The Sleuth Kit (TSK)

Version 4.9.0

UNIVERSITY OF BALTIMORE
TOWSON UNIVERSITY
BJA

Overview

- Autopsy vs. The Sleuth Kit
- The layer of the TSK architecture
- The tools provided by each layer of TSK

Open Source Digital Forensics



Autopsy® is an easy to use, GUI-based program that allows you to efficiently analyze hard drives and smart phones. It has a plug-in architecture that allows you to find add-on modules or develop custom modules in Java or Python.



The Sleuth Kit® is a collection of command line tools and a C library that allows you to analyze disk images and recover files from them. It is used behind the scenes in Autopsy and many other open source and commercial forensics tools.

These tools are used by thousands of users around the world and have community-based e-mail lists and forums. Commercial training, support, and custom development is available from Basis Technology.

Follow @sleuthkit

Input Data

- Analyzes raw (i.e. dd), Expert Witness (i.e. EnCase) and AFF file system and disk images. (Sleuth Kit Informer #11)
- Supports the NTFS, FAT, ExFAT, UFS 1, UFS 2, EXT2FS, EXT3FS, Ext4, HFS, ISO 9660, and YAFFS2 file systems (even when the host operating system does not or has a different endian ordering).
- Tools can be run on a live Windows or UNIX system during Incident Response. These tools will show files that have been "hidden" by rootkits and will not modify the A-Time of files that are viewed. (Sleuth Kit Informer #13)

Search Techniques

- List allocated and deleted ASCII and Unicode file names. (Sleuth Kit Informer #14 (FAT Recovery), #16 (NTFS Orphan Files))
- Display the details and contents of all NTFS attributes (including all Alternate Data Streams).
- Display file system and meta-data structure details.
- Create time lines of file activity, which can be imported into a spread sheet to create graphs and reports. (Sleuth Kit Informer #5)
- Lookup file hashes in a hash database, such as the NIST NSRL, Hash Keeper, and custom databases that have been created with the 'md5sum' tool. (Sleuth Kit Informer #6, Sleuth Kit Informer #7)
- Organize files based on their type (for example all executables, jpegs, and documents are separated). Pages of thumbnails can be made of graphic images for quick analysis. (Sleuth Kit Informer #3, #4, #5)

The Sleuth Kit is written in C and Perl and uses some code and design from The Coroner's Toolkit (TCT). The Sleuth Kit has been tested on:

- Linux
- Mac OS X
- Windows (Visual Studio and mingw)
- CYGWIN
- Open & FreeBSD
- Solaris

TSK Layers

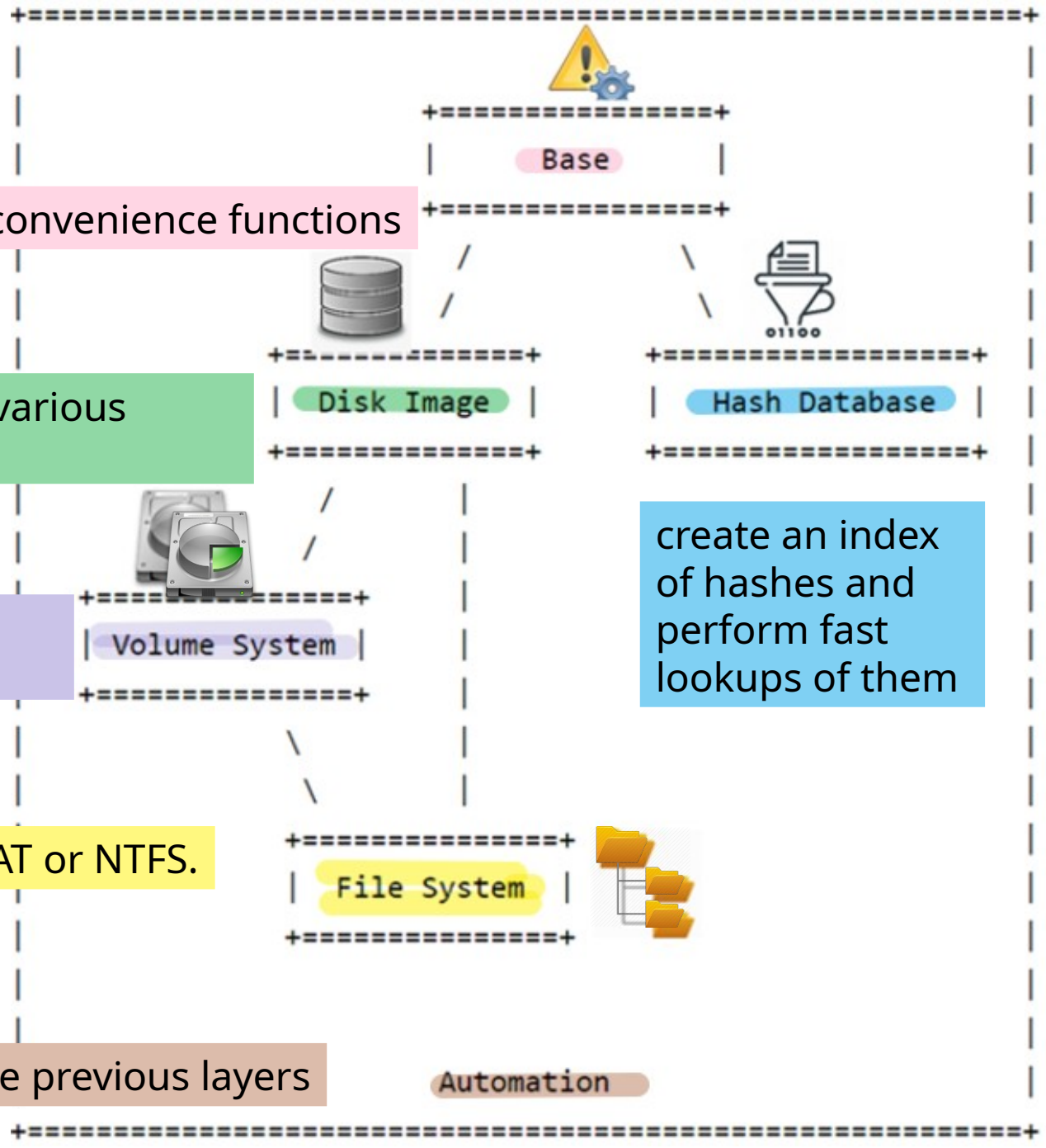
error handling, types, and convenience functions

Can open and process disk images in various formats

Processing data as a volume system, e.g., DOS partition tables

processing data as a file system, such as FAT or NTFS.

integrates all of the previous layers



create an index of hashes and perform fast lookups of them

Storage management vs File System management



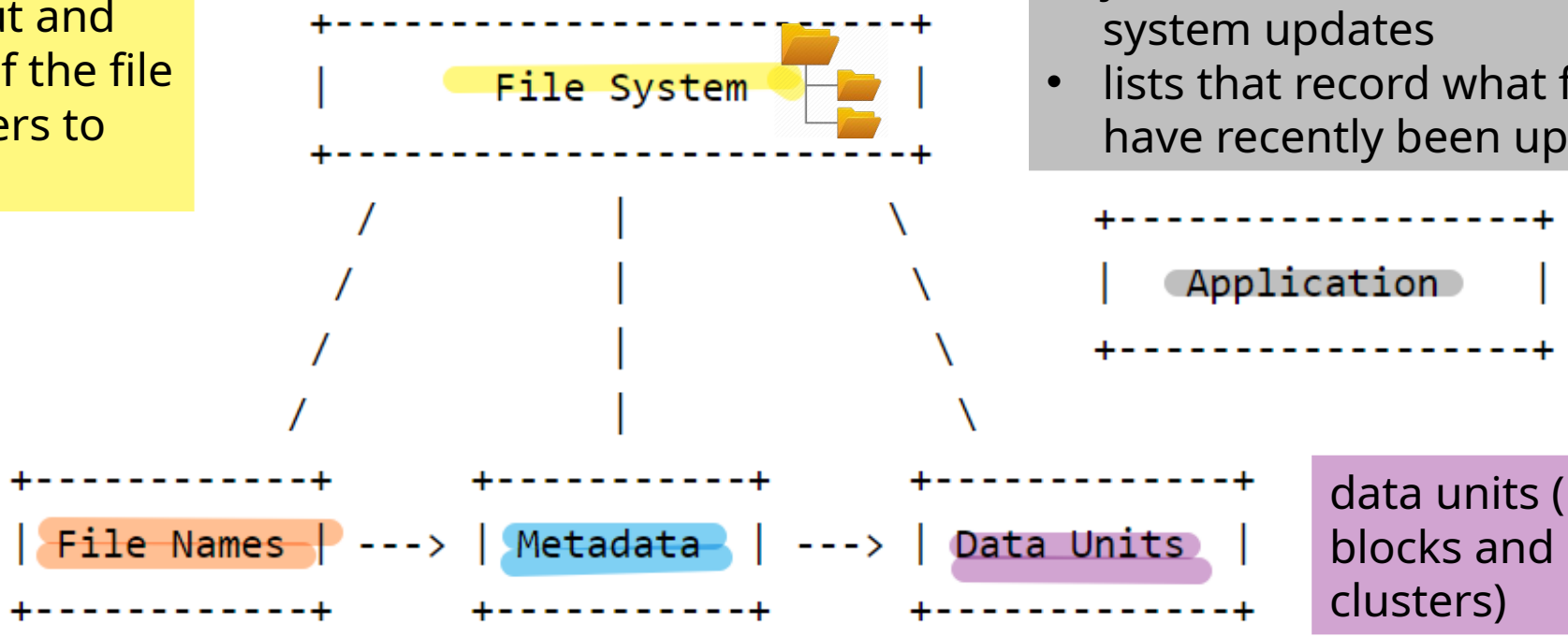
Customer: Frank
in date: 10/01/2021
out date: 11/01/2021
occupied space: 3
location: B3, B4, B5



https://www.geico.com/living/home/home-protection/storage-units/

File System Layers

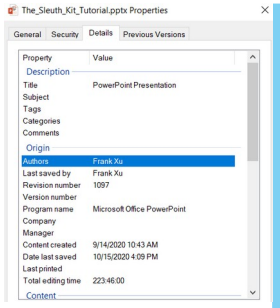
Describe the layout and general features of the file system. Allow others to read, write



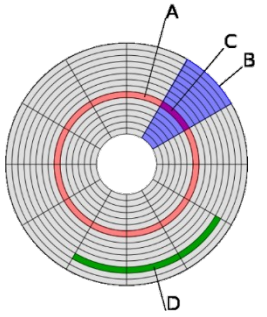
non-essential file system data
- journals that record file system updates
- lists that record what files have recently been updated

File and directory names (FAT file system= FileName+ Metadata)

data units (i.e. blocks and clusters)



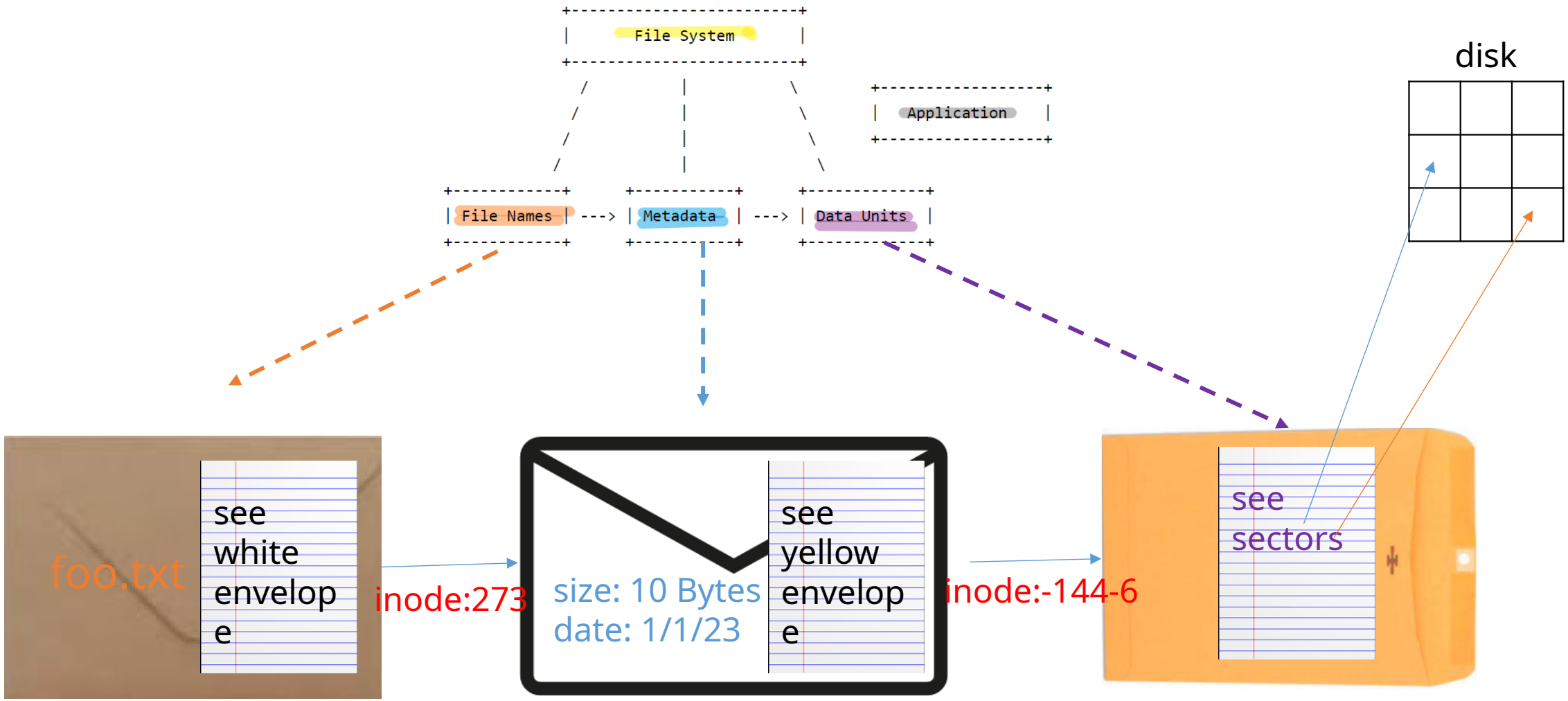
files and directories
- inode structures in UNIX
- MFT entries in NTFS,
- directory entry structures in FAT



Hard Drive Structure:

A = track
B = sector
C = sector of a track
D = cluster

UNIVERSITY OF BALTIMORE
TOWSON UNIVERSITY
BJA




```
root@kali:~/lab# fls -o 206848 cfreds_2015_data_leakage_pc.dd
d/d 273-144-6:  Program Files (x86)
```

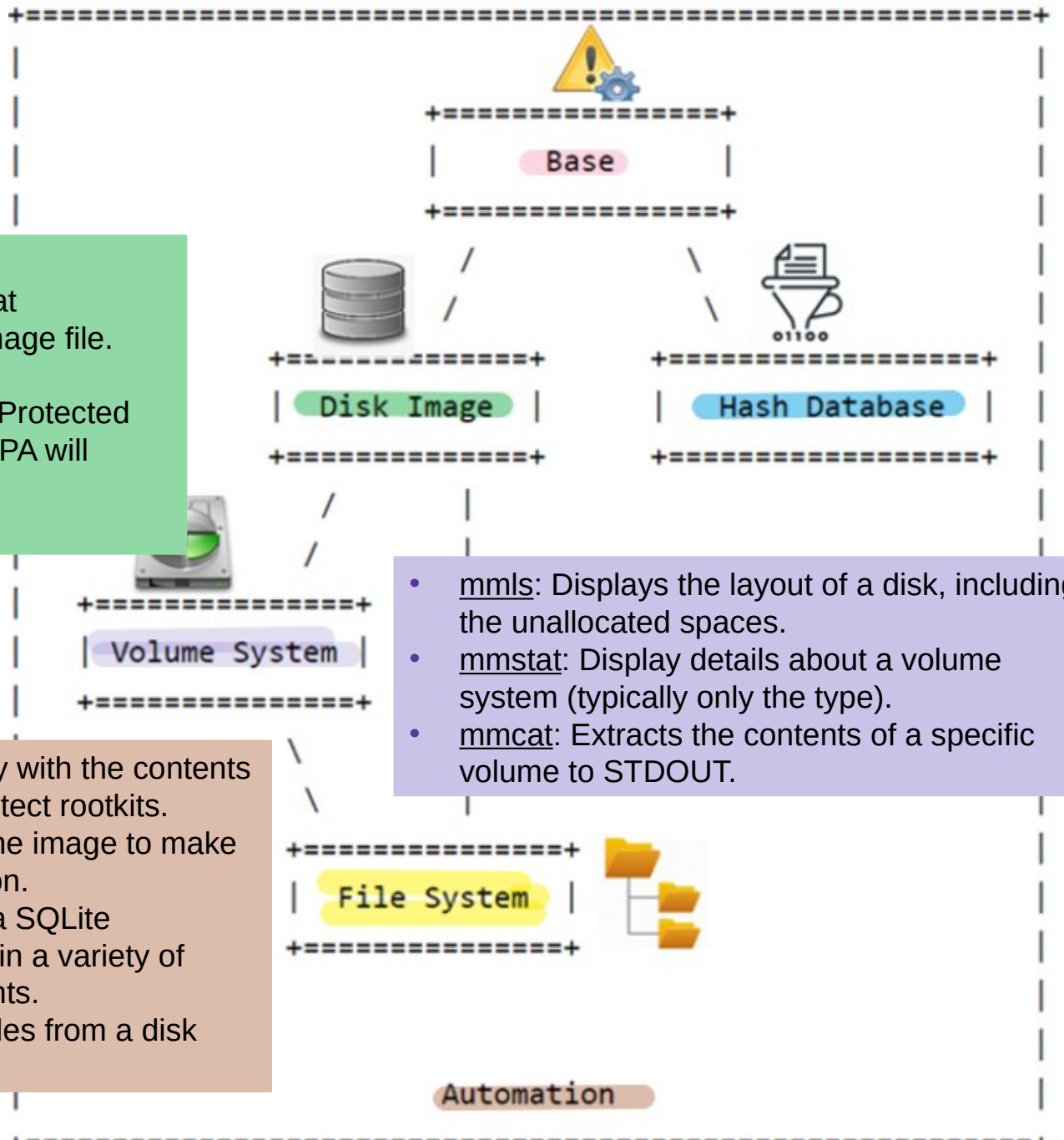
Implication: We can directly read metadata and content without using file systems

https://github.com/sleuthkit/sleuthkit/wiki/The_Sleuth_Kit_commands

TSK Tools

Image Files
- img_stat: tool will show the details of the image format
- img_cat: This tool will show the raw contents of an image file.

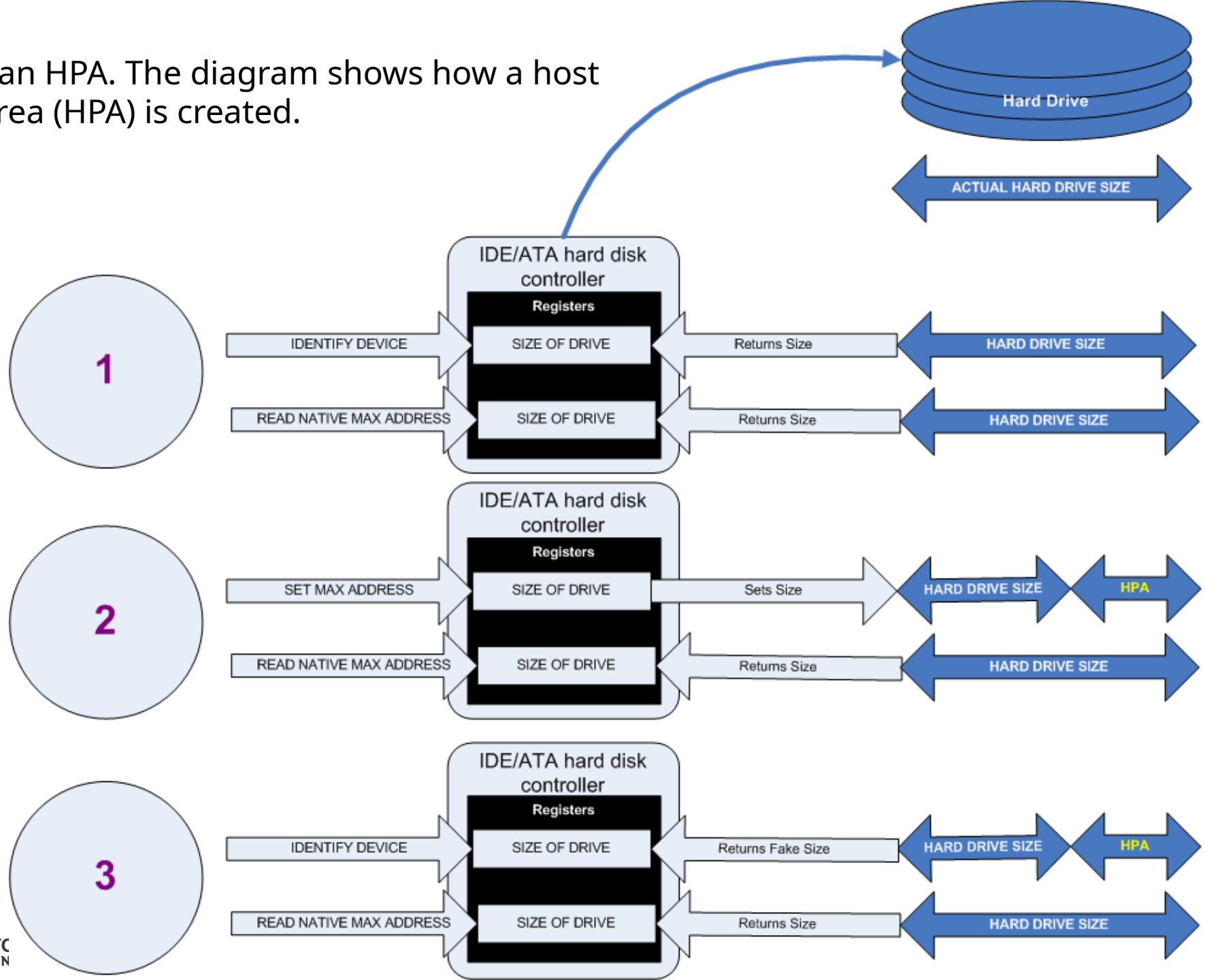
Disk Tools
- disk_sreset: This tool will temporarily remove a Host Protected Area (HPA) if one exists. After the disk is reset, the HPA will return.
- disk_stat: This tool will show if an HPA exists.



- mmls: Displays the layout of a disk, including the unallocated spaces.
- mmstat: Display details about a volume system (typically only the type).
- mmcat: Extracts the contents of a specific volume to STDOUT.

- tsk_comparedir: Compares a local directory hierarchy with the contents of raw device (or disk image). This can be used to detect rootkits.
- tsk_gettimes: Extracts all of the temporal data from the image to make a timeline. Equivalent to running fls with the '-m' option.
- tsk_loaddb: Loads the metadata from an image into a SQLite database. This allows other tools to be easily written in a variety of languages and give them access to the image contents.
- tsk_recover: Extracts the unallocated (or allocated) files from a disk image to a local directory.

Creation of an HPA. The diagram shows how a host protected area (HPA) is created.

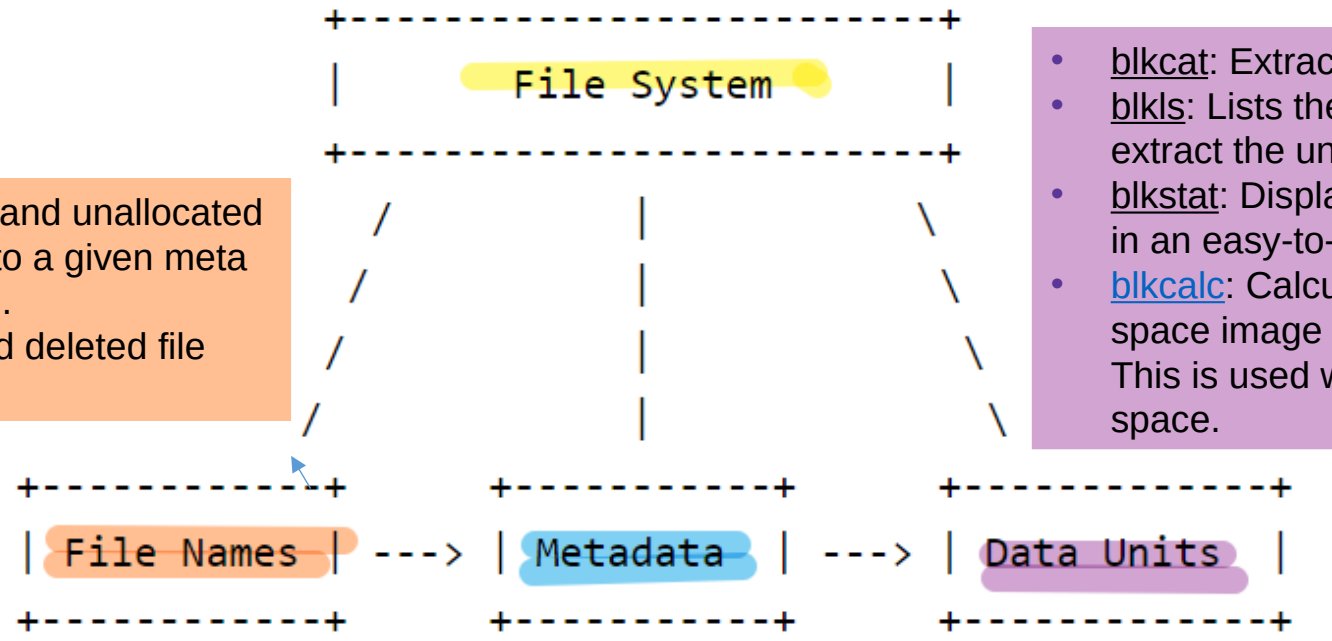


HPA Usages

- firmware company
  - BIOS
- Computer manufacturers
  - Preloaded OS
  - Booting and diagnostic utilities
  - Dell notebooks hide Dell MediaDirect utility in HPA.
- Hacker
  - Some rootkits hide in the HPA to avoid being detected by anti-rootkit and antivirus software
- Government
  - NSA for application persistence.

File System Tools

fsstat: Shows file system details and statistics including layout, sizes, and labels.



- ffind: Finds allocated and unallocated file names that point to a given meta data structure (inode).
- fls: Lists allocated and deleted file names in a directory.

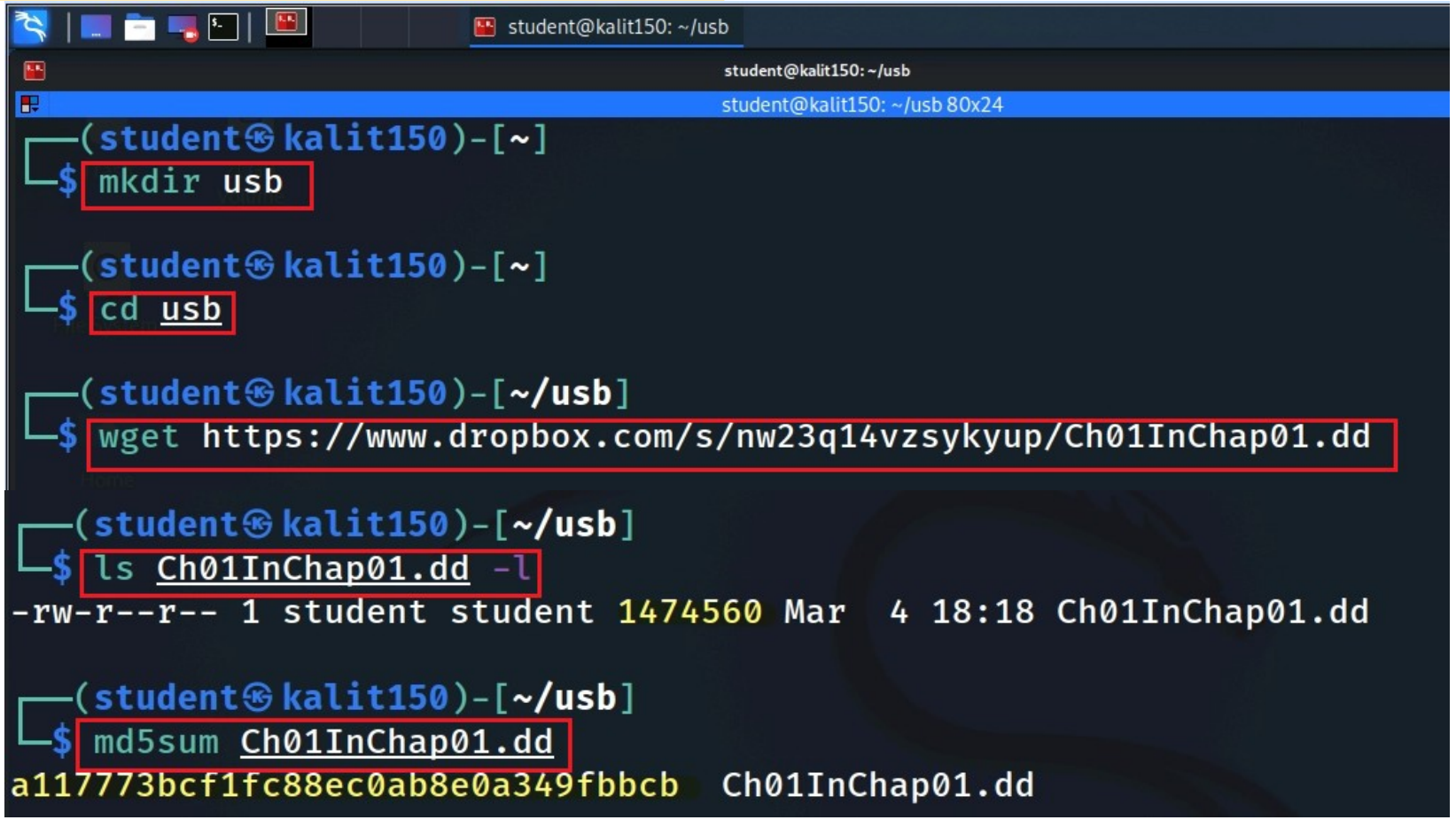
- blkcat: Extracts the contents of a given data unit.
- blkls: Lists the details about data units and can extract the unallocated space of the file system.
- blkstat: Displays the statistics about a given data unit in an easy-to-read format.
- blkcalc: Calculates where data in the unallocated space image (from blkls) exists in the original image. This is used when evidence is found in unallocated space.

- icat: Extracts the data units of a file, which is specified by its meta data address (instead of the file name).
- ifind: Finds the meta data structure that has a given file name pointing to it or the meta data structure that points to a given data unit.
- ils: Lists the meta data structures and their contents in a pipe delimited format.
- istat: Displays the statistics and details about a given meta data structure in an easy-to-read format.

Exam files in a disk image using TSK

a deleted file letter1.txt

Download a USB (Floppy Disk) image



Disk Image Level

FAT VBR (Volume Boot Record) and NTFS MBR (Master Boot Record)

tool will show the details of the image format

```
student@tsurugi01:~/usb$ img_stat Ch01InChap01.dd
IMAGE FILE INFORMATION
--------------------------------------------
Image Type: raw

Size in bytes: 1474560
Sector size:  512
student@tsurugi01:~/usb$
```

```
student@tsurugi01:~/usb$ echo $((0x0b40*512))
1474560
```

reads the first 512 bytes (one sector) of your disk image and prints them as hex

```
00000000: eb34 902c 276b 306e 4948 4300 0201 0100  .4.,'k0nIHC.....
00000010: 02e0 0040 0bf0 0900 1200 0200 0000 0000  ...@............
00000020: 0000 0000 0000 0000 0000 0000 0000 0012  ................
00000030: 0000 0000 0100 fa33 c08e d0bc 007c 1607  .......3.....|..
```



[Ch01InChap01.dd]

FAT VBR (Volume Boot Record)

Windows Image (https://cfreds-archive.nist.gov/data_leakage_case/data-leakage-case.html)

Show the sector 0 of NTFS MBR (Master Boot Record)

```
student@tsurugi01:~/Lab$ img_cat cfreds_2015_data_leakage_pc.dd | head -c 512 | xxd
00000000: 33c0 8ed0 bc00 7c8e c08e d8be 007c bf00  3.....|......|..
00000010: 06b9 0002 fcf3 a450 681c 06cb fbb9 0400  .......Ph.......
00000020: bdbe 0780 7e00 007c 0b0f 850e 0183 c510  ....~..|........
```

...

```
00000150: 10eb f2f4 ebfd 2bc9 e464 eb00 2402 e0f8  ......+..d..$...
00000160: 2402 c349 6e76 616c 6964 2070 6172 7469  $..Invalid parti
00000170: 7469 6f6e 2074 6162 6c65 0045 7272 6f72  tion table.Error
00000180: 206c 6f61 6469 6e67 206f 7065 7261 7469   loading operati
00000190: 6e67 2073 7973 7465 6d00 4d69 7373 696e  ng system.Missin
000001a0: 6720 6f70 6572 6174 696e 6720 7379 7374  g operating syst
000001b0: 656d 0000 0063 7b9a 2057 26f0 0000 8020  em...c{. W&.... 
000001c0: 2100 07df 130c 0008 0000 0020 0300 00df  !.......... ....
000001d0: 140c 07fe ffff 0028 0300 00d0 7c02 0000  .......(....|...
000001e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001f0: 0000 0000 0000 0000 0000 0000 0000 55aa  ..............U.
student@tsurugi01:~/Lab$
```

MBR



Windows Image (https://cfreds-archive.nist.gov/data_leakage_case/data-leakage-case.html)

Volume system tools (partition level)

partition layout (volume

```
student@tsurugi01:~/usb$ mmls Ch01InChap01.dd
student@tsurugi01:~/usb$ mmstat Ch01InChap01.dd
student@tsurugi01:~/usb$
```

Windows Image (https://cfreds-archive.nist.gov/data_leakage_case/data-leakage-case.html)

If the image contains a partition

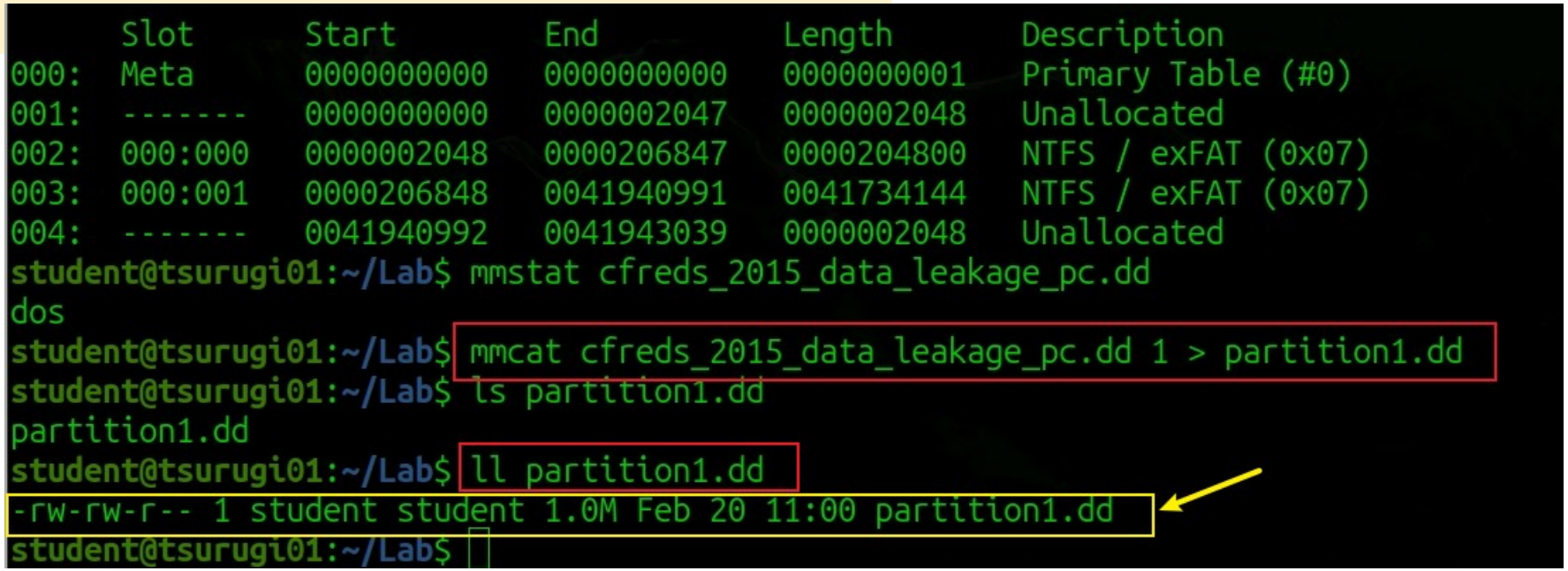
```
student@tsurugi01:~/Lab$ ls
android10  android14  cfreds_2015_data_leakage_pc.dd
student@tsurugi01:~/Lab$ mmls cfreds_2015_data_leakage_pc.dd
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

     Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000002047   0000002048   Unallocated
002:  000:000   0000002048   0000206847   0000204800   NTFS / exFAT (0x07)
003:  000:001   0000206848   0041940991   0041734144   NTFS / exFAT (0x07)
004:  -------   0041940992   0041943039   0000002048   Unallocated
student@tsurugi01:~/Lab$
```

```
student@tsurugi01:~/Lab$ mmstat cfreds_2015_data_leakage_pc.dd
dos
```

MBR Partition

Extracts a specific partition (volume) to



Filesystem tools (inside partition)

Show file system details and statistics including layout, sizes, and labels.

```
student@tsurugi01:~/usb$ fsstat Ch01InChap01.dd
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: FAT12

OEM Name: ,'k0nIHC
Volume ID: 0x0
Volume Label (Boot Sector):
Volume Label (Root Directory):
File System Type Label: ◆3◆◆M

Sectors before file system: 0
```

NTFS File System

```
      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000002047   0000002048   Unallocated
002:  000:000   0000002048   0000206847   0000204800   NTFS / exFAT (0x07)
003:  000:001   0000206848   0041940991   0041734144   NTFS / exFAT (0x07)
004:  -------   0041940992   0041943039   0000002048   Unallocated
student@tsurugi01:~/Lab$ fsstat -o 206848 cfreds_2015_data_leakage_pc.dd
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: NTFS
Volume Serial Number: C8CA0C8DCA0C7A48
OEM Name: NTFS
Version: Windows XP

METADATA INFORMATION
```

lists directory contents in a filesystem inside a disk image

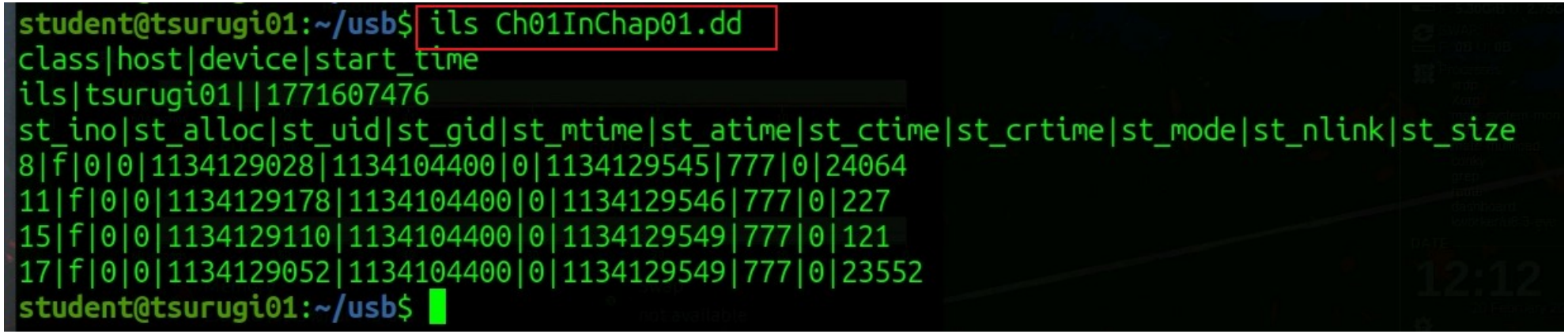
```
student@tsurugi01:~/usb$ fls Ch01InChap01.dd
r/r 5:  Client Info.mdb
r/r * 8:        Billing Letter.doc
r/r * 11:       confirmation.txt
r/r 13: Income.xls
r/r * 15:       letter1.txt
r/r * 17:       Regrets.doc
v/v 45779:      $MBR
v/v 45780:      $FAT1
v/v 45781:      $FAT2
V/V 45782:      $OrphanFiles
student@tsurugi01:~/usb$
```

What is the difference?

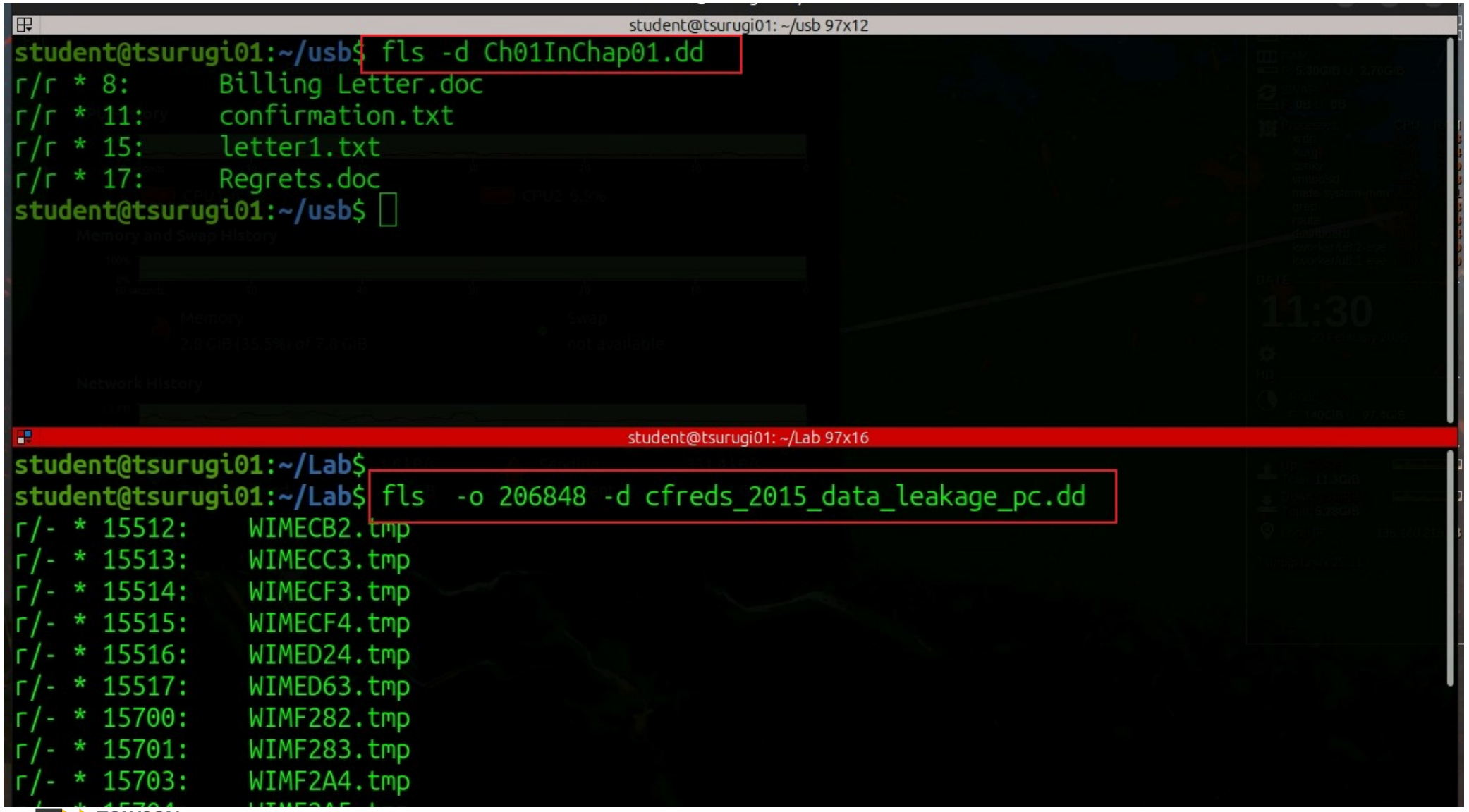
student@tsurugi01: ~/Lab 97x16

```
student@tsurugi01:~/Lab$ fls  -o 206848 cfreds_2015_data_leakage_pc.dd
d/d 486-144-5:  Users
d/d 13797-144-1:        Documents and Settings
d/d 389-144-6:  ProgramData
d/d 273-144-6:  Program Files (x86)
r/r 4-128-4:    $AttrDef
r/r 8-128-2:    $BadClus
r/r 8-128-1:    $BadClus:$Bad
r/r 6-128-4:    $Bitmap
r/r 7-128-1:    $Boot
d/d 11-144-4:   $Extend
r/r 2-128-1:    $LogFile
r/r 0-128-1:    $MFT
```

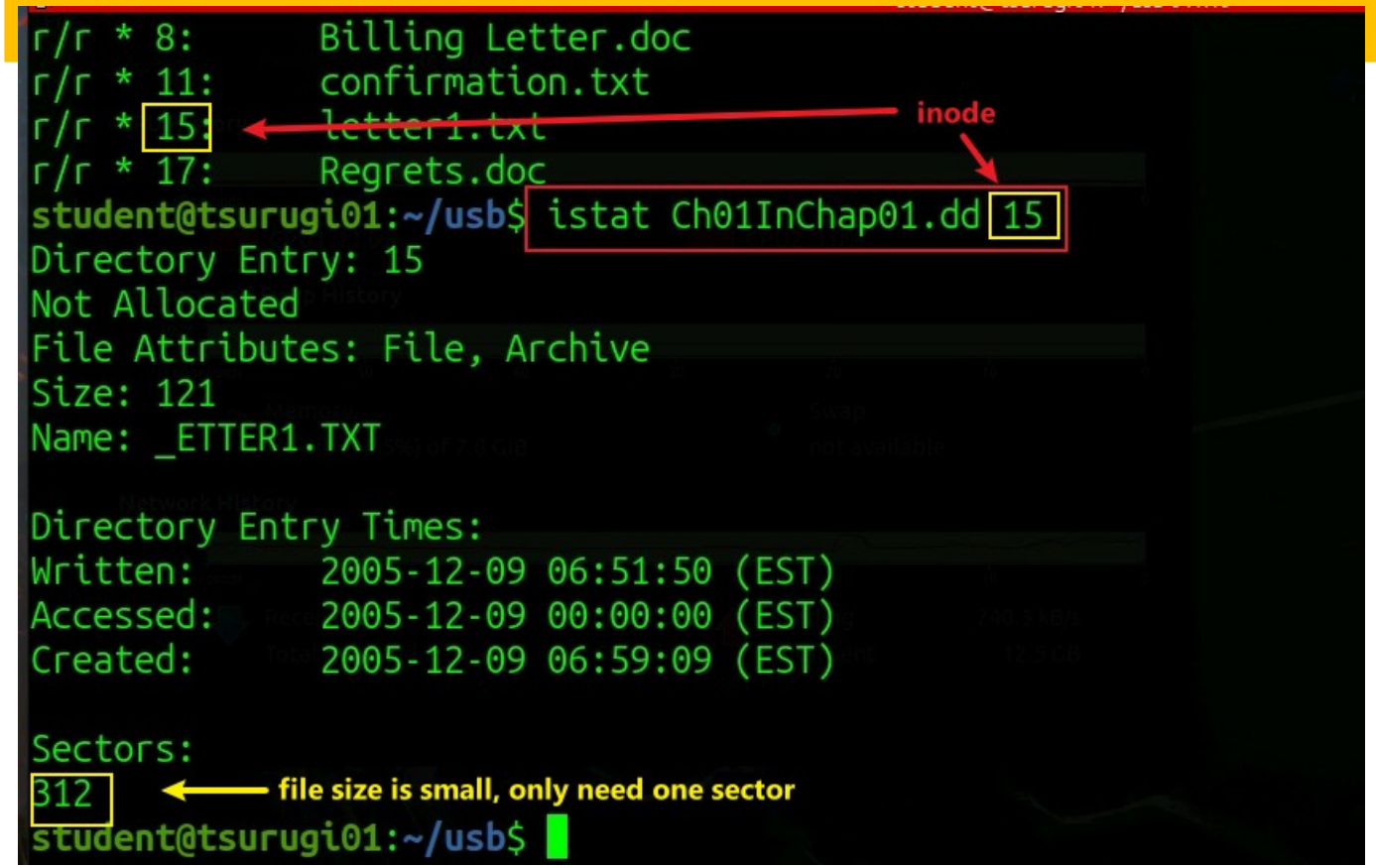
Lists many metadata entries in a raw, pipe delimited table so you can scan or script it.



-d Display deleted entries only



Display the statistics and details about a given meta data

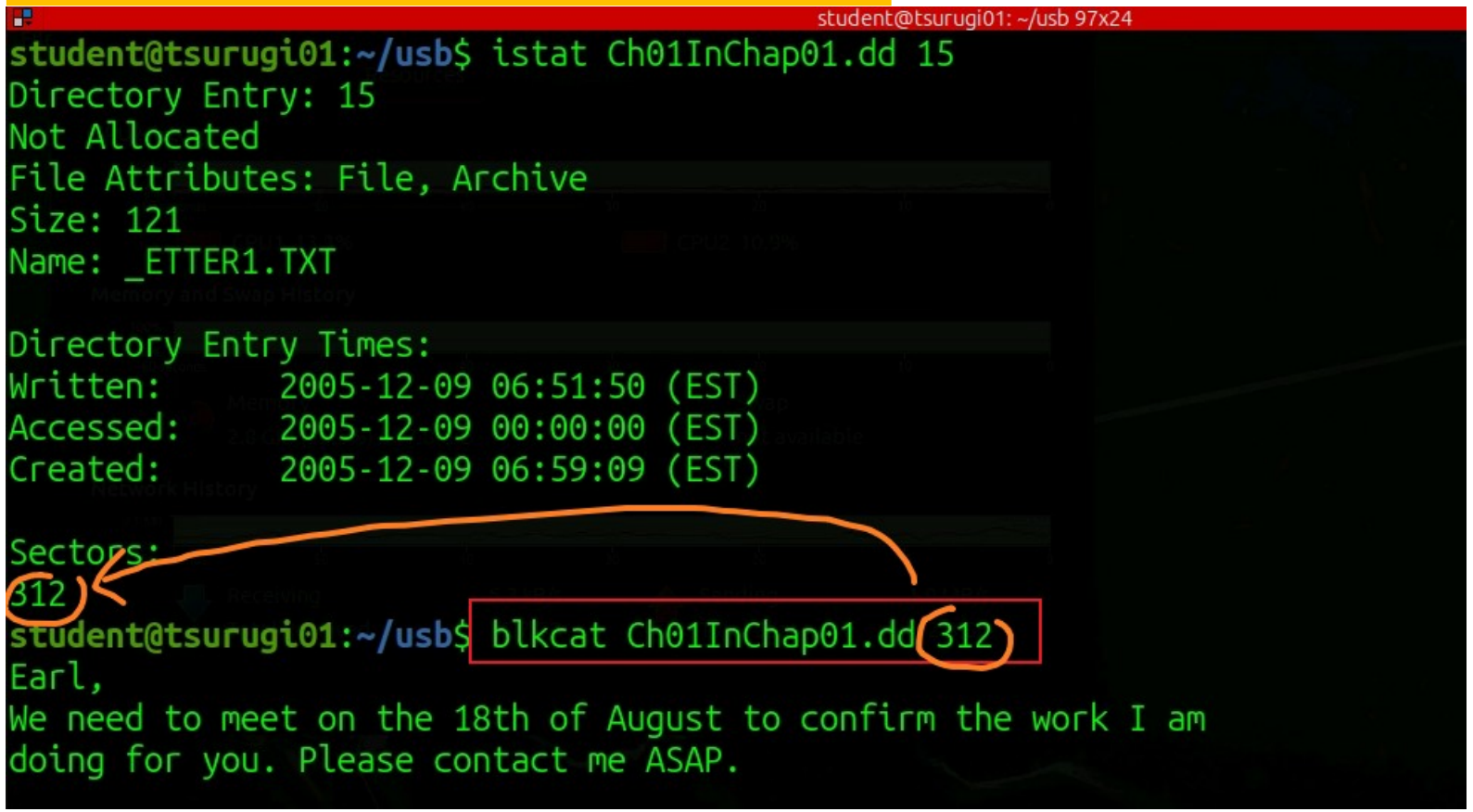


Recover deleted files using *inode* without specifying a file name

```
student@tsurugi01:~/usb$ icat Ch01InChap01.dd 15 > letter1.txt
student@tsurugi01:~/usb$ cat letter1.txt
Earl,
We need to meet on the 18th of August to confirm the work I am
doing for you. Please contact me ASAP.
```

Data Unit

Extract the contents of a given data unit.



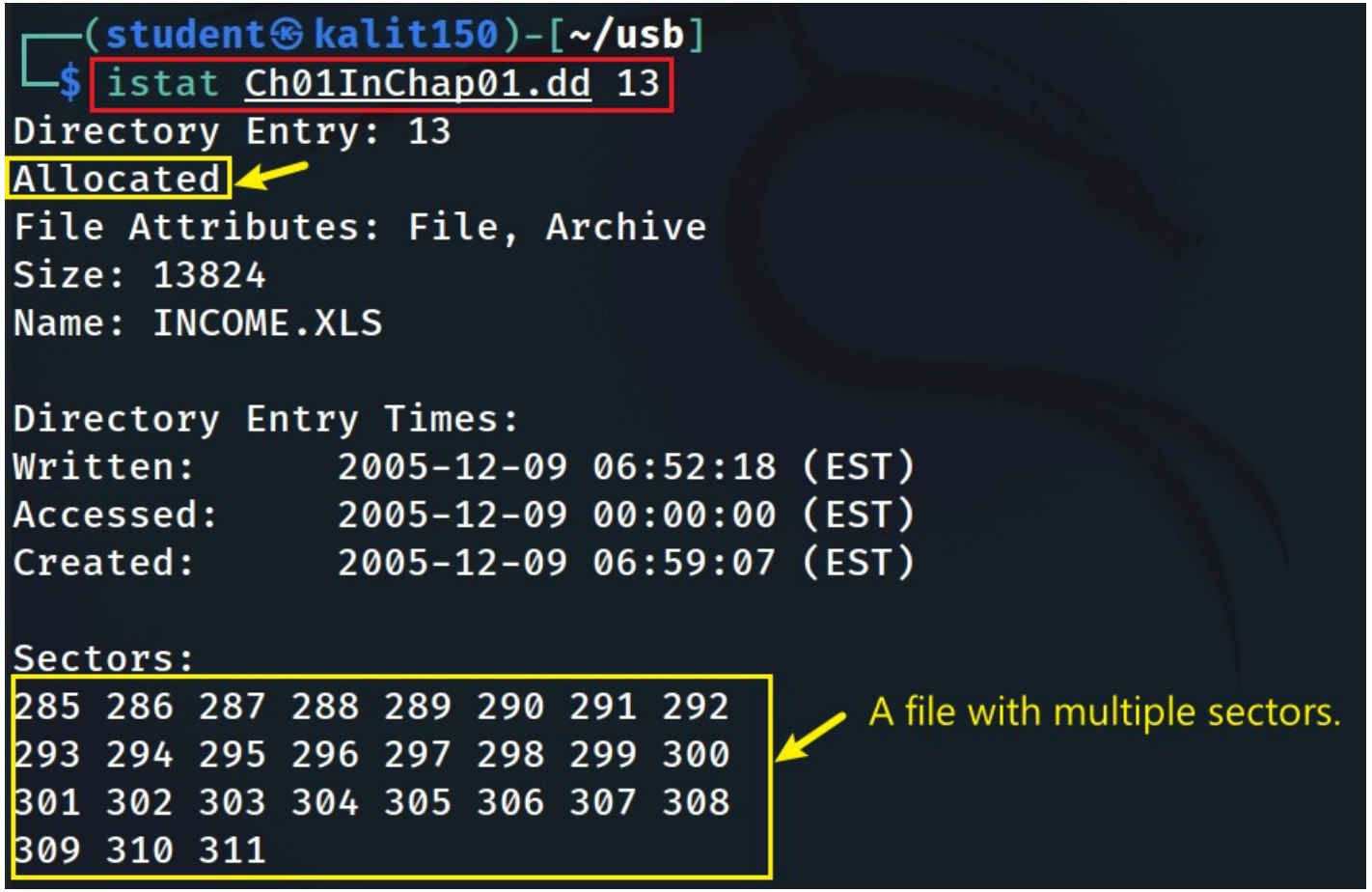
Extract the contents of a given data unit in Hex

What is the size of the sector?

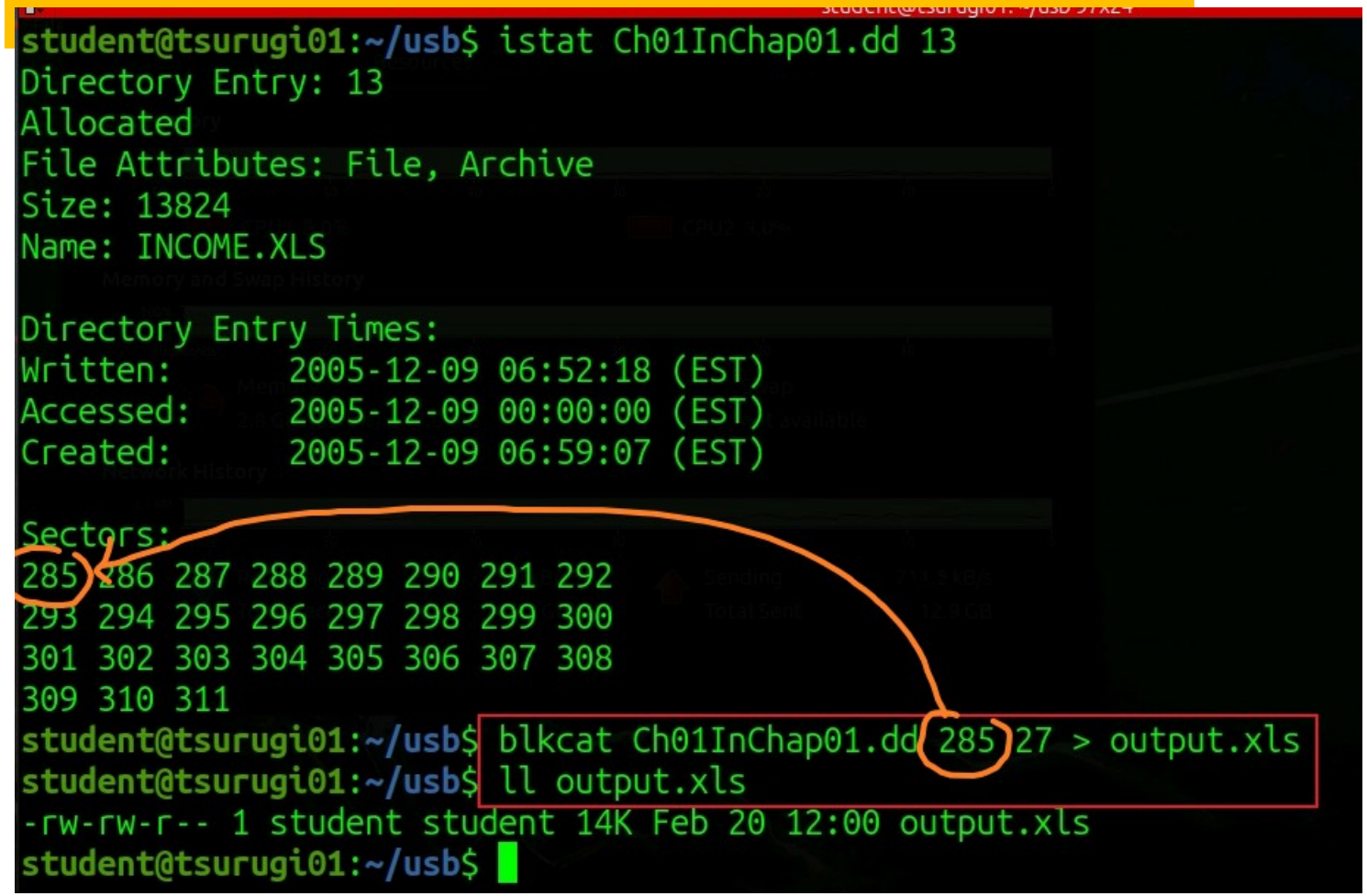
```
┌──(student㉿kalit150)-[~/usb]
└─$ blkcat  Ch01InChap01.dd 312 | xxd
00000000: 4561 726c 2c20 0d0a 5765 206e 6565 6420  Earl, ..We need
00000010: 746f 206d 6565 7420 6f6e 2074 6865 2031  to meet on the 1
00000020: 3874 6820 6f66 2041 7567 7573 7420 746f  8th of August to
00000030: 2063 6f6e 6669 726d 2074 6865 2077 6f72   confirm the wor
00000040: 6b20 4920 616d 200d 0a64 6f69 6e67 2066  k I am ..doing f
00000050: 6f72 2079 6f75 2e20 506c 6561 7365 2063  or you. Please c
00000060: 6f6e 7461 6374 206d 6520 4153 4150 2e0d  ontact me ASAP..
00000070: 0a0d 0a47 656f 7267 6500 0000 0000 0000  ...George.......
00000080: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000090: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000a0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000b0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000c0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000d0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000100: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000110: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000120: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000130: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000140: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000150: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000160: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000170: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000180: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000190: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001a0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001b0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001c0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001d0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

echo "$((16#1ff))"

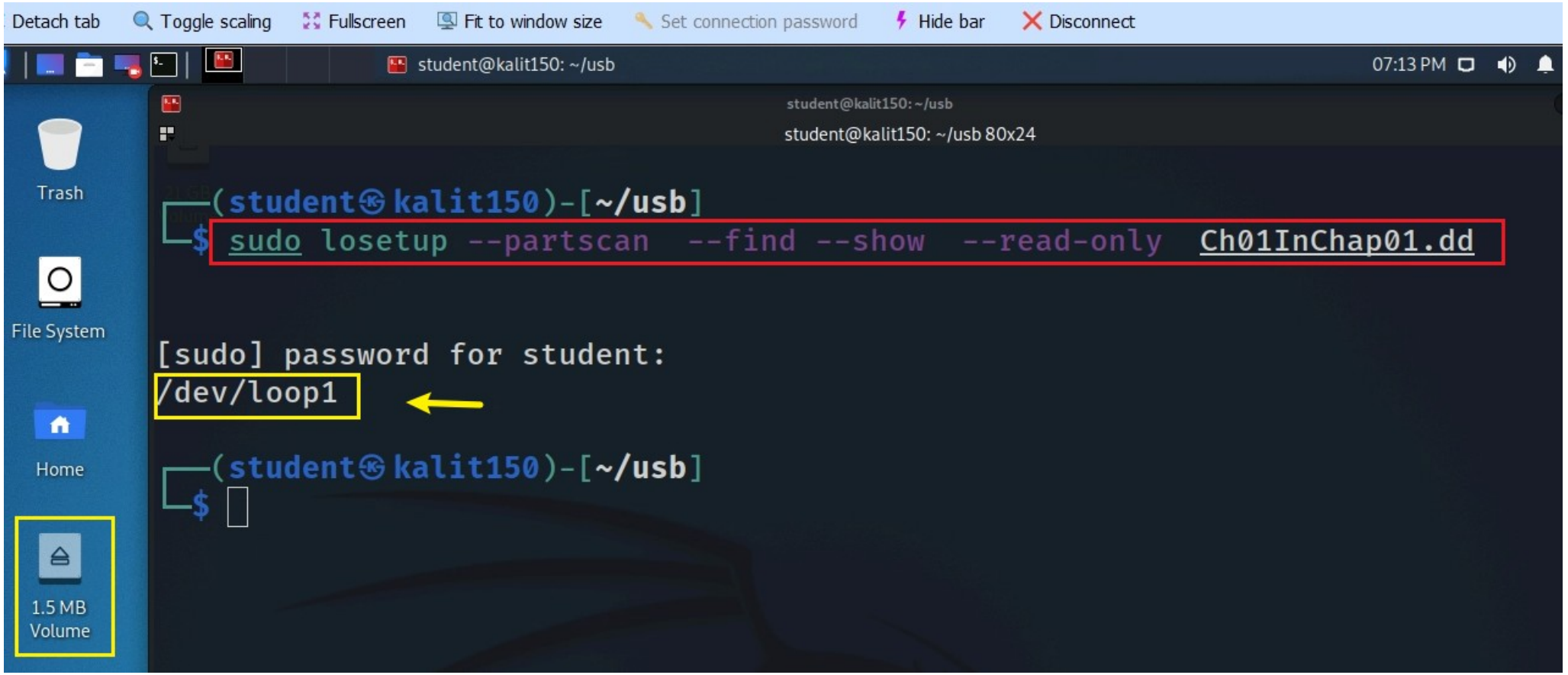
Show A file with multiple sectors



blkcat will read and output 27 consecutive blocks starting from block

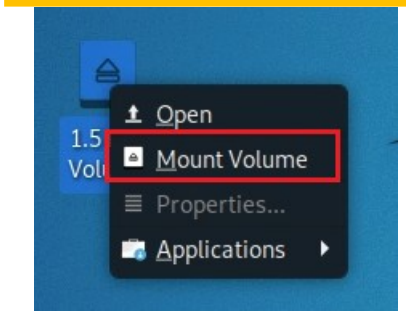


Mounting a USB image

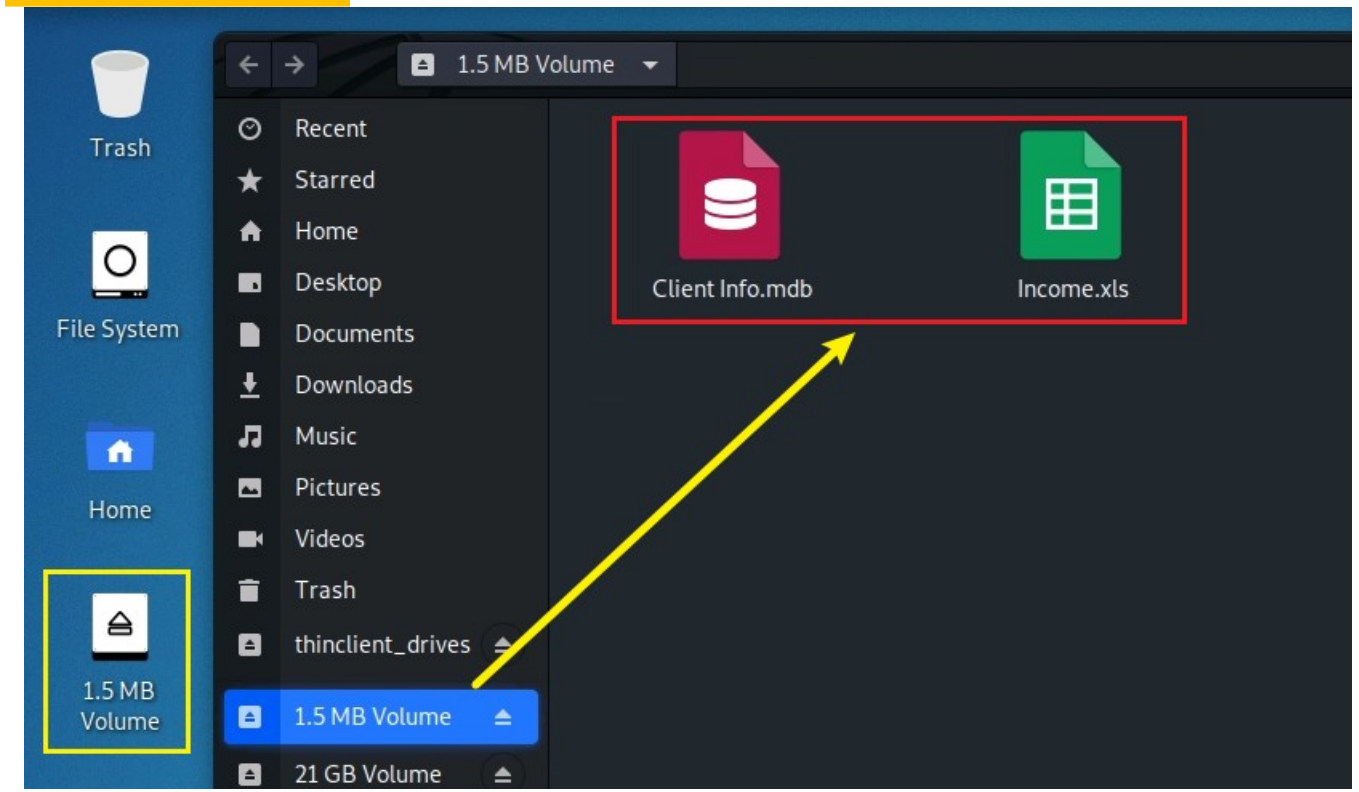


-**partscan**: This option tells losetup to scan the partition table in the specified disk image (Ch01InChap01.dd) and automatically create loop devices for each partition found. This is useful when dealing with disk images that contain multiple partitions.
--**find**: This option instructs losetup to find the first available loop device. When combined with --show, it ensures that the command displays the path of the loop device that is being used.

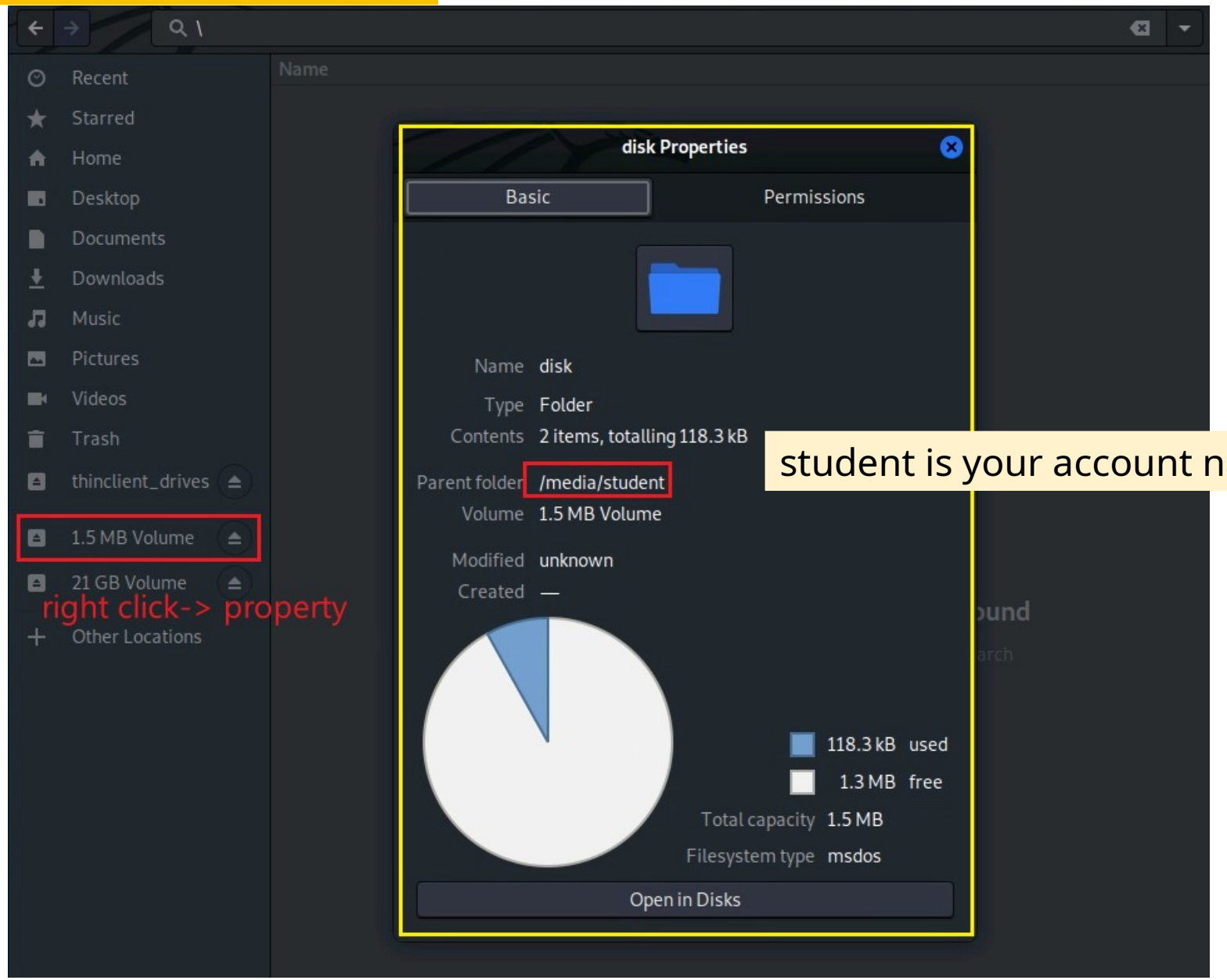
Mount the USB image



Show content



Show the mounted path



right click-> property

student is your account name

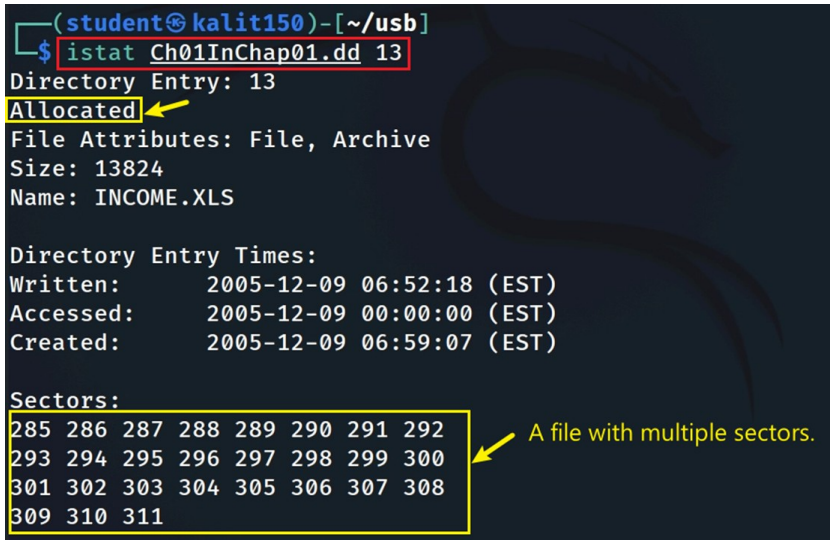
Show the disk content

```
┌──(student㉿kalit150)-[~/usb]
└─$ cd /media/student/disk

┌──(student㉿kalit150)-[/media/student/disk]
└─$ ls -l
total 116
-rw-r--r-- 1 student student 104448 Dec  9  2005 'Client Info.mdb'
-rw-r--r-- 1 student student  13824 Dec  9  2005  Income.xls
```

Assignment (part 1)

1. How many deleted files are in the disk image Ch01InChap01.dd?
   - Show evidence
2. Retrieve `INCOME.XLS` from dd
   - Using `icat`
   - Using `blkcat`
   - Show `.xls` content
   - Verify that recovered files are the same



Assignment (part 2) : Try one new command

Pick up any new TSK command we have not covered in the class

- What are the inputs?
- What are the outputs?
- Explain each command.
- Capture all screenshots.
- If you use your inputs, upload inputs