Digital Forensics
Lecture Week 9a

Disk data

Readings

Objectives

- To classify Disks
- To understand Partitions
- To understand the Boot Process

Disk Drives

- Hard Disk
  - High capacity at a low cost
- USB Flash Drives
  - portable between all Operating systems
- Solid State Drives SSDs
  - No moving parts

Disk Blocks

- The disk is formatted into blocks
  - default is 512 bytes
- The disk file system sees these blocks as sectors
  - default is also 512 bytes
- The file system counts these blocks using a sequential system (Logical block addressing LBA)
- The file system allocates clusters of sectors to a file or other disk object – default size is 4096 byte
- The clusters are allocated by finding unused or deleted blocks
- File Table pointers keep track of the file segments

Formatting

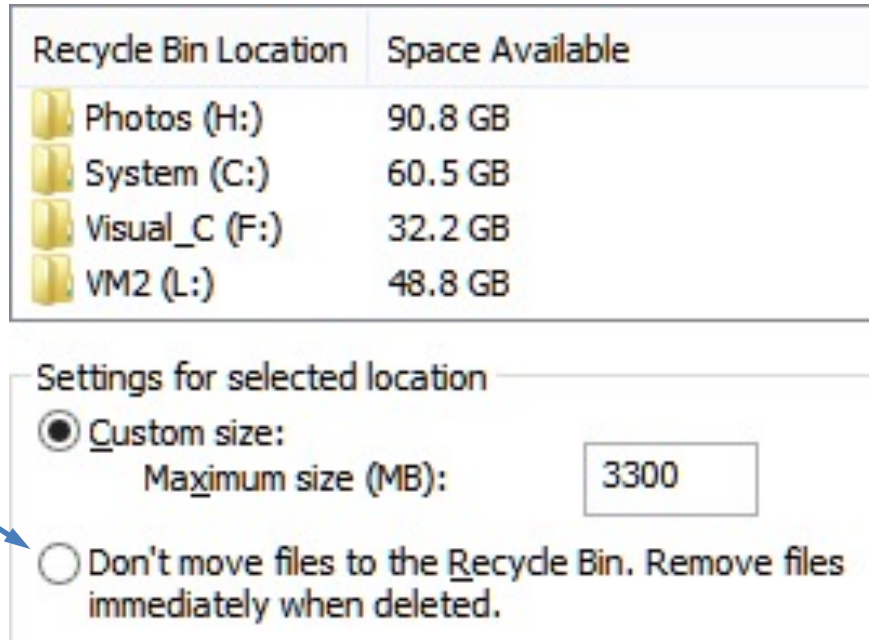
- Low level formatting
  - place disk sectors on the disk
  - done at the disk factory
- Partitioning
  - breaks the disk into sections
  - place data structures on the disk
- High level formatting
  - adds file structures to the partition
  - operating system dependant

The Recycle Bin

- When a file is deleted, it is moved to the Recycle Bin on fixed drives (not USB)
- The recycle bin is great for forensic evidence
- There is a recycle bin for each Drive Letter
- There is a recycle bin for each user
- You can delete files in the recycle bin

Deleting Files

- The suspect may bypass the recycle bin



- We can recover deleted files with TSK tools
  - details in the Lab
- Over time, parts of a a deleted file can be overwritten by new files

Erasing Files

- A high level format or a repartition will NOT erase data
  - it only removes the data pointers
- A low level format will usually erase the data
  - writing zeros may not destroy all previous data
  - specific bit patterns are more effective 01010101
  - some secure systems write random data
  - writing several times improves erasure

File Carving

- Unallocated disk space may contain fragments from previous files
- However the links to the parts of the file are lost
- We need to reassemble the file by hand (or with a tool)
- We start by searching for file headers
  - see file metadata week 4

USB Flash Drives

- Removable Read / Write Storage
- Available capacity 8 – 132 GB
- Low cost, small size, reasonable reliability
- Replace RW DVD
- Power is drawn from the host device
- Serial interface, USB

USB Flash Drives #2

- On chip error checking and wear leveling
  - limited number of erases
  - similar to SSDs
- Typical USB 3.1 rates
  - 700MB/s for sequential reads

- http://en.wikipedia.org/wiki/Universal_Serial_Bus

DRAM

- Dynamic Random Access Memory (DRAM), which is the 'working memory' of computers, as well as the long-term memory in flash drives.
- While writing data to DRAM is fast and low-energy, the data is volatile and must be continuously 'refreshed' to avoid it being lost: this is clearly inconvenient and inefficient. Flash stores data robustly, but writing and erasing is slow, energy intensive and deteriorates it, making it unsuitable for working memory.

SSDs

- Solid State Drives (SSDs) are replacing HDDs
- No moving parts, smaller, lighter, quieter
- Small form factors such as M.2
- Cost more - $200 for 256 GB
- Uses Flash NAND chips
- Faster reads but have trouble writing
- Triple Level Cache (TLC) allows higher density
- Limited erase cycle life so need wear levelling

- http://en.wikipedia.org/wiki/Solid-state_drive

SSD TRIM

- Deletion is handled by the SSD controller, not the OS.
- When the file system wants to delete a file, a TRIM signal is sent to the SSD controller.
- If power is removed, deletion will continue when SSD power is restored, even if removed from the laptop.
- A read after TRIM can be set to return data (DRAT).
- A read after TRIM can be set to return zeros (RZAT).
  - the data may still be on disk

Trim Check

```
TRIM check v0.7 - Written by Vladimir Panteleev
https://github.com/CyberShadow/trimcheck

Loading continuation data from C:\Forensics_Graham\trimcheck-cont.json...
  Drive path   :  \\.\C:
  Offset       :  54400016384
  Random data  :  F0 EA D8 F2 2A 14 1F 63 AD DA 08 71 0E E3 A0 7E...

Reading raw volume data...
  Opening \\.\C:...
  Seeking to position 54400016384...
  Reading 16384 bytes...
  First 16 bytes: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00...
Data is empty (filled with 0x00 bytes).

CONCLUSION: TRIM appears to be WORKING!

Press Enter to exit...
```

SSD Forensics

- Clearing unallocated blocks is slow
- The SSD controller performs random garbage collection independent of the file system
  - even when disconnected from the PC
- Wear levelling means multiple file copies may exist
  - and their location is continually changing

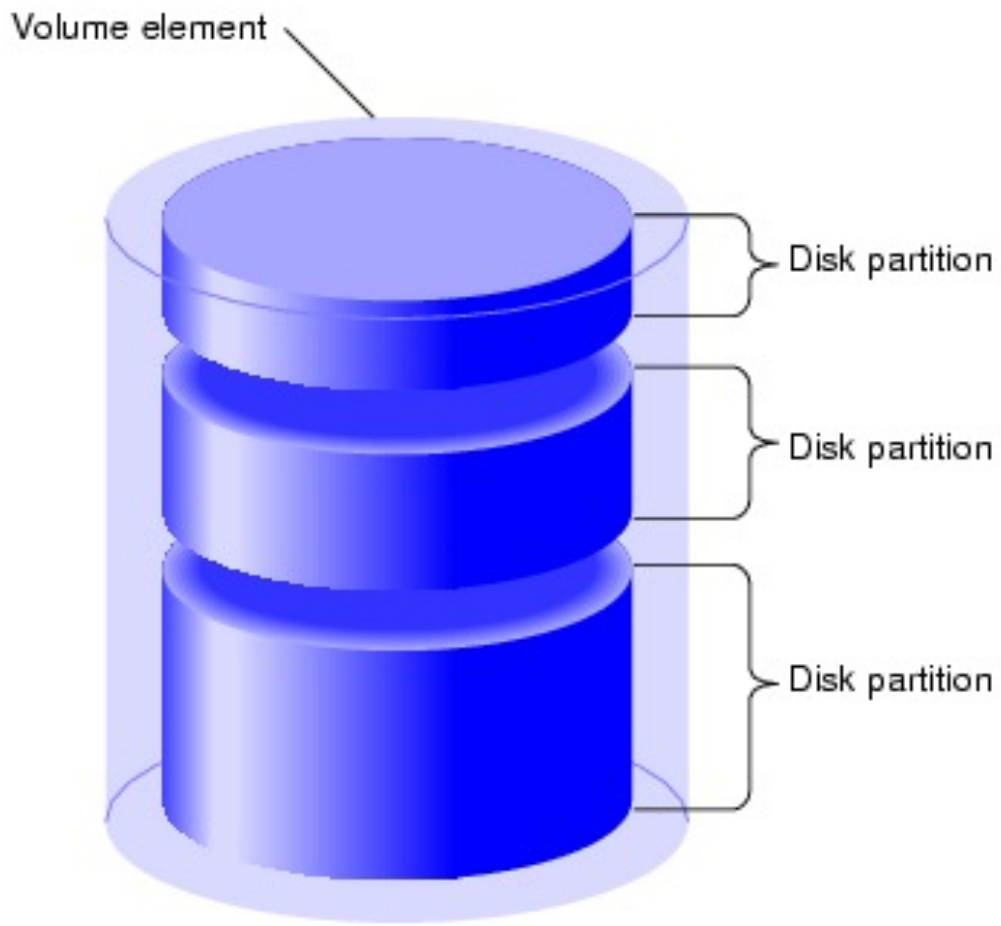
https://forensicswiki.xyz/wiki/index.php?title=Solid_State_Drive_(SSD)_Forensics

Objectives

- To classify Disks
- To understand Partitions
- To understand the Boot Process
- To understand data acquisition principles

Disk volumes and partitions

- Disks are split into disjoint (non overlapping) partitions.
- Each volume has its own file system



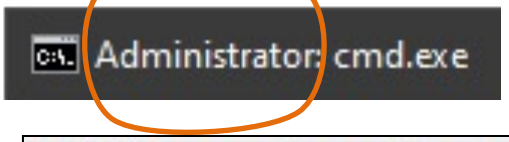
Partitioning schemes

- Partitions can be the older BIOS based
  - Master Boot Record (MBR)
- Each disk may be divided into partitions
- Up to four primary (bootable) partitions
- Typically, the first partition contains the OS
- Partitions can be the newer UEFI based, which is more secure and can use a GUID partition table (GPT)

Unified Extensible Firmware Interface UEFI

- Boots any OS (Windows or Linux)
- Uses a Boot Manager instead of the BIOS Boot Sector
- Can use an EFI system partition instead of the MBR
- CPU independent (Intel or AMD)
- Can load the OS over a network or from USB
- Supports large disks (over 2 TB)
- The OS can talk to the UEFI once loaded

How to detect your disk boot type




```
C:\Forensics_Graham>copy C:\Windows\Panther\setupact.log .
        1 file(s) copied.

C:\Forensics_Graham>find "Detected boot envir" /i setupact.log

---------- SETUPACT.LOG
2017-04-14 18:36:50, Info  IBS Callback_BootEnvironmentDetect:
Detected boot environment: BIOS
```

```
---------- SETUPACT.LOG
2018-09-08 14:02:50, Info IBS Callback_BootEnvironmentDetect:
Detected boot environment: EFI
```

Master Boot Record (MBR)

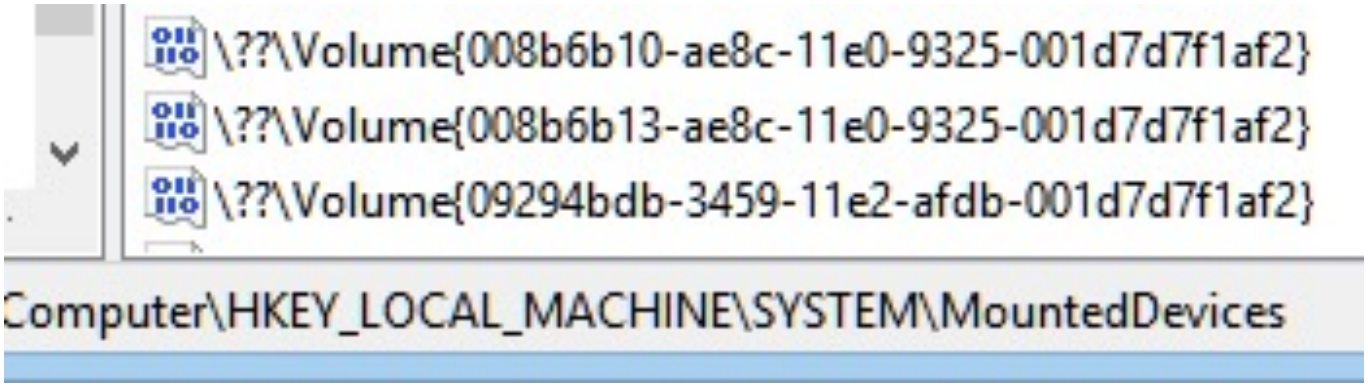
- BIOS style partitions use an MBR
- The first 512 byte sector of a disk is the MBR
- 446 bytes for a boot sector which boots to the OS in a partition. The BIOS boots to this sector
- 64 bytes for 4 partition tables
- 4 bytes (32 bits) for the disk signature

Structure of a classical generic MBR

Address	Description		Size (bytes)
0x0000 (0)	Bootstrap code area		446
0x01BE (446)	Partition entry №1	*Partition table* (for primary partitions)	16
0x01CE (462)	Partition entry №2		16
0x01DE (478)	Partition entry №3		16
0x01EE (494)	Partition entry №4		16
0x01FE (510)	0x55	*Boot signature*[a]	2
0x01FF (511)	0xAA		
Total size: 446 + 4×16 + 2			512

GUID partition tables (GPT)

- An alternate disk signature is a GUID as used by UEFI
  - Global Unique Identifier - a random hash
  - kept in the registry to map drive letters (C:)
  - HKEY_LOCAL_MACHINE\SYSTEM\MountedDevices\



gdisk (fdisk for GPT)

```
Command (? for help): ?
b       back up GPT data to a file
c       change a partition's name
d       delete a partition
i       show detailed information on a partition
l       list known partition types
n       add a new partition
o       create a new empty GUID partition table (GPT)
p       print the partition table
q       quit without saving changes
r       recovery and transformation options (experts
s       sort partitions
t       change a partition's type code
v       verify disk
w       write table to disk and exit
x       extra functionality (experts only)
?       print this menu
```

Viewing the GPT Disk

gdisk64.exe 0: ← 0 is the first disk

```
Command (? for help): p
Disk 0:: 500118192 sectors, 238.5 GiB
Sector size (logical): 512 bytes
Disk identifier (GUID): EC7E7C0B-56E5-4F2E-B37C-DFE192FCC523
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 500118158
Partitions will be aligned on 2048-sector boundaries
Total free space is 2055106 sectors (1003.5 MiB)

Number Start (sector) End (sector)  Size     Code Name
   1            2048       206847   100.0 MiB EF00 EFI system partition
   2          206848       239615   16.0 MiB  0C01 Microsoft reserved .
   3          239616    316718762   150.9 GiB 0700 Basic data partition
   4       316719104    424648703   51.5 GiB  0700 Basic data partition
   5       426698752    498378751   34.2 GiB  0700 Basic data partition
   6       498380800    500117503   848.0 MiB 2700
```

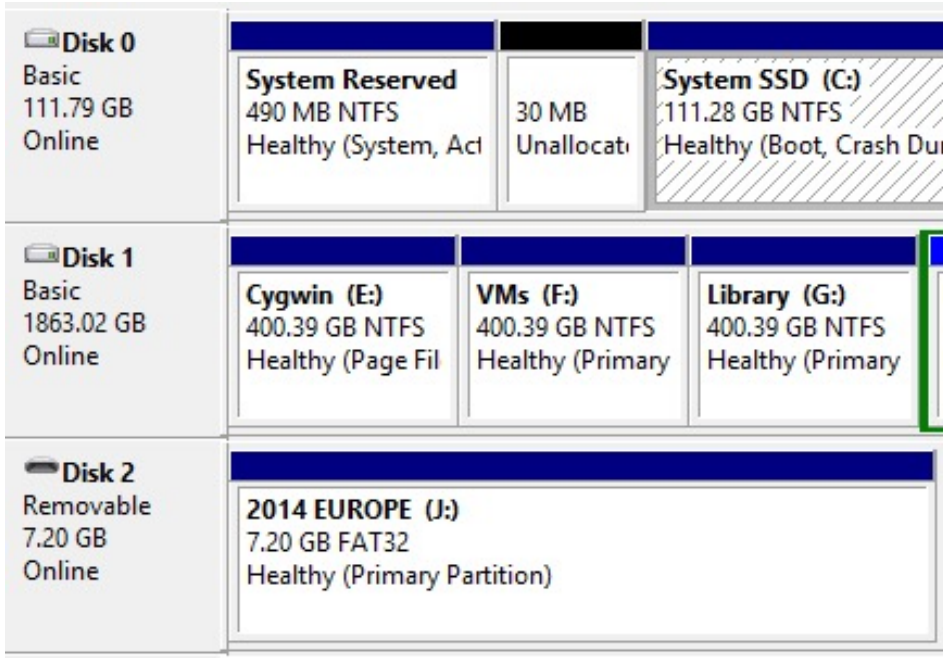
Viewing the GPT partitions

```
Command (? for help): i
Partition number (1-6): 1
Partition GUID code: C12A7328-F81F-11D2-BA4
Partition unique GUID: 0B3FFFDA-A04F-4496-B
First sector: 2048 (at 1024.0 KiB)
Last sector: 206847 (at 101.0 MiB)
Partition size: 204800 sectors (100.0 MiB)
Attribute flags: 8000000000000000
Partition name: 'EFI system partition'
```

Disks as seen by windows

Each partition is identified by a letter C, D, E, … Z

Volume	Layout	Type	File Sys
2014 EUROPE (J:)	Simple	Basic	FAT32
Cygwin (E:)	Simple	Basic	NTFS
HD Unused (I:)	Simple	Basic	NTFS
Library (G:)	Simple	Basic	NTFS
Photos (H:)	Simple	Basic	NTFS
System Reserved	Simple	Basic	NTFS
System SSD (C:)	Simple	Basic	NTFS
VMs (F:)	Simple	Basic	NTFS



Disks as seen by WMIC

- We met WMIC in Week 7
- It is a very handy tool for looking at Windows
- wmic diskdrive list brief

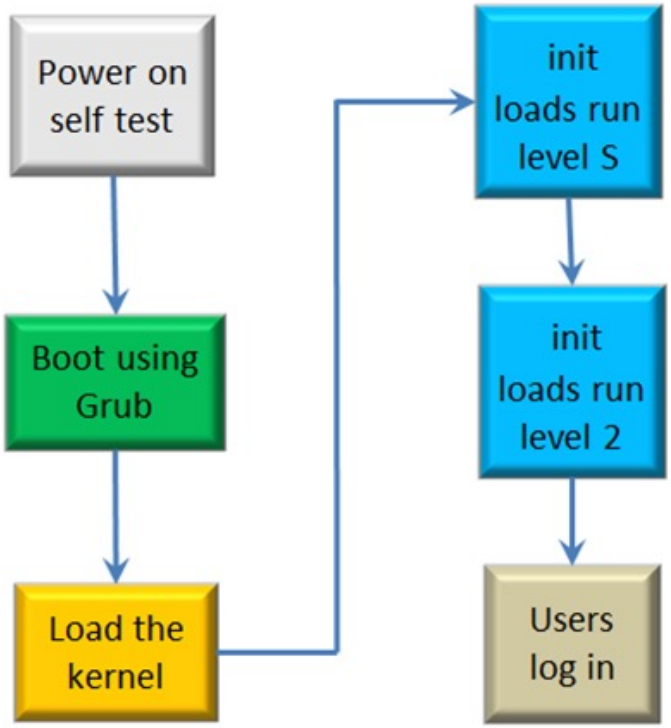
```
C:\Users\graha>wmic diskdrive list brief
Caption                  DeviceID            Model                  Partitions  Size
HFS256G39TND-N210A       \\.\PHYSICALDRIVE0  HFS256G39TND-N210A     5           256052966400
TDK LoR Platinum 3.0 USB Devi  \\.\PHYSICALDRIVE1  TDK LoR Platinum 3.0 USB  2           7723537920
```

Objectives

- To classify Disks
- To understand Partitions
- To understand the Boot Process

The boot process

- To ensure the integrity of the file system we need to guarantee the boot process
- Power On Self Test (POST)
- Basic I/O System (BIOS)
- File System Loader
- Init the Operating System (OS)
- Pass control to the OS



Secure Boot

- A Boot virus rootkit can install and hide from the OS
  - this is very bad
- Secure boot checks a signed certificate in the UEFI
- Microsoft own the Certificate so an issue for Linux
- The Linux distributor uses a shim to allow UEFI to call their boot loader.
- The Linux distributor buys a certificate from Microsoft to sign their boot loader.
- If the boot loader hash matches the certificate, it will load.

FIN