Computational Forensics

File Carving

COMP 8920 - Computer and Network Forensics

Lesson 04

Outlines

- Introduction
- Technical Background
- File Carving Methods
- Challenges
- File Carving Tools

Introduction to File Carving

Definition

What is Carving?

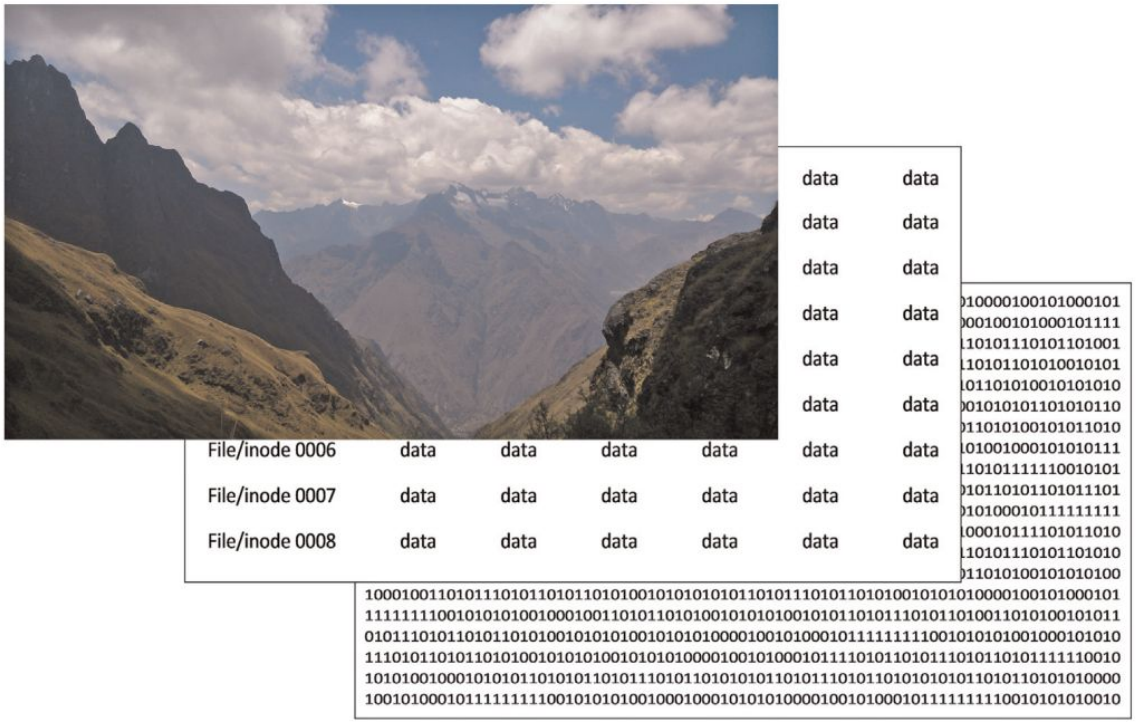
Carving is a general term for extracting structured data (e.g files) out of raw data, based on format-specific characteristics present in the structured data or previous knowledge of the structured data contents.

Done on a disk when the unallocated file system space is analyzed to extract files because data cannot be identified due to missing of allocation info, or on network captures where files are "carved" from the dumped traffic.

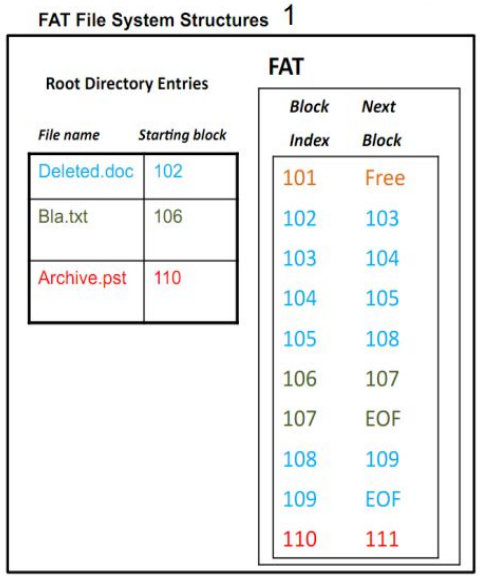
Why file carving?

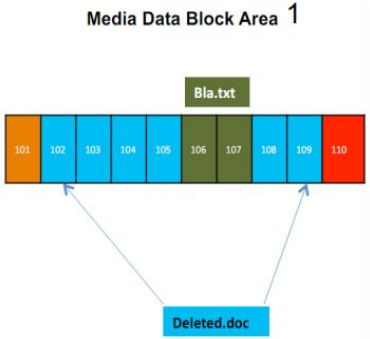
1. Identify and recover files of interest from raw, deleted or damaged file systems, memory, or swap space data.
2. Criminals usually try to hide digital evidence by deleting these evidence.
3. Assist in recovering files and data that may not be accounted for by the operating system and file system.
4. What the user see != OS see != storage media see

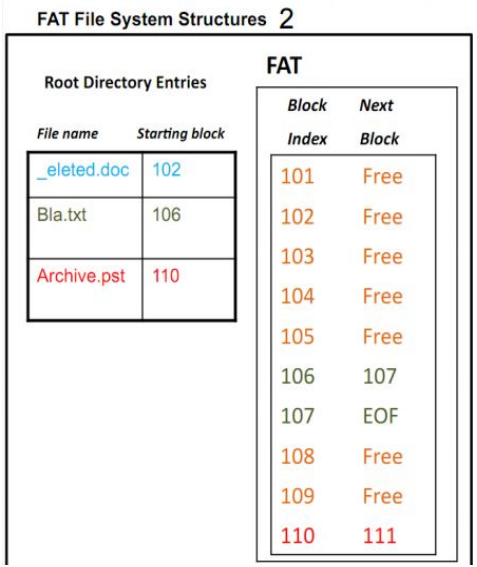
Image as seen by users, by OS, and in hardware

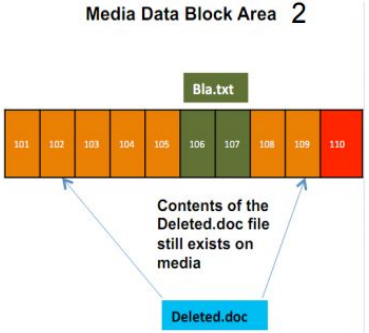


Delete Files









File Carving

Disk Structure

A disk is a device used to store and retrieve data readable by a computer system. The methods of storing the data on a medium are typically

1. **magnetic** – as with a hard disk, tape drive, or floppy disk
2. **optic** – as with a CD or DVD;
3. **electronically** programmable memory – such as with flash drives or memory cards.

Disk Structure

Disk Structure & File Carving

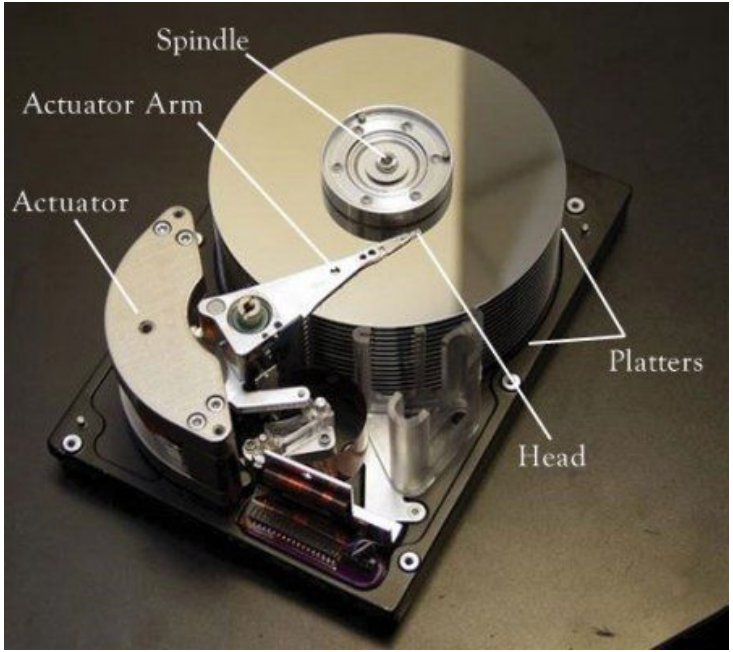
- File carving require good understanding of file systems and file structure.
- Understanding how a file system is stored on a disk is important to understand how file systems work.
- It is important to have a working knowledge of older technology because the examiner may come into contact with older technology for a variety of reasons.

Physical Disk Structures

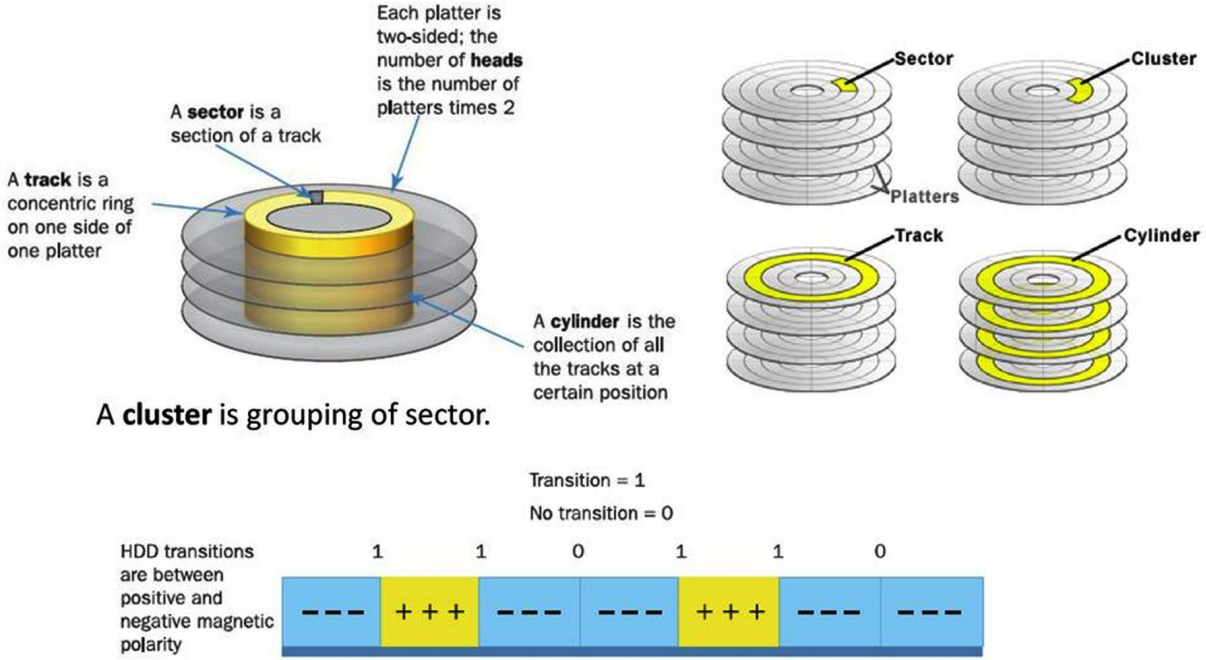
The physical disk structure includes the metal housing, **platters**, **read-write heads**, motor, and electronics.

Data **write/read operations** are through an **interface** that can be interpreted by the computer system.

Common interfaces include small computer system interface (SCSI), serial attached SCSI (SAS), integrated drive electronics (IDE), and serial AT attachment (SATA).



Physical Disk Structures



Magnetic disk drives historically stored data in units known as cylinders, heads, and sectors. These units were then used to assign address to each storage block on a disk.

Sector

- Sectors (or blocks, depending on the file system) are the smallest unit of data a file system can write to.
- Sectors on most drives were 512 bytes, at a strictly physical level, there are additional bytes in use by the drive's electronics that may include metadata such as checksums and block (or sector) number.

Cluster

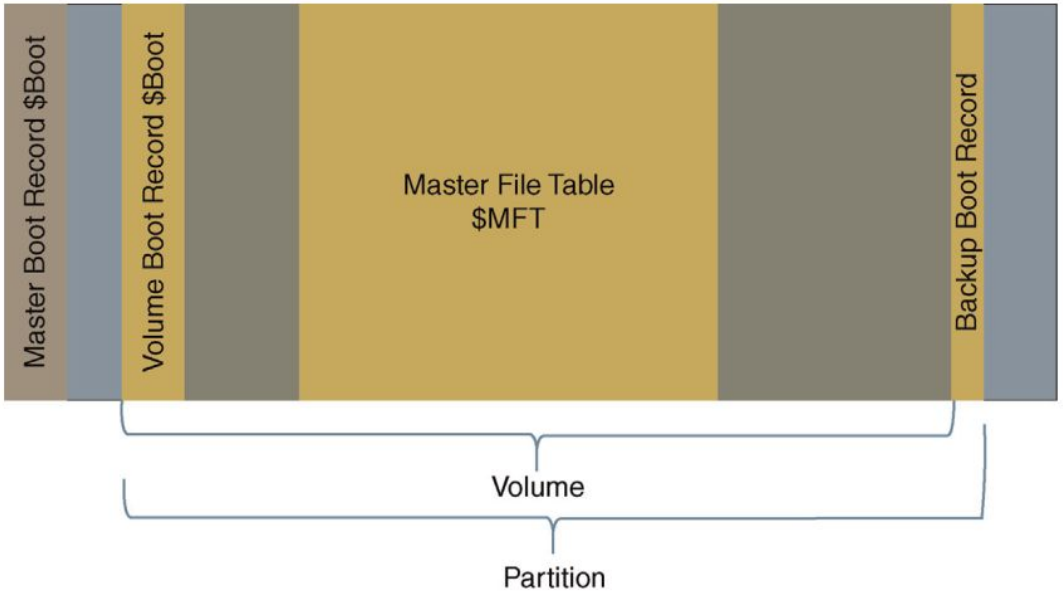
A cluster is the smallest addressable unit on a file system.

During drive format, cluster size is allocated as a multiple of sectors.

On most modern Windows systems with NTFS (New Technology File System) formatting, one cluster (by default) is often composed of eight sectors (4096 bytes, or 4 Kb).

Logical Disk Structures

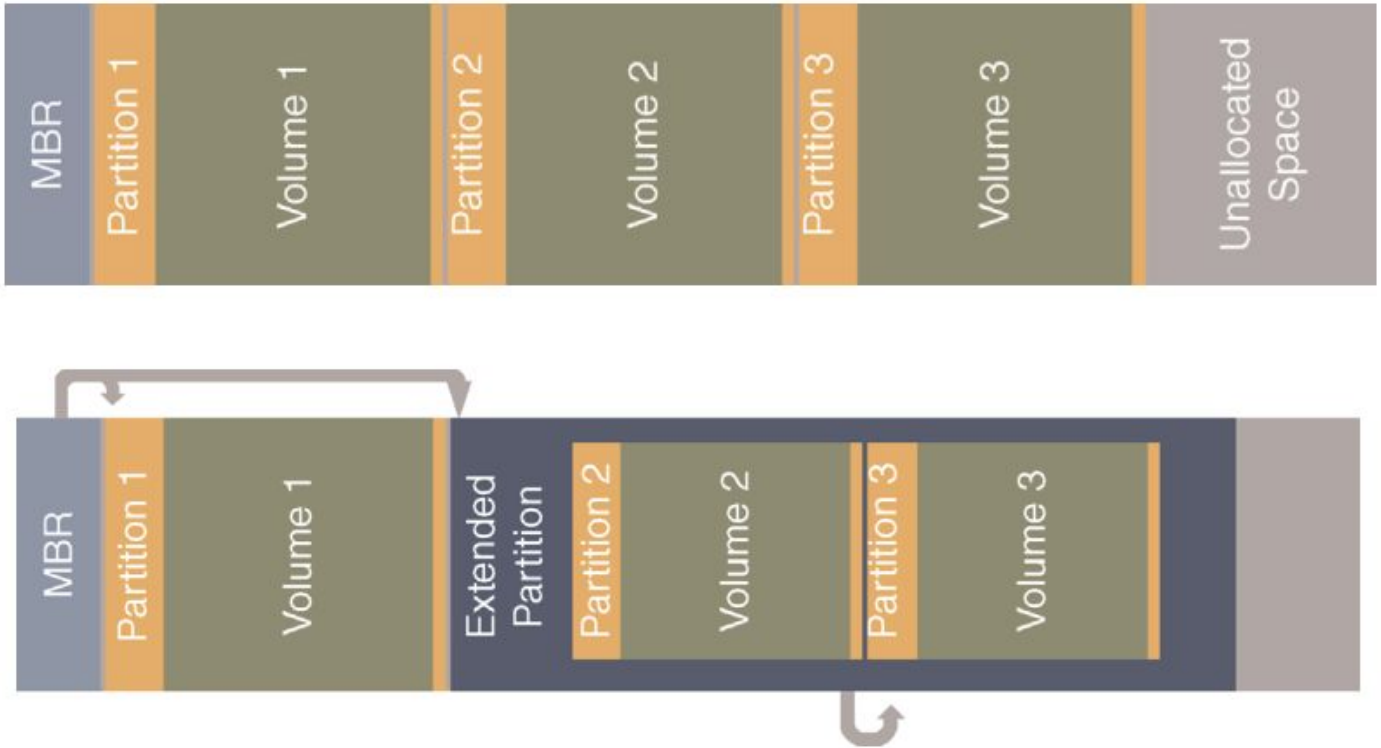
Logical structures on a physical disk are required for the OS to access the storage space on the physical device.



Logical Disk Structures

- A MBR can contain up to four partitions in the boot record all of which can be primary and only one of which can be an extended partition.
- A primary partition can contain one and only one volume
- Only one of these volumes can be bootable (an extended partition cannot be natively bootable).
- An extended partition can contain additional logical partitions.

Logical Disk Structures



File System

- A file system stores data on a device so data can be retrieved by the system or a user.
- File systems are largely independent of an operating system (OS), and different file systems can be supported on different OSs when necessary drivers are installed.
- The terms operating system and file system are often used interchangeably, but this is not technically correct.

File System

- Structure for storing and organizing computer files and the data they contain to make it easy to access and find them.
- Organize disk sectors (typically 512 bytes each) into files and directories and keep track of which sectors belong to which file (allocated) and which are not being used (unallocated).
- Common file systems are: **FAT** (File Allocation Table) / **NTFS** (New Technology File System) on Windows Systems and **UFS/JFS** on Unix Systems.

Common File Systems

Operating system	Native file systems
Windows 98	FAT16, FAT32
Windows 2000	FAT16, FAT32, NTFS
Windows XP	FAT16, FAT32, NTFS
Windows Vista	FAT16, FAT32, NTFS, exFAT (with SP1 and later)
Windows 7	FAT16, FAT32, NTFS, exFAT
Windows 8	FAT16, FAT32, NTFS, exFAT
Windows 10	FAT16, FAT32, NTFS, exFAT
Linux	EXT2, EXT3, EXT4, XFS
Mac OSX	HPFS, exFAT

File System Fragmentation

File system fragmentation occurs when data is not contiguously stored, due to:

1. Low free space;
2. Deletion, truncation or extension of files

Example of fragmentation is the slack space

Given a cluster size of 4 sectors and, what do you think will happen if we want to store a 1 byte file on this cluster.

Computer File

- **File**: term used to indicate a block of stored information (binary digits), e.g., a document in a doc file, an image in a jpg file or a program in an exe file
- Can be **created**, **moved**, **modified**, **copied** and **deleted**, through computer programs
  - use extensions in file names to help identify what they contain (the file type)
- But the user of a computer can also manipulate files if necessary

File Metadata

Metadata is data about data. Metadata can be used to track timestamps, location of data, the exposure setting on a digital image, or any number of arbitrary items that allow a user or the system to locate, sort, or collate data.

File Metadata: List of possible attributes

- Defined in $AttrDef entry of Master File Table (MFT):
  - 0x10 STANDARD_INFORMATION
  - 0x20 $ATTRIBUTE_LIST
  - 0x30 $FILE_NAME0
  - 0x40 (NT) $VOLUME_VERSION (2K) $OBJECT_ID
  - 0x50 $SECURITY_DESCRIPTOR
  - 0x60 $VOLUME_NAME
  - 0x70 $VOLUME_INFORMATION
  - 0x80 $DATA
  - 0x90 $INDEX_ROOT
  - 0xA0 $INDEX_ALLOCATION
  - 0xB0 $BITMAP
  - 0xC0 (NT) $SYMBOLIC_LINK, (2K) $REPARSE_POINT
  - 0xD0 $EA_INFORMATION
  - 0xE0 $EA0xF0NT$PROPERTY_SET
  - 0x100 (2K) $LOGGED_UTILITY_STREAM

File Metadata: Date-Time Stamps Significance

- Usually shows when a file or folder was created
- When an existing file is copied, the File Created date-time stamp of the new copy is set to the current time
- When a file is moved onto a different volume using the Windows command line or drag-and-drop feature, the File Created date-time stamp of the new copy is set to the current time.
- Represents the last time any attribute in the MFT record for the file or folder was modified.

File Metadata: Date-Time Stamps Significance

- When a file is moved onto a different volume using the Cut and Paste menu options, the File Created date-time stamp remains unchanged (the Last Accessed and Entry Modified date-time stamps would most likely change).
- Represents the last time the $DATA attribute of a file was altered.
- Represents the most recent time a file or folder was accessed by the file system.
- Does not necessarily indicate that a file was opened.
- Represents the last time any attribute in the MFT record for the file or folder was modified.

File Signature

- **Magic number:** more reliable way to recognize a file consists of analyzing its structure rather than its extension. Its constant used to identify a file format
- Provides a simple way of distinguishing between file formats
- Rely on the fact that every file has an header and a footer in order to get correctly recognized
- **Note:** File signatures can be changed, resulting in a fake file type recognition

File Signature: Magic Number

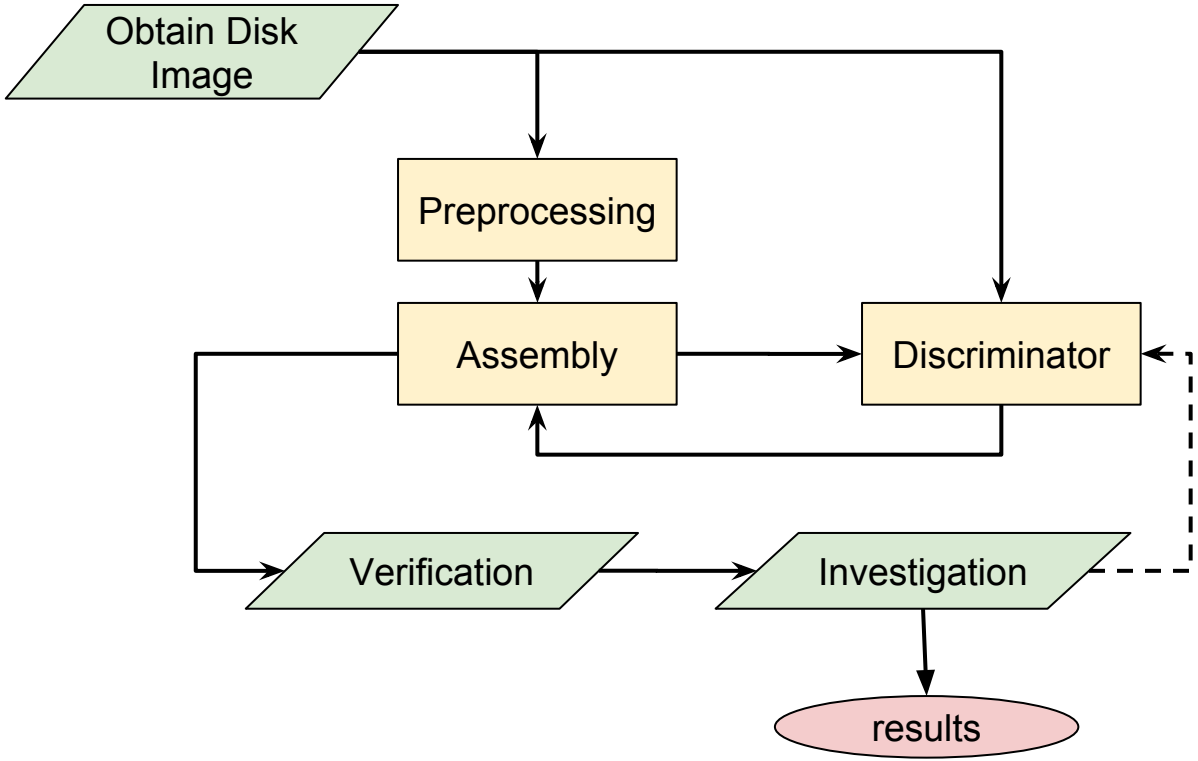
Description	Extension	Magic Number
Adobe Illustrator	.ai	25 50 44 46 [%PDF]
Bitmap graphic	.bmp	42 4D [BM]
Class File	.class	CA FE BA BE
JPEG graphic file	.jpg	FFD8
JPEG 2000 graphic file	.jp2	0000000C6A5020200D0A [....jP..]
GIF graphic file	.gif	47 49 46 38 [GIF89]
TIF graphic file	.tif	49 49 [II]
PNG graphic file	.png	89 50 4E 47 .PNG
WAV audio file	.png	52 49 46 46 RIFF
ELF Linux EXE	.png	7F 45 4C 46 .ELF
Photoshop Graphics	.psd	38 42 50 53 [8BPS]
Windows Meta File	.wmf	D7 CD C6 9A
MIDI file	.mid	4D 54 68 64 [MThd]
Icon file	.ico	00 00 01 00

Complete list of common files: https://asecuritysite.com/forensics/magic

File Carving Techniques

- File Structure Based Carving
  - Uses the internal layout of a file
  - Internal layout is based on elements such as the **header**, **footer**, identifier strings and size information etc.
  - Known carvers which use this technique are **Scalpel** and **PhotoRec**
- Content-based Carving
  - Content structure (XML, HTML, etc)
  - Content characteristics (.Java, .cpp, .py)

The File Carving Process



The File Carving Process

Preprocessing: Extracting information about the file

- Identify file type; identify start and end/length if possible
- Select all sectors which potentially could be part of the file

Assembly: Generate a potential version of the file

- Decide which sectors to include
- Concatenate these sectors in a "sensible" manner
- According to various strategies and based on various data
- Try "best" files first to reduce scope of searching

The File Carving Process

Discriminator: Check whether the result could be correct

- Can this file be "decompressed" (viewed) or does it make "sense"?
- Where in the file is the erroneous position?
- Some parts belonging at an absolute position?
- Usually based on known file viewers or printers

Basic Structure File Carving

Basic Techniques Assume that :

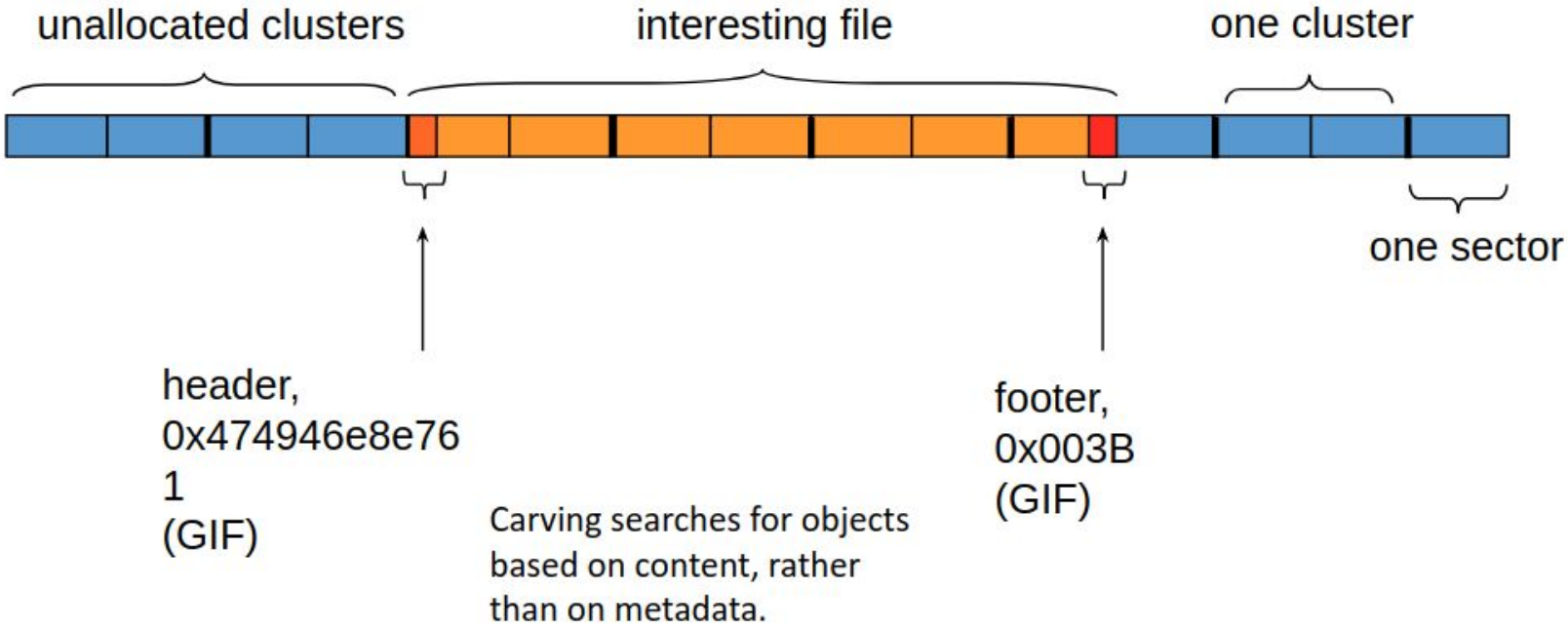
1. The beginning of file is not overwritten
2. The file is not fragmented.
3. The file is not compressed.
4. The file is not encrypted

Basically this type of carving is made with **header** and **footer**.

Structure-based File Carving

- Rely on the header and footer analysis => does not consider the file’s content, which means that sectors inserted, deleted or modified are not considered
- Operate by looking for file headers and footers, and then "carving out" the blocks between these two boundaries.
- By using a database of headers and footers for specific file types, file carver can retrieve files from raw disk images, even if the file system metadata has been destroyed.

Structure File Carving



File Structure Based Carving

- Many carving programs have an option to only look at or near sector or cluster boundaries where headers are found.
- File start is always at a sector boundary, but end is not
- However, searching the entire input can find files that have been embedded into other files, such as JPEGs being embedded into words documents.

File Structure Based Carving

- The majority of file carving programs will only recover files that are contiguous on the media (in other words: files that are not fragmented)
- Files may be incomplete – start, end, middle sectors may have been reused.
- Examples a pdf file starts with “**%PDF**” and ends with “**%EOF**” and a jpeg image file begins with “**0xFFD8**” and ends with “**0xFFD9**”.

Example Carving JPEG files

- JPEG files start with 0xFFD8 and end with 0xFFD9
- To recover a JPEG file
  - Find locations of its header and footer
  - Carve out everything between those two endpoints
  - If no end marker exists: Specify a maximum length.

Hexdump of sample.jpg

```
ff d8 ff e0 00 10 4a 46  49 46 00 01 01 01 00 50  |......JFIF.....P|
... Data ...
28 a2 80 3f ff d9                                 |(..?..|
```

File Structure Based Carving

- Works well only for non-fragmented files.
- One possible improvement is to exclude all sectors in use by other real (allocated) files and previously extracted files.
- The result is hard to verify automatically
- Does not require sophisticated anti-forensics techniques.

Content Based File Carving

Carving of files based on their semantic content (e.g. txt, html, doc, java, c++, etc)

Mainly searching the disk image in several stages

1. Identify all potential sectors
2. Detect language of the file
3. Hierarchy check
4. Boundary check

Content Based File Carving

Identify all potential sectors:

- Recognizing text, programs, etc.
- Programming languages: Idioms (for loop etc.), reserved words
- Natural languages: Check for spaces, letters, non-letters

Detect language of the file:

- Programming language or natural language?
- Natural language: Using ”stop word lists” is fast and easy!
- Programming language: Reserved words, regular expressions » Example C: include "[a-zA-Z\-_0-9]*.h"\n

Content Based File Carving

Hierarchy check

- Nesting for programming languages (indentation) and html files (unopened/unclosed tags)
- Allows excluding certain sequence

Boundary check:

- Check if the first/last word a complete word or only a fragment

Challenges in File Carving

- File carving has a time complexity of NP-complete
- Cannot be solved and verified quickly (time consuming)
- You must try all possible **combinations** of fragments/sectors
- Many unreadable invalid and partial results
- May result in more data as output than input
- Quality of the **tools** are unclear
- File systems become larger (TB disks are inexpensive)

File Carving Tools

- Open Source:
  - **Foremost** - Developed by Jesse Kornblum and Kris Kendall at AFOSI
  - **Scalpel** - Improved version of Foremost, by Golden G. Richard III
  - **CarvFS** - Virtual file system for carving
  - **PhotoRec** - Recovers lost photos from hard drives
- Commercial:
  - **Adroit Photo Recovery** — Amazing, but only works on JPEGs
  - **EnCase** - comes with some eScripts that will carve
  - **DataLifter** - File Extractor Pro

Summary

In this class we covered:

- An introduction to file carving

Note: Make sure to read the reading materials posted on the course website.

What is Next?

In our next Class we continue talking about computational forensics topics and anti-forensics techniques.

References

1. Smith, Jay, Monroe, Klayton, Bair, Andy: Digital Forensics File Carving Advances
2. File carving Forensics Lecture Notes, by Dr. Issa Traore,