Digital Forensics

File System Forensics Masterclass

Fraser Brown

Heriot-Watt University

March 21, 2018

What is this Masterclass?

This masterclass aims to give an overview of digital forensics and provide practical experience through a lecture and lab combination.

The materials will be structured as follows:

1. Lecture is here (ROOM) [TIME]
2. Break [TIME]
3. Lab (EM2.50) [TIME]

Lab Overview I

The Scenario:
You are digital forensics investigators working on a data information leakage case for a client.

Your hypothetical team has given you a USB stick recovered from the person suspected of leaking information. You will analyse the file system and try to prove if they have leaked information.

- Set up a Lab Environment using Kali Linux
- Create/Gather a Forensic image of the USB in question.
- Perform automated data carving
- Gather file system information.
- Keyword searching of unallocated space & extraction of related (deleted) files.
- Recover and confirm evidence using metadata pointers.

Lab Overview II

Reserouce Credits:
The USB resources we will be using in today's lab are from the *National Institute of Standards and Technology (NIST)* as part of their *Computer Forensic Reference Data Sets (CFReDS)*[1].

The rest of the resources and a can be found here:
`https://www.cfreds.nist.gov/data_leakage_case/data-leakage-case.html`

[1] `https://www.cfreds.nist.gov/`

Lecture Overview

1. What is Digital Forensics?
2. Forensic Process
3. Forensic Imaging
4. File Systems
5. Forensic Tools
6. File System Analysis Example
7. Additional Resources
8. Careers

What is Digital Forensics?

Digital Forensics:

"Computer [Digital] Forensics is the practice of *determining the past actions that have taken place on a computer system* using forensic techniques and understanding artefacts." - David Cowen

Artefact:

"An Artefact is a *reproducible* file, setting or system change that occurs every time an application or operating system performs a specific action" - David Cowen

The artefacts we will be dealing with in the lab are files and file systems.

Why File System Analysis? I

- There are many different forms of digital forensic analysis:
  - Network Analysis,
  - Live memory (RAM) Analysis,
  - File system analysis,
  - Database Analysis,
  - Application/OS Analysis
- File system analysis allows:
  - Introduction to a new field (digital forensics).
  - A different perspective on computer science knowledge we already have.
  - Insight into how files relate to memory, and what creation and deletion features actual do.

Why File System Analysis? II

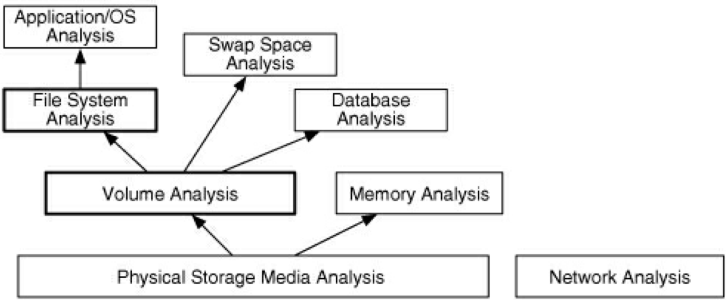


Figure: Layers of Analysis[2]

[2] Carrier, Brian. File System Forensic Analysis (Kindle Location 619). Pearson Education. Kindle Edition.

Forensic Process

Forensic Process

Digital forensics results can be used in a court of law therefore **accuracy**, **integrity** and an **unbiased approach** towards evidence is required. Traditional approaches towards evidence handling and procedure are shared between traditional and digital forensics.

Scientific Method

Defining a hypothesis based on evidence then proceeding search for evidence which disproves our hypothesis.

Digital Forensic Investigation

"A digital forensic investigation is a process that *uses science and technology* to *analyse digital objects* and that **develops and tests theories,** which can be entered into a court of law, to *answer questions about events that occurred.*" - Brian Carrier[a]

[a] Carrier, Brian. File System Forensic Analysis (Kindle Locations 480-481). Pearson Education. Kindle Edition.

Digital Crime Scene Investigation Process Overview

There are three major areas in digital crime scene investigations:

- System Preservation
  - Write Blockers,
  - Cryptographic Hashes,
  - Proper Evidence Handling,
  - Photographic Evidence of a crime scene/workstation.
- Evidence Searching
  - Data Carving,
  - Metadata information gathering.
  - Deleted file recovery.
- Event Reconstruction
  - Time line creation
  - Gathering/providing reliably re-creatable evidence.

PICL Guidelines

While each forensic investigator/team may have their own procedures and work flow the PICL guidelines below provide a good staring structure:

- **Preservation**:
  Preservation of the system being investigated.
- **Isolation**:
  Keeping analysis environment is separate from both the suspect data and the outside world.
- **Correlation**:
  Correlate data with other independent sources. Reduces risk of forged data.
- **Logging**:
  Log/document your actions. This helps identify what searches you have not yet conducted and what your results were.

Analysis Types

Live Analysis:

“A live analysis occurs when you use the operating system or other resources of the system being investigated to find evidence.” - Brian Carrier[a]

[a] Carrier, Brian. File System Forensic Analysis (Kindle Locations 495-496). Pearson Education. Kindle Edition.

Dead Analysis:

“A dead analysis occurs when you are running trusted applications in a trusted operating system to find evidence.” - Brian Carrier[a]

[a] Carrier, Brian. File System Forensic Analysis (Kindle Locations 496-497). Pearson Education. Kindle Edition.

Forensic Imaging

Evidence Acquisition/Imaging

- In order to perform analysis on digital artefacts a forensic **duplicate** of the media must be created.
- *Forensic Duplicates* are **bit-for-bit copies of the original disk** and can encompass the full disk or a single partition.
- This process is known as **imaging** or **acquisition**.
- Contents of a disk are always changing therefore **Write Blockers** are used to preserve the disk state.
- **Cryptographic hash functions** such as `SHA-256, SHA-1, MD5` are used to **verify** the image against the original artefact.

Image Types

- Raw Format (`.dd .raw .img`)
  - Only contain data from the original artifact
  - Metadata on image is not included however can be generated into a separate file by tools. (guymager, TSK)
  - Tools: `dd`, `dcfldd`, `dd_rescue`, `rdd`, `df3dd`, `guymager`
- EnCase Evidence Format (Expert Witness `.E01`)
  - Expert Witness images use headers and footers to hold metadata about the image.
  - Metatdata can include: drive type, source disk OS, timestamps, hashes, CRCs over blocks.

Write Blockers

Write Blockers

Are hardware or software devices that allow gathering of information without damaging the disk contents by blocking write commands but allowing read commands.

- Write Blockers are customisable:
  - Blocking of all or specific commands.
  - Can control the read and write speed.
- Write Blockers come in two forms:
  - *Native*: Same interface for input and output e.g. IDE-to-IDE
  - *Tailgate*: uses different interfaces for input and output e.g. firewire/USB-to-SATA

Imaging Challenges with Solid State Drives (SSD)

While an SSD can be imaged with the same tools as a traditional hard disk drive (HDD), there are technology specific issues that cause problems for forensic investigators.

- **Program-Erase cycles**
  - Sequence of events that result in data being written to a solid state flash memory cell, then erased and rewritten (e.g flash memory USB sticks).
  - These P/E cycles result in a *small amount of physical damage to the medium*, which can result in **bad sectors**.
- **Wear Levelling**
  - Prolongs the life of solid state/flash memory.
  - Distributes rewrites evenly across the medium, so no single block dies prematurely.
- These two technologies due to the evolution of memory results in unallocated space being overwritten earlier than it would on a HDD. This could overwrite valuable hidden information by accident

File Systems

What are File Systems? I

File System:

File systems manage how data is **stored** and **retrieved** in a computer system. They consist of **structural** and **user data** which can be organised and understood by users and computers.

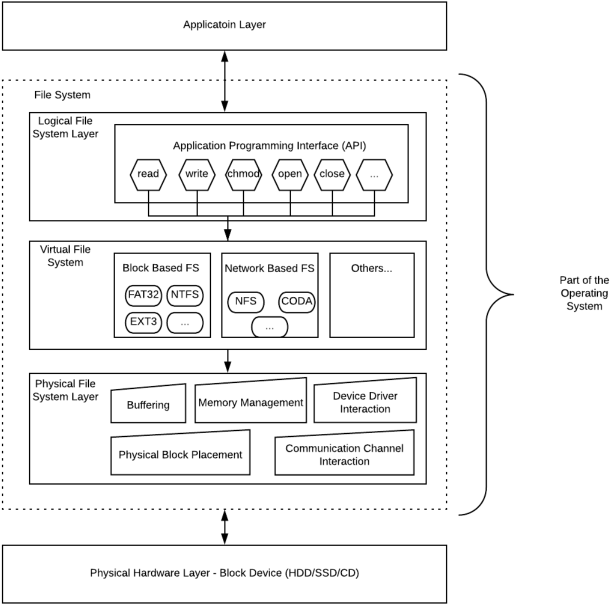
- File system architectures (FAT32, Ext3 etc.) provide *different methods of tracking data on physical media*, each has their own data structures and look up tables.
- Modern operating systems *contain support for many different file systems* providing an interface with physical storage, defining things like allocation methods.

What are File Systems? II

File systems are made up of 2/3 layers:

1. **Logical Layer**: Provides a user application level API for commands such as read, write and chmod etc.
2. **Virtual Layer** (*optional*): Allows access to multiple physical file systems e.g. block based: FAT32, NTFS or network based: NFS
3. **Physical Layer**: Interacts with hardware, performing block and memory management and interacting with device drivers etc.

What are File Systems? III


Applicatoin Layer
File System
Logical File System Layer
Application Programming Interface (API)
read
write
chmod
open
close
...
Virtual File System
Block Based FS
FAT32
NTFS
EXT3
...
Network Based FS
NFS
CODA
...
Others...
Physical File System Layer
Buffering
Memory Management
Device Driver Interaction
Physical Block Placement
Communication Channel Interaction
Part of the Operating System
Physical Hardware Layer - Block Device (HDD/SSD/CD)

File Systems Terminology

- **Sector**:
  Smallest addressable section of memory, which holds static amount of data (512/2048/4096-bytes)
- **INode**:
  Data structure in a file system that contains meta data (a.k.a Meta Data Pointers/Structures)
- **Data Unit**:
  Standard sized container for storing *content* data,which consists of multiple **sectors**. Different file systems have different names for these data units e.g. (Cluster or Block).

File System Data Categories

All data in a file system belongs to a data category listed below. This abstract view will be useful for analysing different elements of a file system for evidence and for visualising structures that make up a file on disk.

- **File System**:
  Contains file system structure overview and where to find other structures and important data.
- **Content**:
  Contains data relating to actual file contents, these are usual organised into containers called data units (block/cluster).
- **Meta data**:
  Data that describes files such as access times, file size, users.
- **File Name**:
  Contains the data that assigns a name to a file, is used by users instead of a meta data address.
- **Application**:
  Special features/additional functionality such as quota data or journalling.

File System Categories Interaction

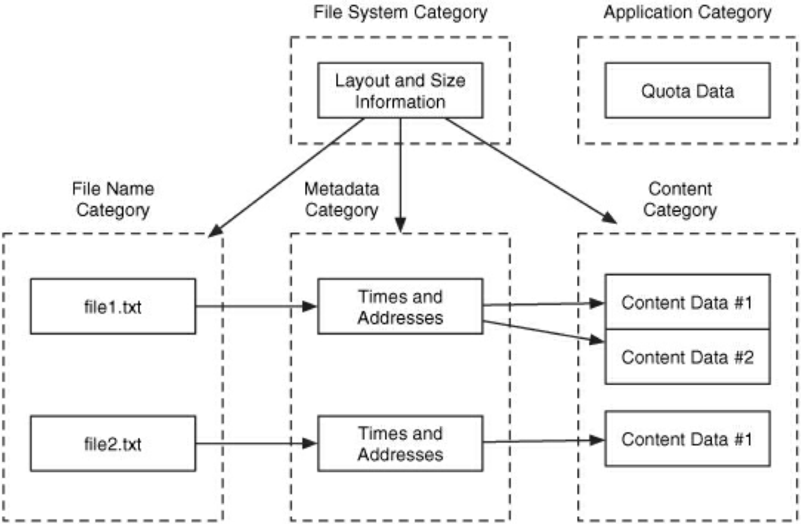
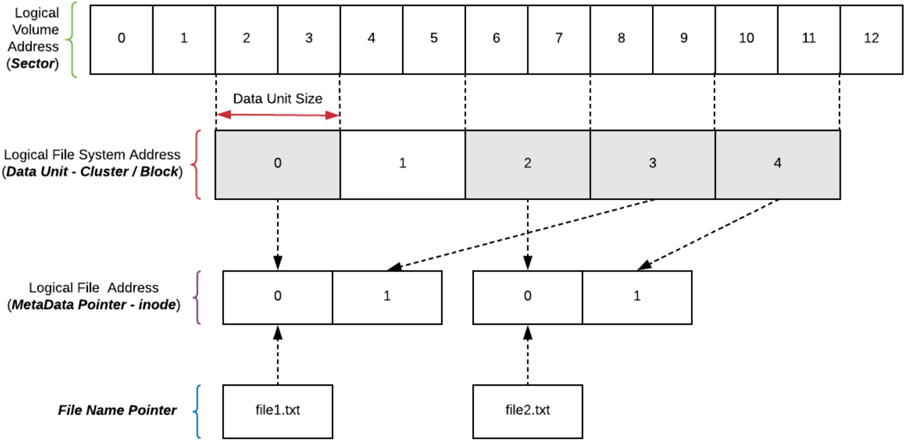


Figure: File System Categories Interaction

File System Categories By Example



Logical Volume Address (***Sector***)

0 1 2 3 4 5 6 7 8 9 10 11 12

Data Unit Size

Logical File System Address (***Data Unit - Cluster / Block***)

0 1 2 3 4

Logical File Address (***MetaData Pointer - inode***)

0 1 0 1

File Name Pointer

file1.txt

file2.txt

Figure: File System Example

File System Architectures

There are numerous file system architectures available, some are operating system specific others are designed to be more universal.

Examples include:

- **FAT** - File Allocation Table (FAT8/16/32) - *Commonly found on Removable Media*
- **NTFS** - New Technology File System - *Default For Windows*
- **Ext** - Extended File System (Ext2/3/4) - *Default for Linux*

We will focus on FAT32 in this lecture due to the lab being structured around applying forensic techniques on FAT32 formatted removable media.

File Allocation Table (FAT) Hisotry

- File Allocation Table (FAT) file system is a simplistic file system developed for Windows DOS and Windows 9* operating systems.
- FAT is supported by both windows and Unix OS's.
- FAT was replaced by NTFS as the primary file system of windows in XP era, FAT is now mainly used in SD Cards and USB flash drives.
- FAT has been extended from the original 8 bit to 32 bit and had derivatives made from it such as exFAT and FAT+

FAT32 - Information I

- Data Units are called **Clusters**
- Increased cluster size over FAT8/FAT16 architectures, due to being stored as 32bit values.
- FAT32 has two overarching data structures:
  - **File Allocaiton Table**:
    Stores next cluster for a given file, holds allocation status of clusters.
  - **Directory Entries**:
    One per file/directory which contains: file name , size, starting address of content and other metadata.
- The *first cluster address* starts at 2.
- **Calculate Sector Address** of Cluster C:
  $(C - 2) \times (NumberSectorsPerCluster) + (SectorOfCluster2)$
- *Calculate cluster address* from sector address S:
  $((SSectorOfCluster2)/(NumberSectorsPerClusterr)) + 2$

FAT32 - Information II

Physical Layout of a FAT FS:

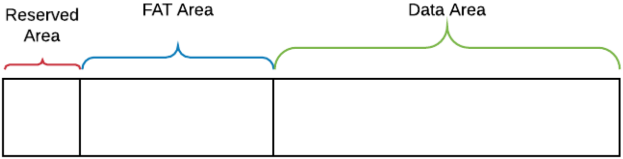


Figure: FAT Physical Layout

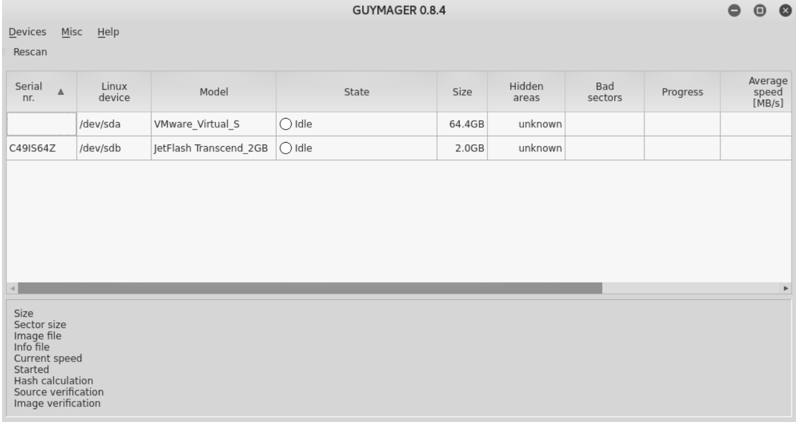
Forensic Tools

(Used in Accompanying Lab)

Guymager - Forensic Imaging

Guymager

Guymager is a GUI based forensic imaging tool, that allows for the creation of various image types such as Raw (dd) and EnCase (E01).



Serial nr. ▲	Linux device	Model	State	Size	Hidden areas	Bad sectors	Progress	Average speed [MB/s]
	/dev/sda	VMware_Virtual_S	○ Idle	64.4GB	unknown			
C49IS64Z	/dev/sdb	JetFlash Transcend_2GB	○ Idle	2.0GB	unknown			

Size
Sector size
Image file
Info file
Current speed
Started
Hash calculation
Source verification
Image verification

Figure: *guymager gui window example*

Foremost - Data Carving

Foremost

Foremost is a command line tool that utilises data carving techniques to recover files.

- **Data Carving** is a process where files are recovered from a disk image based on common information such as file headers, footers and data structures.
- Performing data carving for large forensic images can be rather tedious if done by hand, tools such as **Foremost** have been developed to help digital forensic investigators automating this process.

The Sleuth Kit (TSK)[3]

The Sleuth Kit (TSK)

The Sleuth Kit is a series of command line tools that allow users to inspect and analyse disk images and the file systems therein.

- The tools provided in TSK are divided into the 5 file system categories discussed previously, file system, content (data unit), meta data, file name.
- Due to the wide variety of tools within TSK I will discuss those of which we will use in the accompanying lab.
- Other features of TSK can befound in the tool overview: `http://wiki.sleuthkit.org/index.php?title=TSK_Tool_Overview`

[3] http://www.sleuthkit.org/

File System Analysis

(By Practical Example)

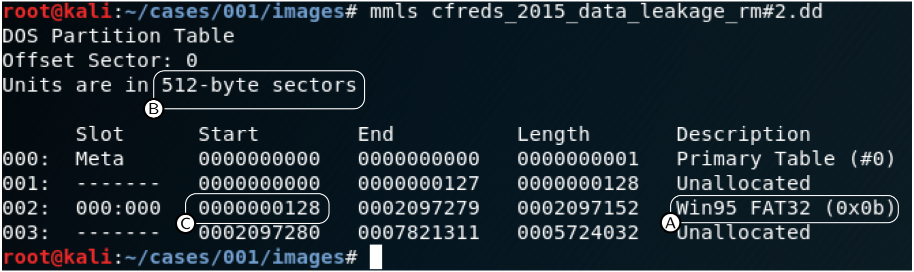
Note About File System Analysis Techniques

There are many different analysis techniques for each of the aforementioned file system data categories. In the respect of time and scope of the lecture I will only cover the ones that are relevant to the lab materials.

You may also notice that sometimes the same analysis technique (logical file system searching and viewing) are doable from different categories in a file system. This added redundancy means effective analysis can be performed even if certain data structures don't exists or were removed.

Initial Volume / Disk Image Enumeration - mmls

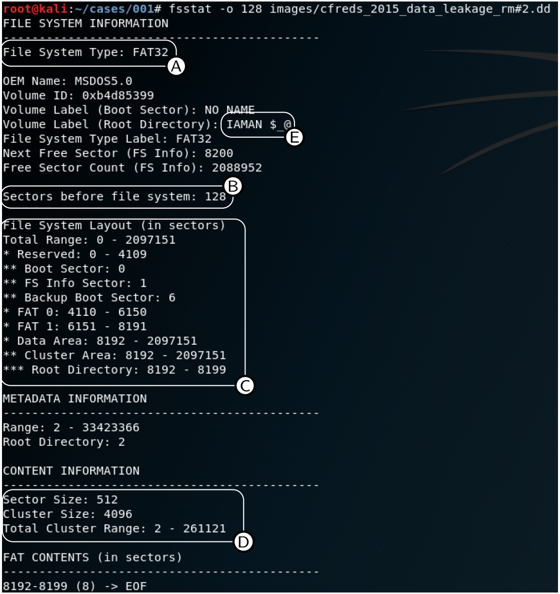
Before we can get into recovering files or gathering evidence we have to understand the problem space we are working in. TSK provides volume system level tools to aid with this. mmls allows us to display the layout of a disk image/volume.



```
root@kali:~/cases/001/images# mmls cfreds_2015_data_leakage_rm#2.dd
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000000127   0000000128   Unallocated
002:  000:000   0000000128   0002097279   0002097152   Win95 FAT32 (0x0b)
003:  -------   0002097280   0007821311   0005724032   Unallocated
root@kali:~/cases/001/images# 
```

Figure: **Initial Volume / Disk Image Enumeration** – File System Type (A), Unit type output is in (B) Starting Sector of File System (C).

File System Information Gathering - `fsstat`



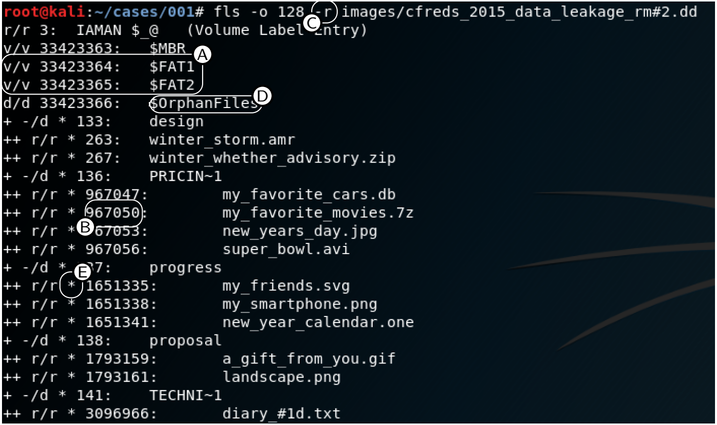
```
root@kali:~/cases/001# fsstat -o 128 images/cfreds_2015_data_leakage_rm#2.dd
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: FAT32

OEM Name: MSDOS5.0
Volume ID: 0xb4d85399
Volume Label (Boot Sector): NO NAME
Volume Label (Root Directory): IAMAN $_@
File System Type Label: FAT32
Next Free Sector (FS Info): 8200
Free Sector Count (FS Info): 2088952

Sectors before file system: 128

File System Layout (in sectors)
Total Range: 0 - 2097151
* Reserved: 0 - 4109
** Boot Sector: 0
** FS Info Sector: 1
** Backup Boot Sector: 6
* FAT 0: 4110 - 6150
* FAT 1: 6151 - 8191
* Data Area: 8192 - 2097151
** Cluster Area: 8192 - 2097151
*** Root Directory: 8192 - 8199

METADATA INFORMATION
--------------------------------------------
Range: 2 - 33423366
Root Directory: 2

CONTENT INFORMATION
--------------------------------------------
Sector Size: 512
Cluster Size: 4096
Total Cluster Range: 2 - 261121

FAT CONTENTS (in sectors)
--------------------------------------------
8192-8199 (8) -> EOF
```

Using TSK's `fsstat` command with the offset of the file system (-o 128) we are able to view information from three of our FS categories (*file system, metadata, and content*).

Figure: **File System Info –**
(A) File System Type,
(B) FS starting Sector,
(C) FS Layout,
(D) Sector & Cluster Address sizes and Cluster Range.
(E) Root Directory of Volume, name of user.

File Name Information Gathering - `fls`

We can extract all the file name pointers, and associated *inode* addresses for files and directories in our disk image.

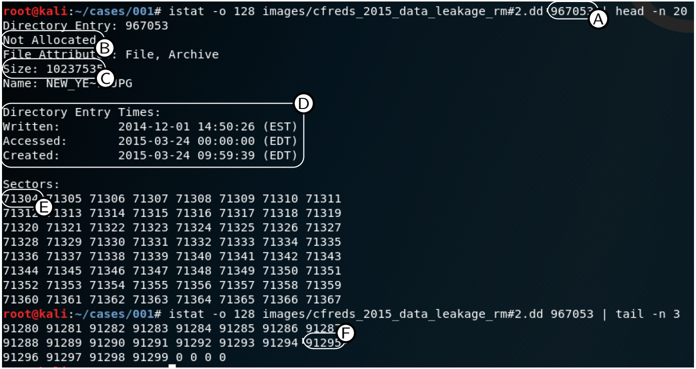


```
root@kali:~/cases/001# fls -o 128 -r images/cfreds_2015_data_leakage_rm#2.dd
r/r 3:  IAMAN $_@   (Volume Label Entry)
v/v 33423363:   $MBR
v/v 33423364:   $FAT1
v/v 33423365:   $FAT2
d/d 33423366:   $OrphanFiles
+ -/d * 133:    design
++ r/r * 263:   winter_storm.amr
++ r/r * 267:   winter_whether_advisory.zip
+ -/d * 136:    PRICIN~1
++ r/r * 967047:        my_favorite_cars.db
++ r/r * 967050:        my_favorite_movies.7z
++ r/r * 967053:        new_years_day.jpg
++ r/r * 967056:        super_bowl.avi
+ -/d * 137:    progress
++ r/r * 1651335:       my_friends.svg
++ r/r * 1651338:       my_smartphone.png
++ r/r * 1651341:       new_year_calendar.one
+ -/d * 138:    proposal
++ r/r * 1793159:       a_gift_from_you.gif
++ r/r * 1793161:       landscape.png
+ -/d * 141:    TECHNI~1
++ r/r * 3096966:       diary_#1d.txt
```

Figure: **File Name Info –** FAT Tables from physical structure (A), inode address (B), recursively print all file names (C), Unallocated or no longer present files directory - Orphaned Files (D), deleted file indicator (E)

Metadata Structure Analysis - istat

TSK commands starting with an `i` allow for metadata structure analysis.
`istat` provides metadata details on a particular file given its inode address.



```
root@kali:~/cases/001# istat -o 128 images/cfreds_2015_data_leakage_rm#2.dd 967053 | head -n 20
Directory Entry: 967053
Not Allocated
File Attributes: File, Archive
Size: 10237535
Name: NEW_YE~1.JPG

Directory Entry Times:
Written:        2014-12-01 14:50:26 (EST)
Accessed:       2015-03-24 00:00:00 (EDT)
Created:        2015-03-24 09:59:39 (EDT)

Sectors:
71304 71305 71306 71307 71308 71309 71310 71311
71312 71313 71314 71315 71316 71317 71318 71319
71320 71321 71322 71323 71324 71325 71326 71327
71328 71329 71330 71331 71332 71333 71334 71335
71336 71337 71338 71339 71340 71341 71342 71343
71344 71345 71346 71347 71348 71349 71350 71351
71352 71353 71354 71355 71356 71357 71358 71359
71360 71361 71362 71363 71364 71365 71366 71367
root@kali:~/cases/001# istat -o 128 images/cfreds_2015_data_leakage_rm#2.dd 967053 | tail -n 3
91280 91281 91282 91283 91284 91285 91286 91287
91288 91289 91290 91291 91292 91293 91294 91295
91296 91297 91298 91299 0 0 0 0
```

Figure: **Metadata Structure info** – use of inode(A), related file has been deleted (B), file size (C) creation and access times (D), start sector of previously attached data (E), end sector of previously attached data(F).

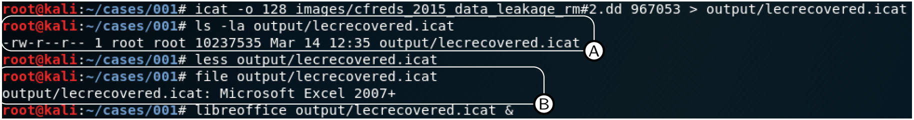
Investigation Sit-rep

What information have we gathered about this disk image?

- *File System Architecture:* FAT32
- *FS starting sector:* 128
- *Sector ranges for data area clusters:* 8192 to 2097151
- *Sector Size:* 512
- *Cluster size:* 4096
- File names for all deleted files in the FS and their inode numbers
- Sector Start and End points for any deleted file
- When the files were created, written and accessed last. (importance of write blockers, and helps with event time-lines).

File Recovery From Metadata Structure - `icat` I

Given the above gathered infomation we can now recover a file, this method uses the data within the metadata structure (inode) to recover the file from disk.

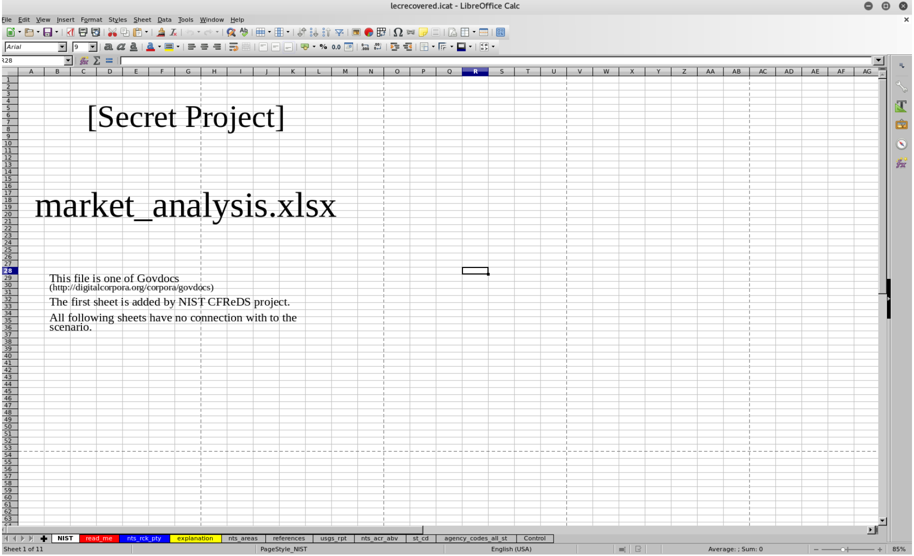


```
root@kali:~/cases/001# icat -o 128 images/cfreds_2015_data_leakage_rm#2.dd 967053 > output/lecrecovered.icat
root@kali:~/cases/001# ls -la output/lecrecovered.icat
-rw-r--r-- 1 root root 10237535 Mar 14 12:35 output/lecrecovered.icat
root@kali:~/cases/001# less output/lecrecovered.icat
root@kali:~/cases/001# file output/lecrecovered.icat
output/lecrecovered.icat: Microsoft Excel 2007+
root@kali:~/cases/001# libreoffice output/lecrecovered.icat &
```

Figure: **File Recovery Using Metadata Structure** – data recovered to file in our OS (A), file type is actually excel, not jpg (B).

NOTE: manual extraction of this file using sector to cluster address conversions and TSK data unit tools (`blk*`) is possible. This helps when data structures are partial or areas have been overwritten and will be explored in the lab.

File Recovery From Metadata Structure - icat II



[Secret Project]

market_analysis.xlsx

This file is one of Govdocs
(http://digitalcorpora.org/corpora/govdocs)

The first sheet is added by NIST CFReDS project.

All following sheets have no connection with to the scenario.

Figure: Opened Recovered File

Content Analysis

- **Data Unit Viewing:** View information in a data unit. We know *where* the evidence could be, but **not** *what* it is.
- **Logical FS level Searching:** Looks for a specific value or phrase in a *data unit*. We know *what* the content is but **not** *where*
- **Data Unit Allocation Status:** Verification of each data units status
- **Consistency Check:** Check all allocated data units have 1 metadata entry.

  none = orphaned data

  2 = unusual & not allowed in file systems.

Metadata Analysis

- **Metadata Lookup:** Reading metadata structure information (istat)
- **Logical File Viewing (data unit):** Looking up the *data units* allocated from metadata and using content lookup to find contents of the file.
  - In TSK, the icat tool allows you to view the contents of the data units that are allocated to a metadata structure.
    -s flag is given, the slack space is shown
    -r flag attempts to recover deleted files.
- **Logical File Searching:** Searches *data units* allocated to a metadata entry
- **Unallocated Metadata Analysis:** Listing unallocated metadata entries (ils)
- **Metadata Attribute Aearching & Sorting**

File Name Analysis

- **File Name Listing:** List file and directories looking for data of interest, then investigating further with metadata techniques like logical file viewing.
- **File Name Searching:** If we don't know partial name info like file extension.

TSK - File System Layer Tools I

- **fsstat**: shows file system details and statistics including layout, sizes and labels

TSK - File Name Tools

Allow for processing of file name structures.

- **ffind**: finds allocated and unallocated file names that point to a given meta data structure
- **fls** lists allocated and deleted file names in a directory

TSK - Meta Data Layer Tools

- **icat**: Extracts the data units of a file, which is specified by its meta data address.
- **ifind**: Finds the meta data structure that has a given file name pointing to it or the meta data structure that points to a given data unit.
- **ils**: Lists the meta data structures and their contents in a pipe delimited format.
- **istat**: Displays the statistics and details about a given meta data structure in an easy to read format.

TSK - Data Unit Tools

These file system tools process the data units where file content is stored.

- **blkcat**: Extracts the contents of a given data unit.
- **blkls**: Lists the details about data units and can extract the unallocated space of the file system.
- **blkstat**: Displays the statistics about a given data unit in an easy to read format.
- **blkcalc**: Calculates where data in the unallocated space image (from blkls) exists in the original image. This is used when evidence is found in unallocated space.

TSK - Volume System Tools

These tools take a disk (or other media) image as input and analyse its partition structures.

- **mmls**: Displays the layout of a disk, including the unallocated spaces.
- **mmstat**: Display details about a volume system (typically only the type).
- **mmcat**: Extracts the contents of a specific volume to STDOUT.

Additional Resources

- **Books**:
  - File System Forensic Analysis- Brian Carrier - http://amzn.eu/1R9U4bz
  - Practical Forensic Imaging - Bruce Nikkel - http://amzn.eu/5y057Ba
- **Reddit**: r/computerforensics https://www.reddit.com/r/computerforensics/wiki/faq
- **DFRWS Conference**: https://www.dfrws.org/
- **SANS DFIR**:
  - **Blog**: https://digital-forensics.sans.org/blog
  - **Crib Sheets/Posters**: https://digital-forensics.sans.org/community/cheat-sheets

Careers

- **GHCQ** - Internet and Systems Investigation: Forensic Technologist[4]
- **Cyfor** - eDiscovery Digital Forensics and Cyber Security - `https://cyfor.co.uk/careers/`

[4] https://recruitmentservices.applicationtrack.com/vx/lang-en-GB/mobile-0/appcentre-3/brand-2/xf-bd8ab30ccf84/candidate/so/pm/1/pl/6/opp/1268