

# File system forensics domain

# File System Forensics

- **this domain examines the organization and structure of file systems to retrieve hidden or deleted data**
- **key points**
  - Slack Space Analysis
    - unused portions of disk sectors might contain residual data.
  - File Carving
    - deleted files extraction by analyzing residual data clusters
  - "Registry" Analysis, OS config analysis
    - recovering system/user activities (Windows Registry, /etc , ~/.config, ~/.bashrc, ... )
- **relevant Tools**
  - Low-level (*stat, istat, debugfs*)
  - High-level (*FTK Imager and Autopsy*)

# File System

- **a file is the smallest logical unit from an user perspective that can be stored**
  - can be stored in bytes, lines, records, ...
- **logical files are mapped into “physical” entities (computer RAM, HD, the Cloud, ...) by the Operating System**
  - memory addresses, disk sectors, remote resources, ...
- **file systems define the organization and structure of files on a computing device**
- **through the OS, a file system define the rules to read, write and maintain the data**

# File Attributes

## ■ name

- a mnemonic (human) id for reference
- DOS legacy: 8 char+3char for the extension (no modern OS still have this limitation, but some names still have)

## ■ type

- categorize the file to indicate how should be manipulated
- “magic number” (few bytes) usually at the file start
  - Note: WinOS often rely on extension to trigger execution/associate applications

# File Attributes

- **protection**

- access control information
- differ depending on OS/FS combination
  - e.g. owner and group (unix-like)
  - read, write, execute (unix-like)

- **location**

- **size**

- **and some other**

# File system formatting

- **the operation that prepare a mass storage for data storage, configuring it with specific file system structures**
  - erases existing information
    - full formatting (slow – erases all sectors with zeros/mark bad ones)
    - quick formatting (eras file system tables)
  - creates one/many partitions
    - each partition can be formatted separately (different logical volumes – primary, extended, logical)
  - selects specific file systems (e.g. NTFS for windows, APFS for MacOSX, ext4 for Linux)
  - creates foundational structures
    - Root directory, FAT/Inode table, superblock/Boot sector

# FAT example

- **when formatting the HD**

- the Boot Record is created
  - OS name and version
  - disk physical characteristics (bytes per sectors, sectors per cluster, root directory entry)
- the Master File Table is created (2 times!)
  - info on clusters (available, allocated, damaged, containing OS files)
- the Directory table is created
  - top level folder file and directory information

# FAT example

- **File Allocation Table**

- i.e. where the OS records the files' position

- **FAT close to the begin of the volume**

- specified in the boot sector
  - 2 copies (optional, for redundancy)

Boot sector	Reserved	FAT 1	FAT 2	Root Folder	Other folders and files
-------------	----------	-------	-------	-------------	-------------------------



# FAT example

## directory entry

<b>file.dat</b>	<b>other ... other</b>	<b>29</b>	<b>3450</b>
<b>file name</b>	<b>other attributes (e.g. RO, hidden)</b>	<b>file start cluster number</b>	<b>file size</b>

## File allocation table (subset)

...	00	00	EOF	30	33	32	EOF	EOF	00
25	26	27	28	29	30	31	32	33	34

```
graph TD; D[Directory Entry] -- "29" --> F29[FAT[29] = 30]; F29 -- "30" --> F30[FAT[30] = 33]; F30 -- "33" --> F33[FAT[33] = EOF];
```

# FAT pros and cons

- **portability** (available to multiple operating systems)
- **migration** (easy switch to richer FS, like NTFS)
- **fast on small volumes** (some GB), due to light "infrastructure" (few metadata, no file index, ...)
- **no integrated advanced features** (on-the-fly compression, user quotas)
- **slow for high number of files** (linked-list structure, fragmentation, no)
- **no security at all** (encryption, access control lists)

# FAT references

- <https://download.microsoft.com/download/1/6/1/161ba512-40e2-4cc9-843a-923143f3456c/fatgen103.doc>
  - [Microsoft Extensible Firmware Initiative FAT32 File System Specification - FAT: General Overview of On-Disk Format](#)
- <https://forensics.wiki/fat/>
  - e.g. recovery in FAT accomplished looking for entries that begin with a sigma 0xe5

# File copy

- **normal commands preserve the file content...**
- **...but alter the file attributes (meta-data)**
  - e.g. creation data
- **required bit-per-bit copy to avoid any modification**
  - “data dump” (*dd*) copy/convert bit-per-bit
    - *dd if=<inputfile> <of=outputfile>*
  - variants (e.g. *dcfldd*, *dc3dd*) with CF added features
    - e.g. on the fly hashing (md5, sha-1, sha-256, and sha-512), pattern wiping, progress report

# File copy

- **clone one hard drive onto another:**
  - *dd if=/dev/sda of=/dev/sdb*
- **clone a hard drive to an image file**
  - *dd if=/dev/hda of=/image.img*
- **clone a hard drive to a zip image in 100Mb blocks**
  - *dd if=/dev/hda bs=100M | gzip -c > /image.img*
- **wipe an hard drive with binary 0s**
  - *dcfldd pattern=00 vf=/dev/hdb*
- **write a binary image and calculate hash**
  - *dc3dd if=/var/log/messages of=/tmp/dc3dd hash=sha512*

# File identification

- **extension is not a reliable source**
  - literally anyone can change them
- **check the metadata**
  - where available
- **check first bytes of the file can act as signature**
  - [https://en.wikipedia.org/wiki/List\\_of\\_file\\_signatures](https://en.wikipedia.org/wiki/List_of_file_signatures)
  - compare the signature with hex dump of the file

# Meta-data example: file system

- **the file system maintains a number of information about file contents**
  - file name, ownership and permission
  - specific data-units allocated to file
  - size of the file
  - time stamps for the file
  - recovery data (e.g. journaling)
  - ...
- **information type and accuracy can vary (very much) depending on file system**
  - FAT32, NTFS, ext2, ext3, ext4, ...

# Slack space (I)

- **data starting from the end of the file written and up to the end of the sectors designated to the file**
- **i.e. leftover space when a file do not fill exactly a sector multiple size**
  - the difference between logical (bytes) and physical (sectors) file size
- **sectors have fixed dimension**
  - e.g. 512 bytes
- **file do not have such fixed dimension**
  - e.g. 392 bytes
  - this file will result in 120 bytes of slack space



# Slack space (II)

- **Slack space is a "dynamic" entity**

- **Example:**

- file1 : 392 Bytes, sector : 512 Bytes, file2 : 192 Bytes
- file1 is created
- ...(other operations on the FS)
- file1 is erased
- what's the final result?
  - (ans: the final 200 bytes of the original file will survive in the FS)

# Recovery process

- **Analysis of the file system structures**

- e.g. Master File Table (MFT) to store metadata for all files (like type of data and file names)
- presence of metadata, such as timestamps, enables creating timelines, essential in criminal cases to determine user actions and verify evidence integrity
- OSs organizes metadata into system files that assist with permissions, integrity, and file tracking, supporting security and data reliability crucial in legal contexts.

- Data can be recovered by "deleted"/orphan files/file signatures

# File analysis - Meta-data

- **data about... data**
  - e.g. EXIF, ODF, DOCX, ...
- **useful to augment knowledge on file**
- **useful to reveal hidden information**
- **useful to correlate various data**
- **meta-data are not “trusted-by-definition”**

# Meta-data example: ODF

- Open document format (ODF) is an OpenOffice.org file format
  - license-free
  - open
  - XML based
- if decompressed (zip file) meta-data are in a separate file (*meta.xml*)

```
<dc:title>Pippo.odt document</dc:title>
<dc:subject>A test of metadata OpenOffice
management</dc:subject>
<meta:initial-creator>Sh</meta:initial-creator>
<meta:creation-date>2009-09-26T11:22:39</meta:creation-date>
<dc:creator>Sh</dc:creator>
```

*meta.xml sample*

# Meta-data example: JPEG

- the most popular format is the Exchangeable Image File (EXIF) one

**Make NIKON CORPORATION**  
**Model NIKON D80**  
**Xresolution 300.00**  
**Yresolution 300.00**  
**ResolutionUnit Inch**  
**Software Ver.1.01**  
**ExifVersion Exif Version 2.21**  
**FocalLengthIn35mmFilm 27**  
**ISOSpeedRatings 125**  
**Orientation top – left**  
**DateTimeOriginal 2009:02:17 12:13:09**  
**PixelXDimension 3872**  
**PixelYDimensio 2592**  
**...**

# exiftool

- **platform independent PERL library**
- **command line tool**
  - *exiftool /media/sdb1/IMG\_1.jpg*
- **read, write, edit meta information**
  - in many different formats
  - for many different products
    - <https://exiftool.org/#supported>

# exiftool support

← → ↺ exiftool.org/#supported

File Type	Support	Description	EXIF	IPTC	XMP	ICC <sup>1</sup>	C2PA	Other
IDML	R	Adobe InDesign Markup Language (ZIP/XML-based)	-	-	-	-	-	R XML ZIP
IIO	R/W	Phase One Intelligent Image Quality RAW (TIFF-based)	R/W/C	R/W/C	R/W/C	R/W/C	R/D	R/W PhaseOne
IND, INDD, INDT	R/W	Adobe InDesign Document/Template	-	-	R/W/C	-	-	-
INSP	R/W	Insta360 Picture (JPEG-based)	R/W/C	R/W/C	R/W/C	R/W/C	R/D	Supported JPEG Meta Information
INSV	R	Insta360 Video (QuickTime-based)	-	-	R	-	R	R QuickTime
INX	R	Adobe InDesign Interchange (XML-based)	-	-	R	-	-	-
ISO	R	ISO 9660 disk image	-	-	-	-	-	R ISO
ITC	R	iTunes Cover Flow artwork	-	-	-	-	-	R ITC
J2C, J2K, JPC	R	JPEG 2000 codestream	R <sup>3</sup>	R <sup>3</sup>	R	R	-	R Jpeg2000 Photoshop <sup>3</sup>
JP2, JPF, JPM, JPX, JPH	R/W	JPEG 2000 image [Compound/Extended/High-throughput]	R/W/C <sup>3</sup>	R/W/C <sup>3</sup>	R/W/C	R	-	R/W/C Jpeg2000, R Photoshop <sup>3</sup>
JPEG, JPG, JPE	R/W	Joint Photographic Experts Group image	R/W/C	R/W/C	R/W/C	R/W/C	R/D	Supported JPEG Meta Information
JSON	R	JavaScript Object Notation	-	-	-	-	-	R JSON
JXL	R/W	JPEG XL (codestream and ISO BMFF) (Jpeg2000-based)	R/W/C	-	R/W/C	-	-	-
K25	R	Kodak DC25 RAW (TIFF-based)	R	R	R	R	R	-
KDC	R	Kodak Digital Camera RAW (TIFF-based)	R	R	R	R	R	R Kodak
KEY, KTH	R	Apple iWork '09 Keynote presentation/Theme	-	-	-	-	-	R XML ZIP
LA	R	Lossless Audio (RIFF-based)	R <sup>3</sup>	-	R	-	R	R RIFF
LFP, LFR	R	Lytro Light Field Picture	-	-	-	-	-	R Lytro
LIF	R	Leica Image File	-	-	-	-	-	R LIF
LNK	R	Microsoft Shell Link (Windows shortcut)	-	-	-	-	-	R LNK
LRV	R/W	Low-Resolution Video (QuickTime-based)	R/W <sup>3</sup>	R/W <sup>3</sup>	R/W/C	-	-	R/W/C QuickTime
M2TS, MTS, M2T, TS	R	MPEG-2 Transport Stream (used for AVCHD video)	-	-	-	-	-	R M2TS H264 MISB
M4A, M4B, M4P, M4V	R/W	MPEG-4 Audio/Video (QuickTime-based)	R/W <sup>3</sup>	R/W <sup>3</sup>	R/W/C	-	R/D	R/W/C QuickTime
MACOS	R	MacOS ". " sidcar file (may have any extension)	-	-	-	-	-	R XAttr RSRG
MAX	R	3D Studio MAX (EPX-like)	-	-	R	R	-	R FlashPix
MEF	R/W	Mamiya (RAW) Electronic Format (TIFF-based)	R/W/C	R/W/C	R/W/C	R/W/C	R/D	-
MIE	R/W/C	Meta Information Encapsulation (MIE specification)	R/W/C	R/W/C	R/W/C	R/W/C	R/D	R/W/C MIE
MIFF, MIF	R	Magick Image File Format	R	R	R	R	-	R MIFF Photoshop
MKA, MKV, MKS	R	Matroska Audio/Video/Subtitle	-	-	-	-	-	R Matroska
MOBI, AZW, AZW3	R	Mobipocket electronic book (Palm-based)	-	-	-	-	-	R Palm MOBI
MODD	R	Sony Picture Motion metadata (XML PLIST-based)	-	-	-	-	-	R PLIST
MOI	R	MOD Information file	-	-	-	-	-	R MOI
MOS	R/W	Creo Leaf Mosaic (TIFF-based)	R/W/C	R/W/C	R/W/C	R/W/C	R/D	R Leaf
MOV, QT	R/W	Apple QuickTime Movie	R/W <sup>3</sup>	R/W <sup>3</sup>	R/W/C	-	R/D	R/W/C QuickTime
MP3	R	MPEG-1 layer 3 audio	-	-	-	-	R	R MPEG ID3 Lyrics3 APE
MP4	R/W	Motion Picture Experts Group version 4 (QuickTime-based)	R/W <sup>3</sup>	R/W <sup>3</sup>	R/W/C	-	R/D	R/W/C QuickTime
MPC	R	Musepack Audio	-	-	-	-	R	R MPC ID3 Lyrics3 APE
MPEG, MPG, M2V	R	Motion Picture Experts Group version 1 or 2	-	-	-	-	R	R MPEG ID3 Lyrics3
MPO	R/W	Extended Multi-Picture format (JPEG with MPF extensions)	R/W/C	R/W/C	R/W/C	R/W/C	R/D	Supported JPEG Meta Information
MQV	R/W	Sony Mobile QuickTime Video	R/W <sup>3</sup>	R/W <sup>3</sup>	R/W/C	-	R/D	R/W/C QuickTime
MRW	R/W	Minolta RAW	R/W/C	R/W/C	R/W/C	R/W/C	R/D	R/W MinoltaRaw Minolta
MRC	R	Medical Research Council	-	-	-	-	-	R MRC
MXF	R	Material Exchange Format	-	-	-	-	-	R MXF
NEF	R/W	Nikon (RAW) Electronic Format (TIFF-based)	R/W/C	R/W/C	R/W/C	R/W/C	R/D	R/W Nikon NikonCapture
NKA	R	Nikon NX Studio Adjustments	-	-	-	-	-	R XML
NKSC	R/W	Nikon Sidecar (XMP-based)	-	-	R/W/C	-	-	-
NMBTEMPLATE	R	Apple iWork '09 Numbers Template	-	-	-	-	-	R XML ZIP
NRW	R/W	Nikon RAW (2) (TIFF-based)	R/W/C	R/W/C	R/W/C	R/W/C	R/D	R/W Nikon NikonCapture
NXD	R	Nikon Capture NX-D adjustments (XMP-based)	-	-	R	-	-	-
NUMBERS	R	Apple iWork '09 Numbers spreadsheet	-	-	-	-	-	R XML ZIP
O	R	Unix compiled code Object	-	-	-	-	-	R EXE
ODB, ODC, ODF, ODD, ODI, ODP, ODS, ODT	R	Open Document Database/Chart/Formula/Graphics/Image/Presentation/Spreadsheet/Text (ZIP/XML-based)	-	-	-	-	-	R XML ZIP
OF8	R	OptimFROG audio (RIFF-based)	R <sup>3</sup>	-	R	-	R	R RIFF
OGG, OGV	R	Ogg bitstream container	-	-	-	-	R	R FLAC ID3 Lyrics3 Theora Vorbis

# hexdump, hexedit, ghex

- **hexdump: command line utility to display a file in a specified format (default hex)**
  - *hexdump <filename>*
- **hexedit: hexadecimal command line file viewer and editor**
  - *hexedit <filename>*
- **ghex: graphic (GNOME) hexadecimal editor**
  - allows user to load data from any file, view and edit it either in hex or ASCII format
  - *ghex <filename>*



# Foremost

- a command line tool that “curves” data from disk images...
- ...i.e. inspect content of an image looking for erased files
- Identify file types on the base of file signature and metadata
  - *foremost -t jpg,gif -l image.dd -o outdir*
    - look for .jpg and .gif data
    - in the image.dd disk image
    - and put the recovered files in outdir

# Photorec

- **Recovering deleted files from a variety of storage media, including hard drives, CD-ROMs, USB flash drives, and memory cards**
  - bypass the file system to scan directly on the storage medium
    - very effective on damaged/formatted storage
  - uses file carving/signature matching techniques (based on header and footer patterns)
  - e.g.
    - *Sudo photorec*
    - choose storage device (like /dev/sda, or HDD image file)
    - choose recovery operation

# File delete

- **overwriting all allocated clusters can be very time consuming...**
- **OSs just delete the location of the file fragments for performance reason**
- **recently deleted files can be recovered partially or even completely**
- **plus, some parts may live for years thanks to slack space**

# Data sanitization tools

## ■ File Shredder Programs

- permanently delete selected files
- overwrite using a specified data sanitization method
- ensures they cannot be undeleted

## ■ Data Destruction Software

- completely erase (delete) all data on a HDD
- one or more data sanitization methods to permanently overwrite all the information
- suitable for virus removal, HDD disposal or recycling

# Data sanitization procedures

## ■ **AFSSI-5020**

- defined in the USAF's Air Force System Security Instruction 5020
- three overwrites of data
  - first Pass: writes a '0'
  - second Pass: writes a '1'
  - third Pass: writes a random character and verifies the write.