CS 798: Digital Forensics and Incident Response

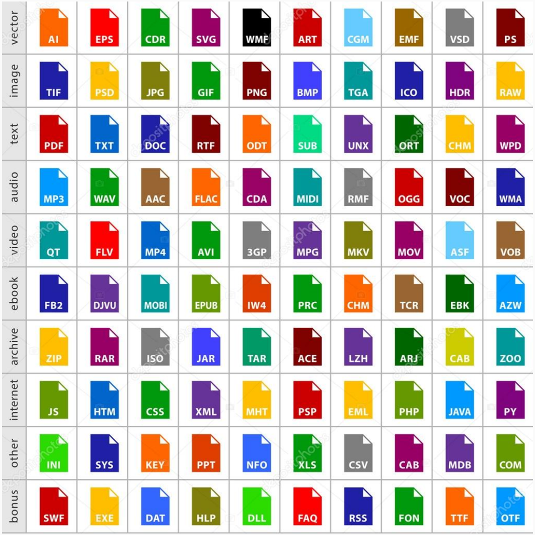
Lecture 7 - Storage and Volume Analysis

Diogo Barradas

Winter 2025

University of Waterloo

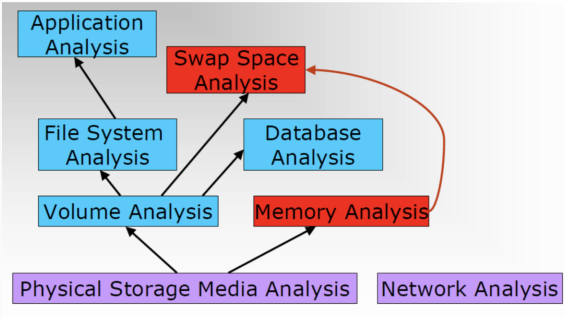
We talked about files...



vector	AI	EPS	CDR	SVG	WMF	ART	CGM	EMF	VSD	PS
image	TIF	PSD	JPG	GIF	PNG	BMP	TGA	ICO	HDR	RAW
text	PDF	TXT	DOC	RTF	ODT	SUB	UNX	ORT	CHM	WPD
audio	MP3	WAV	AAC	FLAC	CDA	MIDI	RMF	OGG	VOC	WMA
video	QT	FLV	MP4	AVI	3GP	MPG	MKV	MOV	ASF	VOB
ebook	FB2	DJVU	MOBI	EPUB	IW4	PRC	CHM	TCR	EBK	AZW
archive	ZIP	RAR	ISO	JAR	TAR	ACE	LZH	ARJ	CAB	ZOO
internet	JS	HTM	CSS	XML	MHT	PSP	EML	PHP	JAVA	PY
other	INI	SYS	KEY	PPT	NFO	XLS	CSV	CAB	MDB	COM
bonus	SWF	EXE	DAT	HLP	DLL	FAQ	RSS	FON	TTF	OTF

Data is organized on different abstraction layers

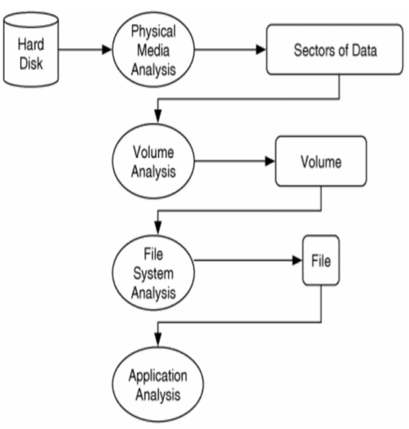
- When performing forensics investigation:
  - We can focus on abstraction layers independently
  - The lower you go, the more information you can get



Source: „File System Forensic Analysis", Brian Carrier

Analysis of persistent storage

- This shows a disk that is analyzed to produce a stream of bytes
- Bytes are analyzed at the volume layer to produce volumes
- The volumes are analyzed at the file system layer to produce a file
- The file is then analyzed at the application layer



Outline

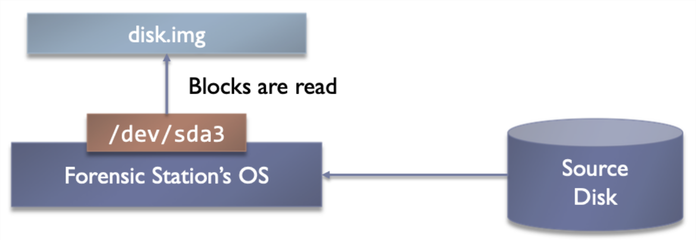
1. Volume analysis

2. Storage media and forensic implications

Volume analysis

Creation of a storage medium bit-stream copy

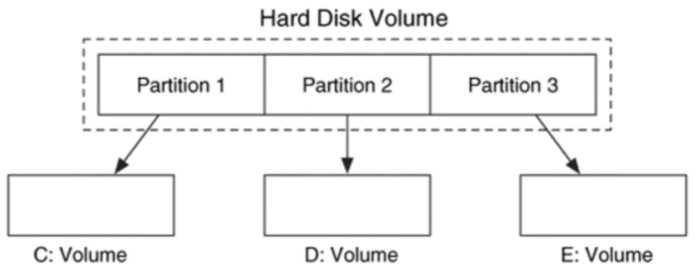
- **Bit-stream copy**
- Exact bit-by-bit copy of the original storage medium
- Capture includes meta-data and both active (known files) as well as inactive data (deleted file fragments)



- Extract a disk image using the `dd` tool family
  - `dc3ddif=/dev/sda3of=/home/forensics/disk.imghash= md5log=/home/forensics/disk.log`

Creation of a storage medium bit-stream copy

- Consider a Microsoft Windows system with one hard disk
  - The hard disk volume is partitioned into three smaller volumes
  - Each volume has a file system
  - Windows assigns the names C, D, and E to each volume



Volumes and partitions

- A volume is a collection of addressable sectors that an OS or application can use for data storage (logical abstraction)
  - Sectors need not be consecutive on a physical storage device
  - A volume may result from assembling / merging smaller volumes (e.g., across machines/disks)
- A partition is a fraction of consecutive sectors in a volume. By definition, a partition is also a volume (logical division of physical space)
- Partitions are used in many scenarios, including:
  - Some file systems have max size smaller than hard disks
  - Many laptops put to sleep store memory on special partition
  - Separate partitions for booting multiple OSes

Basic analysis of volume layout

- In many cases, an investigator acquires an entire hard disk and imports the image into his analysis software
- To identify the volume layout, where the file system starts and ends, the partition tables must be analyzed
  - Not all sectors need to be assigned to a partition, and may contain data from a previous FS or that the suspect was trying to hide
- In some cases, the partition system may become corrupt or erased, and automated tools will not work

Basic analysis of volume layout

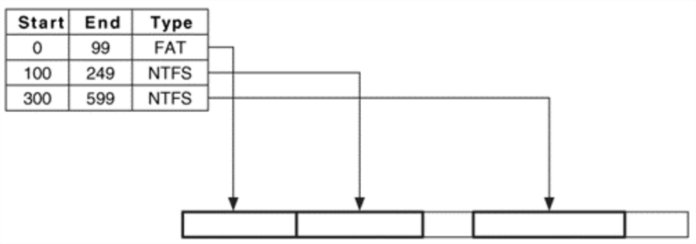
- List the partitions in the volume image

```
 # mmls -t dos disk1.dd
Units are in 512-byte sectors
Slot Start End Length Description
00:  --- 0000000000 0000000000 0000000001 Table #0
01:  --- 0000000001 0000000062 0000000062 Unallocated
02:  00:00 0000000063 0001028159 0001028097 Win95 FAT32 (0x0B)
03:  --- 0001028160 0002570399 0001542240 Unallocated
04:  00:03 0002570400 0004209029 0001638630 OpenBSD (0xA6)
05:  00:01 0004209030 0006265349 0002056320 NTFS (0x07)
```

##:##: This format is used with volume systems that have multiple tables. The first two numbers correspond to the table ID and the second set of numbers correspond to the entry in that table. 00:01 is entry 1 in table 0.

Be aware of partitioning methods

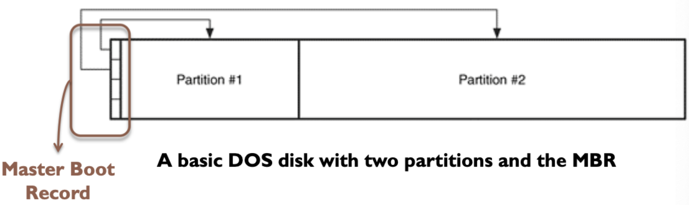
- OS and hardware platform use different partitioning methods
- Typical partition systems have tables; entries describe partitions
- A partition system cannot serve its purpose if those values are corrupt or non-existent



Start	End	Type
0	99	FAT
100	249	NTFS
300	599	NTFS

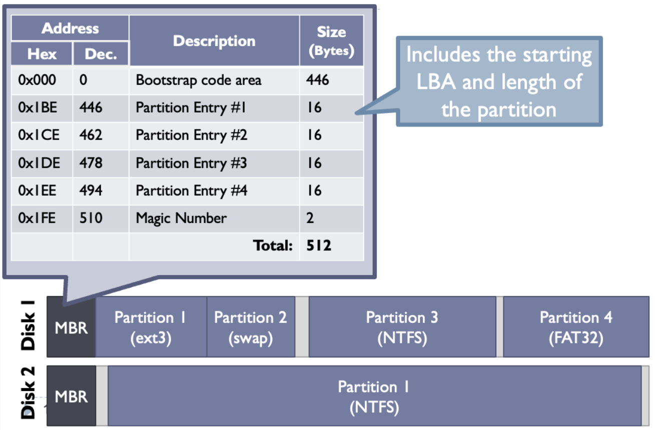
A prevalent partition system: DOS partitions

- A disk that is organized using DOS partitions has a Master Boot Record (MBR) in the first 512-byte sector
- The MBR has a partition table with a maximum of four entries, one for each possible partition



A basic DOS disk with two partitions and the MBR

MBR: High-level example



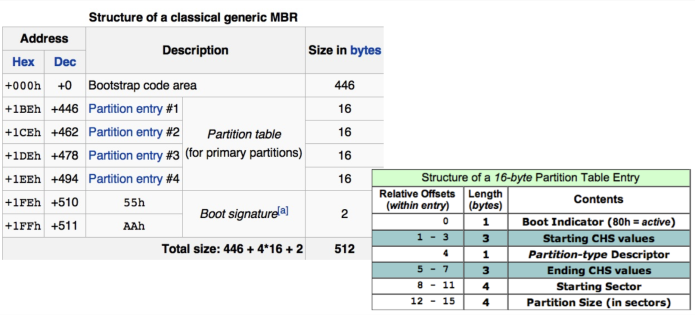
Address		Description	Size (Bytes)
Hex	Dec.		
0x000	0	Bootstrap code area	446
0x1BE	446	Partition Entry #1	16
0x1CE	462	Partition Entry #2	16
0x1DE	478	Partition Entry #3	16
0x1EE	494	Partition Entry #4	16
0x1FE	510	Magic Number	2
		Total:	512

Includes the starting LBA and length of the partition

LBA: logical block address

MBR layout

- The MBR contains bootstrap code, a partition table, and a signature value
- The bootstrap code determines the active partition and fires a second stage boot loader off the boot sector of the active partition



Structure of a classical generic MBR

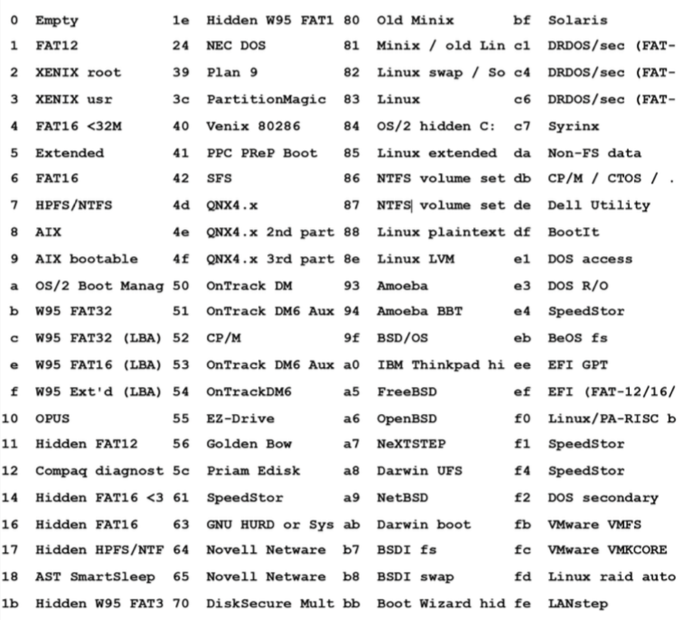
Address		Description		Size in bytes
Hex	Dec			
+000h	+0	Bootstrap code area		446
+1BEh	+446	Partition entry #1	*Partition table* (for primary partitions)	16
+1CEh	+462	Partition entry #2		16
+1DEh	+478	Partition entry #3		16
+1EEh	+494	Partition entry #4		16
+1FEh	+510	55h	*Boot signature*[a]	2
+1FFh	+511	AAh		
			Total size: 446 + 4*16 + 2	**512**

Structure of a *16-byte* Partition Table Entry

Relative Offsets (*within entry*)	Length (*bytes*)	Contents
0	1	Boot Indicator (80h = *active*)
1 - 3	3	Starting CHS values
4	1	*Partition-type* Descriptor
5 - 7	3	Ending CHS values
8 - 11	4	Starting Sector
12 - 15	4	Partition Size (in sectors)

CHS: used to specify the track, head and sector as they physcially exist on the hard drive
Partitions are limited to 32-bit entries – maximum partition size is 2TB

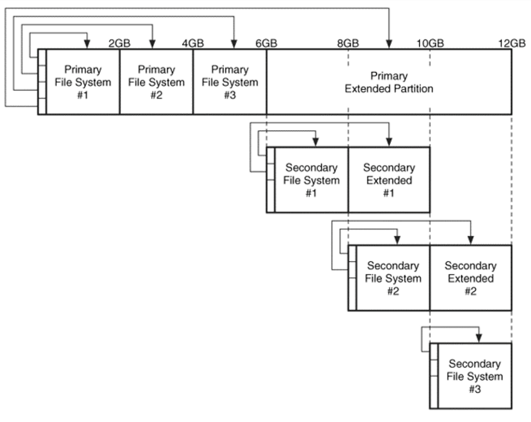
Types of DOS partitions



```
 0  Empty           1e  Hidden W95 FAT1 80  Old Minix       bf  Solaris
 1  FAT12           24  NEC DOS         81  Minix / old Lin c1  DRDOS/sec (FAT-
 2  XENIX root      39  Plan 9          82  Linux swap / So c4  DRDOS/sec (FAT-
 3  XENIX usr       3c  PartitionMagic  83  Linux           c6  DRDOS/sec (FAT-
 4  FAT16 <32M      40  Venix 80286     84  OS/2 hidden C:  c7  Syrinx
 5  Extended        41  PPC PReP Boot   85  Linux extended  da  Non-FS data
 6  FAT16           42  SFS             86  NTFS volume set db  CP/M / CTOS / .
 7  HPFS/NTFS       4d  QNX4.x          87  NTFS volume set de  Dell Utility
 8  AIX             4e  QNX4.x 2nd part 88  Linux plaintext df  BootIt
 9  AIX bootable    4f  QNX4.x 3rd part 8e  Linux LVM       e1  DOS access
 a  OS/2 Boot Manag 50  OnTrack DM      93  Amoeba          e3  DOS R/O
 b  W95 FAT32       51  OnTrack DM6 Aux 94  Amoeba BBT      e4  SpeedStor
 c  W95 FAT32 (LBA) 52  CP/M            9f  BSD/OS          eb  BeOS fs
 e  W95 FAT16 (LBA) 53  OnTrack DM6 Aux a0  IBM Thinkpad hi ee  EFI GPT
 f  W95 Ext'd (LBA) 54  OnTrackDM6      a5  FreeBSD         ef  EFI (FAT-12/16/
10  OPUS            55  EZ-Drive        a6  OpenBSD         f0  Linux/PA-RISC b
11  Hidden FAT12    56  Golden Bow      a7  NeXTSTEP        f1  SpeedStor
12  Compaq diagnost 5c  Priam Edisk     a8  Darwin UFS      f4  SpeedStor
14  Hidden FAT16 <3 61  SpeedStor       a9  NetBSD          f2  DOS secondary
16  Hidden FAT16    63  GNU HURD or Sys ab  Darwin boot     fb  VMware VMFS
17  Hidden HPFS/NTF 64  Novell Netware  b7  BSDI fs         fc  VMware VMKCORE
18  AST SmartSleep  65  Novell Netware  b8  BSDI swap       fd  Linux raid auto
1b  Hidden W95 FAT3 70  DiskSecure Mult bb  Boot Wizard hid fe  LANstep
```

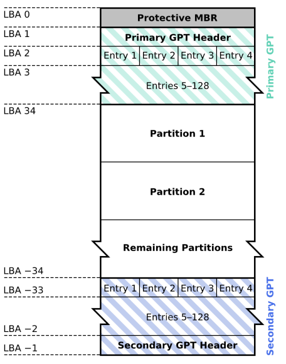
Extended partition concepts

- The MBR supports only up to four partitions
    - Consider a 12GB disk that the user wants to divide into six 2GB partitions because she is using multiple operating systems
- Solution: extended partitions forming a linked list



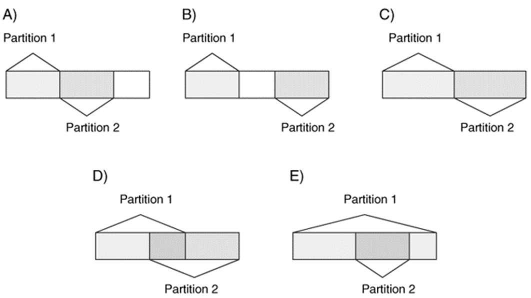
GPT (GUID Partition Table)

- GPT is used by most modern systems
  - Uses UEFI, while MBR uses BIOS
  - Compatible with MBR
  - It is duplicated at the start and end of disk
  - Allows for 128 partitions on Windows (but can be extended)

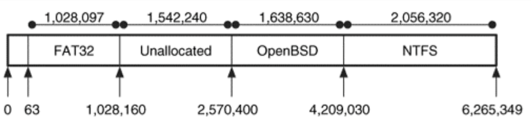


Consistency checking

- Check each partition relative to the other partitions
  - Compares ending of last partition with the end of its parent volume
  - Compares the start and end sectors of consecutive partitions



Extracting the partition contents



- To extract the file system partitions from the disk image, we take the starting sector and size of each partition as shown here:

```
 # dd if=disk1.dd of=part1.dd bs=512 skip=63 count=1028097
# dd if=disk1.dd of=part2.dd bs=512 skip=2570400 count=1638630
# dd if=disk1.dd of=part3.dd bs=512 skip=4209030 count=2056320
```

Recovering deleted partitions

- Common to thwart a forensic investigation by repartitioning a disk or clearing the partition structures
- Partition recovery tools work by assuming that a FS was located in each partition
- Many file systems have data structures with signature values
  - E.g., FAT has values 0x55 and 0xAA in bytes 510 and 511 of first sector
- When the tool finds a signature, additional tests can be conducted on the range of valid values
- Other tools use heuristics
  - `http://www.cgsecurity.org/testdisk.html`

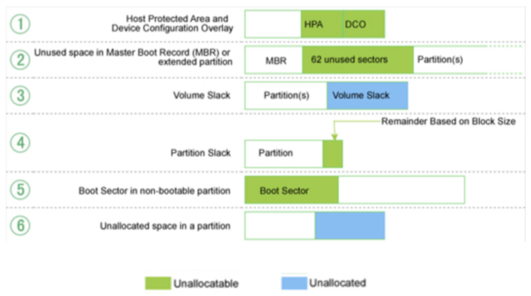
Recovering deleted partitions (cont.)

- E.g., using the gpart tool

```
 # gpart -v disk2.dd
* Warning:  strange partition table magic 0x0000.
Begin scan...
Possible partition(DOS FAT), size(800mb), offset(0mb)
type:  006(0x06)(Primary 'big' DOS (> 32MB))
size:  800mb #s(1638566) s(63-1638628)
chs:  (0/1/1)-(101/254/62)d (0/1/1)-(101/254/62)r
hex:  00 01 01 00 06 FE 3E 65 3F 00 00 00 A6 00 19 00
Possible partition(DOS FAT), size(917mb), offset(800mb) type:  006(0x06)(Primary 'big'
DOS (> 32MB))
size:  917mb #s(1879604) s(1638630-3518233)
chs:  (102/0/1)-(218/254/62)d (102/0/1)-(218/254/62)r
hex:  00 00 01 66 06 FE 3E DA E6 00 19 00 34 AE 1C 00
Possible partition(Linux ext2), size(502mb), offset(1874mb)
type:  131(0x83)(Linux ext2 filesystem)
size:  502mb #s(1028160) s(3839535-4867694)
chs:  (239/0/1)-(302/254/63)d (239/0/1)-(302/254/63)r hex:  00 00 01 EF 83 FE 7F 2E 2F
96 3A 00 40 B0 0F 00
```

Places where data may be hidden

- It is possible that some unused disk space can be used for storing hidden data



1
Host Protected Area and Device Configuration Overlay
HPA
DCO
2
Unused space in Master Boot Record (MBR) or extended partition
MBR
62 unused sectors
Partition(s)
3
Volume Slack
Partition(s)
Volume Slack
Remainder Based on Block Size
4
Partition Slack
Partition
5
Boot Sector in non-bootable partition
Boot Sector
6
Unallocated space in a partition
🟩 Unallocatable
🟦 Unallocated

http://www.berghel.net/publications/data_hiding/data_hiding.php

Dealing with full volume encryption

- Full volume encryption: method for encrypting a single partition, either physical or virtual, on a hard drive
- Implementations:
  - BitLocker
  - FileVault Disk Encryption
  - FreeOTFE
  - TrueCrypt, VeraCrypt (more on this later...)
- Extract encryption key from memory (e.g., cold boots)

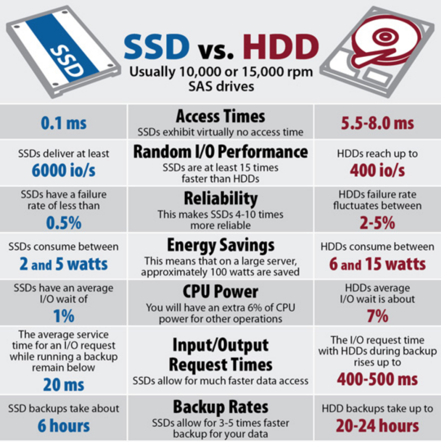
Storage media and forensic implications

Some relevant issues about storage technology

- What factors affect data longevity on the storage device?
- Are there mechanisms for internal data replication?
- Are there security defenses against data extraction?
- Are there potential locations for data hiding on the device?

Common storage technology

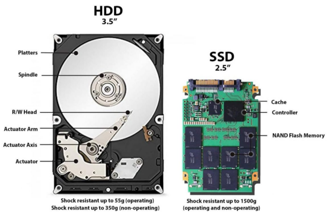
- Solid State Drives (SDD) and Hard Disk Drives (HDD)



SSD vs. HDD

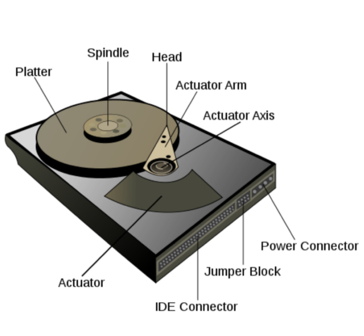
Usually 10,000 or 15,000 rpm
SAS drives

0.1 ms	**Access Times** SSDs exhibit virtually no access time	**5.5-8.0 ms**
SSDs deliver at least **6000 io/s**	**Random I/O Performance** SSDs are at least 15 times faster than HDDs	HDDs reach up to **400 io/s**
SSDs have a failure rate of less than **0.5%**	**Reliability** This makes SSDs 4-10 times more reliable	HDDs failure rate fluctuates between **2-5%**
SSDs consume between **2 and 5 watts**	**Energy Savings** This means that on a large server, approximately 100 watts are saved	HDDs consume between **6 and 15 watts**
SSDs have an average I/O wait of **1%**	**CPU Power** You will have an extra 6% of CPU power for other operations	HDDs average I/O wait is about **7%**
The average service time for an I/O request while running a backup remain below **20 ms**	**Input/Output Request Times** SSDs allow for much faster data access	The I/O request time with HDDs during backup rises up to **400-500 ms**
SSD backups take about **6 hours**	**Backup Rates** SSDs allow for 3-5 times faster backup for your data	HDD backups take up to **20-24 hours**


HDD
3.5″
Platters
Spindle
R/W Head
Actuator Arm
Actuator Axis
Actuator
Shock resistant up to 55g (operating)
Shock resistant up to 350g (non-operating)
SSD
2.5″
Cache
Controller
NAND Flash Memory
Shock resistant up to 1500g
(operating and non-operating)

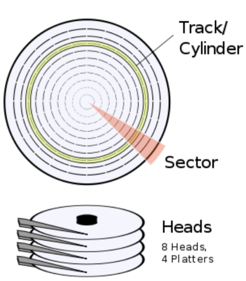
Hard disks

- A lot of data is stored on hard disc drives
    - In commercial use since 1956



Hard disks

- **Head**
  - Device which reads and writes data
- **Track**
  - Individual circles on disk platter where data are located
- **Cylinder**
  - A column of tracks on a disk drive with 2 or more platters
- **Sector**
  - An individual section of data on a track - the smallest amount of data which can be written to the disk – usually 512 bytes
- **Disk Capacity** = #cylinders × #heads × #sectors × sector_size

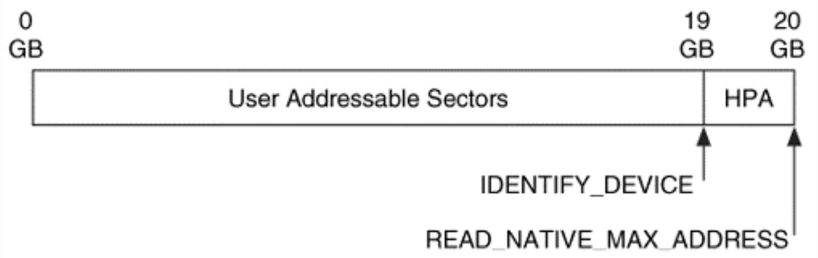


Hard disk formatting and forensic implications

- **Low-level formatting**
  - Physically defines tracks and sectors on disk
  - Does erase data
  - Typically only performed at factory
- **High-level formatting**
  - Performed when initializing a file system on a partition
  - Does not destroy data on disk!
    - Only FS metadata

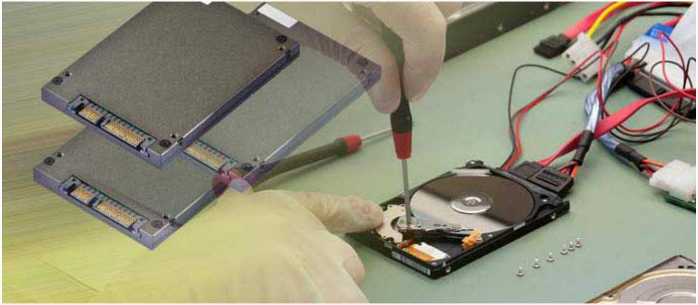
Potential locations for data hiding in HDDs

- Host Protected Area (HPA) was added in ATA-4 spec
    - Computer vendors can store data that would not be erased when a user formats the HDD
- Can be detected by comparing output of ATA commands
    - An HPA can contain system files, hidden information, or both



Hard disk passwords

- ATA-3 spec introduced optional security features
- Passwords can be set to lock the HDD against R/W
- Data recovery is still possible by opening the disk
- Password can be used to wipe the disk

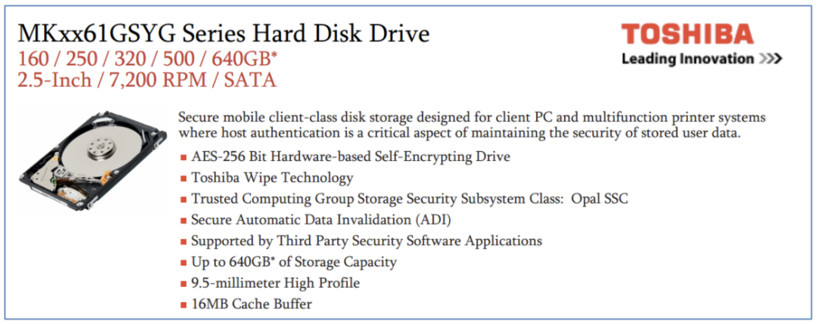


Self-encrypting drives

- Hard drive firmware includes encryption
- Custom firmware simulates unencrypted boot partition
    - But the entire user-accessible portion of the disk is encrypted
- User must enter a key to boot up
- Forensic erase takes less than a second
    - Simply overwrite the key

Self-wiping hard drives

- Wipes out key when drive is moved to another computer
- Makes traditional acquisition impossible



MKxx61GSYG Series Hard Disk Drive

160 / 250 / 320 / 500 / 640GB*
2.5-Inch / 7,200 RPM / SATA

TOSHIBA
Leading Innovation >>>

Secure mobile client-class disk storage designed for client PC and multifunction printer systems where host authentication is a critical aspect of maintaining the security of stored user data.

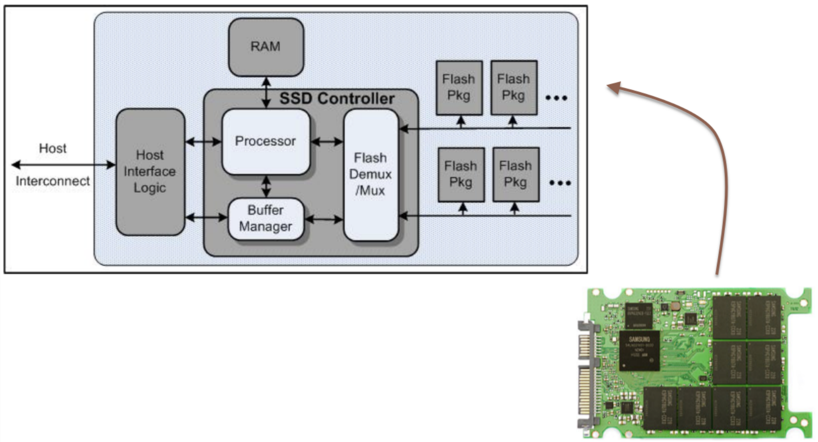
- AES-256 Bit Hardware-based Self-Encrypting Drive
- Toshiba Wipe Technology
- Trusted Computing Group Storage Security Subsystem Class: Opal SSC
- Secure Automatic Data Invalidation (ADI)
- Supported by Third Party Security Software Applications
- Up to 640GB* of Storage Capacity
- 9.5-millimeter High Profile
- 16MB Cache Buffer

For forensics, hard drives are well understood

- Bits of data are placed onto magnetic media via repositionable recording heads
- The data may be randomly accessed by moving the heads over a selected cylinder
- All such operations are easily controlled via drive control commands which, for example, allow a sector of information to be read or written
- All of these are nice properties for forensic analysis, but for SSDs things get tougher...

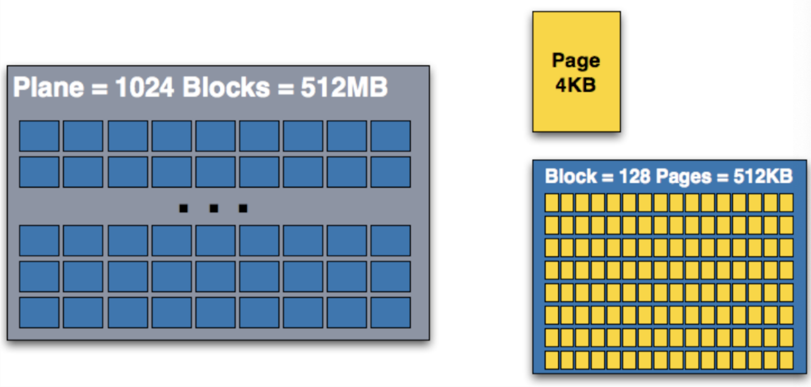
For forensics, hard drives are well understood

- Data is permanently stored in flash memory



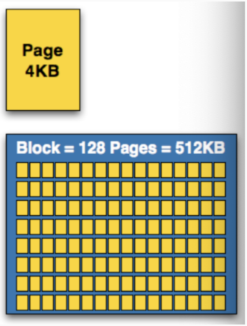
Internal structure of the flash memory

- Components: planes, blocks, and pages



Overview of SSD operations

- SSDs get slower as they fill up
- The smallest structure you can write is a page (4 KB)
- But you cannot write to a page unless it is empty
- The smallest structure you can erase is a block (512 KB)
- Also, you can only erase a block ∼10,000 times before it fails



Challenges of SSDs for forensic investigators

- SSD's internal data structures are hidden
  - Internals not well understood - no accepted standards
- The physical location of any block within the SSD device will almost certainly not match the external Logical Block Address
- Wear leveling algorithms allocated the data in an unpredictable way (non-standard)
  - May distribute multiple copies of data in various locations
- Garbage collection clears blocks marked for deletion
  - When an SSD is powered on, it may automatically initiate a trim operation to clear deleted data
- Hard to read data off chips directly: manufacturer dependent

Takeaways

- Forensic analysis of any piece of digital evidence is entirely dependent on the ability to interpret how the data is represented
- Storage media such as hard drives and solid state drives are major sources of data and create multiple challenges for investigators
- Volume analysis constitutes the first major step at interpreting forensic data from storage media images

Pointers

- **Textbook:**
    - Carrier – Chapters 3 & 4, Luttgens – Chapter 8
- **Other resources:**
    - A comprehensive black-box methodology for testing the forensic characteristics of solid-state drives. Bonetti et al.. ACSAC'13
    - Self-encrypting deception: weaknesses in the encryption of solid state drives. Meijer and van Gastel. S&P'19
- **Acknowledgements:**
    - Slides adapted from Nuno Santos's Forensics Cyber-Security course at Técnico Lisbon