Analysis of an image file

Ayman El Hajjar

March 5, 2018

In this lab you are expected to retrieve the structure of this image. You are also expected to retrieve as much information as possible from the the drive.

1 Download the image File and SleuthKit if not available

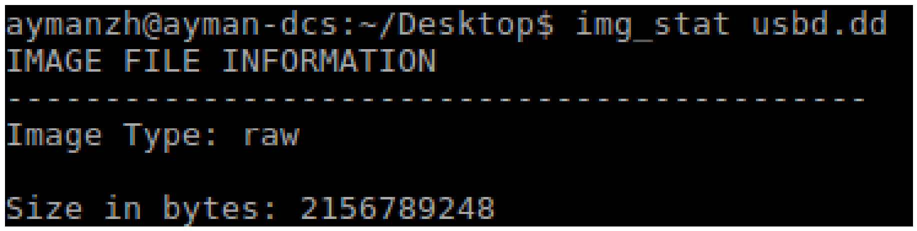
1. You will need to download the image file "**usbd.dd.tar.gz**"
2. Un-compress the file to obtain the original image. From Desktop I used this command, **tar -xvf /home/labadmin/Downloads/usbd.dd.tar.gz**
3. Download Sleuthkit form **https://github.com/sleuthkit/sleuthkit/releases/download/sleuthkit-4.3.0 /sleuthkit-4.3.0.tar.gz**

2 Analyse the image file

2.1 Understand the image file

2.1.1 display basic details of an image file

Lets first display image type and size of image file usbd.dd

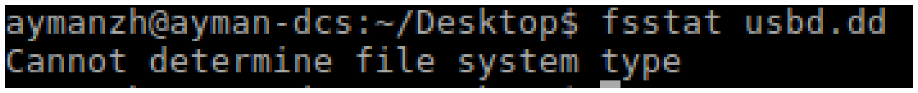


```
aymanzh@ayman-dcs:~/Desktop$ img_stat usbd.dd
IMAGE FILE INFORMATION
--------------------------------------------
Image Type: raw

Size in bytes: 2156789248
```

The information this tool gave us can be divided into two parts. First it gave us that the image is a copy bit by bit and thus the image type is raw. Second it gave us the size of the image

2.1.2 Display details of a file system

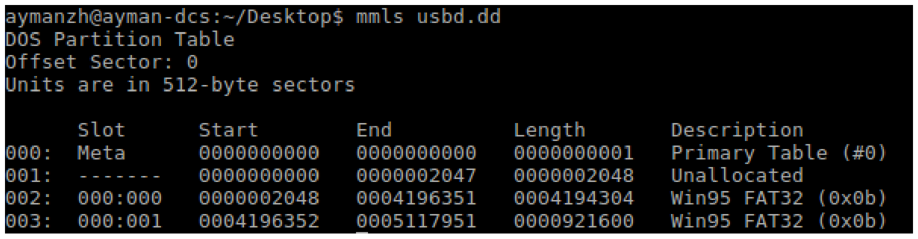
Display detailed file system and metadata information of image file usbd.dd However this tool returned error: Cannot determine file system type. This does not mean it is an error, it only means that this image contains a partition. Therefore no specific file system for the whole image.



```
aymanzh@ayman-dcs:~/Desktop$ fsstat usbd.dd
Cannot determine file system type
```

2.1.3 display the layout of media management systems/partition tables

- Since we know for sure that we now have an image that contains 2 or more partitions. We need to understand the layout of the image.
- Display partition information of image file usbd.dd (Note: If it helps you in understanding the use of mmls, the output of this command in some cases can be similar to "**fdisk -lu ¡device¿**")



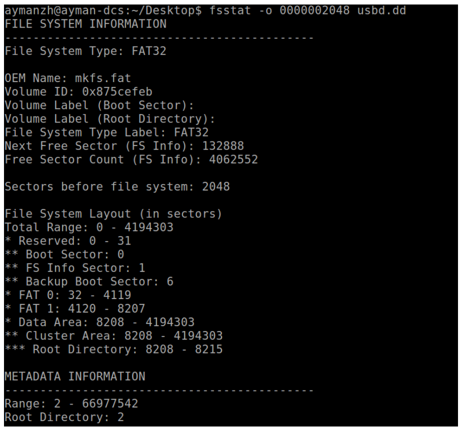
```
aymanzh@ayman-dcs:~/Desktop$ mmls usbd.dd
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000002047   0000002048   Unallocated
002:  000:000   0000002048   0004196351   0004194304   Win95 FAT32 (0x0b)
003:  000:001   0004196352   0005117951   0000921600   Win95 FAT32 (0x0b)
```

- The figure above gave us the information we are looking for. A primary partition on the first byte of the image. That is in Slot 0 Slot 1 contains an unallocated partition of 2kB Slot 2 contains a FAT32 partition that starts in 0000002048 bytes and ends in 0004196351 bytes. Slot 3 contains another FAT32 partition that starts in 0004196352 bytes and ends in 0005117951 bytes
- We can assume that data are not in slot 0. However we can not assume that there is nothing in the unallocated drive.
- Now we know all partitions details we can go back to fsstat to obtain details of file system and its metdata.

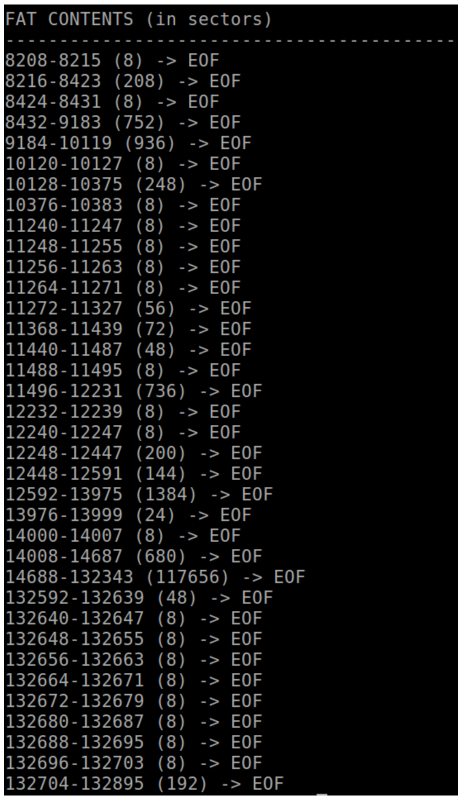
2.1.4 Display details of a file system - Offest

Since we know of all partitions, we can now decided which partition we are looking to obtain information about.

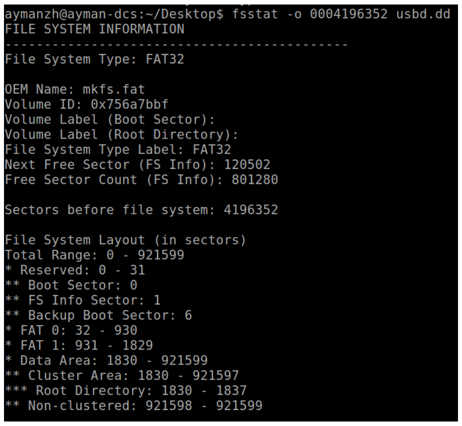
fsstat -o 0000002048 usbd.dd



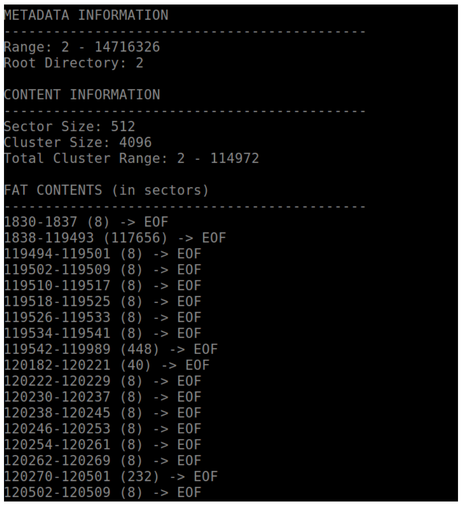
```
aymanzh@ayman-dcs:~/Desktop$ fsstat -o 0000002048 usbd.dd
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: FAT32

OEM Name: mkfs.fat
Volume ID: 0x875cefeb
Volume Label (Boot Sector):
Volume Label (Root Directory):
File System Type Label: FAT32
Next Free Sector (FS Info): 132888
Free Sector Count (FS Info): 4062552

Sectors before file system: 2048

File System Layout (in sectors)
Total Range: 0 - 4194303
* Reserved: 0 - 31
** Boot Sector: 0
** FS Info Sector: 1
** Backup Boot Sector: 6
* FAT 0: 32 - 4119
* FAT 1: 4120 - 8207
* Data Area: 8208 - 4194303
** Cluster Area: 8208 - 4194303
*** Root Directory: 8208 - 8215

METADATA INFORMATION
--------------------------------------------
Range: 2 - 66977542
Root Directory: 2
```



```
FAT CONTENTS (in sectors)
--------------------------------------------
8208-8215 (8) -> EOF
8216-8423 (208) -> EOF
8424-8431 (8) -> EOF
8432-9183 (752) -> EOF
9184-10119 (936) -> EOF
10120-10127 (8) -> EOF
10128-10375 (248) -> EOF
10376-10383 (8) -> EOF
11240-11247 (8) -> EOF
11248-11255 (8) -> EOF
11256-11263 (8) -> EOF
11264-11271 (8) -> EOF
11272-11327 (56) -> EOF
11368-11439 (72) -> EOF
11440-11487 (48) -> EOF
11488-11495 (8) -> EOF
11496-12231 (736) -> EOF
12232-12239 (8) -> EOF
12240-12247 (8) -> EOF
12248-12447 (200) -> EOF
12448-12591 (144) -> EOF
12592-13975 (1384) -> EOF
13976-13999 (24) -> EOF
14000-14007 (8) -> EOF
14008-14687 (680) -> EOF
14688-132343 (117656) -> EOF
132592-132639 (48) -> EOF
132640-132647 (8) -> EOF
132648-132655 (8) -> EOF
132656-132663 (8) -> EOF
132664-132671 (8) -> EOF
132672-132679 (8) -> EOF
132680-132687 (8) -> EOF
132688-132695 (8) -> EOF
132696-132703 (8) -> EOF
132704-132895 (192) -> EOF
```



```
aymanzh@ayman-dcs:~/Desktop$ fsstat -o 0004196352 usbd.dd
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: FAT32

OEM Name: mkfs.fat
Volume ID: 0x756a7bbf
Volume Label (Boot Sector):
Volume Label (Root Directory):
File System Type Label: FAT32
Next Free Sector (FS Info): 120502
Free Sector Count (FS Info): 801280

Sectors before file system: 4196352

File System Layout (in sectors)
Total Range: 0 - 921599
* Reserved: 0 - 31
** Boot Sector: 0
** FS Info Sector: 1
** Backup Boot Sector: 6
* FAT 0: 32 - 930
* FAT 1: 931 - 1829
* Data Area: 1830 - 921599
** Cluster Area: 1830 - 921597
*** Root Directory: 1830 - 1837
** Non-clustered: 921598 - 921599
```



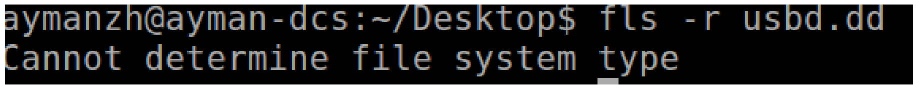
```
METADATA INFORMATION
--------------------------------------------
Range: 2 - 14716326
Root Directory: 2

CONTENT INFORMATION
--------------------------------------------
Sector Size: 512
Cluster Size: 4096
Total Cluster Range: 2 - 114972

FAT CONTENTS (in sectors)
--------------------------------------------
1830-1837 (8) -> EOF
1838-119493 (117656) -> EOF
119494-119501 (8) -> EOF
119502-119509 (8) -> EOF
119510-119517 (8) -> EOF
119518-119525 (8) -> EOF
119526-119533 (8) -> EOF
119534-119541 (8) -> EOF
119542-119989 (448) -> EOF
120182-120221 (40) -> EOF
120222-120229 (8) -> EOF
120230-120237 (8) -> EOF
120238-120245 (8) -> EOF
120246-120253 (8) -> EOF
120254-120261 (8) -> EOF
120262-120269 (8) -> EOF
120270-120501 (232) -> EOF
120502-120509 (8) -> EOF
```

2.2 Look at files and Directories in the usb image file

2.2.1 list file and directory names in a forensic image

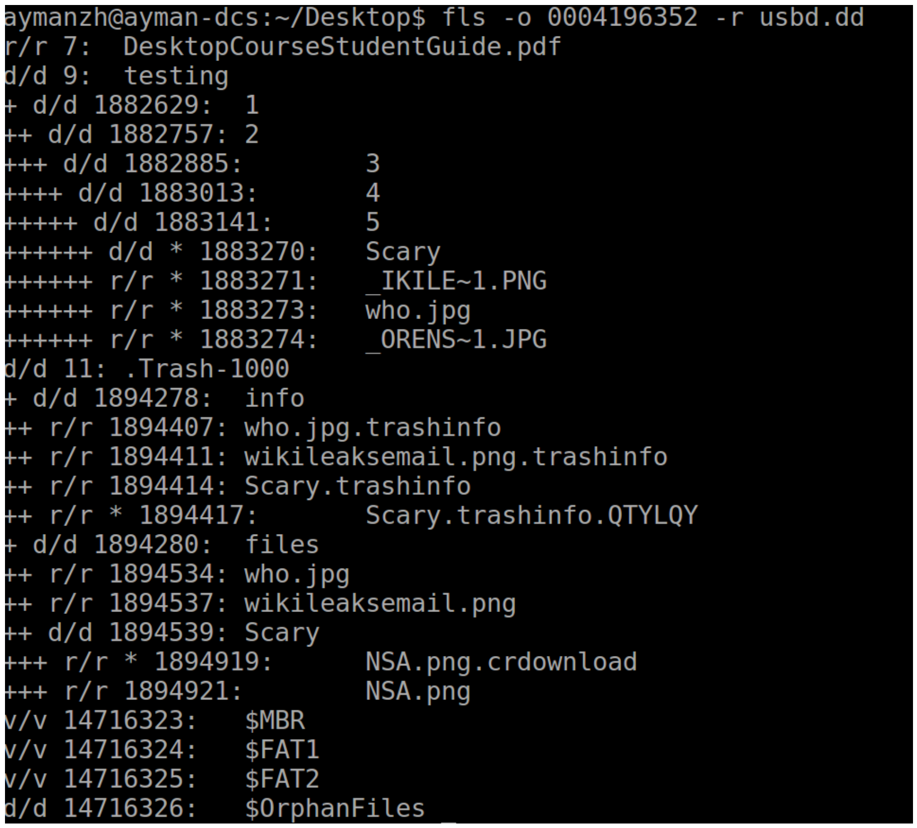
A good tool in TSK to start with is the **fls** tool. **fls** allows forensic experts to identify the structure of files and directories in an image. It also allow them to see any hidden or deleted files.

Similar to the **fsstat** tool, you will need to select the offset of the partition you want to investigate.



```
aymanzh@ayman-dcs:~/Desktop$ fls -r usbd.dd
Cannot determine file system type
```

List file and directory names of image file usbd.dd, and recurse on directories
fls -o 0000002048 -r usbd.dd
fls -o 0004196352 -r usbd.dd



```
aymanzh@ayman-dcs:~/Desktop$ fls -o 0004196352 -r usbd.dd
r/r 7:  DesktopCourseStudentGuide.pdf
d/d 9:  testing
+ d/d 1882629:  1
++ d/d 1882757: 2
+++ d/d 1882885:        3
++++ d/d 1883013:       4
+++++ d/d 1883141:      5
++++++ d/d * 1883270:   Scary
++++++ r/r * 1883271:   _IKILE~1.PNG
++++++ r/r * 1883273:   who.jpg
++++++ r/r * 1883274:   _ORENS~1.JPG
d/d 11: .Trash-1000
+ d/d 1894278:  info
++ r/r 1894407: who.jpg.trashinfo
++ r/r 1894411: wikileaksemail.png.trashinfo
++ r/r 1894414: Scary.trashinfo
++ r/r * 1894417:       Scary.trashinfo.QTYLQY
+ d/d 1894280:  files
++ r/r 1894534: who.jpg
++ r/r 1894537: wikileaksemail.png
++ d/d 1894539: Scary
+++ r/r * 1894919:      NSA.png.crdownload
+++ r/r 1894921:        NSA.png
v/v 14716323:   $MBR
v/v 14716324:   $FAT1
v/v 14716325:   $FAT2
d/d 14716326:   $OrphanFiles
```

- List file and directory names of image file usbd.dd, and display file details in ”long” format. Show the output
- List file and directory names of image file usbd.dd, and display full path of files. Show the output
- List only deleted entries of image file usbd.dd. Show the output

- List file and directory names of image file usbd.dd, and provide verbose output of analysis

2.2.2 list inode information with the following column order

- st_ino: The inode number.
- st_alloc: Allocation status: 'a' for allocated inode, 'f' for free inode.
- st_uid: Owner user ID.
- st_gid: Owner group ID.
- st_mtime : UNIX time (seconds) of last file modification.
- st_atime : UNIX time (seconds) of last file access.
- st_ctime : UNIX time (seconds) of last inode status change.
- st_dtime : UNIX time (seconds) of file deletion.
- st_mode : File type and permissions (octal).
- st_nlink : Number of hard links.
- st_size : File size in bytes.
- st_block0,st_block1 : The first two entries in the direct block address list.

Lists inode information of only deleted files within usbd.dd (again going back to the layout above for usbd.dd, these commands would be done by specifying the offset "for example -o 0000002048"). **ils usbd.dd**

Lists inode information of all files within usbd.dd **ils -e usbd.dd**

Lists inode information of file at inode 54 within usbd.dd **ils usbd.dd 54**

2.2.3 display details of a meta-data structure (i.e. inode)

Display the uid, gid, mode, size, link number, MAC times and all the disk units a structure has allocated for a file at inode 54 within usbd.dd **istat usbd.dd 54**

2.2.4 sort files in an image into categories based on file type

Creates a list in a directory called outputdir of files, filetype & inode information for usbd.dd **sorter -d outputdir usbd.dd**

Same as above but with no requirement of a directory for saving the data (i.e. output to stdout) **sorter -l usbd.dd**

2.3 Files recovery

2.3.1 copy files by inode number

- Recover file at inode 54 in image usbd.dd, even if it was deleted, and save the results to a file called file.bin **icat -r usbd.dd 54 > file.bin**
- Same as above, but also recover the slack space along with the file, and save the results to a file called fileslack.bin **icat -r -s usbd.dd 54 > fileslack.bin**
- Look at the various options you can use with **icat**

3 Conclusion

Show the structure of the image (All partitions) and justify your results.