AUBURN

UNIVERSITY

Midterm

Kate Moreland

kem0149

9/29/25

Executive Summary

The following report shows the process of extracting files from a disk image that was taken from an art theft. The files were recoved from one FAT16 partition and one NTFS partition.

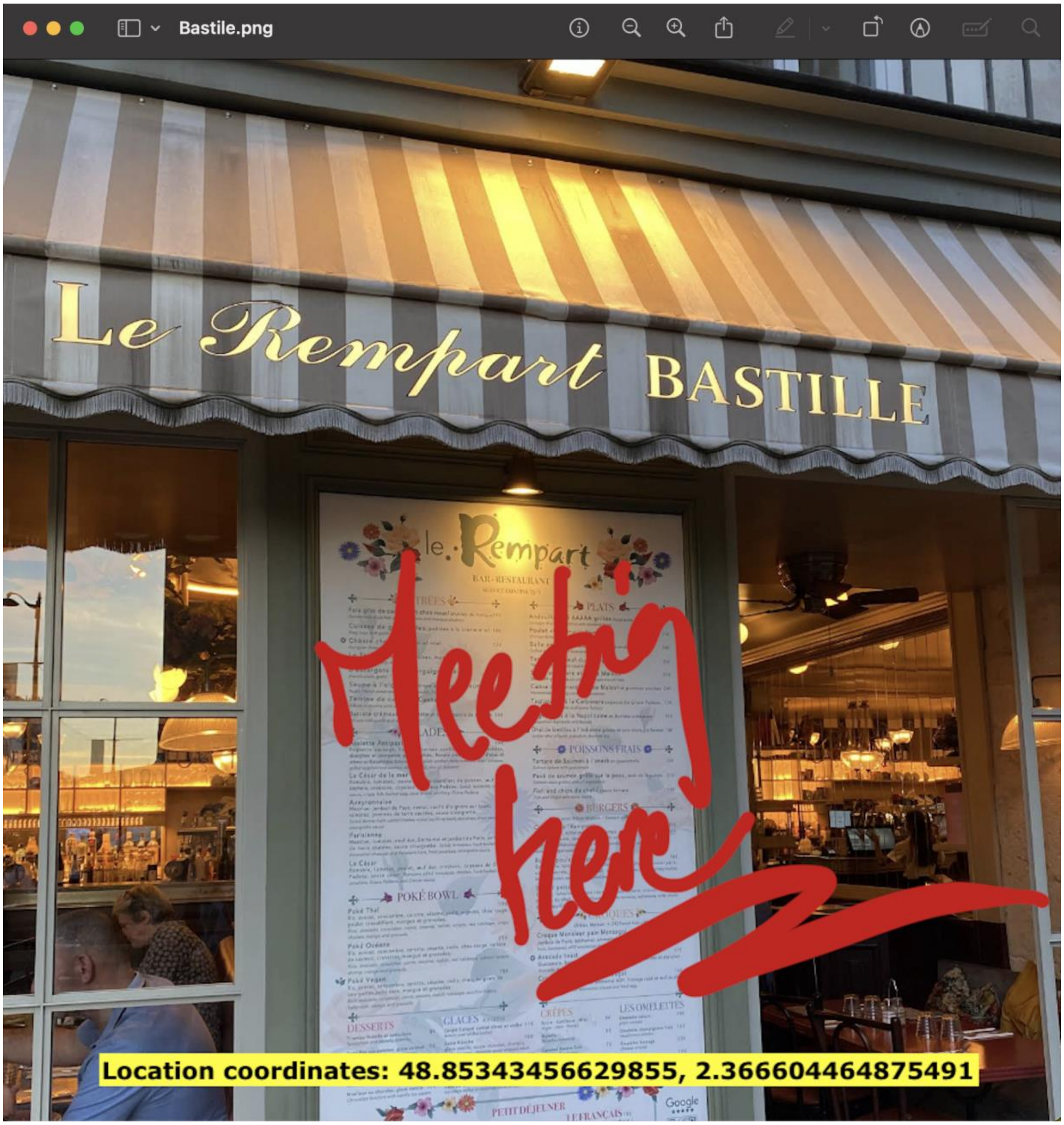
Table of Contents

This section should begin on a new page. The table of contents below is generated using the "References" menu.

Table of Contents

List of Figures

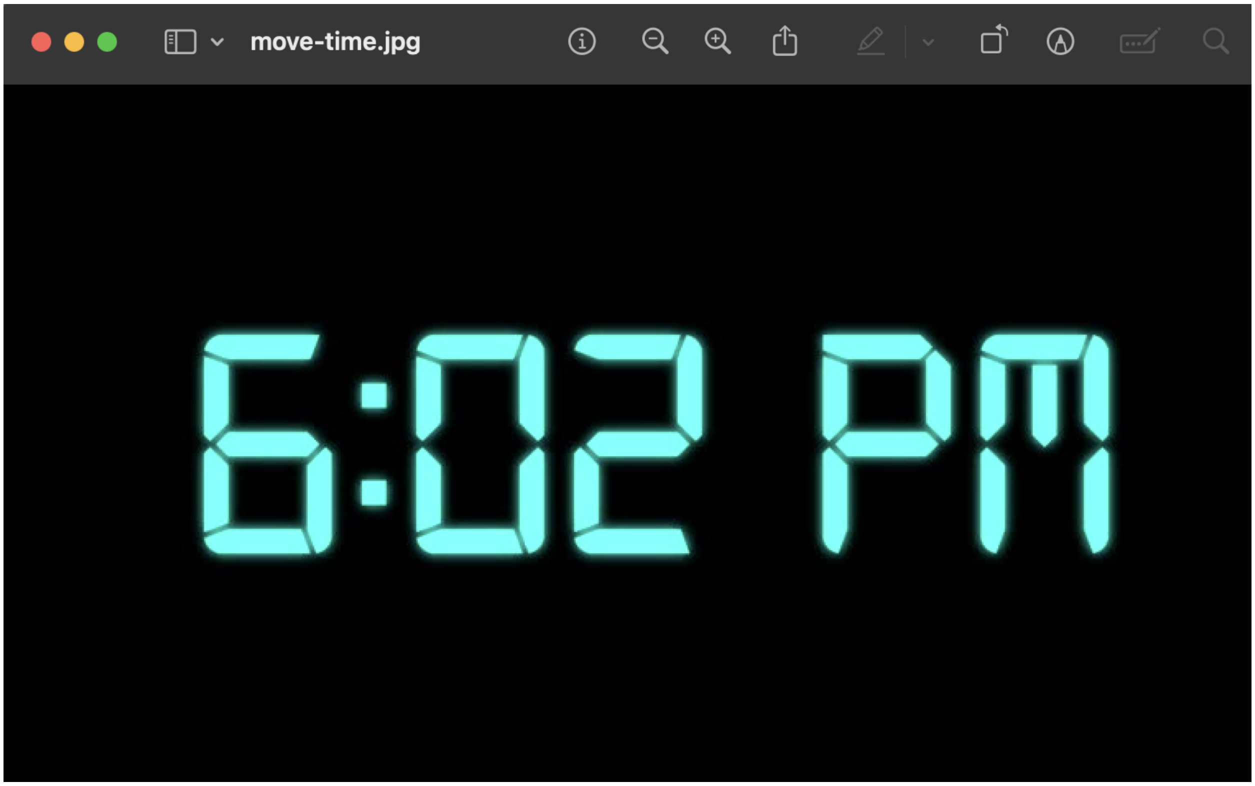
Bastile.png



Location coordinates: 48.85343456629855, 2.366604464875491

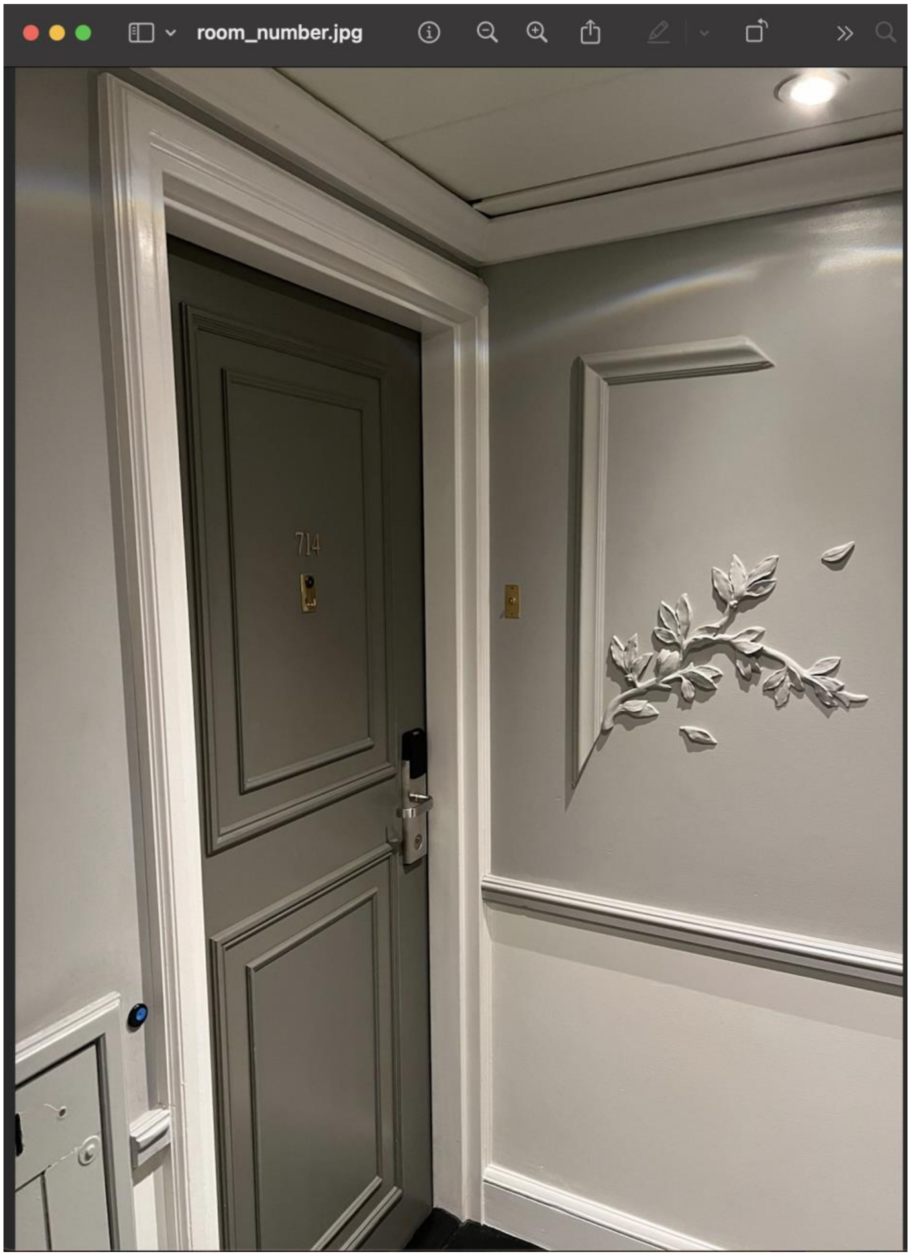
Registrar.zip

move-time.png (in registrar)

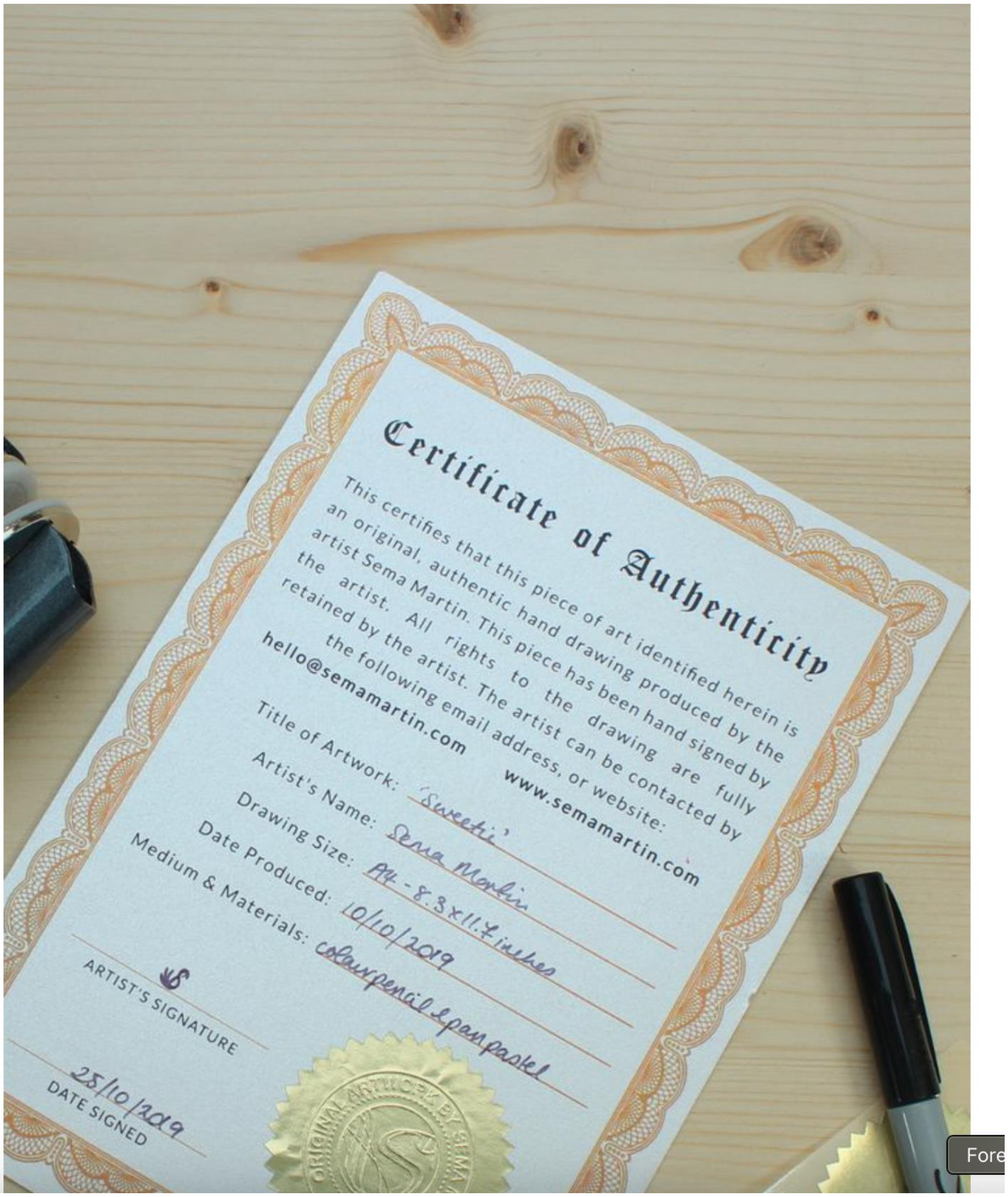


6:02 PM

room_number.jpg (in registrar)



sema_martin.pdf



Certificate of Authenticity

This certifies that this piece of art identified herein is an original, authentic hand drawing produced by the artist Sema Martin. This piece has been hand signed by the artist. All rights to the drawing are fully retained by the artist. The artist can be contacted by the following email address, or website:

hello@semamartin.com **www.semamartin.com**

Title of Artwork: 'Sweeti'

Artist's Name: Sema Martin

Drawing Size: A4 - 8.3x11.7 inches

Date Produced: 10/10/2019

Medium & Materials: colourpencil & panpastel

ARTIST'S SIGNATURE

25/10/2019

DATE SIGNED

ORIGINAL ARTWORK BY SEMA

List of Tables

Disk Information	Reserved Area	1st FAT area	2nd FAT area	Root Discovery	Data Area
0-2048	2048-2052	2052-2152	2152-2252	2252-2284	2284-104448

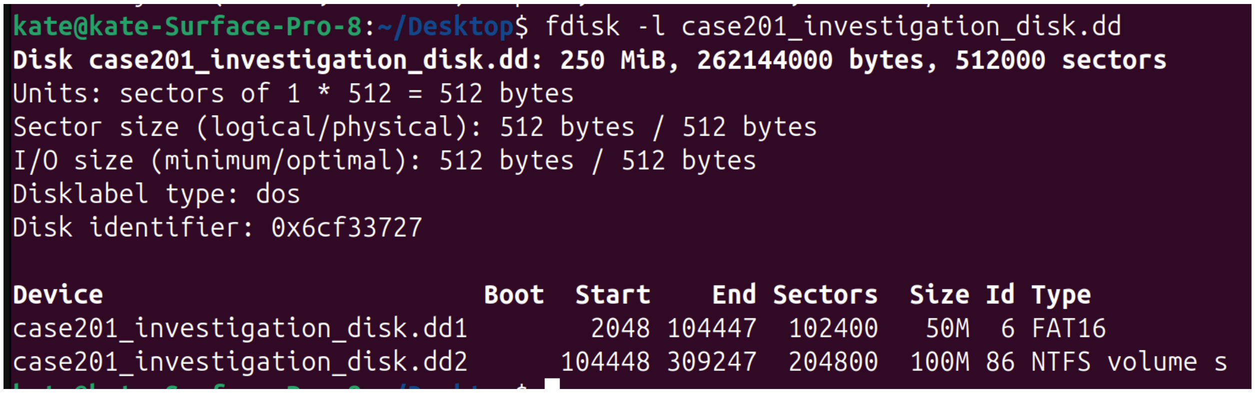
File	Confirmation Command	Recovery Command
Bastile.png	hexdump -C -s $((2288*512)) -n ((2656*512)) case201_investigation_disk.dd	dd if=case201_investigation_disk.dd of=Bastile.png bs=512 skip=2288 count=2656
Registrar.zip	hexdump -C -s $((4944*512)) -n ((402*512)) case201_investigation_disk.dd	dd if=case201_investigation_disk.dd of=Registrar.zip bs=512 skip=4944 count=402
Cert.zip	hexdump case201_investigation_disk.dd -s $((210944*512)) -n $((288*512))	dd if=case201_investigation_disk.dd of=Cert.zip bs=512 skip=210944 count=288
Location.txt		dd if=case201_investigation_disk.dd of=location.txt bs=512 skip=$((53560656+8))

		count=512
		iflag=skip_bytes,count_bytes

File Name	Starting Offset	Ending Offset
Bastile.png	1171456	2531328
Registrar.zip	2531328	2738176
Cert.zip	53559296	53559716

1 Detailed Analysis of Partition Allocation

There are two partitions in the image. One is FAT16 and one is NTFS. The sectors before the FAT16 partition are 2048 and the sectors before the NTFS are 104448. Below is an image of the partitions with the descriptions of where the partitions start and end.

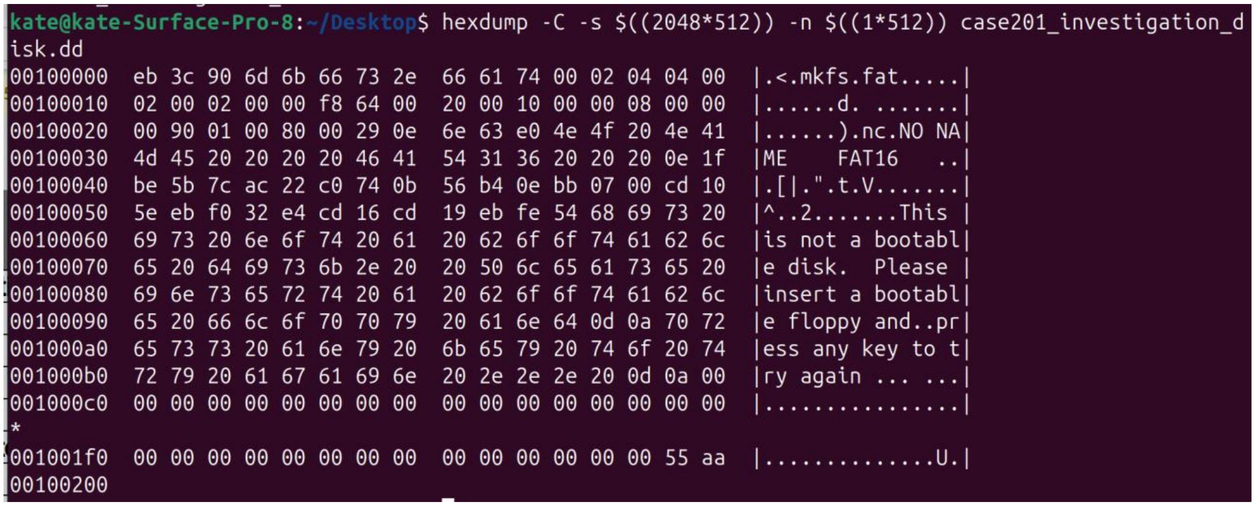


```
kate@kate-Surface-Pro-8:~/Desktop$ fdisk -l case201_investigation_disk.dd
Disk case201_investigation_disk.dd: 250 MiB, 262144000 bytes, 512000 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x6cf33727

Device                          Boot  Start    End Sectors  Size Id Type
case201_investigation_disk.dd1         2048 104447  102400   50M  6 FAT16
case201_investigation_disk.dd2       104448 309247  204800  100M 86 NTFS volume s
```

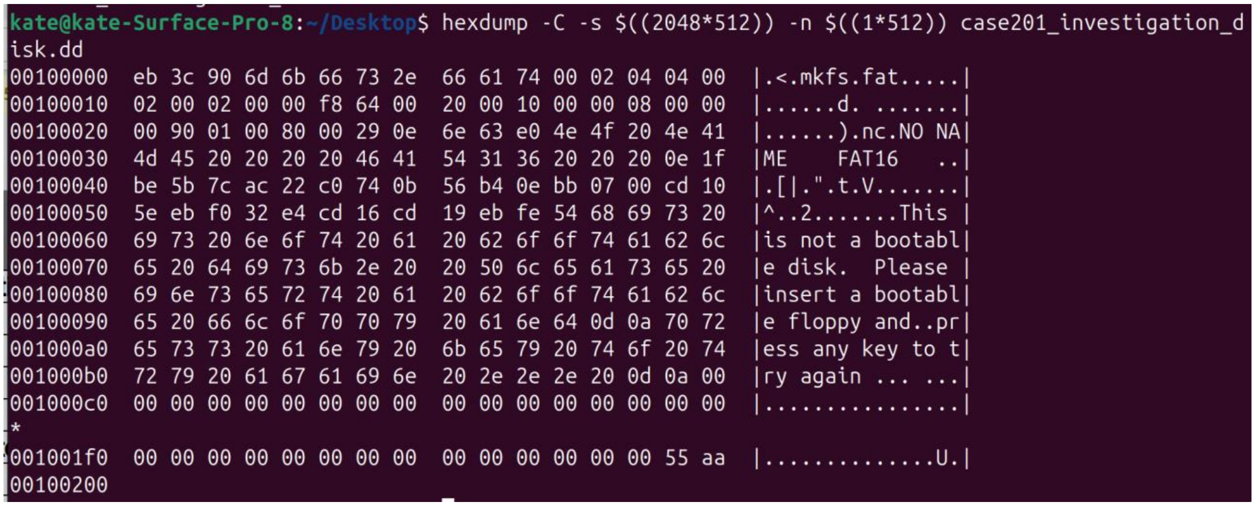
Partition 1

I was able to see the disk information by typing in the following command "hexdump -C –s $((2048*512)) -n $((512*1)) case201_investigation_disk.dd". I found the bytes/sec by looking at the location 000Bh for 2 bytes which is 0002 then do big endian and it is 512(0x0200).



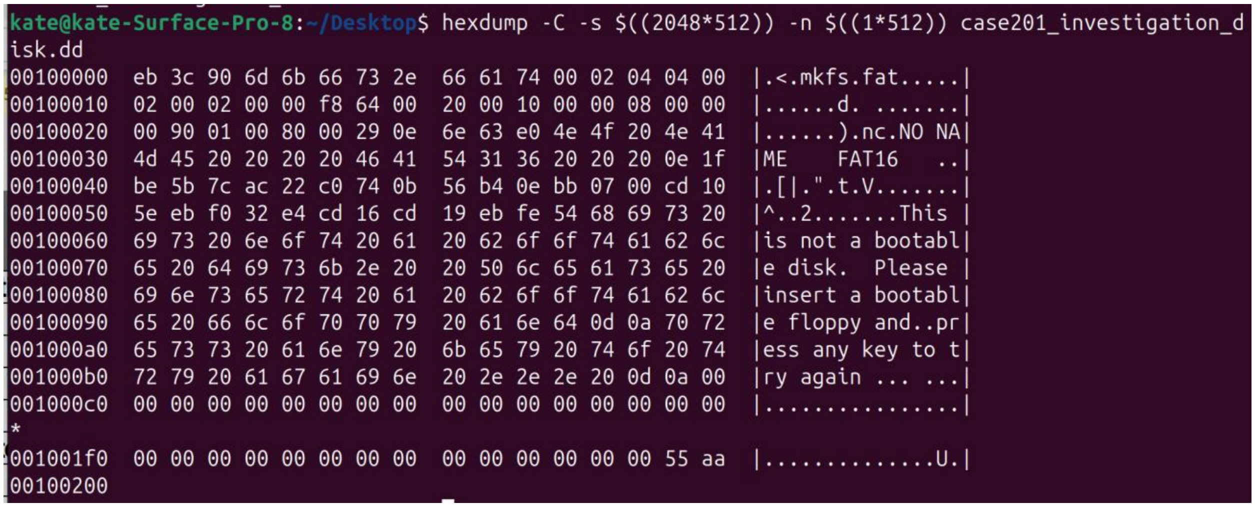
```
kate@kate-Surface-Pro-8:~/Desktop$ hexdump -C -s $((2048*512)) -n $((1*512)) case201_investigation_d
isk.dd
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 00 f8 64 00  20 00 10 00 00 08 00 00  |......d. .......|
00100020  00 90 01 00 80 00 29 0e  6e 63 e0 4e 4f 20 4e 41  |......).nc.NO NA|
00100030  4d 45 20 20 20 20 46 41  54 31 36 20 20 20 0e 1f  |ME    FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
00100200
```

I found the number of sec/cluster by looking at the location 000Dh for 1 byte which is 04. In decimal it is the same so the answer is 4.



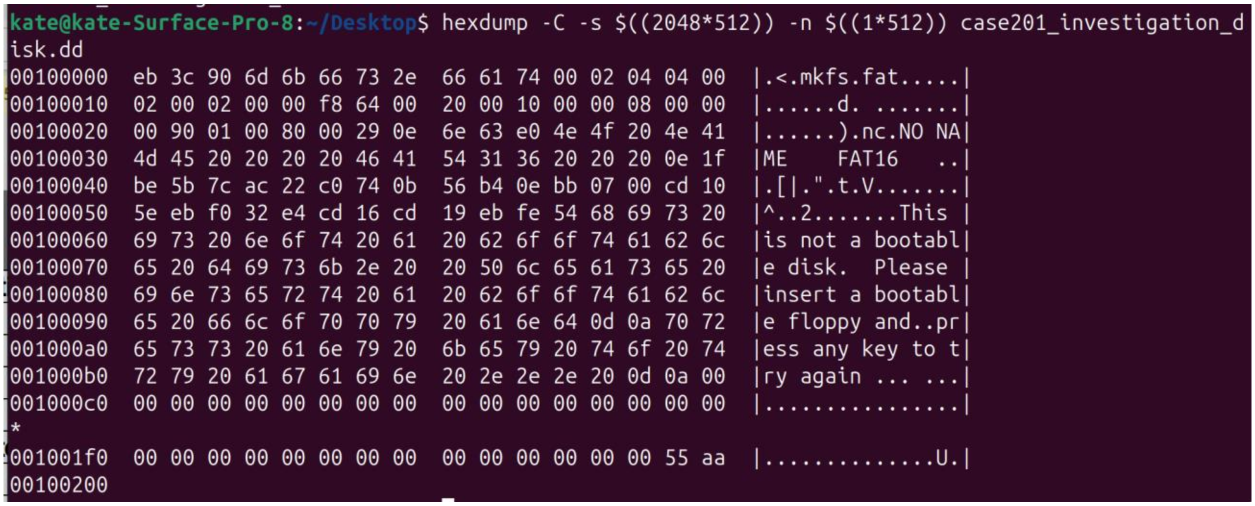
```
kate@kate-Surface-Pro-8:~/Desktop$ hexdump -C -s $((2048*512)) -n $((1*512)) case201_investigation_d
isk.dd
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 00 f8 64 00  20 00 10 00 00 08 00 00  |......d. .......|
00100020  00 90 01 00 80 00 29 0e  6e 63 e0 4e 4f 20 4e 41  |......).nc.NO NA|
00100030  4d 45 20 20 20 20 46 41  54 31 36 20 20 20 0e 1f  |ME    FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
00100200
```

I found the reserved sectors by looking at the location 000Eh for 2 bytes. It is 0400 adn when you do big endian you get 4 (0x0004).



```
kate@kate-Surface-Pro-8:~/Desktop$ hexdump -C -s $((2048*512)) -n $((1*512)) case201_investigation_d
isk.dd
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 00 f8 64 00  20 00 10 00 00 08 00 00  |......d. .......|
00100020  00 90 01 00 80 00 29 0e  6e 63 e0 4e 4f 20 4e 41  |......).nc.NO NA|
00100030  4d 45 20 20 20 20 46 41  54 31 36 20 20 20 0e 1f  |ME    FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
00100200
```

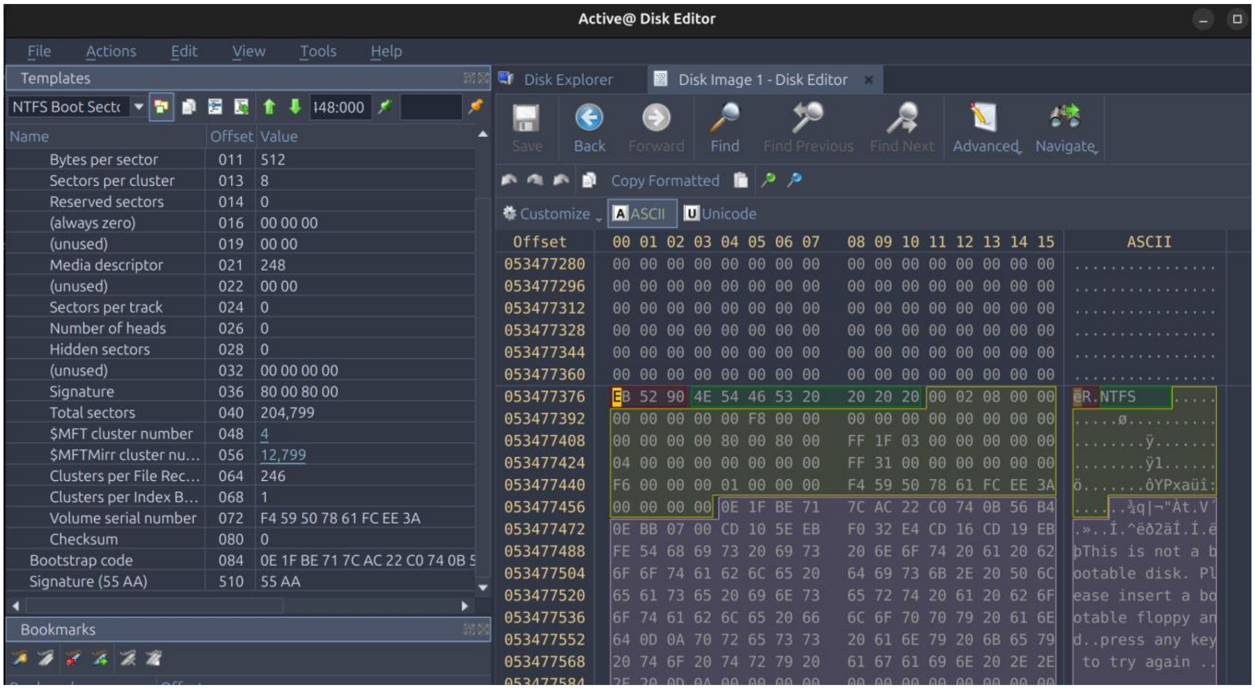
I found the sec/fat by looking at location 0016h for 2 bytes. It is 6400 and when you do big endian endian you get 100 (0x0064).



```
kate@kate-Surface-Pro-8:~/Desktop$ hexdump -C -s $((2048*512)) -n $((1*512)) case201_investigation_d
isk.dd
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 00 f8 64 00  20 00 10 00 00 08 00 00  |......d. .......|
00100020  00 90 01 00 80 00 29 0e  6e 63 e0 4e 4f 20 4e 41  |......).nc.NO NA|
00100030  4d 45 20 20 20 20 46 41  54 31 36 20 20 20 0e 1f  |ME    FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
00100200
```

Partition 2

The 2nd partition is NTFS so the analysis will look a little different. To find our information we need to use Disk Editor. The 2nd partition starts at 104448 so lets start there. The bytes/sec is 512. The sec/custer is 8. The are 0 reserved sectors. The $MFT Cluster Start is 4 and the $MFTMirr Cluster Start is 12799. The #System $MFT Records is standard at 39 and the $MFT Record Size is also standard at 39.



Name	Offset	Value
Bytes per sector	011	512
Sectors per cluster	013	8
Reserved sectors	014	0
(always zero)	016	00 00 00
(unused)	019	00 00
Media descriptor	021	248
(unused)	022	00 00
Sectors per track	024	0
Number of heads	026	0
Hidden sectors	028	0
(unused)	032	00 00 00 00
Signature	036	80 00 80 00
Total sectors	040	204,799
$MFT cluster number	048	4
$MFTMirr cluster nu…	056	12,799
Clusters per File Rec…	064	246
Clusters per Index B…	068	1
Volume serial number	072	F4 59 50 78 61 FC EE 3A
Checksum	080	0
Bootstrap code	084	0E 1F BE 71 7C AC 22 C0 74 0B 5
Signature (55 AA)	510	55 AA

```
 Offset    00 01 02 03 04 05 06 07  08 09 10 11 12 13 14 15      ASCII
053477280  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053477296  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053477312  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053477328  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053477344  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053477360  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053477376  EB 52 90 4E 54 46 53 20  20 20 20 00 02 08 00 00  ëR.NTFS     .....
053477392  00 00 00 00 00 F8 00 00  00 00 00 00 00 00 00 00  .....ø..........
053477408  00 00 00 00 80 00 80 00  FF 1F 03 00 00 00 00 00  ........ÿ.......
053477424  04 00 00 00 00 00 00 00  FF 31 00 00 00 00 00 00  ........ÿ1......
053477440  F6 00 00 00 01 00 00 00  F4 59 50 78 61 FC EE 3A  ö.......ôYPxaüî:
053477456  00 00 00 00 0E 1F BE 71  7C AC 22 C0 74 0B 56 B4  ......¾q|¬"Àt.V´
053477472  0E BB 07 00 CD 10 5E EB  F0 32 E4 CD 16 CD 19 EB  .»..Í.^ëð2äÍ.Í.ë
053477488  FE 54 68 69 73 20 69 73  20 6E 6F 74 20 61 20 62  þThis is not a b
053477504  6F 6F 74 61 62 6C 65 20  64 69 73 6B 2E 20 50 6C  ootable disk. Pl
053477520  65 61 73 65 20 69 6E 73  65 72 74 20 61 20 62 6F  ease insert a bo
053477536  6F 74 61 62 6C 65 20 66  6C 6F 70 70 79 20 61 6E  otable floppy an
053477552  64 0D 0A 70 72 65 73 73  20 61 6E 79 20 6B 65 79  d..press any key
053477568  20 74 6F 20 74 72 79 20  61 67 61 69 6E 20 2E 2E   to try again ..
053477584  2E 20 0D 0A 00 00 00 00  00 00 00 00 00 00 00 00
```

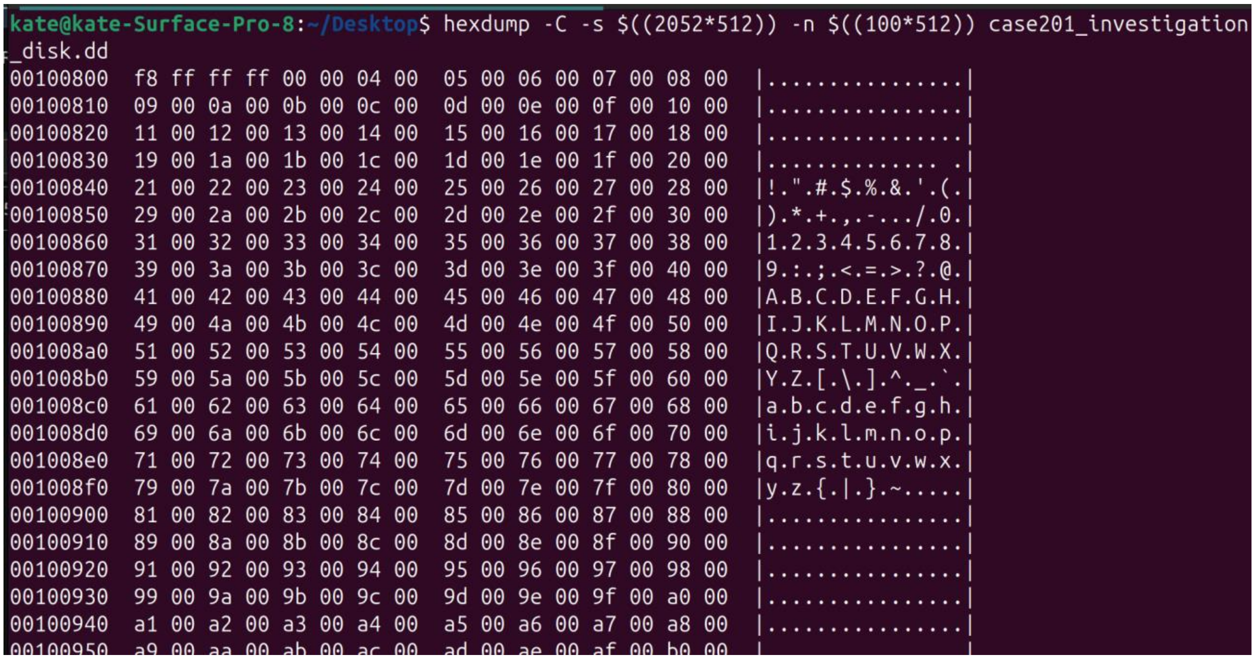
Detailed Analysis of the Recovery Process

After the disk information is analyzed for the first partition, look at the file allocation table and the root directory. The file allocation will show how many buffers you have between he beginning of the data area and the start of the file location. Then look at your root directory, This will say how many files are on the partition, the size of the file, and the status of the file.

For the NTFS partition, after we look at the boot sector we can calculate where the file information is in the MFT. From the MFT it shows the status of the file, the size of the file, and the characteristics of the file.

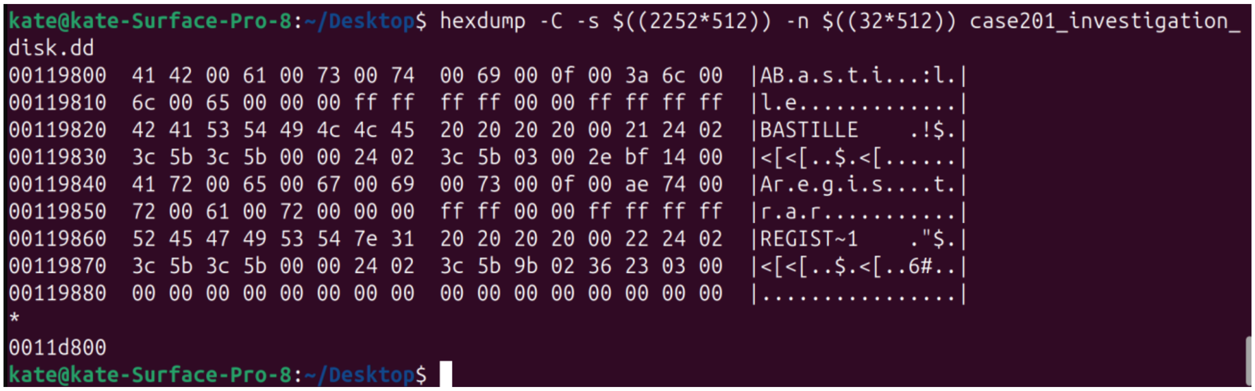
2.2.1 Partition 1 File 1

First take a look at the partition 1 allocation table. This will inficate the size of the buffer. This is crucial because if it is ignored file carving will be done incorrectly and result in incorrect files. The size of the buffer is found by the number of clusters that are found before the files start. The screenshot below shows that there is 1 cluster. Then you multiply it by how many sec/cluster which is 4. 4*1=4



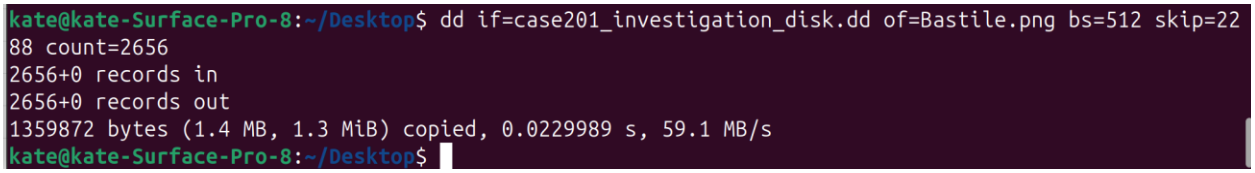
```
kate@kate-Surface-Pro-8:~/Desktop$ hexdump -C -s $((2052*512)) -n $((100*512)) case201_investigation
_disk.dd
00100800  f8 ff ff ff 00 00 04 00  05 00 06 00 07 00 08 00  |................|
00100810  09 00 0a 00 0b 00 0c 00  0d 00 0e 00 0f 00 10 00  |................|
00100820  11 00 12 00 13 00 14 00  15 00 16 00 17 00 18 00  |................|
00100830  19 00 1a 00 1b 00 1c 00  1d 00 1e 00 1f 00 20 00  |.............. .|
00100840  21 00 22 00 23 00 24 00  25 00 26 00 27 00 28 00  |!.".#.$.%.&.'.(.|
00100850  29 00 2a 00 2b 00 2c 00  2d 00 2e 00 2f 00 30 00  |).*.+.,.-.../.0.|
00100860  31 00 32 00 33 00 34 00  35 00 36 00 37 00 38 00  |1.2.3.4.5.6.7.8.|
00100870  39 00 3a 00 3b 00 3c 00  3d 00 3e 00 3f 00 40 00  |9.:.;.<.=.>.?.@.|
00100880  41 00 42 00 43 00 44 00  45 00 46 00 47 00 48 00  |A.B.C.D.E.F.G.H.|
00100890  49 00 4a 00 4b 00 4c 00  4d 00 4e 00 4f 00 50 00  |I.J.K.L.M.N.O.P.|
001008a0  51 00 52 00 53 00 54 00  55 00 56 00 57 00 58 00  |Q.R.S.T.U.V.W.X.|
001008b0  59 00 5a 00 5b 00 5c 00  5d 00 5e 00 5f 00 60 00  |Y.Z.[.\.].^._.`.|
001008c0  61 00 62 00 63 00 64 00  65 00 66 00 67 00 68 00  |a.b.c.d.e.f.g.h.|
001008d0  69 00 6a 00 6b 00 6c 00  6d 00 6e 00 6f 00 70 00  |i.j.k.l.m.n.o.p.|
001008e0  71 00 72 00 73 00 74 00  75 00 76 00 77 00 78 00  |q.r.s.t.u.v.w.x.|
001008f0  79 00 7a 00 7b 00 7c 00  7d 00 7e 00 7f 00 80 00  |y.z.{.|.}.~.....|
00100900  81 00 82 00 83 00 84 00  85 00 86 00 87 00 88 00  |................|
00100910  89 00 8a 00 8b 00 8c 00  8d 00 8e 00 8f 00 90 00  |................|
00100920  91 00 92 00 93 00 94 00  95 00 96 00 97 00 98 00  |................|
00100930  99 00 9a 00 9b 00 9c 00  9d 00 9e 00 9f 00 a0 00  |................|
00100940  a1 00 a2 00 a3 00 a4 00  a5 00 a6 00 a7 00 a8 00  |................|
00100950  a9 00 aa 00 ab 00 ac 00  ad 00 ae 00 af 00 b0 00  |
```

Next is the see what files aare on the partition by looking at the root directory. This shows the number of files and their sizes. Below shows that there are 2 files. The first is called Bastile.png. The file status is 41 which indicates that it is an active file. To find the file size look at the last 4 bytes and perform the big endian operation on them. The last 4 in this case are 2e bf 14 00 so after the big endian it is 00 14 bf 2e which is 1359662. Then to find the number of sections divide that number by 512 so 1359662/523 = 2656.



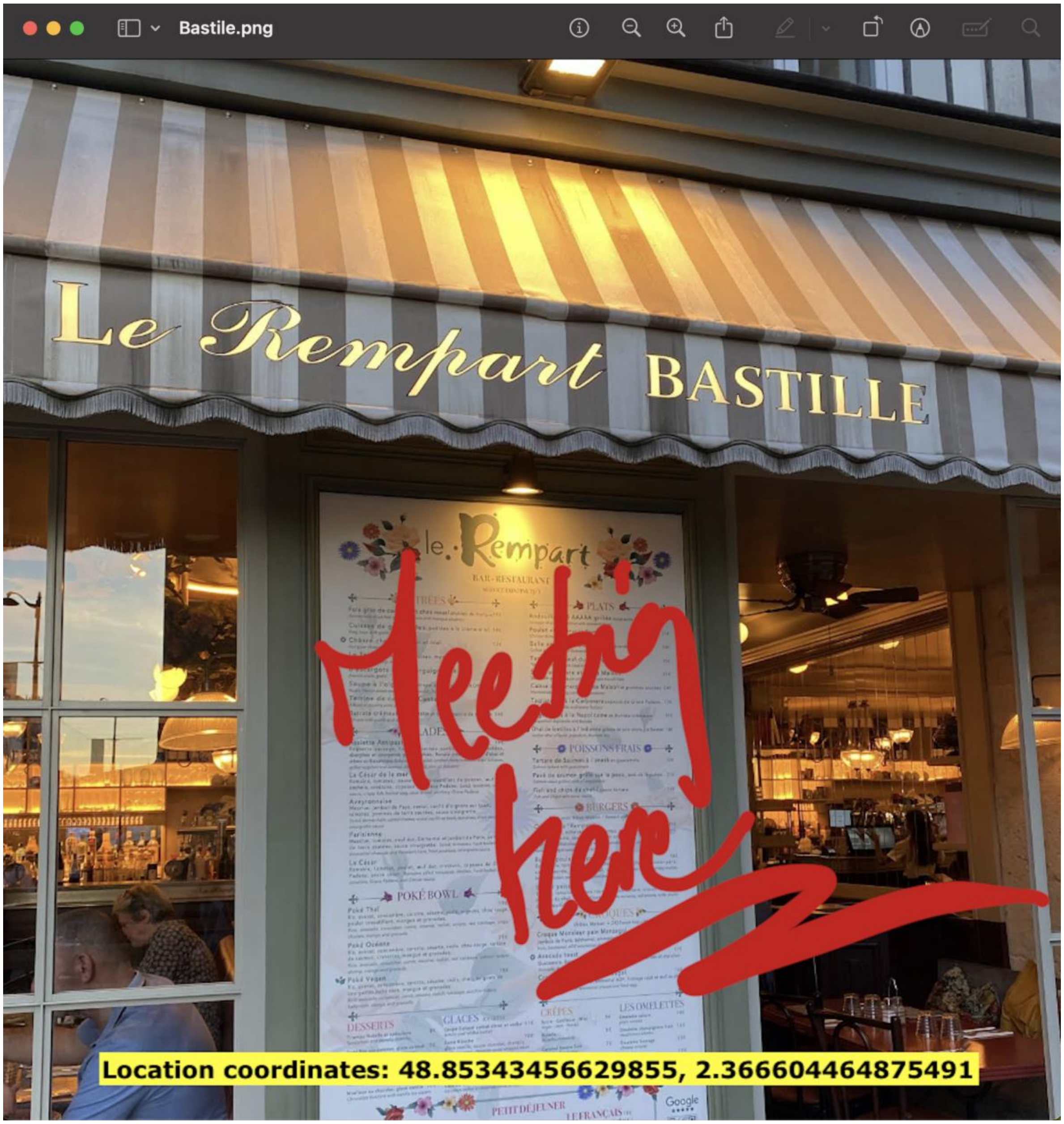
```
kate@kate-Surface-Pro-8:~/Desktop$ hexdump -C -s $((2252*512)) -n $((32*512)) case201_investigation_
disk.dd
00119800  41 42 00 61 00 73 00 74  00 69 00 0f 00 3a 6c 00  |AB.a.s.t.i...:l.|
00119810  6c 00 65 00 00 00 ff ff  ff ff 00 00 ff ff ff ff  |l.e.............|
00119820  42 41 53 54 49 4c 4c 45  20 20 20 20 00 21 24 02  |BASTILLE    .!$.|
00119830  3c 5b 3c 5b 00 00 24 02  3c 5b 03 00 2e bf 14 00  |<[<[..$.<[......|
00119840  41 72 00 65 00 67 00 69  00 73 00 0f 00 ae 74 00  |Ar.e.g.i.s....t.|
00119850  72 00 61 00 72 00 00 00  ff ff 00 00 ff ff ff ff  |r.a.r...........|
00119860  52 45 47 49 53 54 7e 31  20 20 20 20 00 22 24 02  |REGIST~1    ."$.|
00119870  3c 5b 3c 5b 00 00 24 02  3c 5b 9b 02 36 23 03 00  |<[<[..$.<[..6#..|
00119880  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
0011d800
kate@kate-Surface-Pro-8:~/Desktop$
```

Now to carve the file. We have to calculate what to skip which is 2048 + 4 + 100 + 100 + 32 + 4= 2288. So use the dd command to carve it. Type in the following “dd if=case201_investigation_disk.dd of=Bastile.png bs=512 skip=2288 count=2656”



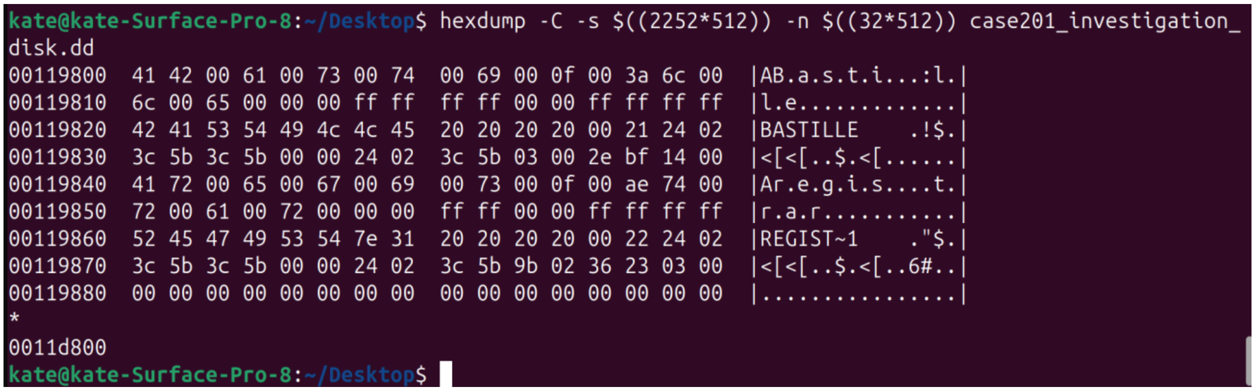
```
kate@kate-Surface-Pro-8:~/Desktop$ dd if=case201_investigation_disk.dd of=Bastile.png bs=512 skip=22
88 count=2656
2656+0 records in
2656+0 records out
1359872 bytes (1.4 MB, 1.3 MiB) copied, 0.0229989 s, 59.1 MB/s
kate@kate-Surface-Pro-8:~/Desktop$
```

Below is an image of the file that we carved.



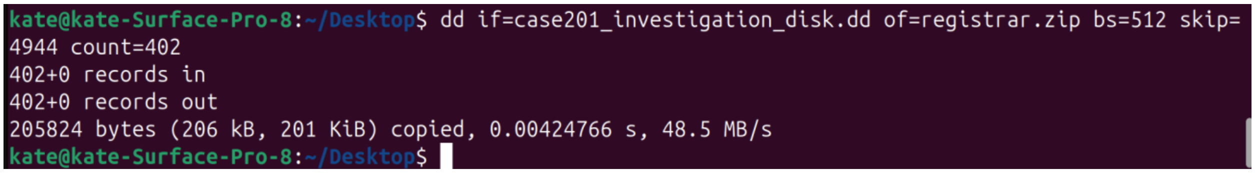
2.2.2 Partition 1 File 2

This is on the same partition so we already know that the data buffer is 4 so now it is time for analysis of the root directory. The next file is called registrar.zipTo find the size we need to look at the last 4 bytes of the file. The bytes are 36 23 03 00 so after the big endian it is 00 03 23 26 (205622). To find the number of sections divide that number by 512 so 205622/512. To find where the file is located add up all of the things that came before so 2048 + 4 + 100 + 100 + 32 + 4 + 2656 = 4944.



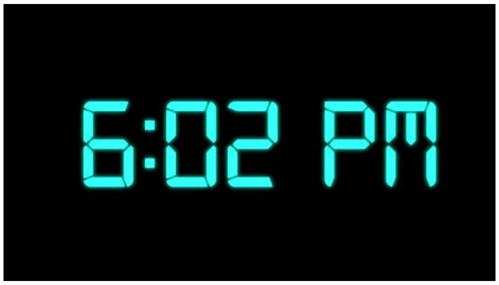
```
kate@kate-Surface-Pro-8:~/Desktop$ hexdump -C -s $((2252*512)) -n $((32*512)) case201_investigation_
disk.dd
00119800  41 42 00 61 00 73 00 74  00 69 00 0f 00 3a 6c 00  |AB.a.s.t.i...:l.|
00119810  6c 00 65 00 00 00 ff ff  ff ff 00 00 ff ff ff ff  |l.e.............|
00119820  42 41 53 54 49 4c 4c 45  20 20 20 20 00 21 24 02  |BASTILLE    .!$.|
00119830  3c 5b 3c 5b 00 00 24 02  3c 5b 03 00 2e bf 14 00  |<[<[..$.<[......|
00119840  41 72 00 65 00 67 00 69  00 73 00 0f 00 ae 74 00  |Ar.e.g.i.s....t.|
00119850  72 00 61 00 72 00 00 00  ff ff 00 00 ff ff ff ff  |r.a.r...........|
00119860  52 45 47 49 53 54 7e 31  20 20 20 20 00 22 24 02  |REGIST~1    ."$.|
00119870  3c 5b 3c 5b 00 00 24 02  3c 5b 9b 02 36 23 03 00  |<[<[..$.<[..6#..|
00119880  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
0011d800
kate@kate-Surface-Pro-8:~/Desktop$
```

To carve the file we need to use dd so use the following command dd if=case201_investigation_disk.dd of=registrar.zip bs=512 skip=4944 count=402. Below is that command.



```
kate@kate-Surface-Pro-8:~/Desktop$ dd if=case201_investigation_disk.dd of=registrar.zip bs=512 skip=
4944 count=402
402+0 records in
402+0 records out
205824 bytes (206 kB, 201 KiB) copied, 0.00424766 s, 48.5 MB/s
kate@kate-Surface-Pro-8:~/Desktop$
```

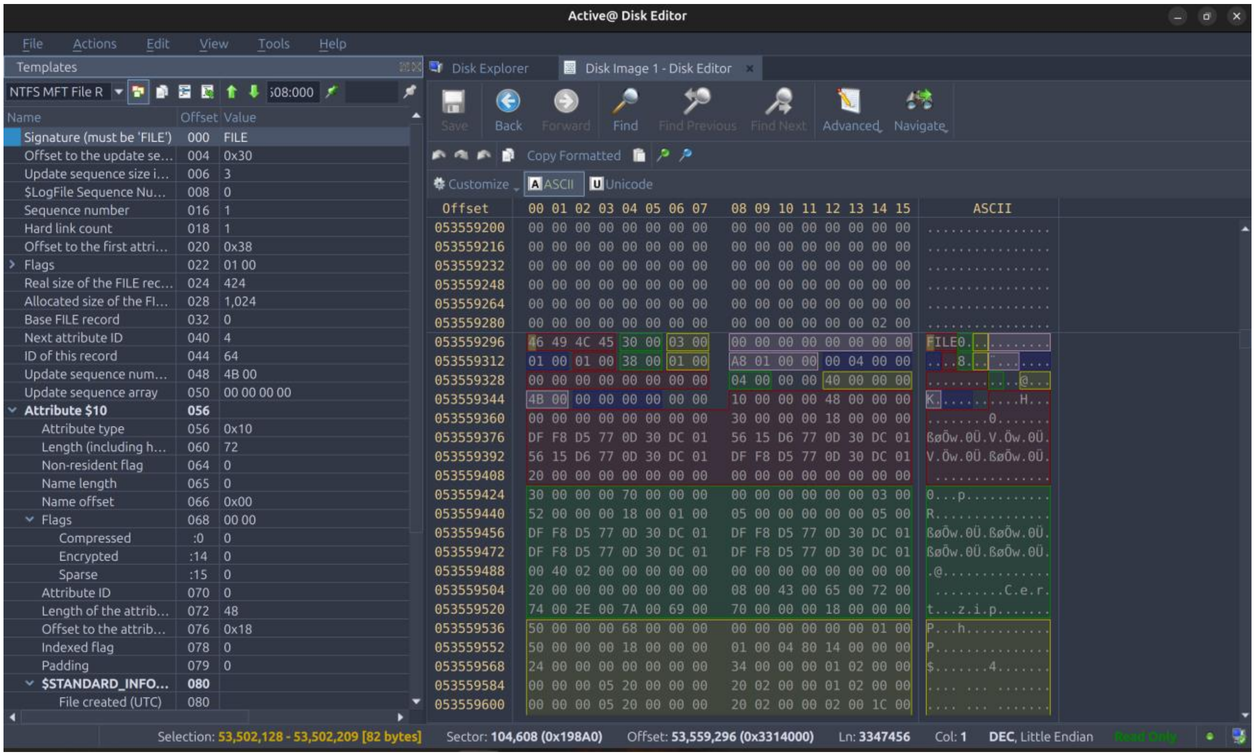
Inside the zip folder there are 2 files move-time.png and room_number.jpg. First is move-time and then second is room_number below



2.2.3 Partition 2 File 3

I ran into some issues with the NTFS partition. I calculated that to sectors to partition was 104448 because earlier when we ran fdisk it showed that was when partition 2 begun. Then the $MFT Cluster Start is 32 sectors because it is 4 clusters and the sec/cluster is 8 so 8*4 = 32. I assumed that the $MFT System Records in sectors is 78 like in class because that is standard for NTFS. So after I added up the sectors to the file I got 104448+32+78 = 104558. In

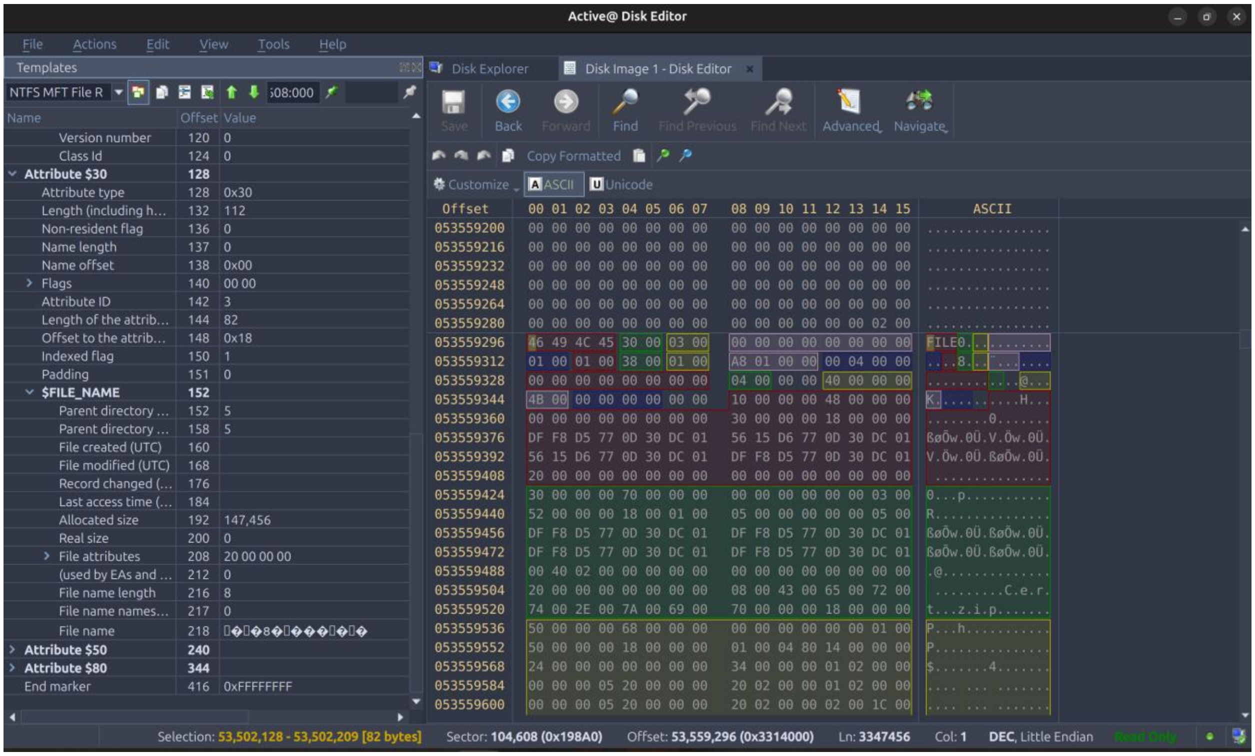
Disk Editor I went there and there was no file. I went back through my calculations and did not find any errors. I went ahead and scrolled through the MFT until I found Cert.zip. Cert.zip is located at sector 104608. After I found that I thought that it was just blank data until the file appeared. Once I found the file I looked through and saw that it was in use and non-resident so I did not have to do cluster carving. The first cluster is 13312 so multiple that by 8 and get 106496 so that is the first sector. Then add the 1st sector plus the stuff before which is 106496 and you get 210944. The number of cluster is 36 so multiple that by 8 and you get 288 which is the number of sectors. It has had $50 which I learned what a security attribute so you could tell the file was password protected.



Name	Offset	Value
Signature (must be 'FILE')	000	FILE
Offset to the update se…	004	0x30
Update sequence size i…	006	3
$LogFile Sequence Nu…	008	0
Sequence number	016	1
Hard link count	018	1
Offset to the first attri…	020	0x38
Flags	022	01 00
Real size of the FILE rec…	024	424
Allocated size of the FI…	028	1,024
Base FILE record	032	0
Next attribute ID	040	4
ID of this record	044	64
Update sequence num…	048	4B 00
Update sequence array	050	00 00 00 00
Attribute $10	**056**	
Attribute type	056	0x10
Length (including h…	060	72
Non-resident flag	064	0
Name length	065	0
Name offset	066	0x00
Flags	068	00 00
Compressed	:0	0
Encrypted	:14	0
Sparse	:15	0
Attribute ID	070	0
Length of the attrib…	072	48
Offset to the attrib…	076	0x18
Indexed flag	078	0
Padding	079	0
$STANDARD_INFO…	**080**	
File created (UTC)	080	

```
 Offset    00 01 02 03 04 05 06 07  08 09 10 11 12 13 14 15        ASCII
053559200  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053559216  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053559232  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053559248  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053559264  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053559280  00 00 00 00 00 00 00 00  00 00 00 00 00 00 02 00  ................
053559296  46 49 4C 45 30 00 03 00  00 00 00 00 00 00 00 00  FILE0...........
053559312  01 00 01 00 38 00 01 00  A8 01 00 00 00 04 00 00  ....8...........
053559328  00 00 00 00 00 00 00 00  04 00 00 00 40 00 00 00  ............@...
053559344  4B 00 00 00 00 00 00 00  10 00 00 00 48 00 00 00  K...........H...
053559360  00 00 00 00 00 00 00 00  30 00 00 00 18 00 00 00  ........0.......
053559376  DF F8 D5 77 0D 30 DC 01  56 15 D6 77 0D 30 DC 01  ßøÕw.0Ü.V.Öw.0Ü.
053559392  56 15 D6 77 0D 30 DC 01  DF F8 D5 77 0D 30 DC 01  V.Öw.0Ü.ßøÕw.0Ü.
053559408  20 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ...............
053559424  30 00 00 00 70 00 00 00  00 00 00 00 00 00 03 00  0...p...........
053559440  52 00 00 00 18 00 01 00  05 00 00 00 00 00 05 00  R...............
053559456  DF F8 D5 77 0D 30 DC 01  DF F8 D5 77 0D 30 DC 01  ßøÕw.0Ü.ßøÕw.0Ü.
053559472  DF F8 D5 77 0D 30 DC 01  DF F8 D5 77 0D 30 DC 01  ßøÕw.0Ü.ßøÕw.0Ü.
053559488  00 40 02 00 00 00 00 00  00 00 00 00 00 00 00 00  .@..............
053559504  20 00 00 00 00 00 00 00  08 00 43 00 65 00 72 00   .........C.e.r.
053559520  74 00 2E 00 7A 00 69 00  70 00 00 00 18 00 00 00  t...z.i.p.......
053559536  50 00 00 00 68 00 00 00  00 00 00 00 00 00 01 00  P...h...........
053559552  50 00 00 00 18 00 00 00  01 00 04 80 14 00 00 00  P...............
053559568  24 00 00 00 00 00 00 00  34 00 00 00 01 02 00 00  $.......4.......
053559584  00 00 00 05 20 00 00 00  20 02 00 00 01 02 00 00  .... ... .......
053559600  00 00 00 05 20 00 00 00  20 02 00 00 02 00 1C 00  .... ... .......
```

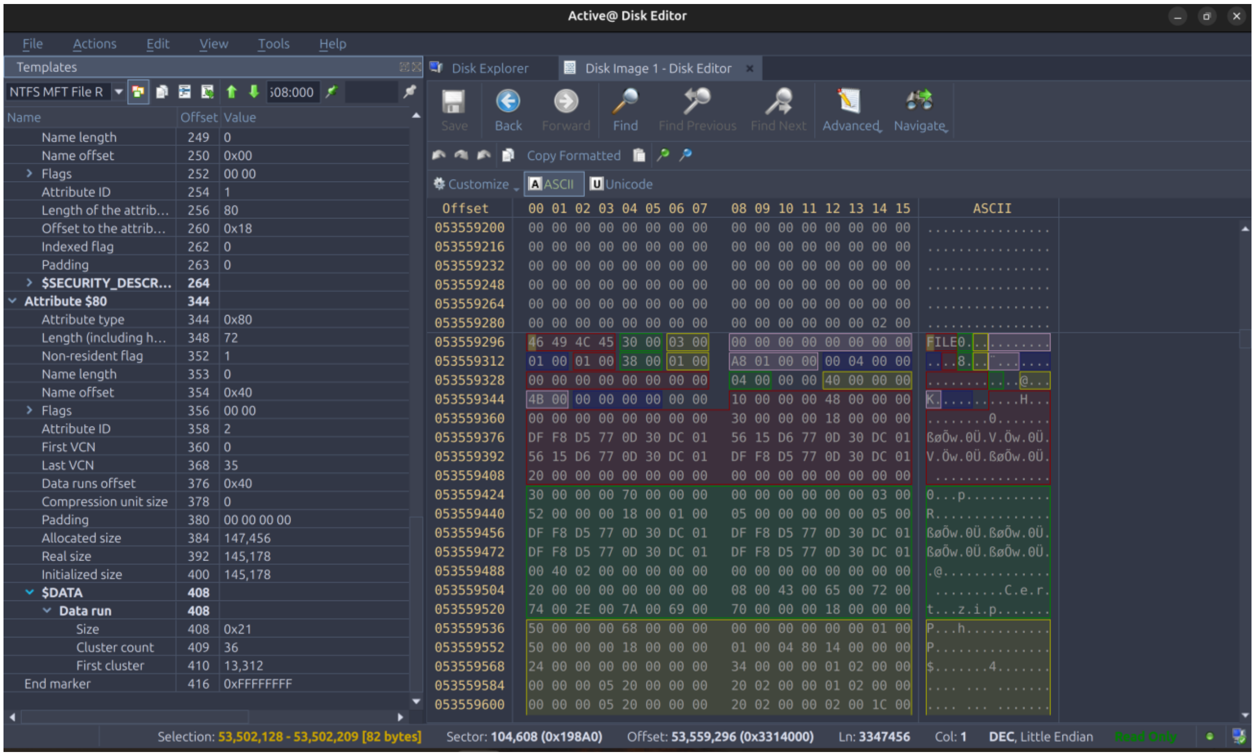
Selection: 53,502,128 - 53,502,209 [82 bytes]   Sector: 104,608 (0x198A0)   Offset: 53,559,296 (0x3314000)   Ln: 3347456   Col: 1   DEC, Little Endian



Name	Offset	Value
Version number	120	0
Class Id	124	0
Attribute $30	**128**	
Attribute type	128	0x30
Length (including h…	132	112
Non-resident flag	136	0
Name length	137	0
Name offset	138	0x00
Flags	140	00 00
Attribute ID	142	3
Length of the attrib…	144	82
Offset to the attrib…	148	0x18
Indexed flag	150	1
Padding	151	0
$FILE_NAME	**152**	
Parent directory …	152	5
Parent directory …	158	5
File created (UTC)	160	
File modified (UTC)	168	
Record changed (…	176	
Last access time (…	184	
Allocated size	192	147,456
Real size	200	0
File attributes	208	20 00 00 00
(used by EAs and …	212	0
File name length	216	8
File name names…	217	0
File name	218	▯�▯�8�▯����▯�▯�
Attribute $50	**240**	
Attribute $80	**344**	
End marker	416	0xFFFFFFFF

```
Offset     00 01 02 03 04 05 06 07  08 09 10 11 12 13 14 15       ASCII
053559200  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053559216  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053559232  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053559248  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053559264  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053559280  00 00 00 00 00 00 00 00  00 00 00 00 00 00 02 00  ................
053559296  46 49 4C 45 30 00 03 00  00 00 00 00 00 00 00 00  FILE0...........
053559312  01 00 01 00 38 00 01 00  A8 01 00 00 00 04 00 00  ....8..¨........
053559328  00 00 00 00 00 00 00 00  04 00 00 00 40 00 00 00  ............@...
053559344  4B 00 00 00 00 00 00 00  10 00 00 00 48 00 00 00  K...........H...
053559360  00 00 00 00 00 00 00 00  30 00 00 00 18 00 00 00  ........0.......
053559376  DF F8 D5 77 0D 30 DC 01  56 15 D6 77 0D 30 DC 01  ßøÕw.0Ü.V.Öw.0Ü.
053559392  56 15 D6 77 0D 30 DC 01  DF F8 D5 77 0D 30 DC 01  V.Öw.0Ü.ßøÕw.0Ü.
053559408  20 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ...............
053559424  30 00 00 00 70 00 00 00  00 00 00 00 00 00 03 00  0...p...........
053559440  52 00 00 00 18 00 01 00  05 00 00 00 00 00 05 00  R...............
053559456  DF F8 D5 77 0D 30 DC 01  DF F8 D5 77 0D 30 DC 01  ßøÕw.0Ü.ßøÕw.0Ü.
053559472  DF F8 D5 77 0D 30 DC 01  DF F8 D5 77 0D 30 DC 01  ßøÕw.0Ü.ßøÕw.0Ü.
053559488  00 40 02 00 00 00 00 00  00 00 00 00 00 00 00 00  .@..............
053559504  20 00 00 00 00 00 00 00  08 00 43 00 65 00 72 00   .........C.e.r.
053559520  74 00 2E 00 7A 00 69 00  70 00 00 00 18 00 00 00  t...z.i.p.......
053559536  50 00 00 00 68 00 00 00  00 00 00 00 00 00 01 00  P...h...........
053559552  50 00 00 00 18 00 00 00  01 00 04 80 14 00 00 00  P...............
053559568  24 00 00 00 00 00 00 00  34 00 00 00 01 02 00 00  $.......4.......
053559584  00 00 00 05 20 00 00 00  20 02 00 00 01 02 00 00  .... ... .......
053559600  00 00 00 05 20 00 00 00  20 02 00 00 02 00 1C 00  .... ... .......
```

Selection: 53,502,128 - 53,502,209 [82 bytes]   Sector: 104,608 (0x198A0)   Offset: 53,559,296 (0x3314000)   Ln: 3347456   Col: 1   DEC, Little Endian   Read Only

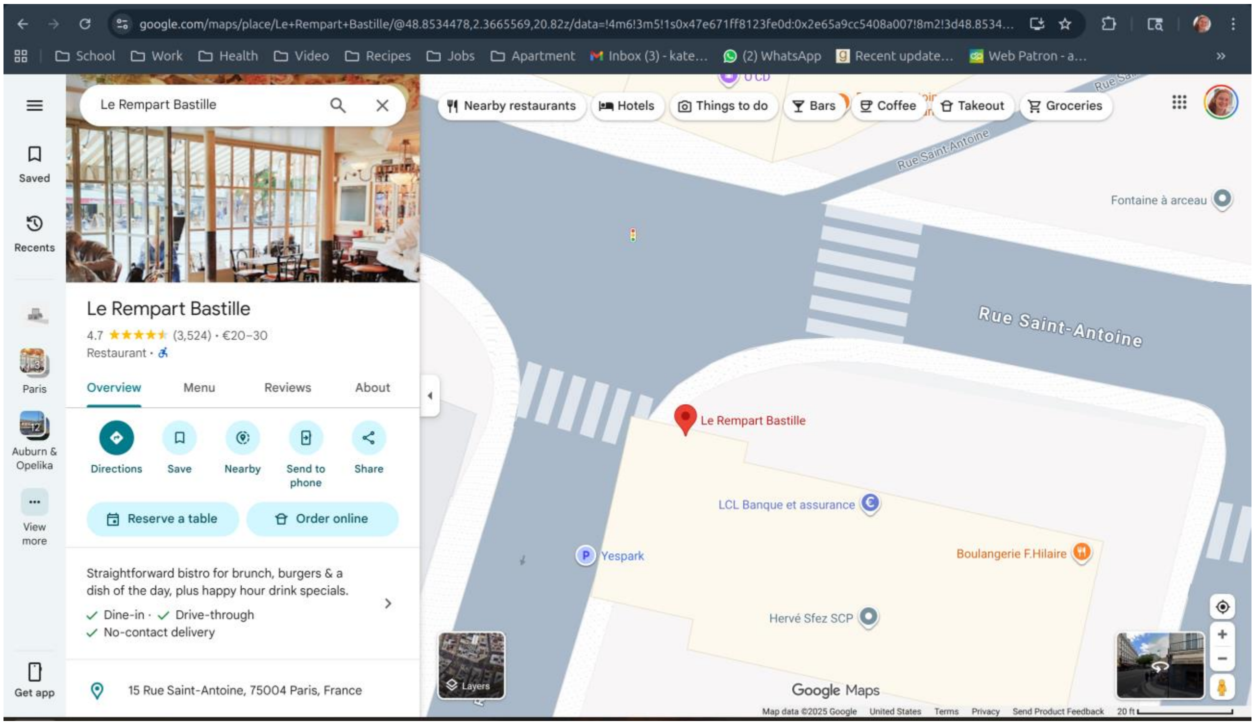


Name	Offset	Value
Name length	249	0
Name offset	250	0x00
Flags	252	00 00
Attribute ID	254	1
Length of the attrib…	256	80
Offset to the attrib…	260	0x18
Indexed flag	262	0
Padding	263	0
$SECURITY_DESCR…	**264**	
Attribute $80	**344**	
Attribute type	344	0x80
Length (including h…	348	72
Non-resident flag	352	1
Name length	353	0
Name offset	354	0x40
Flags	356	00 00
Attribute ID	358	2
First VCN	360	0
Last VCN	368	35
Data runs offset	376	0x40
Compression unit size	378	0
Padding	380	00 00 00 00
Allocated size	384	147,456
Real size	392	145,178
Initialized size	400	145,178
$DATA	**408**	
Data run	**408**	
Size	408	0x21
Cluster count	409	36
First cluster	410	13,312
End marker	416	0xFFFFFFFF

```
Offset     00 01 02 03 04 05 06 07  08 09 10 11 12 13 14 15       ASCII
053559200  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053559216  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053559232  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053559248  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053559264  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
053559280  00 00 00 00 00 00 00 00  00 00 00 00 00 00 02 00  ................
053559296  46 49 4C 45 30 00 03 00  00 00 00 00 00 00 00 00  FILE0...........
053559312  01 00 01 00 38 00 01 00  A8 01 00 00 00 04 00 00  ....8..¨........
053559328  00 00 00 00 00 00 00 00  04 00 00 00 40 00 00 00  ............@...
053559344  4B 00 00 00 00 00 00 00  10 00 00 00 48 00 00 00  K...........H...
053559360  00 00 00 00 00 00 00 00  30 00 00 00 18 00 00 00  ........0.......
053559376  DF F8 D5 77 0D 30 DC 01  56 15 D6 77 0D 30 DC 01  ßøÕw.0Ü.V.Öw.0Ü.
053559392  56 15 D6 77 0D 30 DC 01  DF F8 D5 77 0D 30 DC 01  V.Öw.0Ü.ßøÕw.0Ü.
053559408  20 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ...............
053559424  30 00 00 00 70 00 00 00  00 00 00 00 00 00 03 00  0...p...........
053559440  52 00 00 00 18 00 01 00  05 00 00 00 00 00 05 00  R...............
053559456  DF F8 D5 77 0D 30 DC 01  DF F8 D5 77 0D 30 DC 01  ßøÕw.0Ü.ßøÕw.0Ü.
053559472  DF F8 D5 77 0D 30 DC 01  DF F8 D5 77 0D 30 DC 01  ßøÕw.0Ü.ßøÕw.0Ü.
053559488  00 40 02 00 00 00 00 00  00 00 00 00 00 00 00 00  .@..............
053559504  20 00 00 00 00 00 00 00  08 00 43 00 65 00 72 00   .........C.e.r.
053559520  74 00 2E 00 7A 00 69 00  70 00 00 00 18 00 00 00  t...z.i.p.......
053559536  50 00 00 00 68 00 00 00  00 00 00 00 00 00 01 00  P...h...........
053559552  50 00 00 00 18 00 00 00  01 00 04 80 14 00 00 00  P...............
053559568  24 00 00 00 00 00 00 00  34 00 00 00 01 02 00 00  $.......4.......
053559584  00 00 00 05 20 00 00 00  20 02 00 00 01 02 00 00  .... ... .......
053559600  00 00 00 05 20 00 00 00  20 02 00 00 02 00 1C 00  .... ... .......
```

Selection: 53,502,128 - 53,502,209 [82 bytes]   Sector: 104,608 (0x198A0)   Offset: 53,559,296 (0x3314000)   Ln: 3347456   Col: 1   DEC, Little Endian   Read Only

The Cert.zip file was password protected but the clue in the instructions said that it was the zip code from the restaurant. So I looked up the coordinates for the Bastile restaurant and found the zip code which was 75004.



google.com/maps/place/Le+Rempart+Bastille/@48.8534478,2.3665569,20.82z/data=!4m6!3m5!1s0x47e671ff8123fe0d:0x2e65a9cc5408a007!8m2!3d48.8534...

Le Rempart Bastille

4.7 ★★★★★ (3,524) · €20–30

Restaurant ·

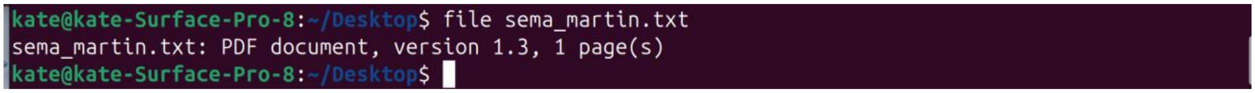
Straightforward bistro for brunch, burgers & a dish of the day, plus happy hour drink specials.

✓ Dine-in · ✓ Drive-through

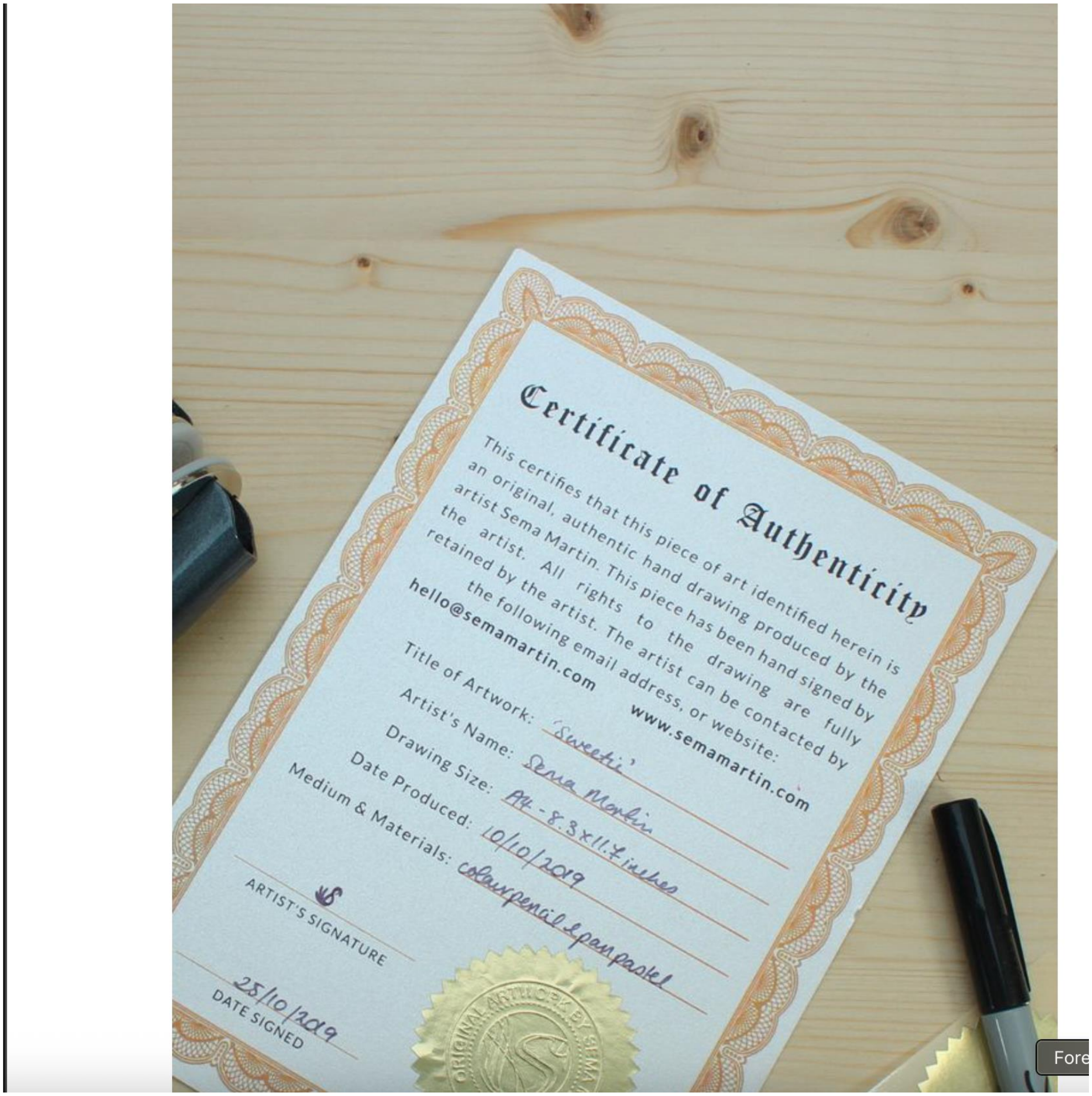
✓ No-contact delivery

15 Rue Saint-Antoine, 75004 Paris, France

After I entered the zip code and unzipped the file. I saw a sema_martin.txt file but it would not open so I ran the file command on it and it showed that it was a pdf file so I renamed it sema_martin.pdf and it opened the certificate.



```
kate@kate-Surface-Pro-8:~/Desktop$ file sema_martin.txt
sema_martin.txt: PDF document, version 1.3, 1 page(s)
kate@kate-Surface-Pro-8:~/Desktop$ 
```



Certificate of Authenticity

This certifies that this piece of art identified herein is an original, authentic hand drawing produced by the artist Sema Martin. This piece has been hand signed by the artist. All rights to the drawing are fully retained by the artist. The artist can be contacted by the following email address, or website:

hello@semamartin.com **www.semamartin.com**

Title of Artwork: 'Sweeti'

Artist's Name: Sema Martin

Drawing Size: A4 - 8.3x11.7 inches

Date Produced: 10/10/2019

Medium & Materials: colourpencil & panpastel

ARTIST'S SIGNATURE

25/10/2019

DATE SIGNED

ORIGINAL ARTWORK BY SEMA

2.2.4 Partition 2 File 2

I continued to scroll down and saw that there was another file called location. I saw that is was a non-resident file. So I attempted to carve it via clusters. I do this dd command but it came back with errors so I believe I did it incorrectly. dd if=case201_investigation_disk.dd of=location.txt bs=512 skip=$((53560656+8)) count=512 iflag=skip_bytes,count_bytes

3 Description of Analysis Techniques Utilized

During the midterm the use of fdisk to analyze the disk image to see how many partitions there were and what type were crucial. I used hexdump to see the hex and ascii results from the deserved sections for the FAT16, without this file recovery would not be possible. I used dd to carve the images with the exact location and size for both partitions. On the NTFS partition I mainly used Disk Editor. Taking use of the templates that are present are useful to quickly find file names, size, etc.

4 Results and Discussion

The results of the data recovery were 3 files. I attempted to get a 4th but was unable to. I was able to discover the restaurant the group met. The location of the registrar and the move time. I was also able to recover the image of the certificate. I was able to walk through the plan of the thieves and discover how they pulled it off.

5 Conclusions and recommendations

The midterm challenges the knowledge of both analyzing FAT16 partitions and NTFS partitions. The basic understanding of the structure was key to solving the puzzle. I think it would have been helpful to go over a non-resident file a little more and dive a little deeper. I think I am still growing my skills and have some more learning to do.

6 Acknowledgements

The lecture notes from Dr. K were used to aid in this assignment. The lessons on FAT16 and NTFS were used the most.

7 References

[1] “Decimal to hex converter [Internet]. RapidTables; c2025 [cited 2025 Sep 29]. Available from: https://www.rapidtables.com/converter/number/decimal-to-hex.html?x=0”.

Appendix A: Place the title of appendix here

Provide appropriate appendices as necessary. Each appendix should begin on a new page.