Bad out of Hell

Digital Forensics 1.0.7

Michael Hamm - CIRCL

TLP:CLEAR

Bad Cluster Forensics - FAT32

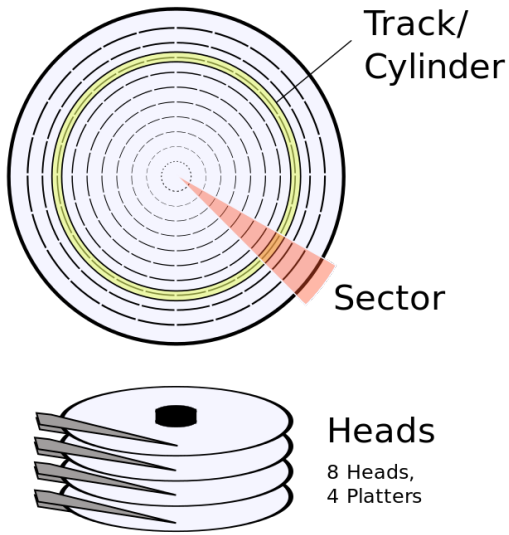
Bad out of Hell

Command Line Cheat Sheet: `https://pastebin.com/vZRSSNtE`

Cover: (c) `https://en.wikipedia.org/wiki/Bat_Out_of_Hell` - Image used solely for illustration purposes

1.1 Warm-Up

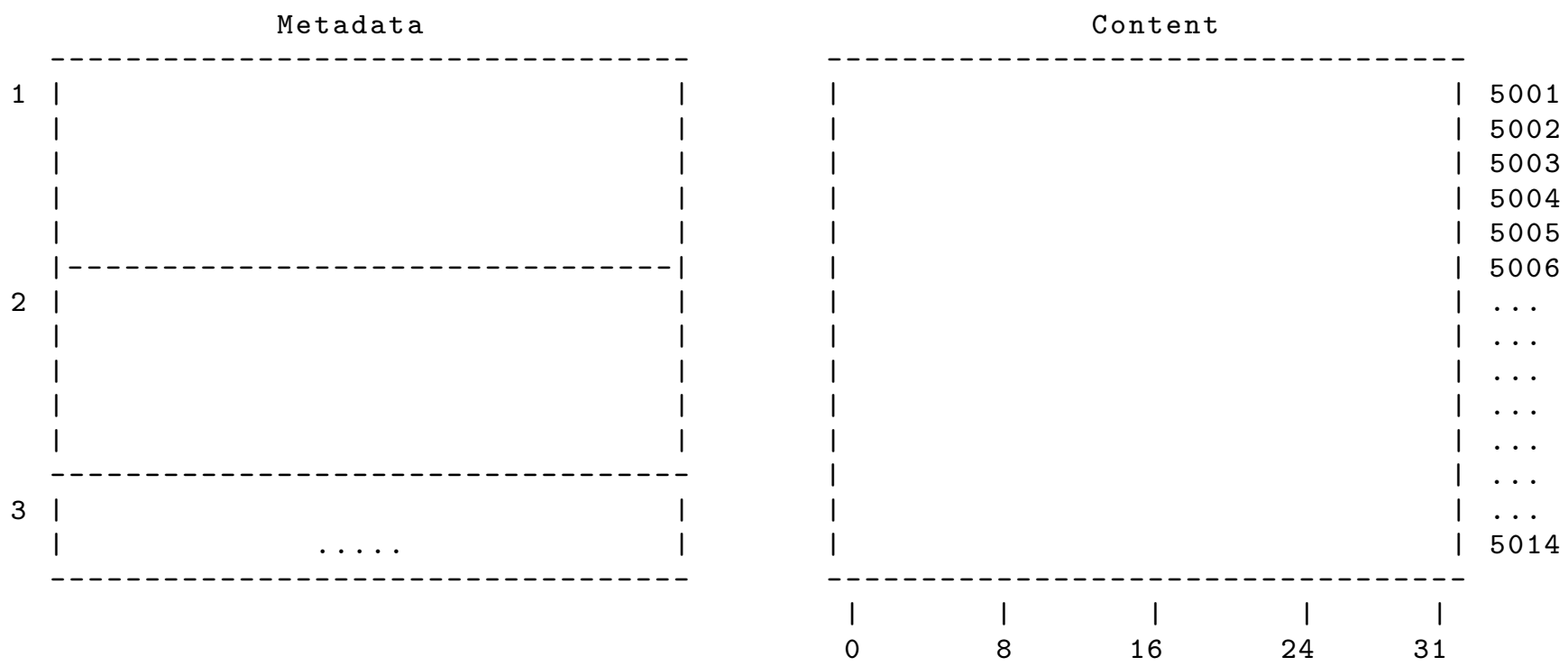
Sector, Head, Track, Cylinder / Cluster, Block / CHS, LBA




```
          0     1     2     3     4     5     6     7     8     9     10    11
        ------------------------------------------------------------------------
Sector: | 512 | 512 | 512 | 512 | 512 | 512 | 512 | 512 | 512 | 512 | 512 | 512 |
Cluster: |         2048          |         2048          |         2048          |
        ------------------------------------------------------------------------
                    0                       1                       2
```

1.2 File System

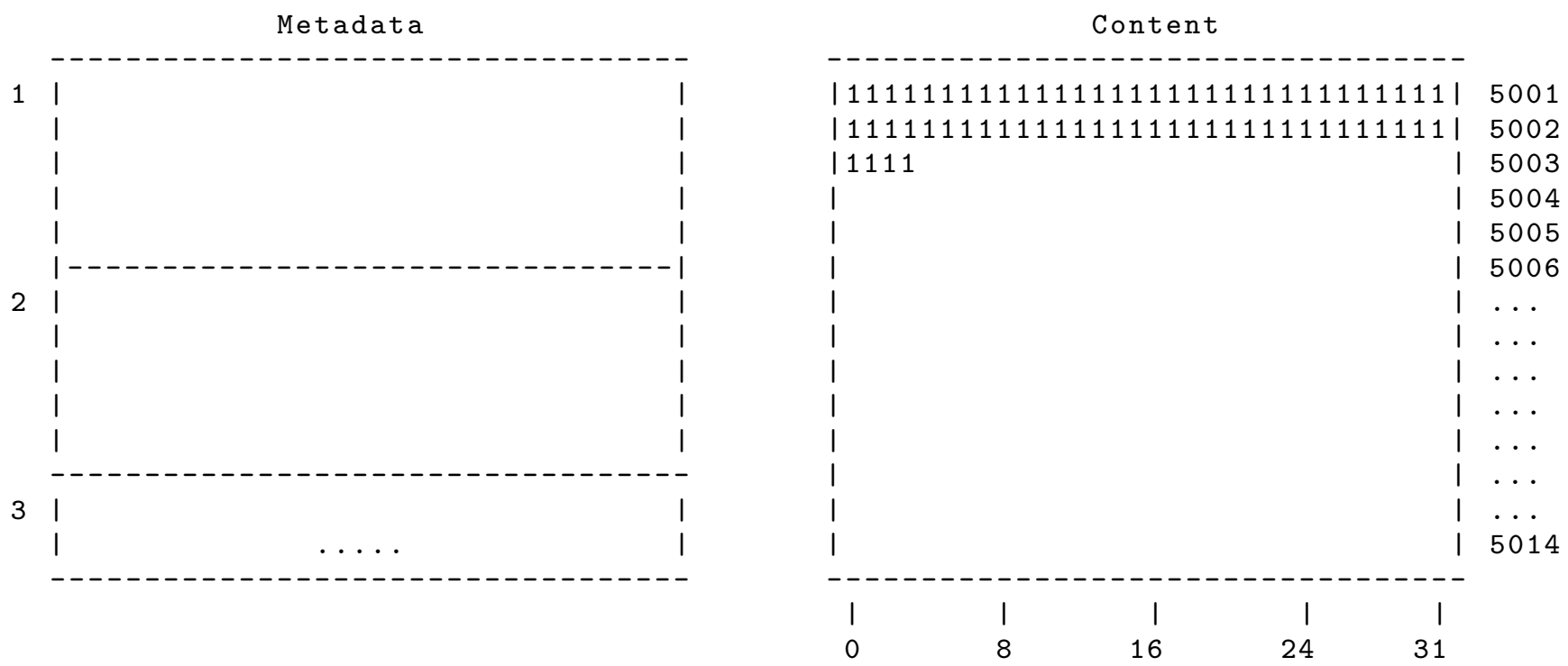
- Organizing data in files
- Maintain file related meta data
- Maintain allocation status of clusters



Allocation table:

1.2 File System

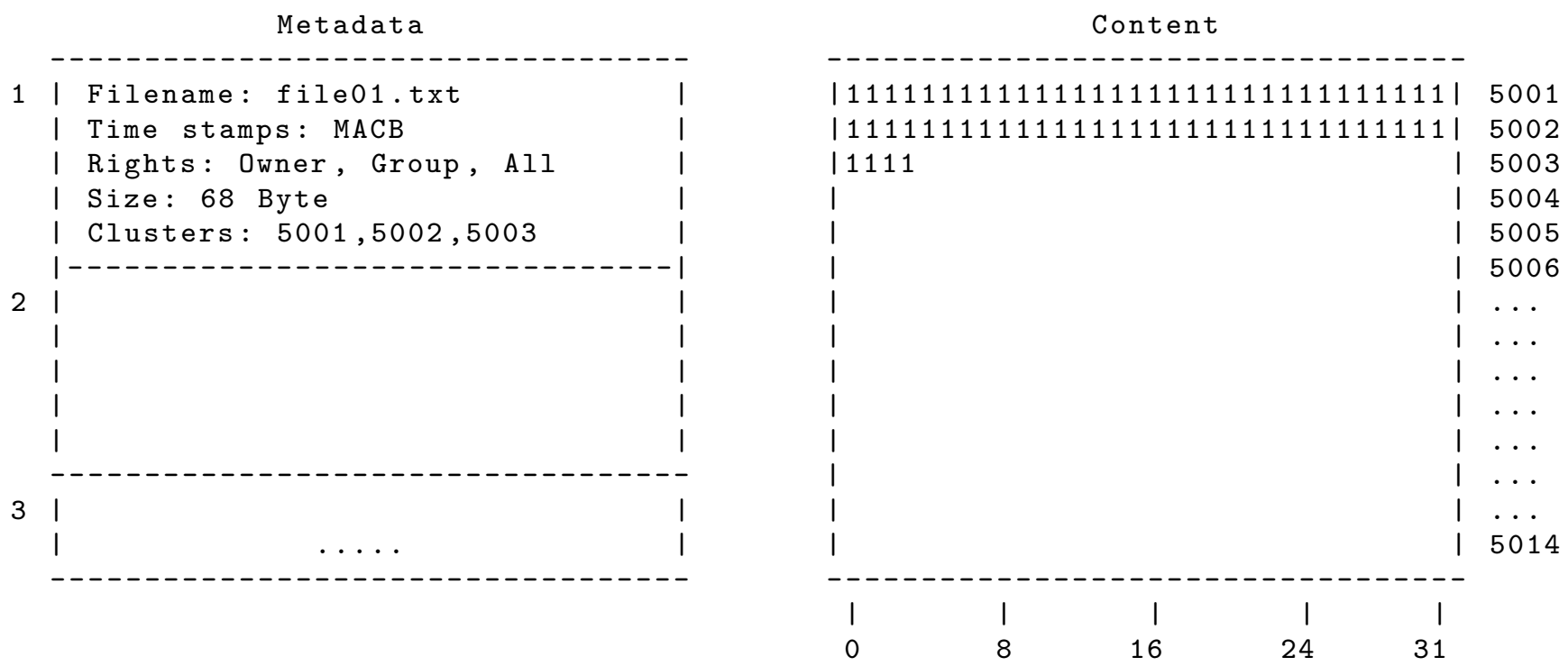
- Organizing data in files
- Maintain file related meta data
- Maintain allocation status of clusters



Allocation table:

1.2 File System

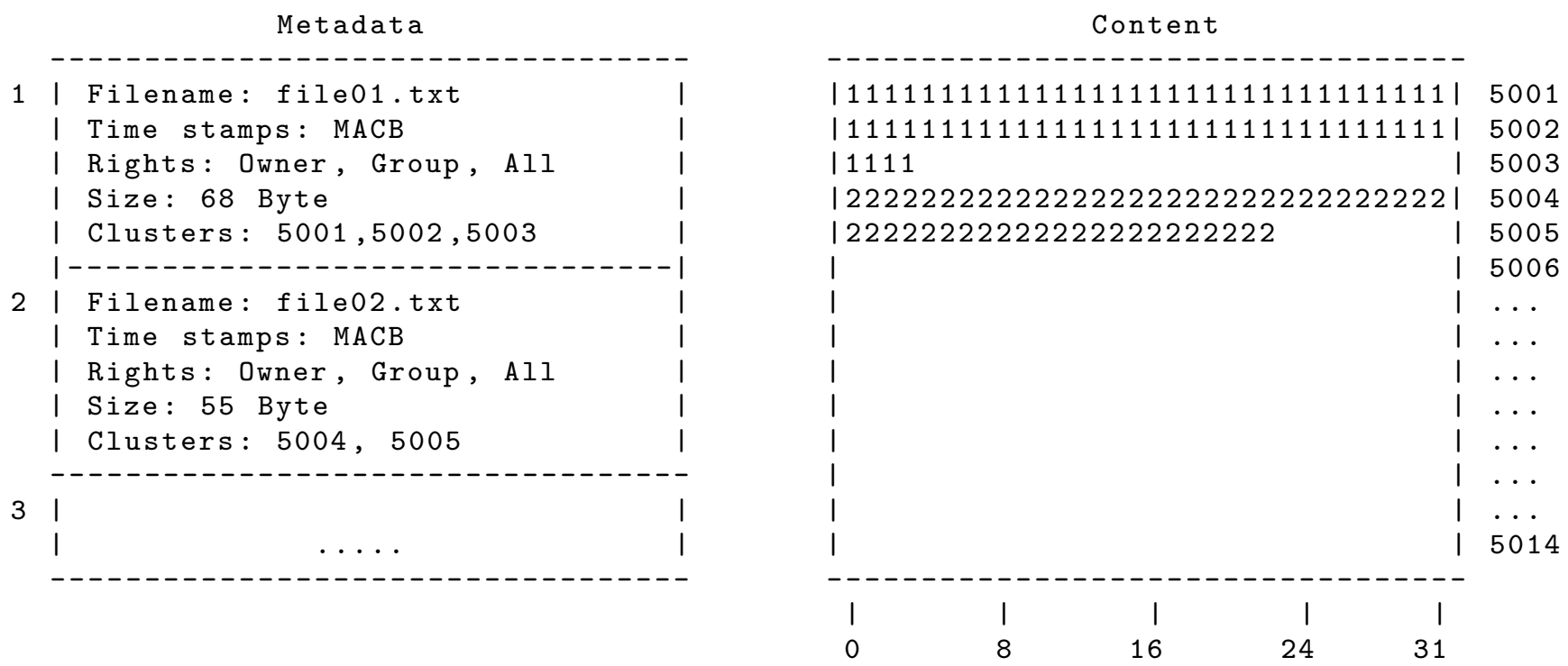
- Organizing data in files
- Maintain file related meta data
- Maintain allocation status of clusters



Allocation table: 5001, 5002, 5003

1.2 File System

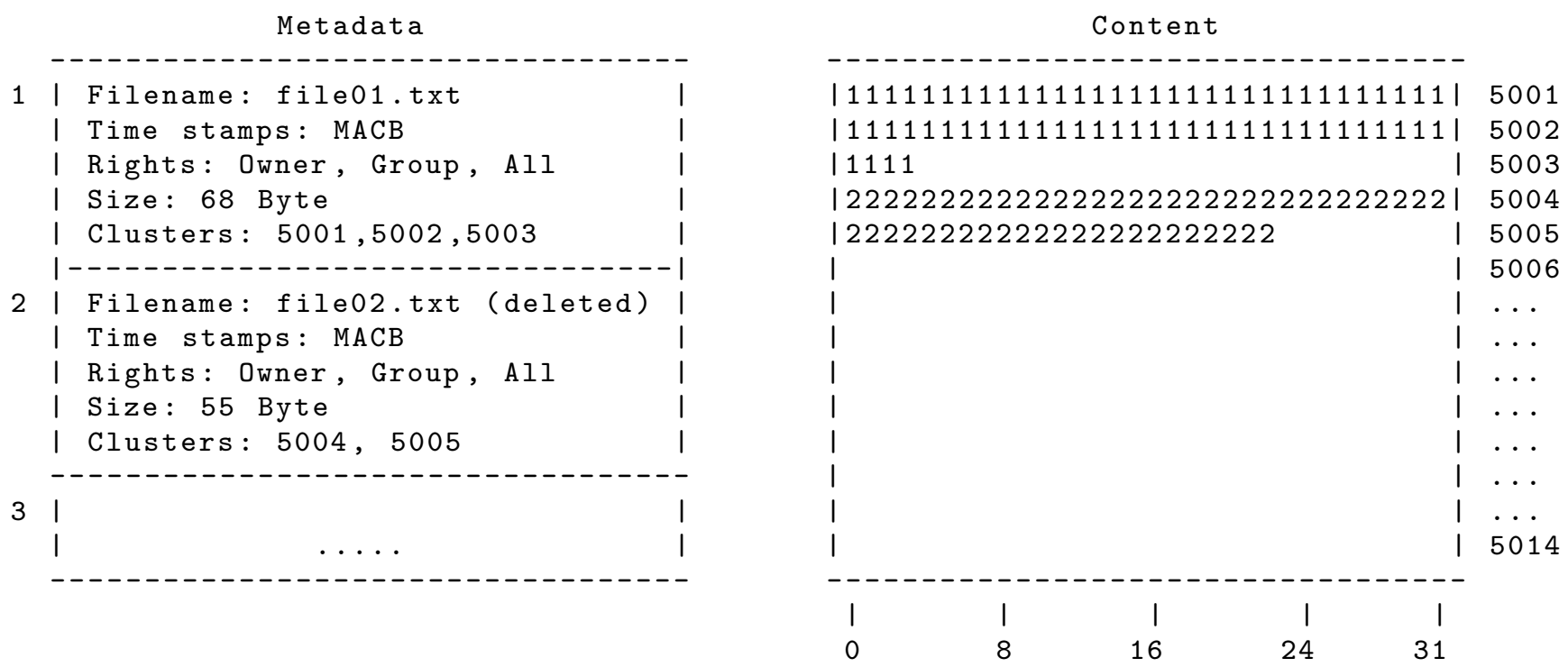
- Organizing data in files
- Maintain file related meta data
- Maintain allocation status of clusters



Allocation table: 5001, 5002, 5003, 5004, 5005

1.2 File System

- Deleting a file
- Maintain file related meta data
- Maintain allocation status of clusters



Allocation table: 5001, 5002, 5003

1.3 Setup your Lab

Create a virtual disk image:

```
dd if=/dev/zero of=fat32.raw bs=4096 count=$((512*256))
dd if=fat32.raw | xxd | less


ls -l fat32.raw
        536870912 Mar 11 11:09 fat32.raw


sudo losetup -fP --show fat32.raw
        /dev/loop15


sudo mkfs.fat -n BooH-FAT32 /dev/loop15
sudo losetup -d /dev/loop15

dd if=fat32.raw | xxd | less
```

Exercise: How many sectors needed to store the FAT?

```
Size of disk image: 536.870.912 Byte
```

-

1.3 Setup your Lab

Create a virtual disk image:

```
dd if=/dev/zero of=fat32.raw bs=4096 count=$((512*256))
dd if=fat32.raw | xxd | less


ls -l fat32.raw
        536870912 Mar 11 11:09 fat32.raw


sudo losetup -fP --show fat32.raw
        /dev/loop15


sudo mkfs.fat -n BooH-FAT32 /dev/loop15
sudo losetup -d /dev/loop15

dd if=fat32.raw | xxd | less
```

Exercise: How many sectors needed to store the FAT?

```
Size of disk image: 536.870.912 Byte

Amount of clusters on the disk:       536870912 / 4096 = 131072
The FAT use 4 bytes for each cluster:       131072 * 4 = 524288
Sector needed to store the FAT:           524288 / 512 =   1024
```

1.3 Setup your Lab

Create a file with 5000 characters:

```
for num in {1..5000}; do echo -n 5; done > 5000.txt
```

Copy file to virtual disk:

```
sudo mkdir /mnt/FAT

sudo mount fat32.raw /mnt/FAT/

sudo cp 5000.txt /mnt/FAT/

ls -lh /mnt/FAT/
              4,9K Mar 11 11:06 5000.txt

sudo umount /mnt/FAT
```

1.4 The Sleuth Kit

```
mmstat          # Volume system information
mmls            # List partition table
mmcat           # Cat a partition

fsstat          # File system information

fls             # List files and directories
fcat            # Cat a file
ffind           # Find filename of an inode

istat           # Inode information
ils             # List inodes
icat            # Cat an inode
ifind           # Find inode of a sector

blkstat         # Information of a data unit
blkls           # Output data units
blkcat          # Cat a data unit

jls             # List content of journal
jcat            # Cat a block from journal

mactime         # File system time line
srch_strings    # Display printable characters
hfind           # Hash database lookup
....
```

2.1 Initial Analysis

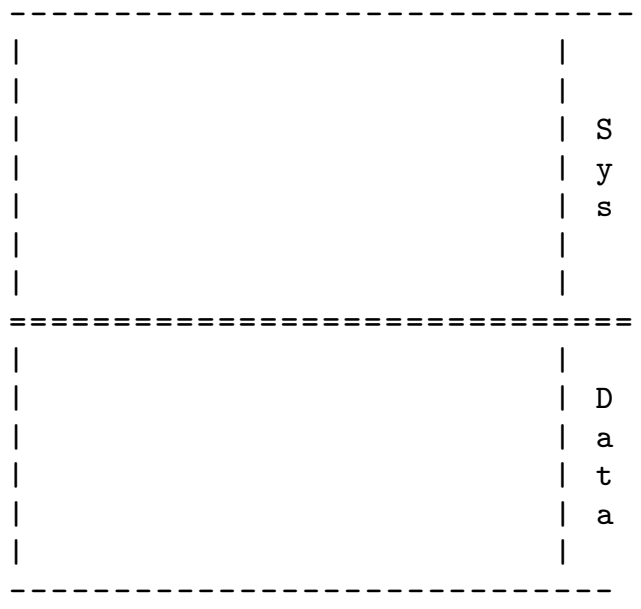
Do initial file system investigation:

```
fsstat fat32.raw

   FILE SYSTEM INFORMATION
   --------------------------------------
   File System Type: FAT32
   Volume Label (Boot Sector): BooH-FAT32

   File System Layout (in sectors)
   Total Range: 0 - 1048571
   * Reserved: 0 - 31
   ** Boot Sector: 0
   ** FS Info Sector: 1
   ** Backup Boot Sector: 6
   * FAT 0: 32 - 1055
   * FAT 1: 1056 - 2079
   * Data Area: 2080 - 1048571
   ** Cluster Area: 2080 - 1048567
   *** Root Directory: 2080 - 2087
    ** Non-clustered: 1048568 - 1048571


   CONTENT INFORMATION
   --------------------------------------
   Sector Size: 512
   Cluster Size: 4096
   Total Cluster Range: 2 - 130812
```



2.1 Initial Analysis

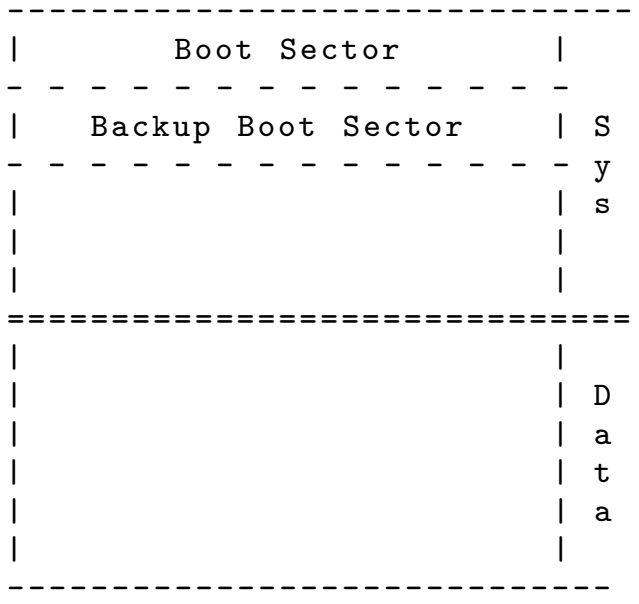
Do initial file system investigation:

```
fsstat fat32.raw

   FILE SYSTEM INFORMATION
   --------------------------------------
   File System Type: FAT32
   Volume Label (Boot Sector): BooH-FAT32

   File System Layout (in sectors)
   Total Range: 0 - 1048571
   * Reserved: 0 - 31
   ** Boot Sector: 0
   ** FS Info Sector: 1
   ** Backup Boot Sector: 6
   * FAT 0: 32 - 1055
   * FAT 1: 1056 - 2079
   * Data Area: 2080 - 1048571
   ** Cluster Area: 2080 - 1048567
   *** Root Directory: 2080 - 2087
    ** Non-clustered: 1048568 - 1048571


   CONTENT INFORMATION
   --------------------------------------
   Sector Size: 512
   Cluster Size: 4096
   Total Cluster Range: 2 - 130812
```



2.1 Initial Analysis

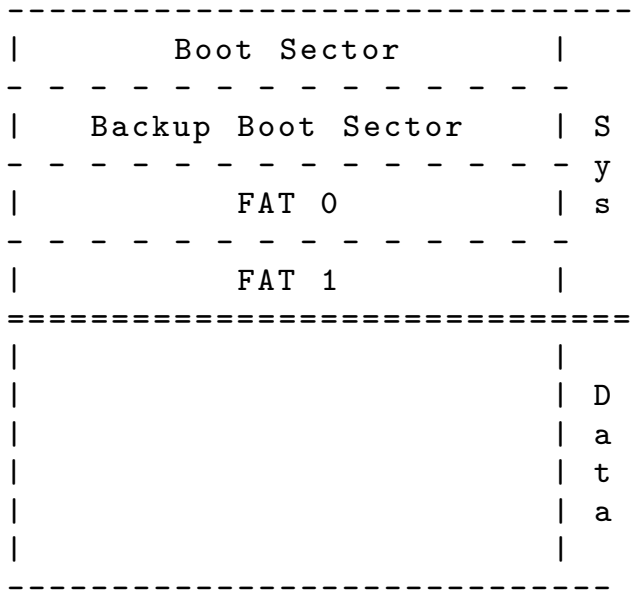
Do initial file system investigation:

```
fsstat fat32.raw

   FILE SYSTEM INFORMATION
   --------------------------------------
   File System Type: FAT32
   Volume Label (Boot Sector): BooH-FAT32

   File System Layout (in sectors)
   Total Range: 0 - 1048571
   * Reserved: 0 - 31
   ** Boot Sector: 0
   ** FS Info Sector: 1
   ** Backup Boot Sector: 6
   * FAT 0: 32 - 1055
   * FAT 1: 1056 - 2079
   * Data Area: 2080 - 1048571
   ** Cluster Area: 2080 - 1048567
   *** Root Directory: 2080 - 2087
    ** Non-clustered: 1048568 - 1048571


   CONTENT INFORMATION
   --------------------------------------
   Sector Size: 512
   Cluster Size: 4096
   Total Cluster Range: 2 - 130812
```



2.1 Initial Analysis

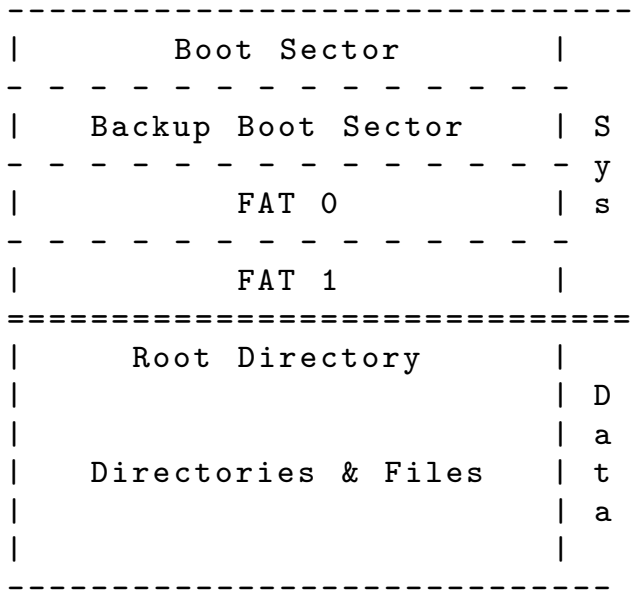
Do initial file system investigation:

```
fsstat fat32.raw

   FILE SYSTEM INFORMATION
   --------------------------------------
   File System Type: FAT32
   Volume Label (Boot Sector): BooH-FAT32

   File System Layout (in sectors)
   Total Range: 0 - 1048571
   * Reserved: 0 - 31
   ** Boot Sector: 0
   ** FS Info Sector: 1
   ** Backup Boot Sector: 6
   * FAT 0: 32 - 1055
   * FAT 1: 1056 - 2079
   * Data Area: 2080 - 1048571
   ** Cluster Area: 2080 - 1048567
   *** Root Directory: 2080 - 2087
    ** Non-clustered: 1048568 - 1048571


   CONTENT INFORMATION
   --------------------------------------
   Sector Size: 512
   Cluster Size: 4096
   Total Cluster Range: 2 - 130812
```



2.2 Boot Sector Analysis

Dump Boot Sector:

```
dd if=fat32.raw count=1 | xxd | less

00000000: eb58 906d 6b66 732e 6661 7400 0208 2000  .X.mkfs.fat... .
00000010: 0200 0000 00f8 0000 3f00 2000 0000 0000  ........?. .....
00000020: fcff 0f00 0004 0000 0000 0000 0200 0000  ................
00000030: 0100 0600 0000 0000 0000 0000 0000 0000  ................
00000040: 8000 2955 58f3 6742 6f6f 482d 4641 5433  ..)UX.gBooH-FAT3
00000050: 3220 4641 5433 3220 2020 0e1f be77 7cac  2 FAT32   ...w|.
00000060: 22c0 740b 56b4 0ebb 0700 cd10 5eeb f032  ".t.V.......^..2
00000070: e4cd 16cd 19eb fe54 6869 7320 6973 206e  .......This is n
00000080: 6f74 2061 2062 6f6f 7461 626c 6520 6469  ot a bootable di
00000090: 736b 2e20 2050 6c65 6173 6520 696e 7365  sk.  Please inse
000000a0: 7274 2061 2062 6f6f 7461 626c 6520 666c  rt a bootable fl
000000b0: 6f70 7079 2061 6e64 0d0a 7072 6573 7320  oppy and..press
.....
```

Boot Sector Data Structure:

```
Offset      Length   Item              Interpretation
00 (0x00)   3        Jump bootstrap    JMP SHORT 88 (+2); NOP
03 (0x03)   8        OEM name          mkfs.fat
11 (0x0B)   2        Bytes/sector      0x0002  -->  0x0200 =  512 Bytes
13 (0x0D)   1        Sectors/Cluster   0x08                = 4096 Bytes
14 (0x0E)   2        Reserved Area     0x2000  -->  0x0020 =   32 Sectors
16 (0x10)   1        Number of FATs    0x02                =    2 FATs
32 (0x20)   4        Sectors in FS     0xfcff0f -> 0xffffc = 1.048.572
.....
```

2.3 Root Directory Analysis

Dump Root Directory:

```
dd if=fat32.raw skip=2080 count=1 | xxd | less
```

```
00000000: 426f 6f48 2d46 4154 3332 2008 0000 f553  BooH-FAT32 ....S
00000010: 6b5c 6b5c 0000 f553 6b5c 0000 0000 0000  k\k\...Sk\......
00000020: 4135 0030 0030 0030 002e 000f 00d2 7400  A5.0.0.0......t.
00000030: 7800 7400 0000 ffff ffff 0000 ffff ffff  x.t.............
00000040: 3530 3030 2020 2020 5458 5420 0043 d050  5000    TXT .C.P
00000050: 6b5c 6b5c 0000 d050 6b5c 0300 8813 0000  k\k\...Pk\......
00000060: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000070: 0000 0000 0000 0000 0000 0000 0000 0000  ................
.....
```

Interpretation of the data:

Offset	Length	Item	Interpretation
00 (0x00)	11	File Name	BooH-FAT32
11 (0x0B)	1	Attribute	0x08 --> Volume Lable
00 (0x00)	11	File Name	5000 TXT
11 (0x0B)	1	Attribute	0x20 --> Archive
.....			
20 (0x14)	2	Start FAT High	0x0000 --> Empty
26 (0x1A)	2	Start FAT Low	0x0300 --> 03
28 (0x1C)	4	Size in Byes	0x8813 --> 0x1388 == 5000

2.4 FAT Analysis

Dump FAT 0:

```
dd if=fat32.raw skip=32 count=1024 | xxd | less

00000000: f8ff ff0f ffff ff0f f8ff ff0f 0400 0000  ................
00000010: ffff ff0f 0000 0000 0000 0000 0000 0000  ................
```

Dump FAT 1:

```
dd if=fat32.raw skip=1056 count=1024 | xxd | less
00000000: f8ff ff0f ffff ff0f f8ff ff0f 0400 0000  ................
00000010: ffff ff0f 0000 0000 0000 0000 0000 0000  ................
```

Interpretation of values:

Little Endian:	Big Endian:	Cluster:
0000 0000	0000 0000	Not Allocated
0400 0000	0000 0004	Allocated
f8ff ff0f	0fff fff8	End of file
ffff ff0f	0fff ffff	End of file
f7ff ff0f	0fff fff7	Damaged

2.4 FAT Analysis

Dump FAT 0:

```
dd if=fat32.raw skip=32 count=1024 | xxd | less

00000000: f8ff ff0f ffff ff0f f8ff ff0f 0400 0000  ................
00000010: ffff ff0f 0000 0000 0000 0000 0000 0000  ................
```

Interpretation of the data:

Entry Number:	Little Endian:	Big Endian:	Cluster:
0	f8ff ff0f	0fff fff8	Reserved
1	ffff ff0f	0fff ffff	Reserved
2	f8ff ff0f	0fff fff8	0 Allocated; Last
3	0400 0000	0000 0004	1 Allocated; Next: 4
4	ffff ff0f	0fff ffff	2 Allocated; Last
5	0000 0000	0000 0000	3 Not Allocated
6	0000 0000	0000 0000	4 Not Allocated

2.5 File Content Analysis

Calculate Sector Offset:

```
Data/Cluster Area start at sector: 2080
FAT entry 3 - 2 --> Cluster 1
1 Cluster = 8 Sector

To dump cluster 1-2: (1 * 8) + 2080
--> dd if=fat32.raw skip=2088 count=16 | xxd | less
```

Dump File Content:

```
dd if=fat32.raw skip=2088 count=16 | xxd | less

00000000: 3535 3535 3535 3535 3535 3535 3535 3535  5555555555555555
00000010: 3535 3535 3535 3535 3535 3535 3535 3535  5555555555555555
00000020: 3535 3535 3535 3535 3535 3535 3535 3535  5555555555555555
00000030: 3535 3535 3535 3535 3535 3535 3535 3535  5555555555555555
.....
00001360: 3535 3535 3535 3535 3535 3535 3535 3535  5555555555555555
00001370: 3535 3535 3535 3535 3535 3535 3535 3535  5555555555555555
00001380: 3535 3535 3535 3535 0000 0000 0000 0000  55555555........
00001390: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000013a0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
.....
00001fc0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001fd0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001fe0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001ff0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

2.6 Bring it all together

Analyze Boot Sector:

```
FAT 0: 32 - 1055
FAT 1: 1056 - 2079
Data Area: 2080 - 1048571
Root Directory: 2080 - 2087
```

Analyze Direcoty Entry:

```
26 (0x1A)    2       Start FAT Low    0x0300 --> 03
28 (0x1C)    4       Size in Byes     0x8813 --> 0x1388 == 5000
```

Analyze FAT:

```
00000000: f8ff ff0f ffff ff0f f8ff ff0f 0400 0000  ................
00000010: ffff ff0f 0000 0000 0000 0000 0000 0000  ................
```

Review Data:

```
00000000: 3535 3535 3535 3535 3535 3535 3535 3535  5555555555555555
00000010: 3535 3535 3535 3535 3535 3535 3535 3535  5555555555555555
.....
.....
00001fe0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001ff0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

3.1 Hiding Data: Kick-Start

What need to be done:

- Which cluster we like to hide the data
- Identify corresponding FAT entry
- Mark FAT entry as defect
- Hide data in selected cluster
  - `hexeditor`
  - `dd`

What we have:

- FAT

```
FAT32:     0    1    2    3    4    5    6    7    8    9   10   11   12
          ---------------------------------------------------------------
          |Res |Res | F8 | 04 | FF |    |    |    |    |    |    |    |
          ---------------------------------------------------------------
```

- Cluster area

```
Cluster:       0         1          2         3         4         5         6
          ---------------------------------------------------------------------
          |VolumLab|55555555||5555    |         |         |         |         |
          ---------------------------------------------------------------------
```

3.1 Hiding Data: Kick-Start

What need to be done:

- Which cluster we like to hide the data
- Identify corresponding FAT entry
- Mark FAT entry as defect
- Hide data in selected cluster
  - `hexeditor`
  - `dd`

What we want:

- FAT

```
FAT32:     0    1    2    3    4    5    6    7    8    9   10   11   12
          ---------------------------------------------------------------
          |Res |Res | F8 | 04 | FF |    | F7 |    |    |    |    |    |
          ---------------------------------------------------------------
```

- Cluster area

```
Cluster:       0         1          2         3         4         5         6
          ---------------------------------------------------------------------
          |VolumLab|55555555||5555    |         | Secret |         |         |
          ---------------------------------------------------------------------
```

3.2 Hiding Data: On the road

Manipulating the FAT:

```
hexedit fat32.raw

.....
00003FF0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00004000   F8 FF FF 0F  FF FF FF 0F  F8 FF FF 0F  04 00 00 00  ................
00004010   FF FF FF 0F  00 00 00 00  F7 FF FF 0F  00 00 00 00  ................
00004020   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
.....
.....
00083FF0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
00084000   F8 FF FF 0F  FF FF FF 0F  F8 FF FF 0F  04 00 00 00  ................
00084010   FF FF FF 0F  00 00 00 00  F7 FF FF 0F  00 00 00 00  ................
00084020   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
.....
```

Most important `hexedit` commands:

```
      F2  - Save File
<ctrl>-G  - Go to address
<ctrl>-X  - Save and Exit
<ctrl>-C  - Exit without saving
```

3.2 Hiding Data: Defect sector data hiding

Calculate Sector Offset:

```
Data/Cluster Area start at sector: 2080
FAT entry 6 - 2 --> Cluster 4
--> 1 Cluster = 8 Sector

Start-Sector to use: (4 * 8) + 2080 = 2112
We can use max. 8 sectors
```

Write data to disk:

```
echo -n "My secret message is: The fox is sleeping" |
            dd of=fat32.raw seek=2112 count=1 conv=notrunc
```

Validate your activity:

```
dd if=fat32.raw skip=2112 | xxd | less
```

3.3 Investigate actual state of the disk

Discuter: Outcome of 'The Sleuth Kit' commands

```
$ fsstat fat32.raw

$ fls -r fat32.raw
    r/r 3:      BooH-FAT32  (Volume Label Entry)
    r/r 5:      5000.txt

$ istat fat32.raw 3
    Allocated
    File Attributes: Volume Label
    Size: 0
    Name: BooH-FAT32

$ istat fat32.raw 4
    Allocated
    Size: 0
    Name: 5000.txt

$ istat fat32.raw 5
    Size: 5000
    Name: 5000.TXT

    Sectors:
    2088 2089 2090 2091 2092 2093 2094 2095
    2096 2097 0 0 0 0 0 0

Other commands to test: ifind, ffind, icat, .....
```

3.4 Create additional files

Create a 2 cluster file with 8000 characters:

```
for num in {1..8000}; do echo -n 8; done > 8000.txt
```

Create a 1 cluster file with 100 characters:

```
for num in {1..100}; do echo -n 1; done > 100.txt
```

Copy files to virtual disk:

```
sudo mount fat32.raw /mnt/FAT/


sudo cp 8000.txt /mnt/FAT/
sudo cp 100.txt /mnt/FAT/


ls -lh /mnt/FAT/
                100 Apr 10 14:41 100.txt
              4,9K Mar 11 11:06 5000.txt
              7,9K Apr 10 14:41 8000.txt


sudo umount /mnt/FAT
```

3.5 Investigate new state of the disk

```
$ fls -r fat32.raw
    r/r 3:      BooH-FAT32  (Volume Label Entry)
    r/r 5:      5000.txt
    r/r 7:      8000.txt
    r/r 9:      100.txt
```

```
dd if=fat32.raw skip=2080 count=8 | xxd | less

00000040: 3530 3030 2020 2020 5458 5420 0043 d050  5000    TXT .C.P
00000050: 6b5c 6b5c 0000 d050 6b5c 0300 8813 0000  k\k\...Pk\......
00000060: 4138 0030 0030 0030 002e 000f 008b 7400  A8.0.0.0......t.
00000070: 7800 7400 0000 ffff ffff 0000 ffff ffff  x.t.............
00000080: 3830 3030 2020 2020 5458 5420 00a0 3465  8000    TXT ..4e
00000090: 8a5c 8a5c 0000 3465 8a5c 0500 401f 0000  .\.\..4e.\..@...
000000a0: 4131 0030 0030 002e 0074 000f 0071 7800  A1.0.0...t...qx.
000000b0: 7400 0000 ffff ffff ffff 0000 ffff ffff  t...............
000000c0: 3130 3020 2020 2020 5458 5420 0054 3865  100     TXT .T8e
000000d0: 8a5c 8a5c 0000 3865 8a5c 0800 6400 0000  .\.\..8e.\..d...
```

FAT32:	0	1	2	3	4	5	6	7	8	9	10	11	12
	Res	Res	F8	??		??	F7		??				

3.5 Investigate new state of the disk

```
$ fls -r fat32.raw
    r/r 3:      BooH-FAT32  (Volume Label Entry)
    r/r 5:      5000.txt
    r/r 7:      8000.txt
    r/r 9:      100.txt
```

```
dd if=fat32.raw skip=32 count=1 | xxd | less

    00000000: f8ff ff0f ffff ff0f f8ff ff0f 0400 0000  ................
    00000010: ffff ff0f 0700 0000 f7ff ff0f ffff ff0f  ................
    00000020: ffff ff0f 0000 0000 0000 0000 0000 0000  ................
    .....
```

FAT32:	0	1	2	3	4	5	6	7	8	9	10	11	12
	Res	Res	F8	04	FF	07	F7	FF	FF				

3.5 Investigate new state of the disk

```
To dump cluster linked to FAT 5: (3 * 8) + 2080 = 2104
------------------------------------------------------
dd if=fat32.raw skip=2104 count=8 | xxd | less
    00000000: 3838 3838 3838 3838 3838 3838 3838 3838  8888888888888888

To dump cluster linked to FAT 6: (4 * 8) + 2080 = 2112
------------------------------------------------------
dd if=fat32.raw skip=2112 count=8 | xxd | less
    00000000: 4d79 2073 6563 7265 7420 6d65 7373 6167  My secret messag
    00000010: 6520 6973 3a20 5468 6520 666f 7820 6973  e is: The fox is
    00000020: 2073 6c65 6570 696e 6700 0000 0000 0000   sleeping.......

To dump cluster linked to FAT 7: (5 * 8) + 2080 = 2120
------------------------------------------------------
dd if=fat32.raw skip=2120 count=8 | xxd | less
    00000000: 3838 3838 3838 3838 3838 3838 3838 3838  8888888888888888

To dump cluster linked to FAT 8: (6 * 8) + 2080 = 2128
------------------------------------------------------
dd if=fat32.raw skip=2128 count=8 | xxd | less
    00000000: 3131 3131 3131 3131 3131 3131 3131 3131  1111111111111111
```

FAT32:	0	1	2	3	4	5	6	7	8	9	10	11	12
	Res	Res	F8	04	FF	07	F7	FF	FF				

3.6 Investigate with The Sleuth Kit Tools

```
fls fat32.raw
    r/r 3:      BooH-FAT32  (Volume Label Entry)
    r/r 5:      5000.txt
    r/r 7:      8000.txt
    r/r 9:      100.txt


istat fat32.raw 5
    Name: 5000.TXT

    Sectors:
    2088 2089 2090 2091 2092 2093 2094 2095
    2096 2097 0 0 0 0 0 0


istat fat32.raw 7
    Name: 8000.TXT

    Sectors:
    2104 2105 2106 2107 2108 2109 2110 2111
    2120 2121 2122 2123 2124 2125 2126 2127


istat fat32.raw 9
    Name: 100.TXT

    Sectors:
    2128 0 0 0 0 0 0 0
```

Bad out of Hell

Digital Forensics 1.0.7

Michael Hamm - CIRCL

TLP:CLEAR

Bad Cluster Forensics - FAT32