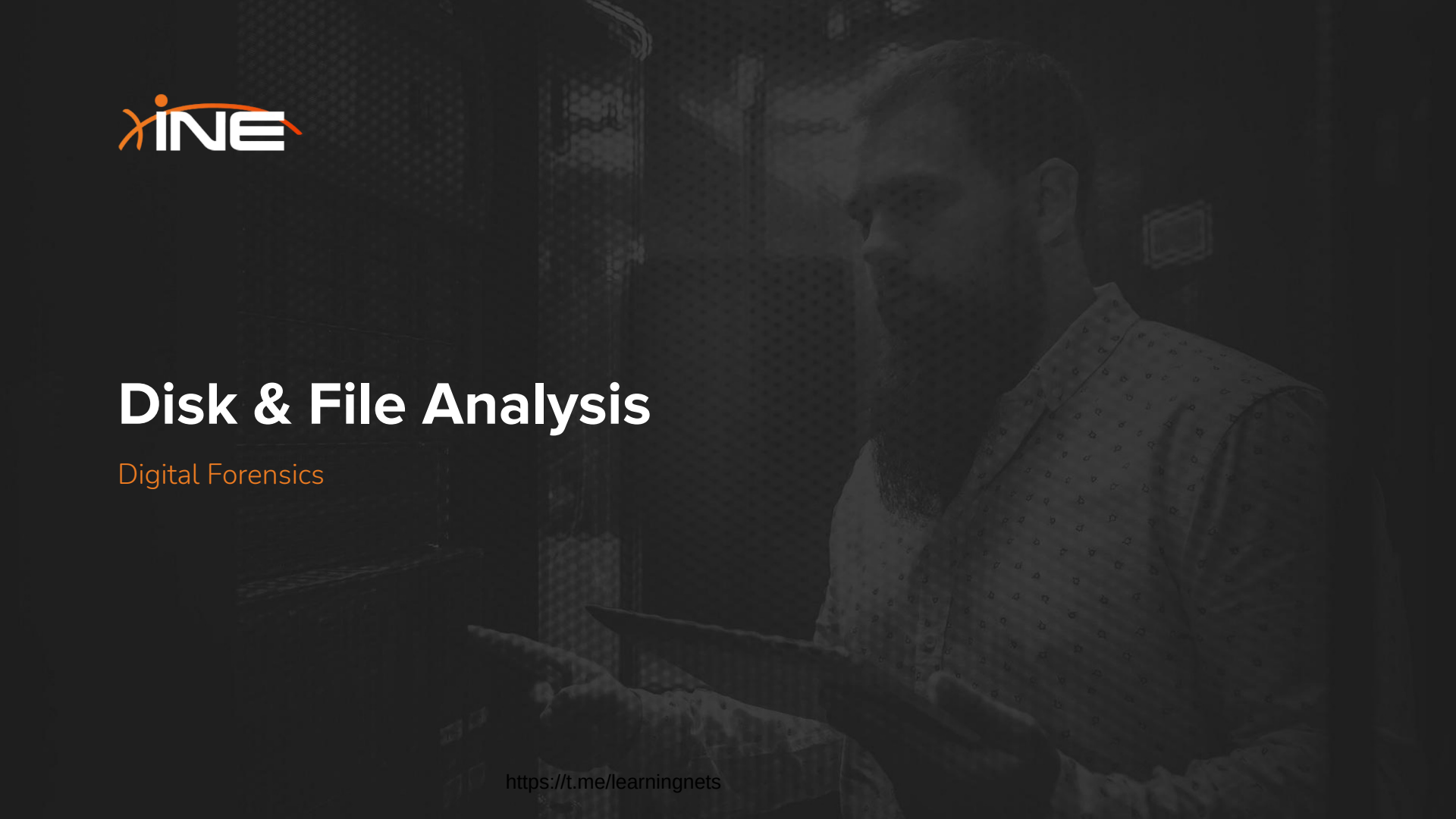


Disk & File Analysis

Digital Forensics

Course Introduction

Course Layout:

- Persistent Storage Principles
  - Module Introduction
  - Definition & History
  - Persistent Storage Devices
  - Volumes & Partitions
  - Disk Partitioning
- Fundamentals of File Systems
  - Module Introduction
  - Definition & History
  - FAT File System Analysis
  - NTFS File System Analysis
  - File Carving
  - The Sleuth Kit (TSK)
  - File Identification & Structure
  - Temporary File Overview

At the end of this course, you will be able to:

- Understand the physical and logical technologies used today,
- Define the relationship between storage media and file systems, and
- Perform disk and file system analysis.

Persistent Storage Principles

Module Layout:

- Module Introduction
- Definition & History
- Persistent Storage Devices
- Volumes & Partitions
- Disk Partitioning

At the end of this module, you will be able to:

- Understand the common types of persistent storage used,
- Define the components of persistent storage mechanisms, and
- Perform disk and volume analysis.

Definition & History

Where is the Data?

Since analyzing evidence is the primary objective of digital forensics, we need to know where to find evidentiary data.

Learning the Lingo

In everyday life, the terms ‘data’ and ‘information’ are often used interchangeably. To many digital forensic practitioners, however, they mean different things.

Layers of Analysis

The types of data analyzed can be categorized into four basic layers:

- Physical
- Volume
- File System
- OS & Application

Layers of Analysis

Data become information once the bits and bytes are processed.

Example

- The data we need can be just a few bytes on a disk
- How do we read it and benefit from the information inside it

Processing

Information

```
00109B00  80 35 81 35 82 35 83 35  84 35 85 35 86 35 87 35  88 35 89 35 8A 35 8B 35  8C 35 8D 35 8E 35 8F 35
00109B20  90 35 91 35 92 35 93 35  94 35 95 35 96 35 97 35  98 35 99 35 9A 35 9B 35  9C 35 9D 35 9E 35 9F 35
00109B40  A0 35 A1 35 A2 35 A3 35  A4 35 A5 35 A6 35 A7 35  A8 35 A9 35 AA 35 AB 35  AC 35 AD 35 AE 35 AF 35
00109B60  B0 35 B1 35 B2 35 B3 35  B4 35 B5 35 B6 35 B7 35  B8 35 B9 35 BA 35 BB 35  BC 35 BD 35 BE 35 BF 35
00109B80  C0 35 C1 35 C2 35 C3 35  C4 35 C5 35 C6 35 C7 35  C8 35 C9 35 CA 35 CB 35  CC 35 CD 35 CE 35 CF 35
00109BA0  D0 35 D1 35 D2 35 D3 35  D4 35 D5 35 D6 35 D7 35  D8 35 D9 35 DA 35 DB 35  DC 35 DD 35 DE 35 DF 35
00109BC0  E0 35 E1 35 E2 35 E3 35  E4 35 E5 35 E6 35 E7 35  E8 35 E9 35 EA 35 EB 35  EC 35 ED 35 EE 35 EF 35
```

*Untitled - Notepad

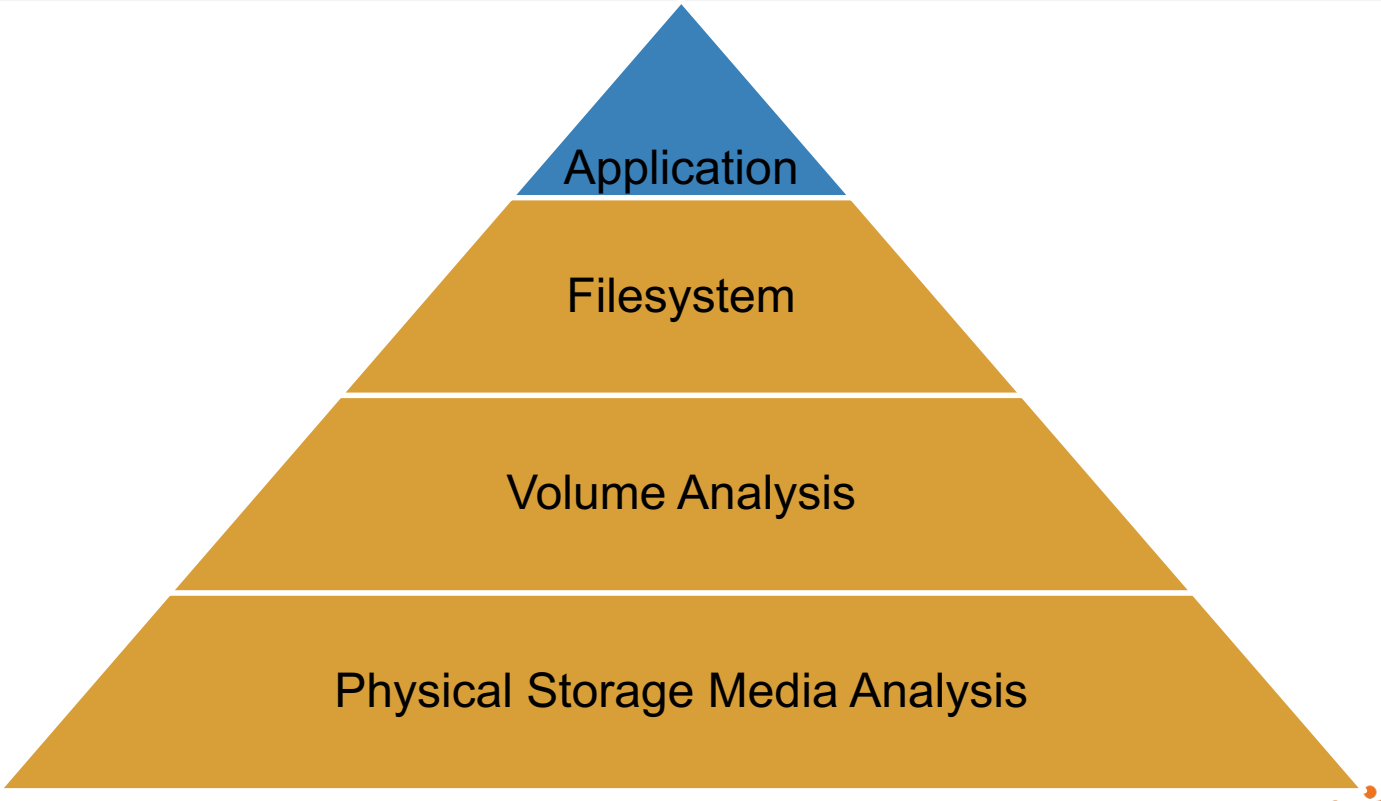
File Edit Format View Help

Hello World

Layers of Media Analysis

- Physical Media
- Volume
- File System
- Application and OS

Data Layers of Analysis



Persistent Storage Devices

What is Persistent Storage?

Mostly, we're talking about:

+ Hard disk drives
+ Solid state drives

Why the distinction?

Hard Disk Drives

+ Introduced by IBM in 1956
+ Use magnetic storage on spinning metal platters.

Platters, Spindles, and Heads, Oh My!

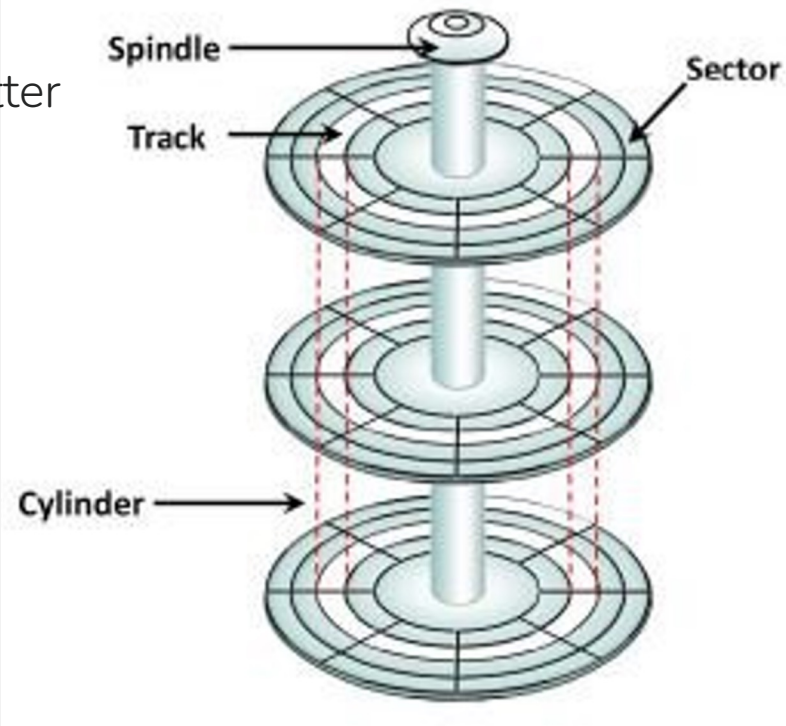
Hard Disk Drives Components:

1. Platter
2. Spindle
3. Head

Place holder for hdd demo

A Bit More About Platters

- Track: a stripe around a platter
- Sectors: a segment of a track on one platter
- Cylinder: the same track across multiple platters (e.g., track 2 of platters 1-3)



HDD Addressing

CHS

- Cylinder, Head, Sector
- Coordinate System
- Based on disk geometry
- First sector on a disk- (0,0,1)

LBA

- Address a drive capacity limitation
- Uses a single number to address each sector
- Not related to geometry
- Modern HDDs using LBA

LBA Address = (((Cylinder x HeadsPerCylinder) + Head) xSectorsPerTrack) + Sector - 1

Sector Sizes

- 512b or 4kb?
- 2010
- More efficient
- 4k fully supported

https://www.seagate.com/tech-insights/advanced-format-4k-sector-hard-drives-master-ti/

Solid State Drives

- Use flash storage and have no moving parts.
- It is important to understand how they differ from HDDs.

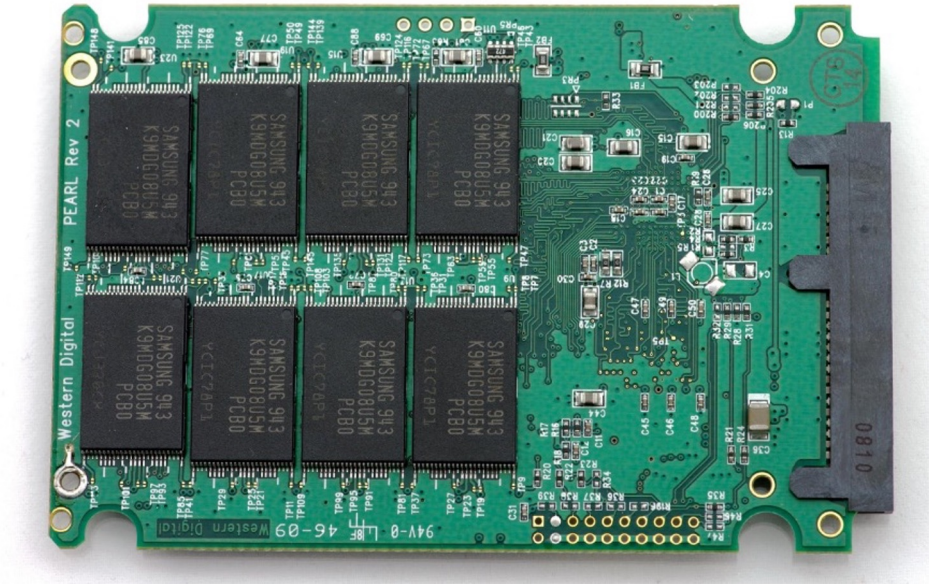
SSD Internals

SSDs have no moving parts.

Use NANDflash memory.

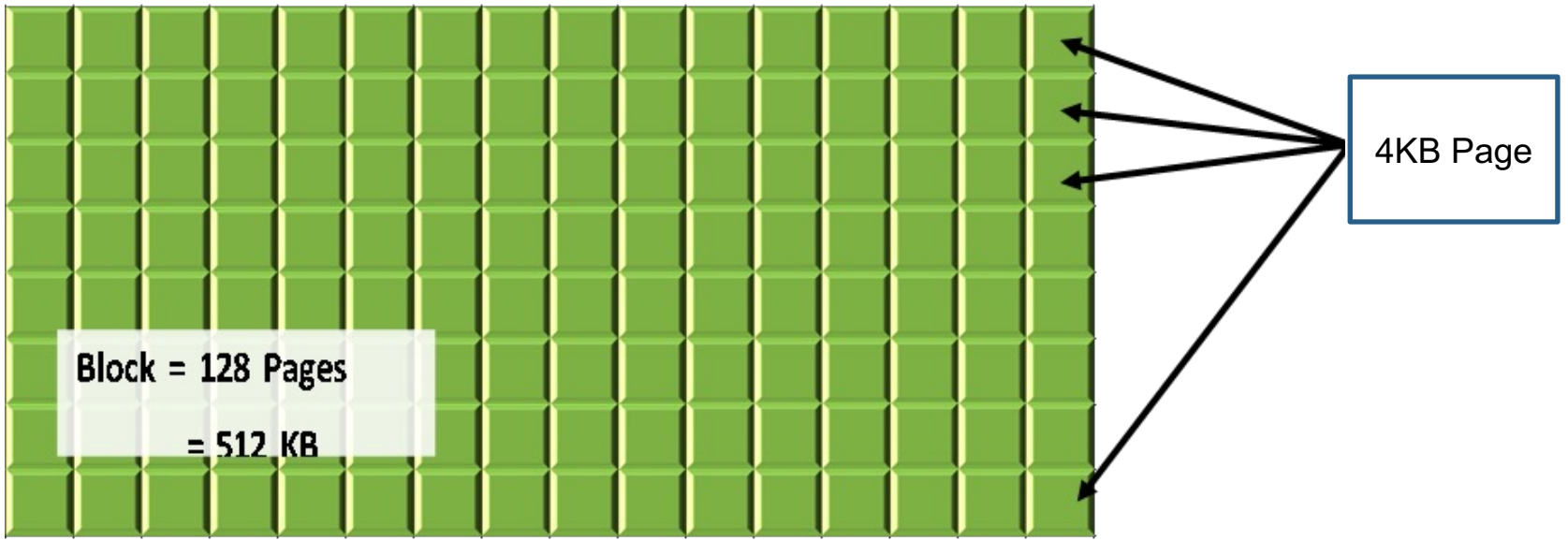
Non-volatile flash memory

Uses structures called pages and blocks.



Page by Page, and Block by Block

- Pages are the smallest unit of readable and writable memory
- Grouped into segments called blocks
- Smallest erasable unit.
- Typically, there are 128 pages in a block



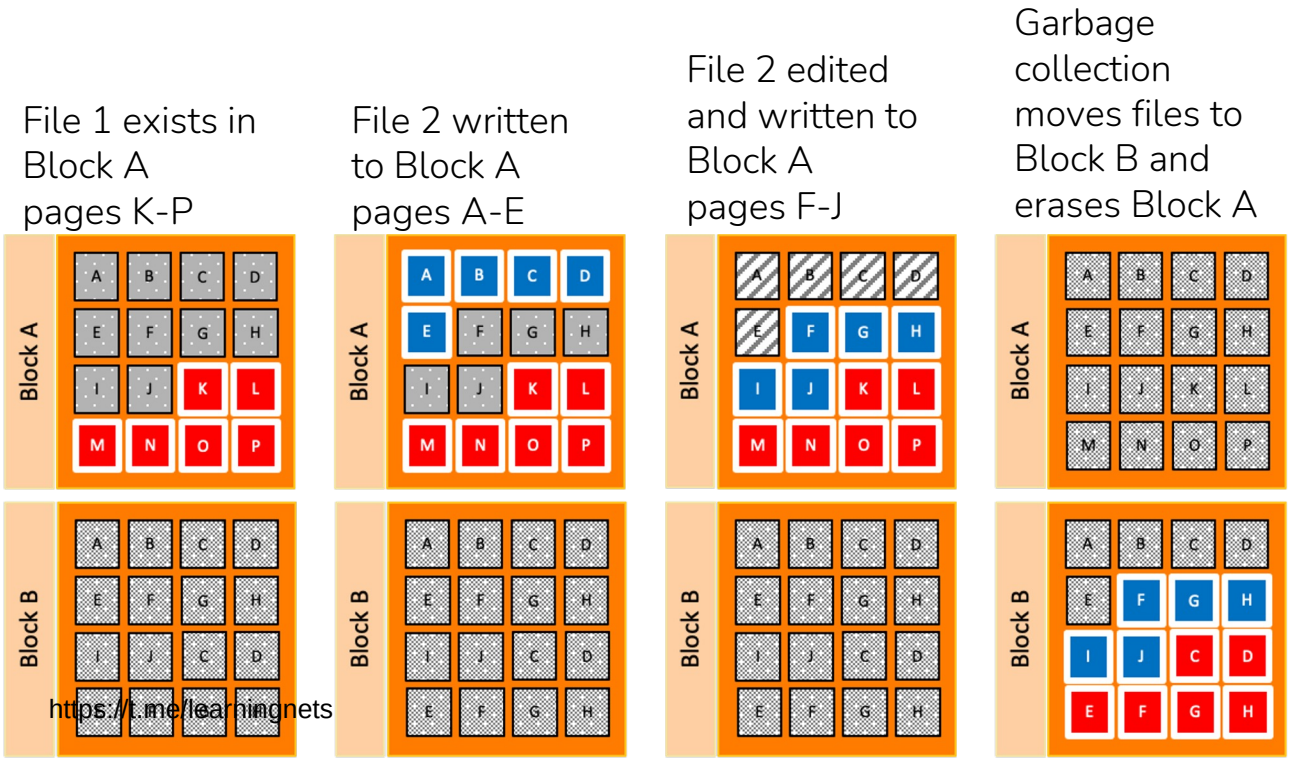
SSD Erasure

SSDs can only erase by the block, not by the page.

Also can’t directly overwrite data

That also triggers a process called ‘garbage collection.’

Garbage Collection?!

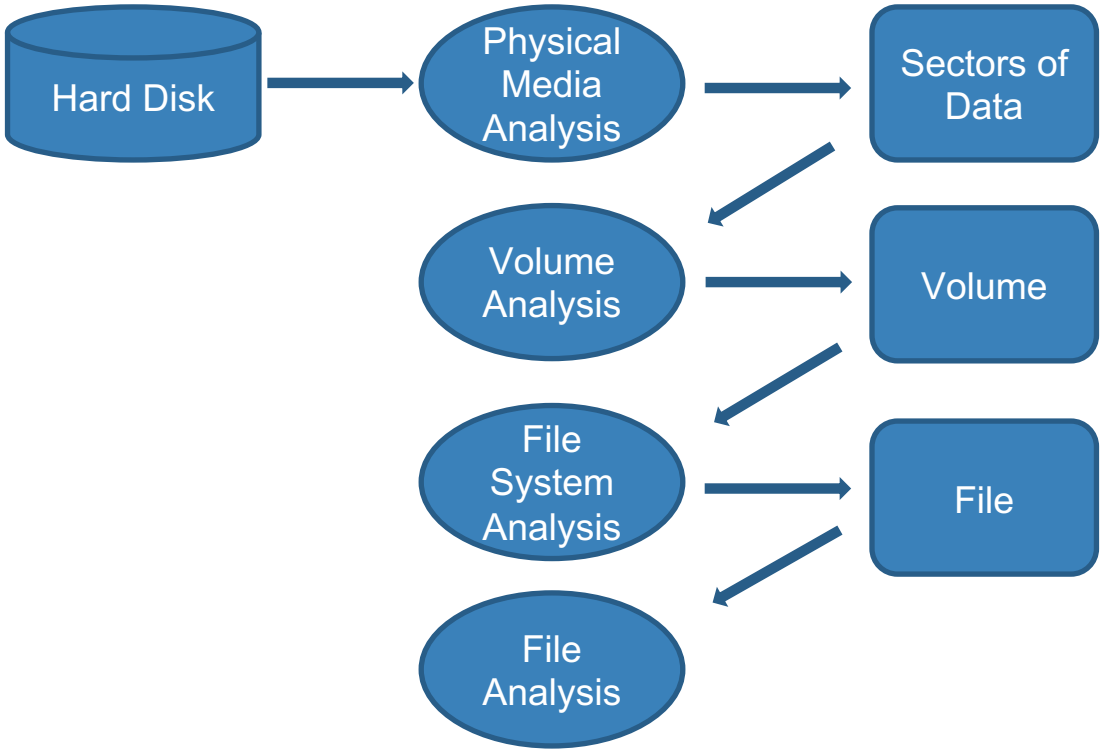


Plug Me In, Scotty!

Some of the common connections are:

- SATA
- M.12
- SCSI
- PATA / IDE

Abstraction Layers



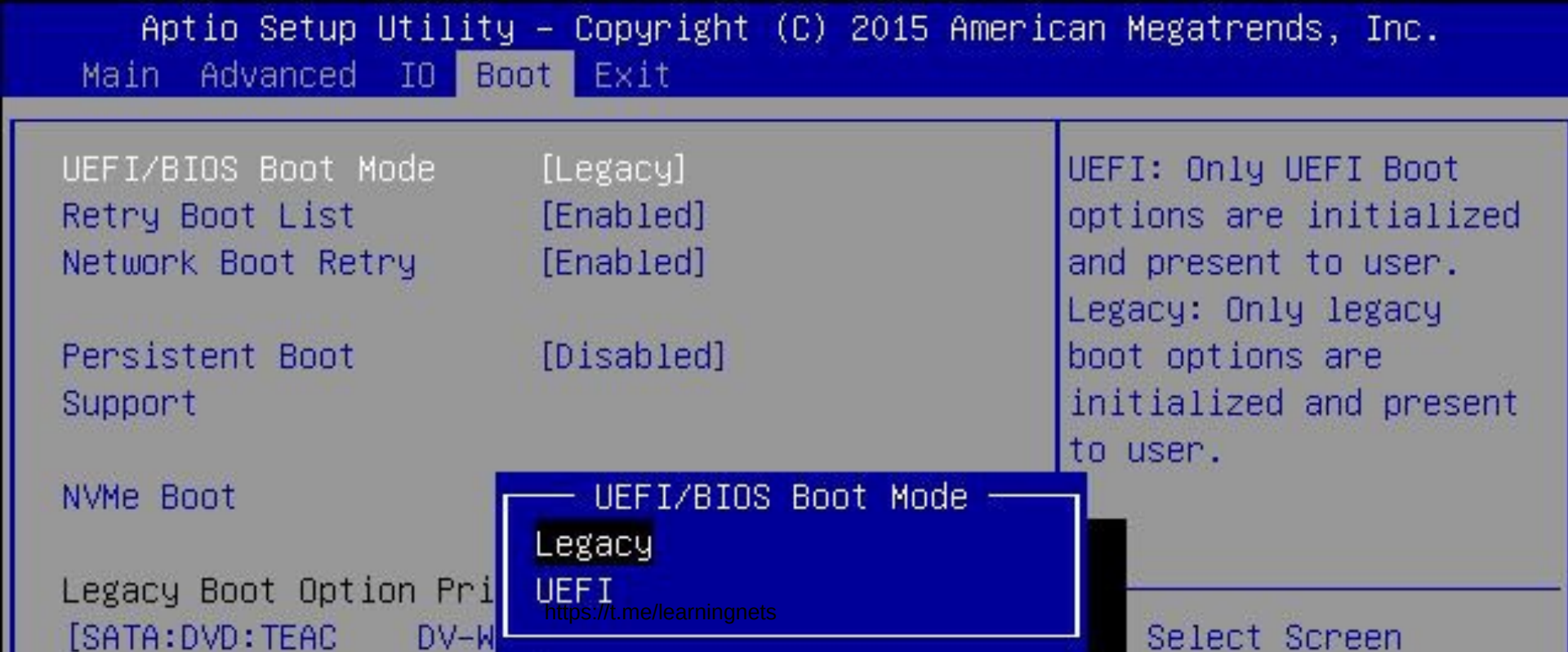
Booting to BIOS

Okay, so you have a disk with an operating system installed. But how does the machine hardware know that?

The short answer – BIOS

The Basic Input/Output System is firmware that serves as a rudimentary operating system and lives on the motherboard. BIOS helps the hardware figure out what it needs to do, providing runtime services to make things start properly. It is the first software to run when you hit the power button.

BIOS ... And then what?



Why does that matter?

Forensic tools can run from a bootable flash drive or CD/DVD!

Volumes & Partitions

In this module

+ Disk Structures
+ Partitions

What is a Partition?

A partition is a logical division of a hard disk, and there can be 1 or more partitions on a single disk.

Disks require at least one partition in order to be used.

Okay, then what is a volume?

Can be differentiated by allocated sectors are on the same disk.

A volume can span multiple disks.

But it doesn’t really matter.

Challenge

So now that you know a disk can have multiple volumes or partitions, how can you go about finding them all?

Disk Partitioning

Partition Tables

Below is an example partition table with four partitions, totalling just shy of 16gb of space.

Partition #	Start	End	Size
1	63	126	64
2	127	4193344	4193218
3	4193345	4194353	1009
4	4194354	16514130	12319777

The two most common partition table schemas in use today are the Master Boot Record (MBR) and GUID Partition Table (GPT).

Master Boot Record

MBRs allocate the first sector of the hard drive and holds three important pieces of information:

- Boot code
- Partition table
- Signature

Basic Structure of the MBR Sector

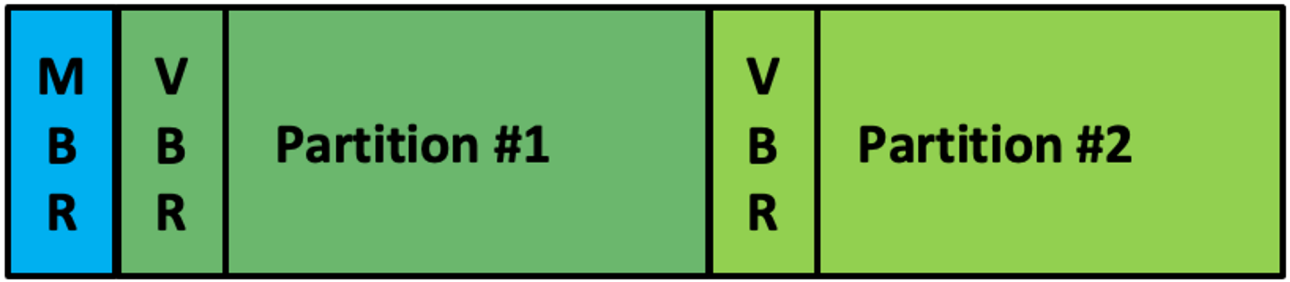
Offsets here are in decimal notation, and the total size is 1 sector (or 512 bytes).

Offset (start)	Offset (end)	Length (in bytes)	Description
000	445	446	**Boot Code**
446	509	64	**Partition Table**
510	511	2	**Signature**

MBR Sector – Boot Code

The boot code is 446 bytes, and was previously used to start the operating system.

MBR vs. VBR



MBR Sector – Partition Table

Offset relative to MBR (decimal)		Length in bytes	Content
Start	End		
446	461	16	Table for Partition #1
462	477	16	Table for Partition #2
478	493	16	Table for Partition #3
494	509	16	Table for Partition #4

MBR Sector – Partition Table Entries

Offset (relative)	Length (bytes)	Content
0	1	Boot indicator (0x80 = active)
1 – 3	3	Starting CHS value
4	1	Partition-type descriptor
5 – 7	3	Ending CHS value
8 – 11	4	Starting sector (LBA)
12 – 15	4	Partition size (in sectors)

Arithmetic Time!

2^{32} maximum partition size in sectors x 512 bytes/sector=

2,199,023,255,552 bytes or 2TB

Do I need to know partitioning schema before acquiring?

In short, no. Acquisition can be performed the same way whether it's an MBR or GPT disk. This knowledge is just another tool to prepare you for unexpected challenges, such as if the partition table is corrupted.

Module Review

This module covered:

- Definition & History
- Persistent Storage Devices
- Volumes & Partitions
- Disk Partitioning

Now you are able to:

- Understand the physical and logical technologies used today,
- Define the relationship between storage media and file systems, and
- Perform disk and file system analysis.

File System Fundamentals

Module Layout:

- Module Introduction
- Definition & History
- FAT File System Analysis
- NTFS File System Analysis
- File Carving
- The Sleuth Kit (TSK)
- File Identification & Structure
- Temporary File Overview

At the end of this module, you will be able to:

- Understand the role of file systems in modern computing,
- Define common file system structures, and
- Perform basic file system analysis.

Definition & History

What are File Systems?

File systems provide a framework for organizing data.

Quite simply, a file system is what makes this...

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI ASCII
00000000000	EB	52	90	4E	54	46	53	20	20	20	20	00	02	08	00	00	ëR NTFS
00000000016	00	00	00	00	00	F8	00	00	3F	00	FF	00	00	F8	0A	00	ø ? ÿ ø
00000000032	00	00	00	00	80	00	80	00	FF	FF	74	0C	00	00	00	00	€ € ÿÿt
00000000048	00	00	0C	00	00	00	00	00	02	00	00	00	00	00	00	00	
00000000064	F6	00	00	00	01	00	00	00	D7	AC	1D	48	E1	1D	48	AA	ö ×¬ Há Hª
00000000080	00	00	00	00	FA	33	C0	8E	D0	BC	00	7C	FB	68	C0	07	ú3ÀŽÐ¼ |ûhÀ
00000000096	1F	1E	68	66	00	CB	88	16	0E	00	66	81	3E	03	00	4E	hf Ë^ f > N
00000000112	54	46	53	75	15	B4	41	BB	AA	55	CD	13	72	0C	81	FB	TFSu ´A»ªUÍ r û
00000000128	55	AA	75	06	F7	C1	01	00	75	03	E9	DD	00	1E	83	EC	Uªu ÷Á u éÝ ƒì
00000000144	18	68	1A	00	B4	48	8A	16	0E	00	8B	F4	16	1F	CD	13	h ´HŠ ‹ô Í
00000000160	9F	83	C4	18	9E	58	1F	72	E1	3B	06	0B	00	75	DB	A3	ŸƒÄ žX rá; uÛ£
00000000176	0F	00	C1	2E	0F	00	04	1E	5A	33	DB	B9	00	20	2B	C8	Á. Z3Û¹ +È
00000000192	66	FF	06	11	00	03	16	0F	00	8E	C2	FF	06	16	00	E8	fÿ ŽÂÿ è
00000000208	4B	00	2B	C8	77	EF	B8	00	BB	CD	1A	66	23	C0	75	2D	K +Èwï¸ »Í f#Àu-
00000000224	66	81	FB	54	43	50	41	75	24	81	F9	02	01	72	1E	16	f ûTCPAu$ ù r
00000000240	68	07	BB	16	68	52	11	16	68	09	00	66	53	66	53	66	h » hR h fSfSf
00000000256	55	16	16	16	68	B8	01	66	61	0E	07	CD	1A	33	C0	BF	U h¸ fa Í 3À¿
00000000272	0A	13	B9	F6	0C	FC	F3	AA	E9	FE	01	90	90	66	60	1E	¹ö üóªéþ f`
00000000288	06	66	A1	11	00	66	03	06	1C	00	1E	66	68	00	00	00	f¡ f fh
00000000304	00	66	50	06	53	68	01	00	68	10	00	B4	42	8A	16	0E	fP Sh h ´BŠ
00000000320	00	16	1F	8B	F4	CD	13	66	59	5B	5A	66	59	66	59	1F	‹ôÍ fY[ZfYfY
00000000336	0F	82	16	00	66	FF	06	11	00	03	16	0F	00	8E	C2	FF	, fÿ ŽÂÿ
00000000352	0E	16	00	75	BC	07	1F	66	61	C3	A1	F6	01	E8	09	00	u¼ faÃ¡ö è
00000000368	A1	FA	01	E8	03	00	F4	EB	FD	8B	F0	AC	3C	00	74	09	¡ú è ôëý‹ð¬< t

...Look like this!

Drive C:

\

Name	Ext.	Size	Created	Modified
Path unknown				
$Extend		0.5 KB	02/11/2017 02:56:29	02/11/2017 02:56:29
$Recycle.Bin	Bin	328 B	08/22/2013 08:36:31	02/11/2017 03:06:42
(Root directory)		8.2 KB	08/22/2013 06:31:02	08/18/2017 02:13:33
Documents and Settings		60 B	08/22/2013 07:45:52	08/22/2013 07:45:52
MSOCache		256 B	04/10/2017 15:16:53	04/10/2017 15:16:53
PDFStreamDumper		8.2 KB	07/13/2017 17:12:15	07/13/2017 17:12:18
PerfLogs		48 B	08/22/2013 08:36:30	08/22/2013 08:22:35
Program Files		8.2 KB	08/22/2013 06:36:15	07/28/2017 06:01:06
Program Files (x86)		8.2 KB	08/22/2013 06:36:15	08/08/2017 01:38:54
ProgramData		4.1 KB	08/22/2013 06:36:15	08/08/2017 01:39:13
Python27		4.1 KB	08/04/2017 02:35:53	08/04/2017 02:36:12
System Volume Information		4.1 KB	02/11/2017 02:59:45	08/25/2017 04:53:34
Users		4.1 KB	08/22/2013 06:36:15	02/11/2017 03:06:20
Windows		24.1 KB	08/22/2013 06:36:15	08/18/2017 02:13:33
$AttrDef		2.5 KB	02/11/2017 02:56:29	02/11/2017 02:56:29
$BadClus		0 B	02/11/2017 02:56:29	02/11/2017 02:56:29
$Bitmap		3.1 MB	02/11/2017 02:56:29	02/11/2017 02:56:29
$Boot		8.0 KB	02/11/2017 02:56:29	02/11/2017 02:56:29
$LogFile		64.0 MB	02/11/2017 02:56:29	02/11/2017 02:56:29

Why analyzed the file system?

The Rumors Are True – Deleted Data Can Be Recovered

But How?!

Here’s a simple way of thinking about the three step process of file deletion:

X Marks the Spot

Much like URLs can be mapped to IP addresses, file names are mapped to file locations on disk (that is, the sectors that the file is stored in).

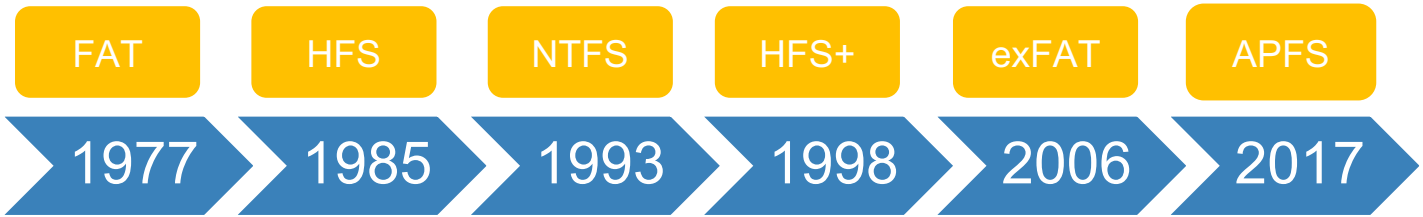
This map is created and maintained by the file system – adding new entries when files are created, updating entries as their names or locations change, and removing entries when files are deleted.

The Common Ones

There are several (if not many) file systems in use today, however a few are more common than the rest:

- File Allocation Table (FAT)
- Extended File Allocation Table (exFAT)
- New Technology File System (NTFS)
- Hierarchical File System Plus (HFS+)
- APple File System (APFS)

Just Some History



FAT File System Analysis

The FAT File System

- Why is it important to learn an out-of-date file system that hasn't been used in years.
- Simple, but Robust
- Compatible across all operating systems
- Commonly use on small storage devices

Types of FAT

While FAT32 is most commonly used today, the original FAT was 8-bit and was followed by FAT12 and a couple iterations of FAT16.

FAT12	2^{12}	4KB
FAT16	2^{16}	64KB
FAT32	2^{28}	256 KB

Cluster Size

Let’s Get Logical

While clusters and sectors have similarities in concept, in practice they are used differently.

File Allocation

When disks are formatted, the number of sectors per cluster is defined. 1:1 is the minimum, but the default would be based on the cluster size and sector size. FAT12 would default to 8:1 on a disk with a 512 byte sector size.

File Allocation

Performance is mentioned here because clusters are made up of contiguous sectors.

What the Slack?

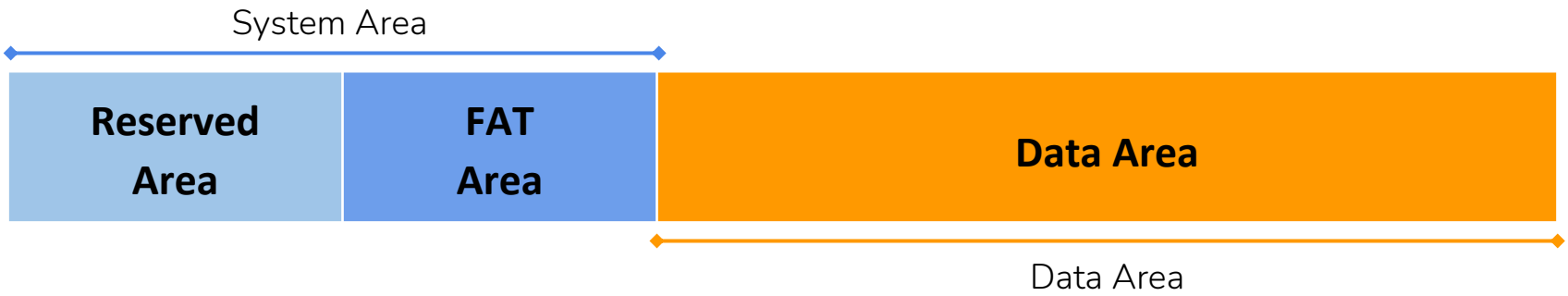
‘Slack space’ is the term used to describe the amount of space between the end of a file and the end of the cluster allocated to it.

Looking High and Low

The terms ‘high-level’ and ‘low-level’ are used when discussing cluster allocation and sector usage.

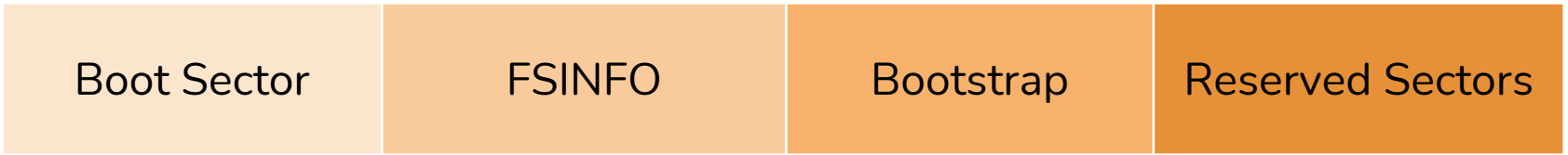
FAT Structure

FAT breaks the partition into three segments: reserved area, FAT area, and data area. The reserved and FAT areas together are the system area.



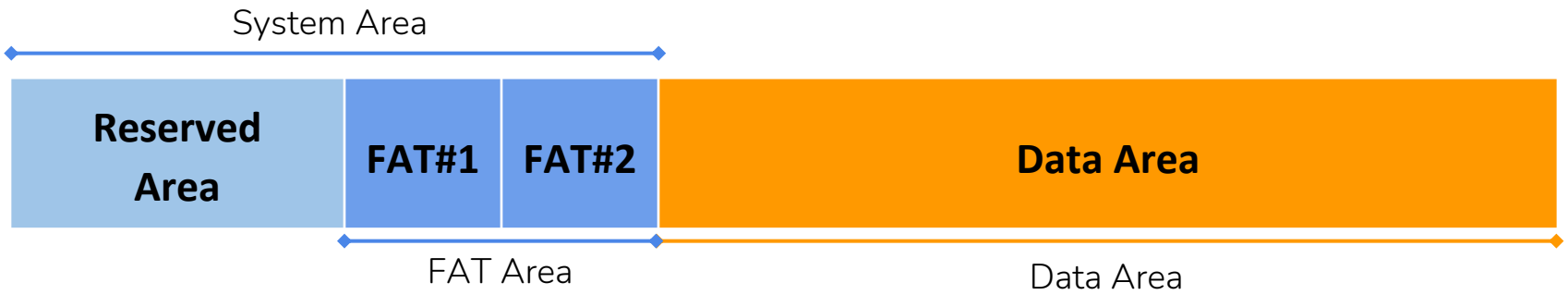
The Reserved Area

The reserved area for FAT12 and FAT16 is just a single sector called the boot sector. However, with FAT32, 32 sectors are allocated to the reserved area and are broken down as seen below.



The FAT Area

The FAT area of all types contains the File Allocation Table itself. Note, however, that FAT32 actually maintains two copies of the File Allocation Table, as seen below.



The Data Area

NTFS Analysis

The New Technology File System

NTFS was developed by Microsoft to support increasing capability requirements. In addition to supporting more files and storage capacities, NTFS added performance and security features that had not been available with FAT.

Some NTFS Features

Here's a brief list of the many features NTFS included that previous file systems hadn't had:

- Journaling
- Scalability
- Hard links
- Alternate Data Streams (ADS)
- File compression
- Sparse files
- Volume shadow copy
- Transactions
- Permissions & ACLs
- Encryption
- Quotas
- Reparse points
- Resizing

Size Limitations?

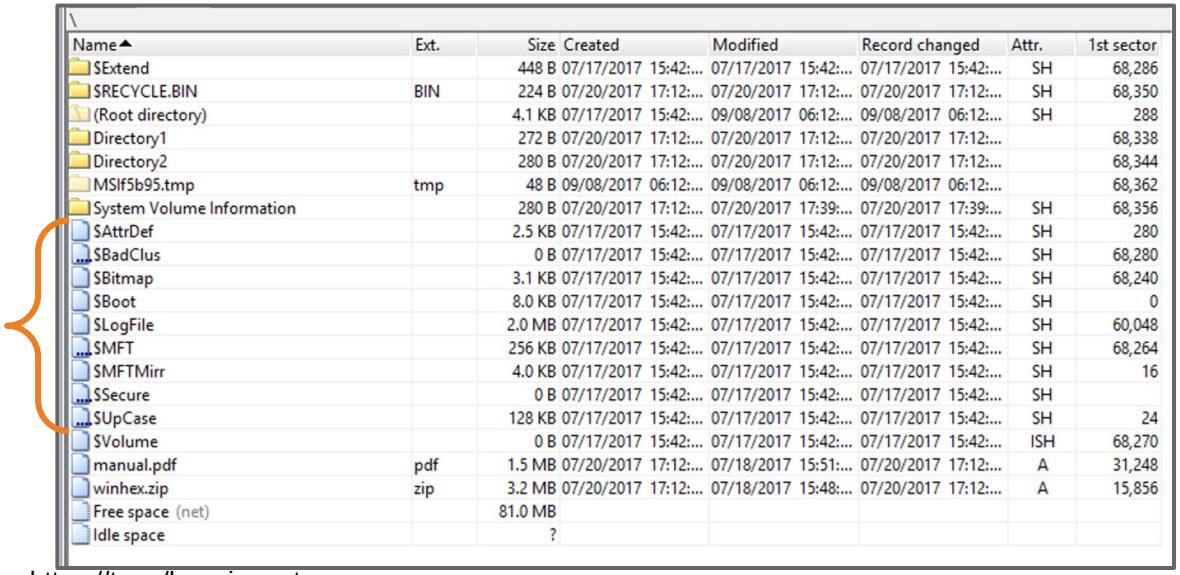
Every file system has some size limitation

All the Files, All the Time

NTFS doesn’t separate the partition by category, unlike FAT with its system and data areas.

It’s Systemic

- System files start with $



\

Name	Ext.	Size	Created	Modified	Record changed	Attr.	1st sector
$Extend		448 B	07/17/2017 15:42:...	07/17/2017 15:42:...	07/17/2017 15:42:...	SH	68,286
$RECYCLE.BIN	BIN	224 B	07/20/2017 17:12:...	07/20/2017 17:12:...	07/20/2017 17:12:...	SH	68,350
(Root directory)		4.1 KB	07/17/2017 15:42:...	09/08/2017 06:12:...	09/08/2017 06:12:...	SH	288
Directory1		272 B	07/20/2017 17:12:...	07/20/2017 17:12:...	07/20/2017 17:12:...		68,338
Directory2		280 B	07/20/2017 17:12:...	07/20/2017 17:12:...	07/20/2017 17:12:...		68,344
MSIf5b95.tmp	tmp	48 B	09/08/2017 06:12:...	09/08/2017 06:12:...	09/08/2017 06:12:...		68,362
System Volume Information		280 B	07/20/2017 17:12:...	07/20/2017 17:39:...	07/20/2017 17:39:...	SH	68,356
$AttrDef		2.5 KB	07/17/2017 15:42:...	07/17/2017 15:42:...	07/17/2017 15:42:...	SH	280
$BadClus		0 B	07/17/2017 15:42:...	07/17/2017 15:42:...	07/17/2017 15:42:...	SH	68,280
$Bitmap		3.1 KB	07/17/2017 15:42:...	07/17/2017 15:42:...	07/17/2017 15:42:...	SH	68,240
$Boot		8.0 KB	07/17/2017 15:42:...	07/17/2017 15:42:...	07/17/2017 15:42:...	SH	0
$LogFile		2.0 MB	07/17/2017 15:42:...	07/17/2017 15:42:...	07/17/2017 15:42:...	SH	60,048
$MFT		256 KB	07/17/2017 15:42:...	07/17/2017 15:42:...	07/17/2017 15:42:...	SH	68,264
$MFTMirr		4.0 KB	07/17/2017 15:42:...	07/17/2017 15:42:...	07/17/2017 15:42:...	SH	16
$Secure		0 B	07/17/2017 15:42:...	07/17/2017 15:42:...	07/17/2017 15:42:...	SH	
$UpCase		128 KB	07/17/2017 15:42:...	07/17/2017 15:42:...	07/17/2017 15:42:...	SH	24
$Volume		0 B	07/17/2017 15:42:...	07/17/2017 15:42:...	07/17/2017 15:42:...	ISH	68,270
manual.pdf	pdf	1.5 MB	07/20/2017 17:12:...	07/18/2017 15:51:...	07/20/2017 17:12:...	A	31,248
winhex.zip	zip	3.2 MB	07/20/2017 17:12:...	07/18/2017 15:48:...	07/20/2017 17:12:...	A	15,856
Free space (net)		81.0 MB					
Idle space		?					

Master File Table

- The primary system file
- MFT can be compared to the File Allocation Tables

In the Zone...
...The MFT Zone, that is.

Since the MFT is the go-to resource for determining what files live where, it's best if it doesn't get fragmented. To prevent this, NTFS drives reserve at least 12.5% of the total volume storage for the MFT to grow into. That percentage can be increased if needed. This space is called the MFT Zone.

Primary System Files

System File	File Name	MFT Record #	Purpose
Master file table	$MFT	0	Holds a record for each file and directory on the volume.
Master file table mirror	$MFTMirror	1	For recovery in case of MFT failure.
Log file	$LogFile	2	Holds information for file system metadata changes and helps with recovery.
Volume	$Volume	3	Contains information about the volume and its label.
Attribute definition	$AttrDef	4	Holds information about all attributes used within the file system.
Root file name index	.	5	The root directory of the volume.
Cluster bitmap	$Bitmap	6	Tracks free, unused clusters within the volume.
Boot sector	$Boot	7	Mounts the volume and other bootstrap code when the volume is bootable.
Bad cluster file	$BadClus	8	Tracks bad clusters within the volume.
Security file	$Secure	9	Stores the security descriptors for all files in the volume.
Upcase table	$Upcase	10	Converts lowercase characters to the matching Unicode uppercase.
NTFS extension directory	$Extended	11	Holds optional and extended features.
		12 - 15	Reserved for future use

NTFS Volume Structure

NTFS doesn’t preemptively segment the volume like FAT does, and the MFT and MFTMirror files can live anywhere on the volume.

Volume Boot Record Structure

Here's a quick look into the structure of the VBR.

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td colspan="3">Jump Instruction</td><td colspan="8">OEM ID</td><td colspan="2">Bytes per sector</td><td>Sectors Per Cluster</td><td colspan="2">Reserved No. of Sectors</td></tr>
<tr><td colspan="5">Unused</td><td>Media Descr.</td><td colspan="2">Must be zero</td><td colspan="8">Unused</td></tr>
<tr><td colspan="8">Unused</td><td colspan="8">Total number of sectors</td></tr>
<tr><td colspan="8">Logical cluster number for the $MFT file</td><td colspan="8">Logical cluster number for the $MFTMirror file</td></tr>
<tr><td>Clusters per MFT Record</td><td colspan="3">Unused</td><td>Clusters per Index</td><td colspan="3">Unused</td><td colspan="8">Volume Serial Number</td></tr>
<tr><td colspan="4">Unused</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Keys to the Kingdom

The most important offsets for our purposes are usually the ones that tell us where the MFT files are.

MFT Entries

Every file in NTFS is given at least one MFT entry, which follows a structure similar to what is seen here.

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td colspan="4">Entry Signature</td><td colspan="2">Offset of the Fixup Array</td><td colspan="2">Entries in Fixup Array</td><td colspan="8">$LogFile Sequence Number (LSN)</td></tr>
<tr><td colspan="2">Sequence Number</td><td colspan="2">Hard Link Count</td><td colspan="2">Offset of the First Attribute</td><td colspan="2">Flags</td><td colspan="4">Real size of the MFT Entry</td><td colspan="4">Allocated size of the MFT Entry</td></tr>
<tr><td colspan="8">File reference to the base MFT Entry</td><td colspan="2">Next Attribute ID</td><td colspan="2">Alignment to 4 Byte Boundary</td><td colspan="4">Number of this MFT Entry</td></tr>
</table>

Of particular note is the entry signature. This will be FILE (or 0x46494C45) unless the file is unusable, in which case it'll be BAAD (or 0x42414144).

MFT Entry Attributes

Each entry within the MFT can contain a number of different pieces of information, called attributes.

One thing to know about attributes is that they can be either ‘resident’ or ‘nonresident’.

MFT Entry Attributes

Attribute Type	Name	Purpose
Standard Information	$STANDARD_INFORMATION	Stores information regarding access mode (eg, read-only, read/write), timestamp, and link count.
Attribute List	$ATTRIBUTE_LIST	Holds the location of other attributes that were not able to fit in a single entry.
File Name	$FILE_NAME	Stores the file names. There will be more than one if a long file name or POSIX file name is used.
Data	$DATA	Stores file data. NTFS allows a single file to have more than one $DATA attribute.
Object ID	$OBJECT_ID	A unique file identifier across the whole volume.
Reparse Point	$REPARSE_POINT	Used for mounting drives.
Index Root	$INDEX_ROOT	Implementation of directories and other indexes.
Index Allocation	$INDEX_ALLOCATION	Implementation of B-tree structure for large directories and large indexes.
Bitmap	$BITMAP	Represents the status of an entity.
Volume Information	$VOLUME_INFORMATION	Only found in $Volume system file, and includes volume version.
Volume Name	$VOLUME_NAME	Contains the name of the volume.
Security Descriptor	$SECURITY_DESCRIPTOR	Stores file security information (e.g. Access Control Lists).

File Carving

What is File Carving?

File carving

The process of extracting files that are not accessible to a user.

When is it necessary?

File carving can be used whenever the file system cannot provide access to the file.

Types of File Carving

Carving for files can be summarized as four different types:

- File Header Carving
- Header – Footer Carving
- File Structure Carving
- Metadata File Carving

Back to Basics

File carving is simplest to perform when the files or clusters of interest are:

A. Not compressed
B. Not overwritten
C. Consecutive / Contiguous

The Sleuth Kit

The Sleuth Kit (TSK)

The Sleuth Kit is a set of free and open-source CLI-based forensic tools.

Scripts on Scripts

TSK provides about 25 CLI tools across a variety of use cases. These include:

- Media or volume: mmls, mmstat, and mmcat
- Image file: img_stat and img_cat
- File system: fsstat
- File name: fls and ffind
- Metadata: icat, ils, istat, and ifind
- Content: blkcat, blkls, blkstat, and blkcalc
- File system journal: jcat and jls
- Miscellaneous: hfind, mactime, sorter, sigfind
- Automation: tsk_comparedir, tsk_gettimes, tsk_loaddb, tsk_recover

TSK Tip

You may have noticed that the commands follow a memorable naming convention – i.e. starting with fs for ‘file system,’ mm for ‘media,’ and j for ‘journal.’

After that, you see familiar terms like ‘ls,’ ‘cat,’ and ‘stat’.

This can help to keep track of which command is used for what.

File Identification & Structure

Starting from Scratch

The simplest way to determine a file's type is by its extension.

Extension Mismatch

When a file extension is changed to something incompatible, the default application will fail to open it.

What’s a File Signature?

File signatures – which can be headers or a combination of headers+footers – are specific hexadecimal strings used by the application to read the file correctly.

Finally, some consistency!

- File structures remain the same across operating systems.
- Allows files to be shared across operating systems.

Temporary File Overview

What the Temp?

- Temporary files created by an application or OS for a short period of time.
- Temp files can be used for many things
- The OS uses temp files swap data between the file system and RAM.

Forensic Usefulness

+ May be available for analysis at the time of acquisition
+ Many temp files during normal usage
+ Forensic tools support reading and indexing temp files

Module Review

This module covered:

- Definition & History
- FAT File System Analysis
- NTFS File System Analysis
- File Carving
- The Sleuth Kit (TSK)
- File Identification & Structure
- Temporary File Overview

Now you are able to:

- Understand the role of file systems in modern computing,
- Define common file system structures, and
- Perform basic file system analysis.

Course Conclusion

Course Layout:

- Persistent Storage Principles
  - Module Introduction
  - Definition & History
  - Persistent Storage Devices
  - Volumes & Partitions
  - Disk Partitioning
- Fundamentals of File Systems
  - Module Introduction
  - Definition & History
  - FAT File System Analysis
  - NTFS File System Analysis
  - File Carving
  - The Sleuth Kit (TSK)
  - File Identification & Structure
  - Temporary File Overview

Now you are able to:

- Understand the physical and logical technologies used today,
- Define the relationship between storage media and file systems, and
- Perform disk and file system analysis.