**Disk Image Analysis**

**Objective:**

Exploring the files residing in the disk images using linux utilities

Understand the different layers of information associated with disk storage and to use sleuthkit which offers plethora of tools for undermining the information from these layers.

**Jargon / Common terminologies used :**

Mounting - interpret the contents of the disk using local file system / Make the file system accessible through the directory tree of Linux.

Loop device - Virtual device which allows access to the raw image in the same way as that of actual disk.

FUSE - linux kernel module used for interpreting different file system

File system - virtual organization of data (road map for OS to find data on disk)

**Native access to raw image:**

Though there are many forensic tools available for parsing the information from the raw images. It is always useful if the raw image is accessible natively by Linux. Linux offers this functionality through the concept of loop devices and mounting

**Step 1: Read the partition table using mmls command to find the volumes associated with file systems**

mmls /media/sf\_Forensics/Disk\ Image\ Analysis/Disk\ Images/Cfreds001A001.dd

DOS Partition Table

Offset Sector: 0

Units are in 512-byte sectors

Slot Start End Length Description

00: Meta 0000000000 0000000000 0000000001 Primary Table (#0)

01: ----- 0000000000 0000000062 0000000063 Unallocated

02: 00:00 0000000063 0000199583 0000199521 DOS FAT16 (0x06)

03: ----- 0000199584 0000201599 0000002016 Unallocated

00 -> primary partition table

01 -> standard spacing of 63 sectors

02 -> partition associated with the file system

**Step 2: Associate loop device with partition**

(loop device - virtual device allows disk image to be treated as a actual disk)

sudo losetup -r -o 32256 /dev/loop0 /media/sf\_Forensics/Disk\ Image\ Analysis/Disk\ Images/Cfreds001A001.dd

-r (readonly)

-o (offset starting byte position )

Starting sector - 63 Sector

Sector size - 512 bytes

Starting byte position = 512\*63 =32256

**Step 3: Creating a mount point (So that entire directory could be accessed)**

mount -o ro /dev/loop0 /mnt/raw

/dev/loop0 cannot be accessed /mnt/raw (directory) which is created using mkdir for the purpose of mounting

**Mounting the Volumes in Forensic Containers:**

Sometimes, the disk images obtained may be wrapped using forensic containers such as AFF (Advanced Forensic Format) or EWF (Express Witness Format) which cannot be directly mounted. But useful packages are available in Linux to convert these images wrapped in forensic container to raw images using FUSE (linux kernel module used for interpreting the various file systems in linux) .

AFFUSE - Convert image from .aff format to .raw format

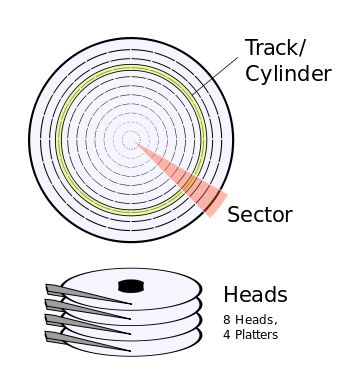
affuse /media/sf\_Forensics/Disk\ Image\ Analysis/Disk\ Images/ntfs.aff /sample

<affuse> <path to aff image> <path to copy the raw image>

MountEWF - Convert image from .E01 format to .raw format

**Understanding the Disk storage Mechanism:**

Disk Image Analysis is concerned with identification, extraction, and analysis of files and the file systems that lie upon the disk image.File system presents logical view of stored information on disk generates information at various level. Let us look at the jargons associate with disk storage and interpret the meaning



**Disk Layer**

Physical storage device

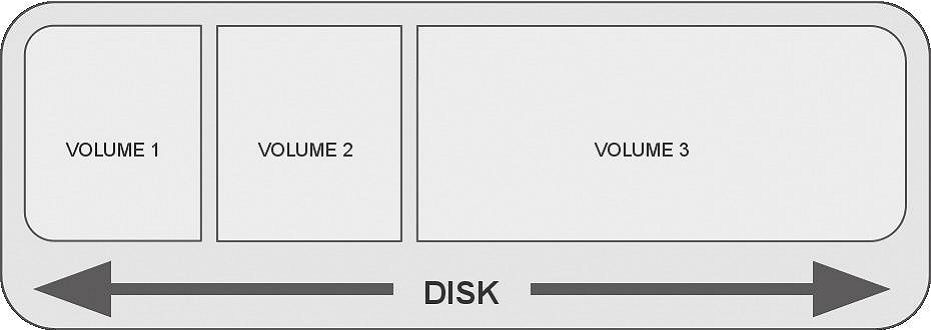
Analyzing information at this level is bit complex

Track - concentric circle that stores information

Sector - section of track with specified size

Cylinder - column of track across platter

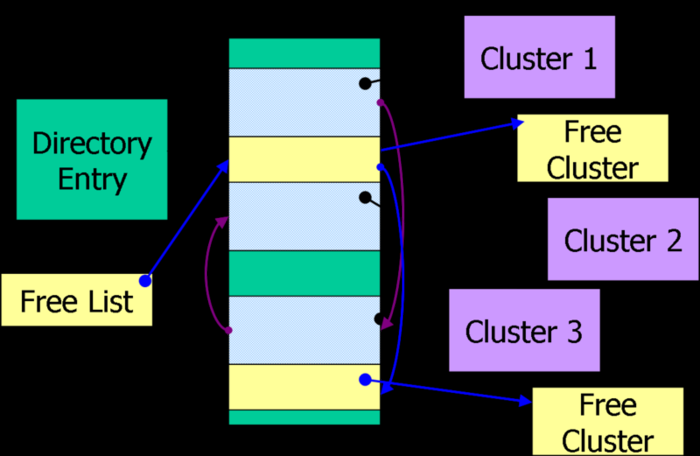
Sector - physical address / firmware



**Partition Layer**

Disk is divided into partitions

Each partition/volume can use different file system



**File system Layer**

Metadata specific to file system

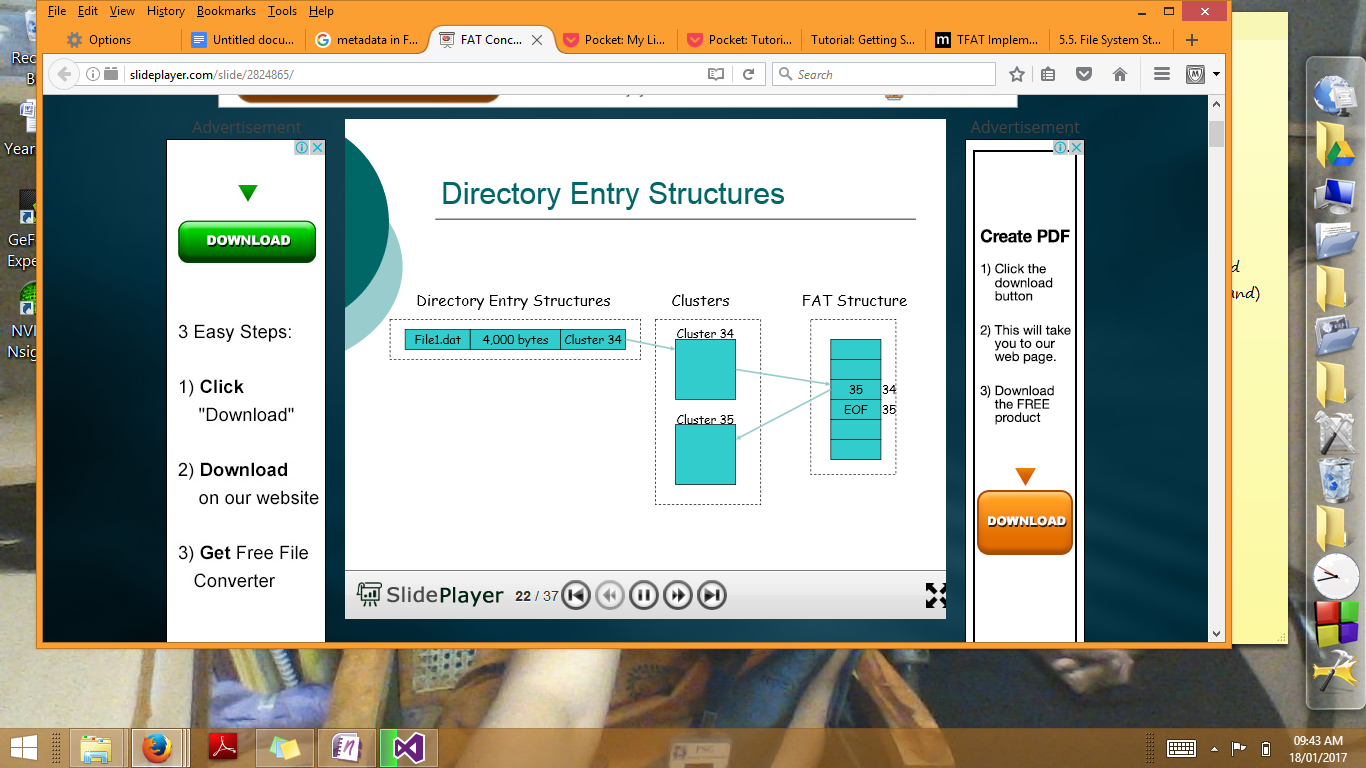
Describes the layout of files

Volume is organized effectively by file systems



**Data unit**

Contains the data



**Metadata**

Data about the data layer

Directory entry for files and directories

Information regarding file timestamps, data units corresponding to a file and file ownership information

**File layer**

Contains filenames and directory name

Pointers to metadata units

**Sleuth Kit (Autopsy)**

Extensive command line utility and web interface to access information across all these layers are provided.

Earlier, it was available as Coroner's toolkit. (Not portable to non-unix systems)

**Parsing the command line utility**

The common prefixes found in the Sleuth Kit tools that indicate the file system layer of the tool are:

• **“mm-”:** tools that operate on volumes (aka “media management”)

• **“fs-”**: tools that operate on file system structures

• **“blk-”**: tools that operate at the data unit (or “block”) layer

• **“i-”**: tools that operate at the metadata (or “inode”) layer

• **“f-”**: tools that operate at the file name layer

Common suffixes found in Sleuth Kit tools that indicate the expected function of the tool are:

**• “-stat”**: displays general information about the queried item

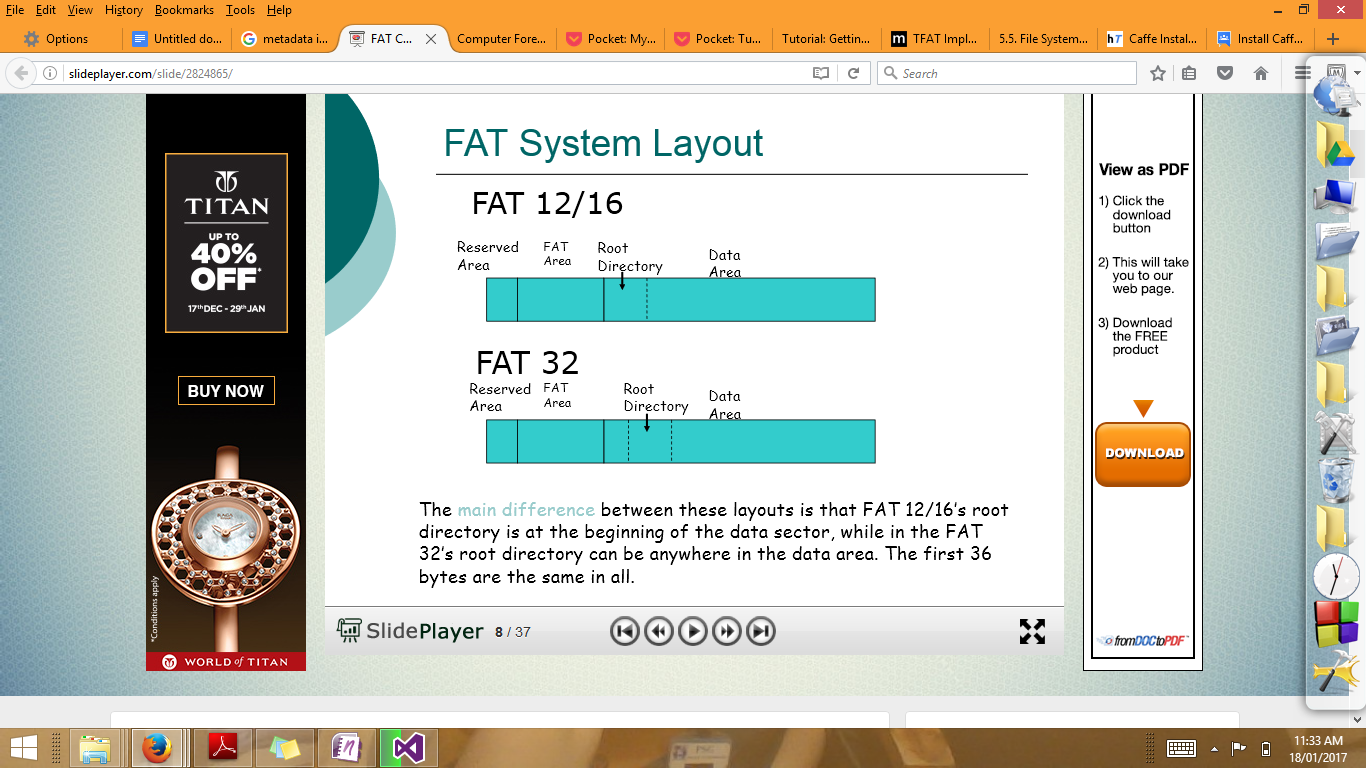
**• “-ls”**: lists the contents of the queried layer

**• “-cat”**: dumps/extracts the content of the queried layer

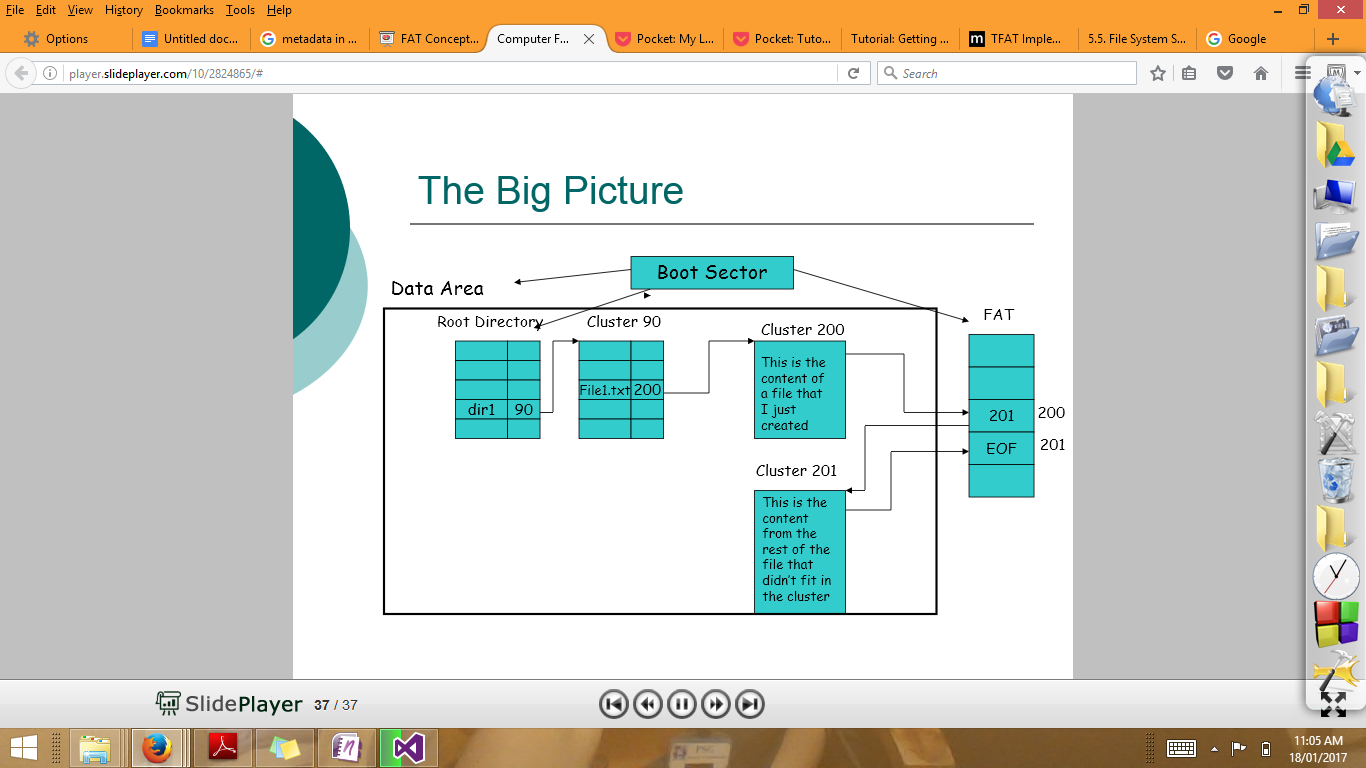
|  |  |
| --- | --- |
| **Layers** | **SleuthKit tools** |
| **Partition layer** | mmls - List the partition table  mmcat - streams the contents of specified volume to stdout |
| **File System Layer** | fsstat - displays the file system information |
| **Data Unit** | blkstat - allocation status of the block (specify the sector number)  blkcat - send the data of the specified sector to a file  blkls-extract the unallocated space of the file system  blkcalc- used in conjuction with unallocated space extracted from the blkls  to map the unallocated space to the actual location in the image |
| **Metadata** | ils - lists the metadata structures  istat - lists information about specific metadata structure  icat -(same as cat [filename] - streams the data to the console)  icat streams the data referenced by metadata address  ifind - find the metadata structure associated with filename / data unit address(Sector number) |
| **File Layer** | fls - Lists the file names traversing the root directory  ffind - finds the file name associated with metadata structure |
| **Miscellanous Tools** | mactime - generating timeline  sigfind - looks for hexadecimal string  hfind - faster searching on hash databases |

**Task :- Parsing Information from Disk Image with FAT volume**

FAT organizes the data as clusters. Clusters are group of one or more sectors. Clusters are numbered by the file system and are specific to the logical volume. Clusters are numbered from 2. First sector (oth sector) is used for storing info about file structure database, boot area.



Reserved Area contains the boot sector. It contains information about the various areas, allocated clusters, cluster size, sector size. Usually FAT area contains the FAT allocation table, backup of FAT table is also included. Root Directory can be present in any sector of the data area. The directory entry contains all information about the creation time, starting cluster (data block of the file).



**Parse the contents of file system**

**fsstat -o 63 /media/sf\_Forensics/Disk\ Image\ Analysis/Disk\ Images/Cfreds001A001.dd**

**-o (image offset ) Provide the sector information**

**File System Layout (in sectors)**

**Total Range: 0 - 199520 (range of sector)**

**\* Reserved: 0 - 0**

**\*\* Boot Sector: 0**

**\* FAT 0: 1 - 195**

**\* FAT 1: 196 - 390**

**\* Data Area: 391 - 199520**

**\*\* Root Directory: 391 - 422**

**\*\* Cluster Area: 423 - 199518 ( 199096 addressable sectors /4 =49774)**

**\*\* Non-clustered: 199519 - 199520**

**METADATA INFORMATION**

**--------------------------------------------**

**Range: 2 - 3186086**

**Root Directory: 2**

**CONTENT INFORMATION**

**--------------------------------------------**

**Sector Size: 512**

**Cluster Size: 2048 (2048/512 = 4 sectors in cluster)**

**Total Cluster Range: 2 - 49775**

**FAT CONTENTS (in sectors) (no fragmented files)**

**--------------------------------------------**

**423-426 (4) -> EOF**

**427-430 (4) -> EOF**

**431-434 (4) -> EOF**

**435-438 (4) -> EOF**

**439-478 (40) -> EOF**

**479-482 (4) -> EOF**

**483-486 (4) -> EOF**

**487-490 (4) -> EOF**

**491-494 (4) -> EOF**

**495-498 (4) -> EOF**

**499-502 (4) -> EOF**

**503-506 (4) -> EOF**

**507-510 (4) -> EOF**

**511-514 (4) -> EOF**

**515-518 (4) -> EOF**

**519-534 (16) -> EOF**

**535-538 (4) -> EOF**

**539-542 (4) -> EOF**

**543-546 (4) -> EOF**

**List the directory entries**

**ils -m -al -f fat16 -o 63 /media/sf\_Forensics/Disk\ Image\ Analysis/Disk\ Images/Cfreds001A001.dd**

**md5|file|st\_ino|st\_ls|st\_uid|st\_gid|st\_size|st\_atime|st\_mtime|st\_ctime|st\_crtime**

**0|<Cfreds001A001.dd--alive-2>|2|-/d---------|0|0|16384|0|0|0|0**

**0|<Cfreds001A001.dd-CFRED001 -alive-3>|3|-/rrwxrwxrwx|0|0|0|0|1098729328|0|0**

**0|<Cfreds001A001.dd-SHORT.TXT-alive-5>|5|-/rrwxrwxrwx|0|0|8|1098849600|1098890582|0|1098890582**

**0|<Cfreds001A001.dd-SUB-alive-7>|7|-/drwxrwxrwx|0|0|2048|1098849600|1098890628|0|1098890628**

**0|<Cfreds001A001.dd-UNICODE-alive-9>|9|-/drwxrwxrwx|0|0|2048|1098849600|1098890836|0|1098890836**

**0|<Cfreds001A001.dd-RUSSIA~1-alive-12>|12|-/drwxrwxrwx|0|0|2048|1098849600|1098933752|0|1098933751**

**0|<Cfreds001A001.dd-SHORT.TXT-alive-582>|582|-/rrwxrwxrwx|0|0|8|1098849600|1098890628|0|1098890628**

**0|<Cfreds001A001.dd-ALPHA.DOC-alive-710>|710|-/rrwxrwxrwx|0|0|19456|1098849600|1098890836|0|1098890836**

**0|<Cfreds001A001.dd-ALPHA-~1.UTX-alive-713>|713|-/rrwxrwxrwx|0|0|740|1098849600|1098890836|0|1098890836**

**0|<Cfreds001A001.dd-ALPHA-~2.UTX-alive-716>|716|-/rrwxrwxrwx|0|0|740|1098849600|1098890836|0|1098890836**

**0|<Cfreds001A001.dd-ALPHA-~3.UTX-alive-719>|719|-/rrwxrwxrwx|0|0|927|1098849600|1098890836|0|1098890836**

**0|<Cfreds001A001.dd-ALPHA-~4.UTX-alive-722>|722|-/rrwxrwxrwx|0|0|687|1098849600|1098890836|0|1098890836**

**0|<Cfreds001A001.dd-JAP-VO~1.UTX-alive-725>|725|-/rrwxrwxrwx|0|0|252|1098849600|1098890836|0|1098890836**

**0|<Cfreds001A001.dd-JAP-VO.UTX-alive-727>|727|-/rrwxrwxrwx|0|0|191|1098849600|1098890836|0|1098890836**

**0|<Cfreds001A001.dd-MIX.TXT-alive-729>|729|-/rrwxrwxrwx|0|0|589|1098849600|1098890836|0|1098890836**

**0|<Cfreds001A001.dd-RUSSIA~1.TXT-alive-732>|732|-/rrwxrwxrwx|0|0|316|1098849600|1098890836|0|1098890836**

**0|<Cfreds001A001.dd-RUSSIAN.TXT-alive-734>|734|-/rrwxrwxrwx|0|0|333|1098849600|1098890836|0|1098890836**

**0|<Cfreds001A001.dd-DS\_STO~1-alive-1990>|1990|-/rrwxrwxrwx|0|0|6148|1098849600|1098929408|0|1098933751**

**0|<Cfreds001A001.dd-\_958B~1.DS\_-alive-1992>|1992|-/rrwxrwxrwx|0|0|82|1098849600|1098929408|0|1098933751**

**0|<Cfreds001A001.dd-snack.txt-alive-1993>|1993|-/rrwxrwxrwx|0|0|16|1098849600|1098929408|0|1098933751**

**0|<Cfreds001A001.dd-808C~1.TXT-alive-1995>|1995|-/rrwxrwxrwx|0|0|280|1098849600|1098933856|0|1098933751**

**0|<Cfreds001A001.dd-$MBR-alive-3186083>|3186083|-/v---------|0|0|512|0|0|0|0**

**0|<Cfreds001A001.dd-$FAT1-alive-3186084>|3186084|-/v---------|0|0|99840|0|0|0|0**

**0|<Cfreds001A001.dd-$FAT2-alive-3186085>|3186085|-/v---------|0|0|99840|0|0|0|0**

**0|<Cfreds001A001.dd-$OrphanFiles-alive-3186086>|3186086|-/d---------|0|0|0|0|0|0|0**

**Know the files residing in the image**

**fls -l -o 63 /media/sf\_Forensics/Disk\ Image\ Analysis/Disk\ Images/Cfreds001A001.dd**

**r/r 3: CFRED001 (Volume Label Entry) 2004-10-25 14:35:28 (EDT) 0000-00-00 00:00:00 (UTC) 0000-00-00 00:00:00 (UTC) 0000-00-00 00:00:00 (UTC) 0 0 0**

**r/r 5: short.txt 2004-10-27 11:23:02 (EDT) 2004-10-27 00:00:00 (EDT) 0000-00-00 00:00:00 (UTC) 2004-10-27 11:23:02 (EDT) 80 0**

**d/d 7: sub 2004-10-27 11:23:48 (EDT) 2004-10-27 00:00:00 (EDT) 0000-00-00 00:00:00 (UTC) 2004-10-27 11:23:48 (EDT) 2048 00**

**d/d 9: unicode 2004-10-27 11:27:16 (EDT) 2004-10-27 00:00:00 (EDT) 0000-00-00 00:00:00 (UTC) 2004-10-27 11:27:16 (EDT) 2048 00**

**d/d 12: russian-utf-16 2004-10-27 23:22:32 (EDT) 2004-10-27 00:00:00 (EDT) 0000-00-00 00:00:00 (UTC) 2004-10-27 23:22:31 (EDT) 2048 0 0**

**v/v 3186083: $MBR 0000-00-00 00:00:00 (UTC) 0000-00-00 00:00:00 (UTC) 0000-00-00 00:00:00 (UTC) 0000-00-00 00:00:00 (UTC) 512 0 0**

**v/v 3186084: $FAT1 0000-00-00 00:00:00 (UTC) 0000-00-00 00:00:00 (UTC) 0000-00-00 00:00:00 (UTC) 0000-00-00 00:00:00 (UTC) 99840 0 0**

**v/v 3186085: $FAT2 0000-00-00 00:00:00 (UTC) 0000-00-00 00:00:00 (UTC) 0000-00-00 00:00:00 (UTC) 0000-00-00 00:00:00 (UTC) 99840 0 0**

**Questions?**

1. **List down the file systems commonly used and are supported by various OS?**
2. **What are the generic ways in which content is hidden in the disk?**
3. **Why analysis using hex editors are preferred in forensic examination?**

**Submit an extensive report on parsing disk images using Slueth kit.**

**Sample disk images can be found at the following link: http://www.dftt.org/**