01 00 40 00 00 00 0B 00	@	
00000000416	00 00 02 00 00 00 00 00 40 F8 01 00 00 00 00 00 40 F8 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00	21 1E 07 1F 41 01 F6 DA	
00000000448	49 04 31 01 3B 4B 2B 00 FF FF FF FF 00		
00000000512	00 00		
00000000544	00 00		
00000000576	00 00		
00000000608	00 00		
00000000640	00 00		
00000000672	00 00		
0000000704	00 00		
0000000736	00 00		
0000000768	00 00		
0000000800	00 00		
0000000832	00 00		
0000000864	00 00		
0000000896	00 00		
0000000928	00 00		
0000000960	00 00		
0000000992	00 00		
0000001024	46 49 4C 45 30 00 03 00 40 14 00 02 00		

Figure 9. \$MFT is open in a separate window.

The next task is to find the smallsmallsmall file. In the \$MFT, characters in the file name is usually separated by hexadecimal value 00. The hexadecimal value for the word small is 73 6d 61 6c 6c. Separating each character with 00 it becomes 73 00 6d 00 61 00 6c 00 6c. By repeating this three times to get smallsmallsmall it then becomes 73006D0061006C006C0073006D0061006C006C00

Knowing this we can then search for the following hex values in the \$MFT file. To do so click on Search -> Find Hex Values.

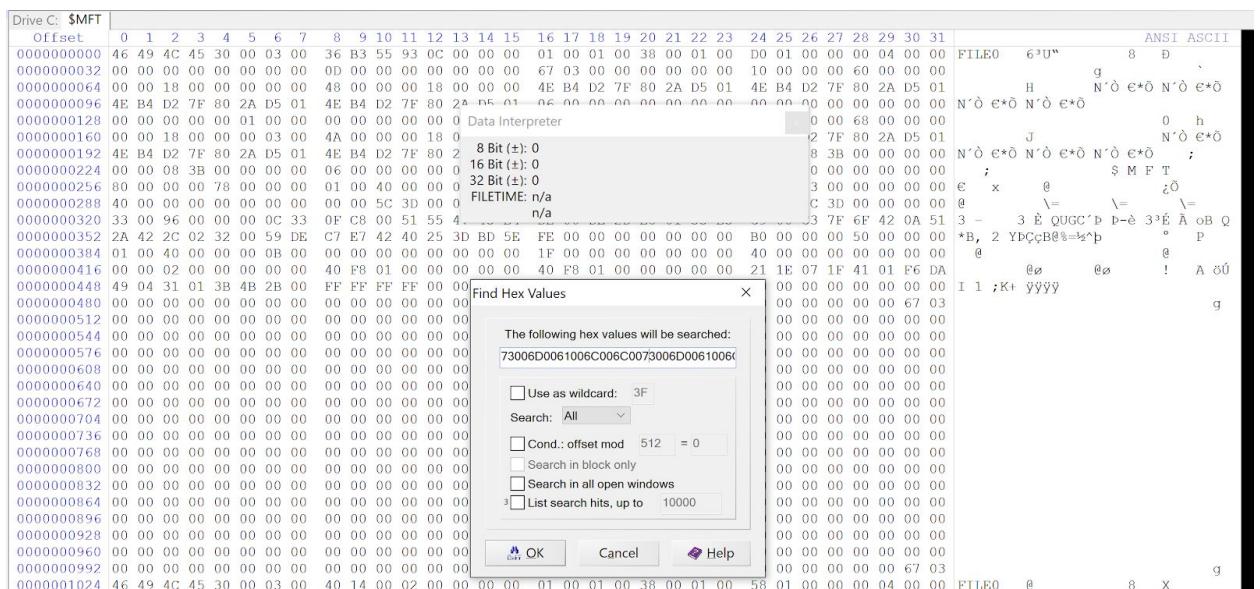


Figure 10. Searching for the hexadecimal title for smallsmallsmall.

After the search, we can see that we got a hit.

Figure 11. Hit for hexadecimal value.

By following the same methodology as in hands-on activity 5 and lab 2 to examine this \$MFT file record. You need to find out the data runs for this file. In the demo, the first data run information is “41 05 37 52 55 04”. This means the starting position of the first data run is “0x 04 55 52 37” and the size is “0x05”. In this case, we can find the start of the data run for this file by looking at offset 0x40 from 0x80.

Figure 12. First data run info is 31 01 07 A5 04 00.

The first data run is 31 01 07 A5 04 00, which means the starting position is 0x 00 04 A5 07, and the size 0x05. So next, we open the window for the C: drive once again, and click Navigation then go to offset. We'll ender the offset appending three zeros to the end of it.

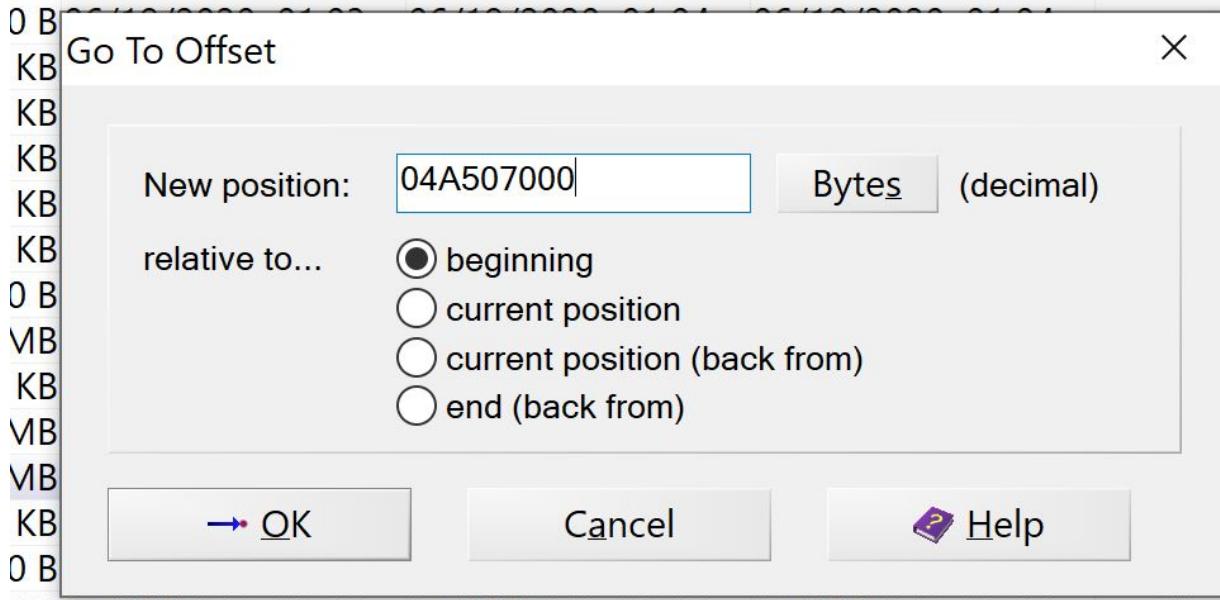


Figure 13. The first data run at offset 04A507000.

Now we're taken to the beginning pg the smallsmallsmall file.

Figure 14. Beginning of smallsmallsmall file.

Since the size of the first data run is 0x05, we can click on Navigation then go to offset again to find the end of the first data run. Setting 05000 as our position and choosing the current position will take us to the ending point for the first data run. In our case, the end of the first data run is at offset 0x004A50C000

\$MFT												982 MB 06/24/2019 03:32... 06/24/2019 03:32... 06/24/2019 03:32... SH 6,291,456																				
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
004A50C000	63	6F	70	79	72	69	67	68	74	3D	54	68	65	20	53	6F	66	74	77	61	72	65	20	69	73	20	74	72	61	64	65	
004A50C000	61	72	6B	65	64	20	61	6E	64	20	43	6F	70	79	72	69	67	68	74	20	A9	20	32	30	31	37	20	47	65	6E	75	69
004A50C000	74	65	63	2C	20	AC	4C	43	2F	20	32	32	32	31	20	4A	75	73	74	69	6F	20	52	64	20	23	31	31	39	2D	33	34

Figure 15. End of the data run.

Starting from the beginning of the file again, we click on the first byte of the file and drag until the offset is 5000, which is the size of the data run.

\$MFT	982 MB 06/24/2019 03:32... 06/24/2019 03:32... 06/24/2019 03:32... SH 6,291,456																																	
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	v	ANSI ASCII
004A507000	5B	5A	6F	6E	65	54	72	61	6E	73	66	65	72	5D	0D	0A	5A	6F	6E	65	49	64	3D	33	0D	0A	52	65	66	65	72	72	[ZoneTransfer] ZoneId=3 Referr	
004A507020	65	72	55	72	6C	3D	68	74	74	70	73	3A	2F	2F	63	73	75	73	2B	69	6E	73	74	72	75	63	74	75	72	65	2B	63	erUrl=https://csus.instructure.c	
004A507040	6F	6D	2F	0D	0A	48	62	73	74	55	72	6C	3D	68	74	74	70	73	3A	2F	2F	63	64	6E	2E	69	6E	73	74	2D	66	73	cm/ HostUrl=https://cdn.instruc	
004A507060	2D	69	61	64	2D	70	72	6F	64	2E	69	6E	73	63	6C	6F	75	64	67	61	74	65	2E	66	65	74	2F	31	65	62	31	65	-iad-prod.inscloudgate.net/leble	
004A507080	33	61	32	2D	30	64	37	65	2D	34	34	61	64	2D	38	38	39	32	2D	32	64	61	39	38	34	39	63	34	63	31	34	2F	3a2-0d7e-44ad-8892-2da9849c4c14/	
004A5070A0	73	6D	61	6C	6C	73	6D	61	6C	6C	73	6D	61	6C	6C	3F	74	6F	6B	65	6E	3D	65	79	4A	68	62	47	63	69	4F	69	smallsmallsmall?token=eyJhbGciOi	
004A5070C0	42	49	55	7A	55	78	4D	69	49	73	49	65	52	35	63	43	49	36	49	6B	70	58	56	43	49	73	49	6D	74	70	5A	43	JIUz0xM1sInR5cCI61kpXVCIsImtpZC	
004A5070E0	49	36	49	6D	4E	6B	62	69	4A	39	2E	65	79	4A	79	5A	58	4E	76	64	58	4A	6A	53	49	36	49	63	38	78	5A	I61mNkb1j9.eyJyZXNvdXJkJZSI6i18xZ		
004A507100	57	49	78	5A	54	4E	68	4D	69	30	77	5A	44	60	6C	4C	54	51	30	59	57	51	74	4F	44	67	35	4D	69	30	79	5A	WiZTNhMiowZDdlLTQ0YWQtODg5M1oYz	
004A507120	47	45	35	4F	44	51	35	59	7A	52	6A	4D	54	51	76	63	32	31	68	62	47	78	7A	62	57	46	73	62	48	4E	74	59	GE50DQ5YzRjMTQvc21hbGxzbfWsHbHN	
004A507140	57	78	73	49	69	77	64	47	56	75	59	57	35	30	49	6A	6F	69	59	32	46	75	64	6D	46	7A	49	69	77	69	64	WxsIwidGVuYW50Ijo1Y2FudmFzIwid		
004A507160	58	4B	6C	63	6C	39	70	5A	43	49	36	49	65	45	78	4D	68	60	35	4D	44	41	77	44	41	74	47	80	4D	40	Xnlc1p9ZC161jExMjk5MDAwMDAwMDAxM			
004A507180	7A	55	34	4D	79	49	73	49	6D	6C	68	64	43	49	36	4D	54	59	77	4E	64	53	40	44	59	78	4F	73	77	69	5A	ZU4MyIsImlhCI6MTWnjEOMDYxOCwIZ		
004A5071A0	58	68	77	49	6A	6F	78	4E	6A	41	32	4D	6A	49	33	44	45	34	66	51	2E	50	64	74	49	67	58	33	59	4D	34	XbwIjoxNjA2MjI3MDExf4q.PdtIgX3YM4		
004A5071C0	67	55	76	50	64	55	52	50	62	55	6F	49	52	61	69	71	53	67	58	58	46	62	6F	68	4C	50	46	53	55	38	5F	4A	gUvPdURPBu0IRaiqSgXXFbhlPFCU8	
004A5071E0	68	33	34	76	63	50	38	4E	42	33	4B	33	5F	6A	41	73	35	49	70	48	65	37	6A	42	4C	32	75	69	44	4C	33	4C		
004A507200	45	50	64	36	66	35	4D	69	54	45	47	78	77	26	64	6F	77	6E	6C	6F	61	64	3D	31	26	63	6F	74	65	6E	74	5F	EPdf5M1TEwx&download=1&content	
004A507220	74	79	70	70	65	3D	6E	75	6C	0D	0A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	type=null	
004A507240	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

Figure 15. Selecting everything from the first data run.

Next, we right-click and choose edit copy block into a new file.

\$MFT	982 MB 06/24/2019 03:32... 06/24/2019 03:32... 06/24/2019 03:32... SH 6,291,456																																	
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	v	ANSI ASCII
004A507000	5B	5A	6F	6E	65	54	72	61	6E	73	66	65	72	5D	0D	0A	5A	6F	6E	65	49	64	3D	33	0D	0A	52	65	66	65	72	72	[ZoneTransfer] ZoneId=3 Referr	
004A507020	65	72	55	72	6C	3D	68	74	74	70	73	3A	2F	2F	63	73	75	73	2E	69	6E	73	74	72	75	63	74	75	72	65	2E	63	erUrl=https://csus.instructure.c	
004A507040	6F	6D	2F	0D	0A	48	6F	73	74	55	72	6C	3D	68	74	74	70	73	3A	2F	63	64	6E	2E	69	6E	73	74	2D	66	73	cm/ HostUrl=https://cdn.instruc		
004A507060	2D	69	61	64	2D	70	72	6F	64	2E	69	6E	73	63	6C	6F	75	64	67	61	74	65	2E	66	65	74	2F	31	65	62	31	65	-iad-prod.inscloudgate.net/leble	
004A507080	33	61	32	2D	30	64	37	65	2D	34	34	61	64	2D	38	38	39	32	2D	32	64	61	39	38	34	39	63	34	63	31	34	2F	3a2-0d7e-44ad-8892-2da9849c4c14/	
004A5070A0	73	6D	61	6C	6C	73	6D	61	6C	6C	73	6D	61	6C	6C	3F	74	6F	6B	65	6E	3D	65	79	4A	68	62	47	63	69	4F	69	smallsmallsmall?token=eyJhbGciOi	
004A5070C0	42	49	55	7A	55	78	4D	69	49	73	49	65	52	35	63	43	49	36	49	6B	70	58	56	43	49	73	49	6D	74	70	5A	43	JIUz0xM1sInR5cCI61kpXVCIsImtpZC	
004A5070E0	49	36	49	6D	4E	6B	62	69	4A	39	2E	65	79	4A	79	5A	58	4E	76	64	58	4A	6A	53	49	36	49	63	38	78	5A	I61mNkb1j9.eyJyZXNvdXJkJZSI6i18xZ		
004A507100	57	49	78	5A	54	4E	68	4D	69	30	77	5A	44	64	6C	4C	54	51	30	59	57	51	74	4F	44	67	35	4D	69	30	79	5A	WiZTNhMiowZDdlLTQ0YWQtODg5M1oYz	
004A507120	47	45	35	4F	44	51	35	59	7A	52	6A	4D	54	51	76	63	32	31	68	62	47	78	7A	62	57	46	73	62	48	4E	74	59	GE50DQ5YzRjMTQvc21hbGxzbfWsHbHN	
004A507140	57	78	73	49	69	77	64	47	56	75	59	57	35	30	49	6A	6F	69	59	32	46	75	64	6D	60	46	74	49	69	77	69	64	WxsIwidGVuYW50Ijo1Y2FudmFzIwid	

Figure 16. Copy block to a new file.

Drive C:\ \$MFT new	982 MB 06/24/2019 03:32... 06/24/2019 03:32... 06/24/2019 03:32... SH 6,291,456																																	
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	v	ANSI ASCII
00000000	5B	5A	6F	6E	65	54	72	61	6E	73	66																							

Drive C: \$MFT	new	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
00000000	89 50 40 47	65	54	72	61	6E	73	66	65	72	5D	0D	0A	5A	6F	6E	65	49	64	3D	33	0D	0A	52	65	66	65	72	72				
00000020	65	72	55	72	6C	3D	68	74	74	70	73	3A	2F	2F	63	73	75	73	2E	69	6E	73	74	72	75	63	74	75	72	65	2E	63	
00000040	6F	6D	2F	0D	0A	48	6F	73	74	55	72	6C	3D	68	74	74	70	73	3A	2F	2F	63	64	6E	2E	69	6E	73	74	2D	66	73	
00000060	2D	69	61	64	2D	70	72	6F	64	2E	69	6E	73	63	6C	6F	75	64	67	61	74	65	2E	6E	65	74	2F	31	65	62	31	65	
00000080	33	61	32	2D	30	64	37	65	2D	34	34	61	64	2D	38	38	39	32	2D	32	64	61	39	38	34	39	63	34	63	31	34	2F	
000000A0	73	6D	61	6C	73	6D	61	6C	6C	73	6D	61	6C	6C	3F	74	6F	6B	65	6E	3D	65	79	4A	68	62	47	63	69	4F	69		
000000C0	4A	49	55	7A	55	78	4D	69	49	73	49	6E	52	35	63	43	49	36	49	6B	70	58	56	43	49	73	49	6D	74	70	5A	43	
000000E0	49	36	49	6D	4E	6B	62	69	4A	39	2E	65	79	4A	79	5A	58	4E	76	64	58	4A	6A	5A	53	49	36	49	69	38	78	5A	
00000100	57	49	78	5A	54	4E	68	4D	69	30	77	5A	44	64	6C	4C	54	51	30	59	57	51	74	4F	44	67	35	4D	69	30	79	5A	
00000120	47	45	35	4F	44	51	35	59	7A	52	6A	4D	54	51	76	63	32	31	68	62	47	78	7A	62	57	46	73	62	48	4E	74	59	
00000140	57	78	73	49	69	77	69	64	47	56	75	59	57	35	30	49	6A	6F	69	59	32	46	75	64	6D	46	7A	49	69	77	69	64	
00000160	58	4E	6C	63	6C	39	70	5A	43	49	36	49	6A	45	78	4D	6A	6B	35	4D	44	41	77	4D	44	41	78	4F	44	77	69	5A	
00000180	7A	55	34	4D	79	49	73	49	6D	6C	68	64	43	49	36	4D	54	59	77	4E	6A	45	30	4D	44	59	78	4F	43	77	69	57	

Figure 18. Replace the file header with the proper header for PNG.

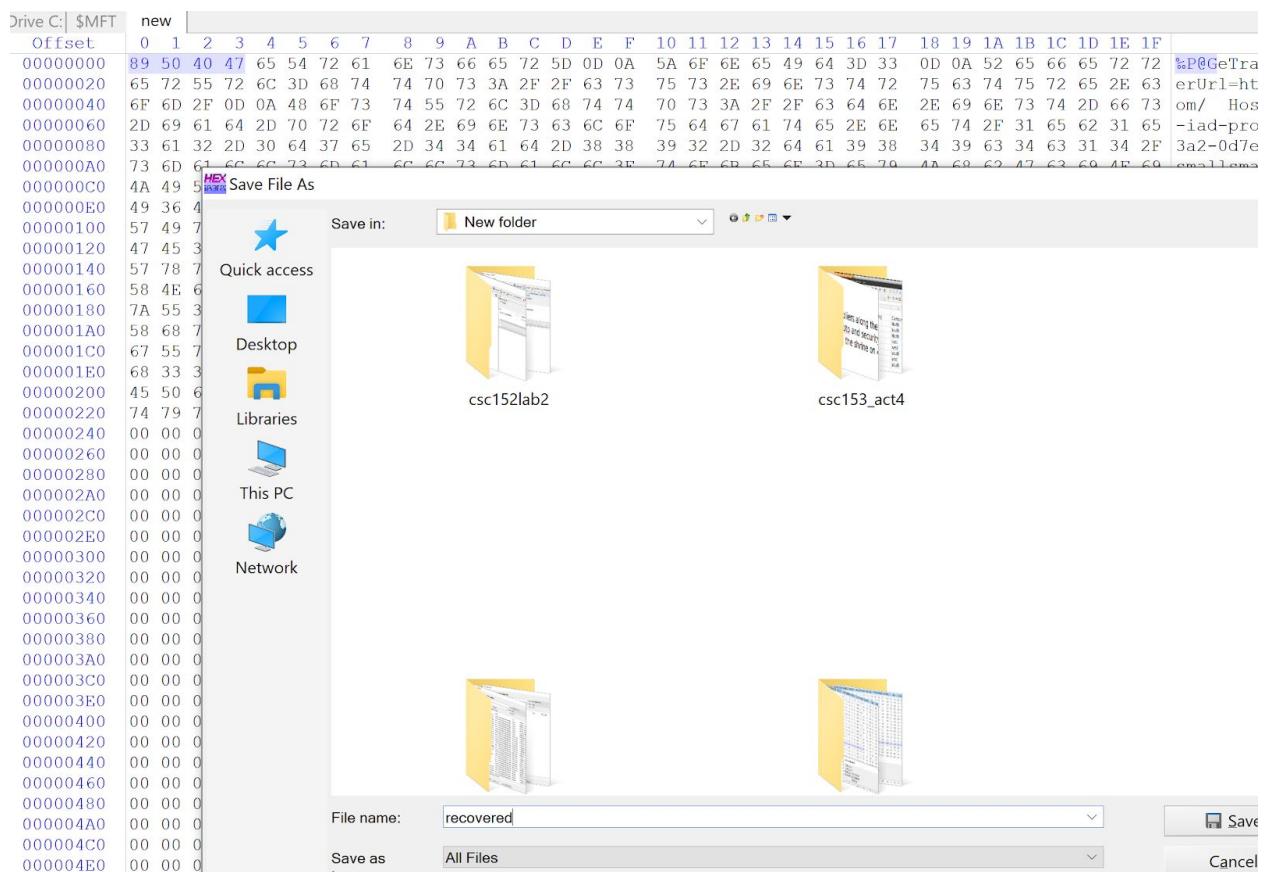


Figure 19. Saving the file as recovered.

Now once we add the .png extension, we can open the file recovered.png and see that it's a 58x75 pixel version of the image from Part 1.

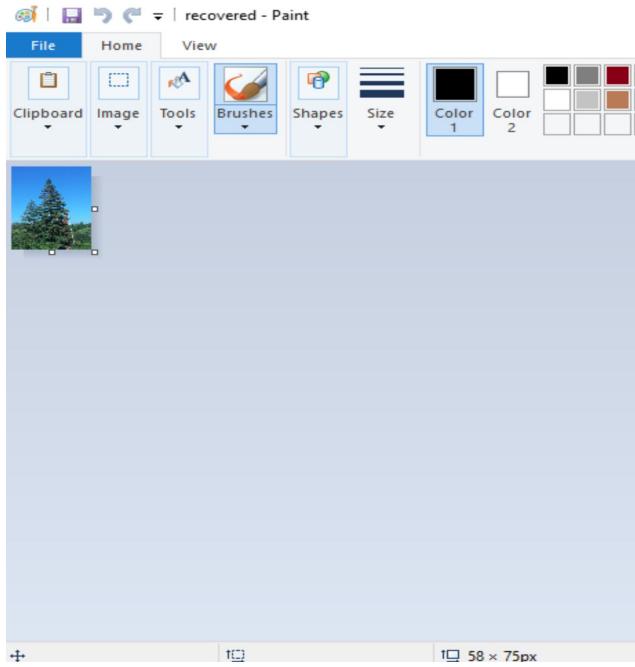


Figure 20. Recovered png.

Questions:

1. What hexadecimal values did you use to search for the file “smallsmallsmall” in step 13 and 15?
 - a. The hexadecimal value searched for was

73006D0061006C006C0073006D0061006C006C0073006D0061006C00

6C00

Figure 21. Searching for smallsmallsmall.

2. Is the file a resident file or a non-resident file? How do you know?

- a. This is a non-resident file, we can tell by looking at offset 0x08 from attribute 0x80 and see the flag is set to 0x01.

Figure 22. None resident flag.

3. How many data runs does this file have?

- a. Only had one data run for the activity.

```

0053203904 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 Data Interpreter
0053203936 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0053203968 46 49 AC 45 30 00 03 00 88 D8 27 1F D0 0 8 Bit (±): 0
0053204000 00 00 00 00 00 00 00 00 00 00 00 F5 C 16 Bit (±): 0
0053204032 00 00 00 00 00 00 00 00 48 00 00 00 18 0 32 Bit (±): -16.777.216
0053204064 35 A4 B5 50 F1 C1 D6 01 E8 0B 58 3B DA C ? FILETIME: ?
0053204096 00 00 00 00 E3 0B 00 00 00 00 00 00 00 00 00 00
0053204128 00 00 00 00 00 00 0C 00 52 00 00 00 18 00 01 00 7E 8E 08 00 00 00 00 D0 00 DE D5 E0 39 DA C1 D6 01
0053204160 EB 08 58 3B DA C1 D6 01 E8 08 58 3B DA C1 D6 01 E8 08 58 3B DA C1 D6 01 00 50 00 00 00 00 00 00 00 00 00
0053204192 41 47 00 00 00 00 00 00 20 00 00 00 00 00 00 00 08 02 53 00 4D 00 41 00 4C 00 4C 00 53 00 7E 00
0053204224 31 00 00 00 00 00 00 00 30 00 00 00 78 00 00 00 00 00 00 00 00 00 00 00 00 0B 00 60 00 00 00 18 00 01 00
0053204256 7E 8E 08 00 00 00 0D DF D5 E0 39 DA C1 D6 01 E8 0B 58 3B DA C1 D6 01 E8 0B 58 3B DA C1 D6 01
0053204288 EB 08 58 3B DA C1 D6 01 00 50 00 00 00 00 00 00 41 47 00 00 00 00 00 00 20 00 00 00 00 00 00 00 00 00
0053204320 0F 01 73 00 0D 60 01 6C 00 60 00 73 00 60 00 61 00 6C 00 6C 00 73 00 6D 00 61 00 6C 00 6C 00
0053204352 80 00 00 00 50 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0053204384 40 00 00 00 00 00 00 00 00 50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0053204416 21 01 80 3A 41 03 64 49 70 01 31 01 31 01 3B 5D 15 00 80 00 00 68 00 00 01 0F 40 00 00 00 00 0A 00
0053204448 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 60 00 00 00 00 00 00 00 00 10 00 00 00 00 03 00
0053204480 2B 02 00 00 00 00 00 00 2B 02 00 00 00 00 00 00 5A 00 6F 00 6E 00 65 00 2E 00 49 00 64 00 65 00
0053204512 06 E6 74 00 69 00 66 00 69 00 65 00 72 00 00 31 01 07 A5 04 00 00 00 FF FF FF FF 82 79 47 11
0053204544 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
0053204576 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
FILEO ^`* x 8 @
          öß
          H Böða9úóð è x;úáð
5uþPnñáð è x;úáð
          a e #~ 0 p
          R ~Z Böða9úóð
          è x;úáð è x;úáð P
          AG S M A L L S ~
          l 0 x
          -ž Böða9úóð è x;úáð è x;úáð
          è x;úáð P AG
          s m a l l s m a l l s m a l l
          € P
          @ P AG AG
          ! :A dI p 1 ; ) h @
          + + Zone . I d e
          n t i f i e r 1 Y yyyý, yG

```

Figure 23. One data run.

4. What is the starting position for the first data run?

- a. The starting position for the data run is 0x0004A507

```

0053203904 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 Data Interpreter 0 00 00 00 00 00
0053203936 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0053203968 46 49 AC 45 30 00 03 00 88 D0 27 1F OD 0 8 Bit(±): 0 0 00 00 00 20 00
0053204000 00 00 00 00 00 00 00 00 00 00 00 F5 C 16 Bit(±): 0 0 00 00 04 00
0053204032 00 00 00 00 00 00 00 00 00 48 00 00 18 32 Bit(±): -16,777,216 0 00 00 60 00
0053204064 35 A4 B5 50 F1 C1 D6 01 E8 0B 58 3B DA C FILETIME: ? 8 3B DA C1 D6 01
0053204096 00 00 00 00 E3 0B 00 00 00 00 00 00 00 00 00 00 ? 0 00 00 70 00
0053204128 00 00 00 00 00 00 0C 00 52 00 00 18 00 01 00 7E 8E 08 00 00 00 00 D0 00 DE D5 E0 39 DA C1 D6 01
0053204160 EB 08 58 3B DA C1 D6 01 E8 08 58 3B DA C1 D6 01 E8 0B 58 3B DA C1 D6 01 00 50 00 00 00 00 00 00 00 00
0053204192 41 47 00 00 00 00 00 00 20 00 00 00 00 00 00 00 08 02 53 00 4D 00 41 00 4C 00 45 00 53 00 7E 00
0053204224 31 00 00 00 00 00 00 00 30 00 00 00 78 00 00 00 00 00 00 00 00 00 00 00 0B 00 60 00 00 00 18 00 01 00
0053204256 7E 8E 08 00 00 00 00 DF D5 E0 39 DA C1 D6 01 E8 0B 58 3B DA C1 D6 01 E8 0B 58 3B DA C1 D6 01
0053204288 E8 0B 58 3B DA C1 D6 01 00 50 00 00 00 00 41 47 00 00 00 00 00 00 20 00 00 00 00 00 00 00 00 00 00
0053204320 0F 01 73 00 6D 00 61 00 6C 00 6D 00 73 00 6D 00 61 00 6C 00 6C 00 73 00 6D 00 61 00 6C 00 6C 00
0053204325 80 00 00 00 50 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0053204384 40 00 00 00 00 00 00 00 00 50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0053204416 21 01 80 3A 41 03 64 49 70 01 31 01 31 01 3B 5D 15 00 80 00 00 68 00 00 00 01 0F 40 00 00 00 00 0A 00
0053204448 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 60 00 00 00 00 00 00 00 00 10 00 00 00 00 03 00
0053204480 2B 02 00 00 00 00 00 00 2B 02 00 00 00 00 00 00 5A 00 6F 00 6E 00 65 00 2E 00 49 00 64 00 65 00
0053204512 6E 06 74 00 69 00 66 00 69 00 65 00 72 00 00 31 01 07 A5 04 00 00 00 FF FF FF FF 82 79 47 11
0053204544 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
0053204576 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
FILEO ^` x 8 @
          öß
          H Bööäööäö è x;üöö
5uµPñäöö è x;üöö
          a € #^ 0 p
          R ~Z Bööäööäö
          è x;üöö è x;üöö P
          AG S M A L L S ~
          l 0 x
          ~Z Bööäööäö è x;üöö è x;üöö
          è x;üöö P AG
          s m a l l s m a l l s m a l l
          € P AG AG
          @ ! : A d I p 1 ; ) h @
          + + Zone. I de
          n t i f i e r 1 Y yyyy,yG

```

b.

c. Figure 24. The starting position for the first data.

5. What is the size of the first data run?

- a. The size of the first data run is 0x05. On the bottom of WinHex size would be 5000.

6. In step 20, what is the header of the file “smallsmallsmall” which you downloaded from Canvas? Please provide the first 4 bytes of the hexadecimal values. Please take a screenshot to show the evidence.

- a. The header of the file downloaded from canvas was 5B 5A 6F 6E.

Drive C:	\$MFT	new		ANSI ASCII																													
		Offset		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
00000000	5B	5A	6F	6E	65	54	72	61	6E	73	66	65	72	5D	0D	0A	5A	6F	6E	65	49	64	3D	33	0D	0A	52	65	66	65	72	72	
00000020	65	72	55	72	6C	3D	68	74	74	70	73	3A	2F	2F	63	73	75	73	2E	69	66	73	74	72	75	63	74	75	72	66	2E	63	
00000040	6F	6D	2F	0D	0A	48	6F	73	55	72	6C	3D	68	74	70	73	3A	2F	2F	63	64	6E	2E	69	6E	73	74	2D	66	73	om/	HostUrl=https://cdn.inst-fs	
00000060	2D	69	61	64	2D	70	72	6F	64	2E	69	63	73	63	6C	6F	75	64	67	61	74	65	2E	66	65	74	2F	31	65	62	31	55	
00000080	33	61	32	3D	20	64	37	65	2D	34	3A	61	62	3D	38	39	32	2D	32	64	61	39	38	34	39	63	34	33	31	34	2F	32-0d+2e-44d=8892-2da984c14	
000000A0	73	6D	61	6C	73	6D	61	6C	73	6D	61	6C	63	74	6F	6B	65	6E	3D	65	79	4A	68	62	47	63	69	4F	69	smallsmallsmall1?token=eyJhbGciOi			
000000C0	4A	49	55	7A	55	78	4D	69	49	73	49	6B	52	35	63	43	49	36	49	6B	70	58	56	43	49	73	49	6D	70	54	JIU2uRnR5C1L6jVXCs1ImpTzC		
000000E0	49	36	49	6D	4E	6B	62	69	4A	39	2E	65	79	74	59	58	4E	76	64	58	4A	6A	5A	53	49	36	49	38	78	54	I6lNmKnbj9J.eyJyJxZNSwNxJxZS16i16gj		
00000100	57	49	78	5A	54	4E	68	4D	69	30	77	5A	44	64	6C	4C	54	51	30	59	57	51	74	4F	44	67	35	4D	69	30	79	5A	
00000120	47	45	35	4F	41	51	35	59	7A	52	6A	4D	51	76	63	32	31	60	62	47	78	7A	57	46	73	62	48	4E	54	69	56	G95ED0G5YzR MTQyc21hbzbwFsFbHNy	
00000140	57	78	73	49	69	77	69	64	47	76	75	59	57	35	30	49	6A	6F	69	59	32	46	75	64	6D	46	7A	69	49	77	69	64	WxsLiwidGvW50J1oyZyF2mdPfliwN

Figure 25. The header for small small small, could be wrong.

7. What should be the correct header of a PNG file?

- The correct header for a PNG file is 89 50 4E 47.

Drive C:\ \$MFT	new	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	
Offset																																		
00000000		8	9	50	40	47	65	54	72	61	6E	73	66	65	72	5D	0D	0A	5A	6F	6E	65	49	64	3D	33	0D	0A	52	65	66	65	72	72
00000020		65	72	55	72	6C	3D	68	74	74	70	73	3A	2F	2F	63	73	75	73	2E	69	6E	73	74	72	75	63	74	75	72	65	2E	63	
00000040		6F	6D	2F	0D	0A	48	6F	73	74	55	72	6C	3D	68	74	74	70	73	3A	2F	2F	63	64	6E	2E	69	6E	73	74	2D	66	73	
00000060		2D	69	61	64	2D	70	72	6F	64	2E	69	6E	73	63	6C	6F	75	64	67	61	74	65	2E	6E	65	74	2F	31	65	62	31	65	
00000080		33	61	32	2D	30	64	37	65	2D	34	34	61	64	2D	38	38	39	32	2D	32	64	61	39	38	34	39	63	34	63	31	34	2E	
000000A0		73	6D	61	6C	6C	73	6D	61	6C	6C	73	6D	61	6C	6C	3F	74	6F	6B	65	6E	3D	65	79	4A	68	62	47	63	69	4F	69	
000000C0		4A	49	55	7A	55	78	4D	69	49	73	49	6E	52	35	63	43	49	36	49	6B	70	58	56	43	49	73	49	6D	74	70	5A	43	
000000E0		49	36	49	6D	4E	6B	62	69	4A	39	2E	65	79	4A	79	5A	58	4E	76	64	58	4A	6A	5A	53	49	36	49	69	38	78	5A	
00000100		57	49	78	5A	54	4E	68	4D	69	30	77	5A	44	64	6C	4C	54	51	30	59	57	51	74	4F	44	67	35	4D	69	30	79	5A	
00000120		47	45	35	4F	44	51	35	59	7A	52	6A	4D	54	51	76	63	32	31	68	62	47	78	7A	62	57	46	73	62	48	4E	74	59	
00000140		57	78	73	49	69	77	69	64	47	56	75	59	57	35	30	49	6A	6F	69	59	32	46	75	64	6D	46	7A	49	69	77	69	64	
00000160		58	4E	6C	63	6C	39	70	5A	43	49	36	49	6A	45	78	4D	6A	6B	35	4D	44	41	77	4D	44	41	77	4D	44	41	78	4I	
00000180		7A	55	34	4D	79	49	73	49	6D	6C	68	64	43	49	36	4D	54	59	77	4E	6A	45	30	4D	44	59	78	4F	43	77	69	5I	

Figure 26. Correct header for PNG.

8. Take a screenshot to show the recovered file.



Figure 27. recovered.png