

Forensic Evidence After Moving a File

736233

Proposed by: Mr R May

Abstract

The collection of evidence on computers is a non-trivial task, requiring a good technical understanding of the operating system and underlying filesystem. The availability of different filesystems (FAT and NTFS) can also increase the analysis difficulty or offer better features to track activities.

1 Introduction

The digital-lifestyle has dawned over the last 30 years and computers are now commonplace in homes and offices worldwide, becoming a commodity more than a rarity. The ability to automate tasks, store incredible amounts of data, connect computers via networks and a plethora of other capabilities have lead to increased productivity and unrivaled access to data. As with all technological advancements, the digital age is open to all and unfortunately becoming another aid in illegal activity. The perceived anonymity and protection offered by digital device use, especially over the Internet, can help encourage people to break laws where they may not do so in everyday life.

The role of forensics comes in a case where it is suspected that a computer has been involved in a crime either directly or indirectly. As with other crimes, evidence collection is paramount and a report could be written on that a topic alone. In computer forensics the main source of evidence is data, usually in the form of files; user-created files or system files, alternatively data areas on a hard drive may hold valuable information.

For the purpose of this report it is assumed that one is looking for evidence on a computer hard disk that has been correctly imaged to protect the data it contains. We are specifically looking for evidence of file movement from one directory to another. A brief introduction to hard disk properties and filesystems will be necessary to explain how such evidence can be found.

2 Hard Disk Drives

The smallest unit of space on the hard drive that any software program can access is a sector[1], which usually consists of 512 bytes With the size of hard disks today, managing so

many individual sectors introduces difficulties, large files become split over many hundreds of sectors and managing a number of files becomes a difficult task.

To ease this situation clusters are used, a cluster being a collection of sectors ranging from 4 sectors (2048 bytes) upwards eg. 128 sectors (65536 bytes). The number of sectors per cluster is usually dependent on the maximum size of the disk.

Each cluster is a continuous block of space on the disk and each can be occupied by only one file eg a file of size 1024 byte will occupy one cluster (for a 4 sector cluster as above) with the other 1024 bytes wasted. If a file is to be written and its size exceeds that of the cluster size, it will span to another cluster eg. A file of size 3012 will occupy 2 clusters (for a 4 sector cluster as above) note that 1080 bytes are spare in the second cluster. The allocation and management of clusters is the duty of the filesystem.

3 Filesystems

A computer operating system (OS) provides the interface between user and hardware, for a hard drive this relies on the filesystem in use. The filesystem dictates how the OS can create/modify/delete/read/write data on the hard drive. There exists numerous filesystems two of the most popular being: FAT (File Allocation Table) filesystem used in MS-DOS and Microsoft Windows 9.xx and NTFS (New Technology File System) used in the NT family of Microsoft Windows.

3.1 FAT Filesystem

There are several versions of FAT all sharing similar characteristics, for this report we consider the most recent FAT32. In a FAT filesystem files and folders are mostly indistinguishable. A FAT filesystem partition is composed of four different sections[2].

The Boot Sector This is always the first sector of the partition and includes some basic file system information (in particular, its type), pointers to the location of the other sections and the operating system's boot loader code.

The FAT Region. This contains two copies of the File Allocation Table for the sake of redundancy. These are maps of the partition, indicating how the clusters are allocated.

The Root Directory Region. This is a Directory Table that stores information about the files and directories in the root directory. This can be stored anywhere in the partition.

The Data Region. This is where the actual file and directory data is stored and takes up most of the partition. The size of files and subdirectories can be increased arbitrarily (as long as there are free clusters) by simply adding more links to the file's chain in the FAT.

3.1.1 File Allocation Table

The File Allocation Table is a list of entries that map to each cluster on the partition. Each entry can record one of five properties of the cluster:

- The address of the next cluster in a chain
- A character that indicates the end of a chain
- A character to mark a bad cluster
- A character to mark a reserved cluster
- A zero to note that that cluster is unused

3.1.2 Directory Table

A Directory table can be stored in the root directory region and the data region. Each file or directory stored on the partition is represented by a 32 byte entry in the table. This stores information in the form:

Bytes	Description
8	Name
3	Extension
1	Attribute
1	Reserved
5	Time of creation
2	Last access time
2	High 2 bytes of first cluster number
2	Last modification time
4	Low 2 bytes of first cluster
4	File Size

The first bit of the attribute byte can have values corresponding to archive, directory, hidden, read-only, system and volume.

When a file is deleted in FAT filesystem, it is not actually removed from the disk, rather the filename is changed. The first byte is changed to 0xE5 (hex) or 229 (ASCII).

A file may occupy one or more clusters on a disk. These clusters are 'chained' together, in more technical terms, the file is represented by a linked list. When a file is accessed, the directory table will be read containing a reference to the first cluster, this will take it to the FAT where it will read the address of the next cluster, this will continue until the last cluster is reached that will have a character (usually all 1's) to represent the end of file. Shown in figure 1. It is not necessary for the data to remain in one contiguous block, but any cluster on disk, leading to fragmentation.

A directory is identical to a file apart from the attribute bit and the first two entries being reserved, they must always contain sub-directory entries called "." dot and ".." dotdot entries (ie the filename is . or .. This is not true however for the root directory, does not have any date or time stamps, does not have a file name (other than the implied file name), and does not contain . and .. files as the first two directory entries in the directory[3].

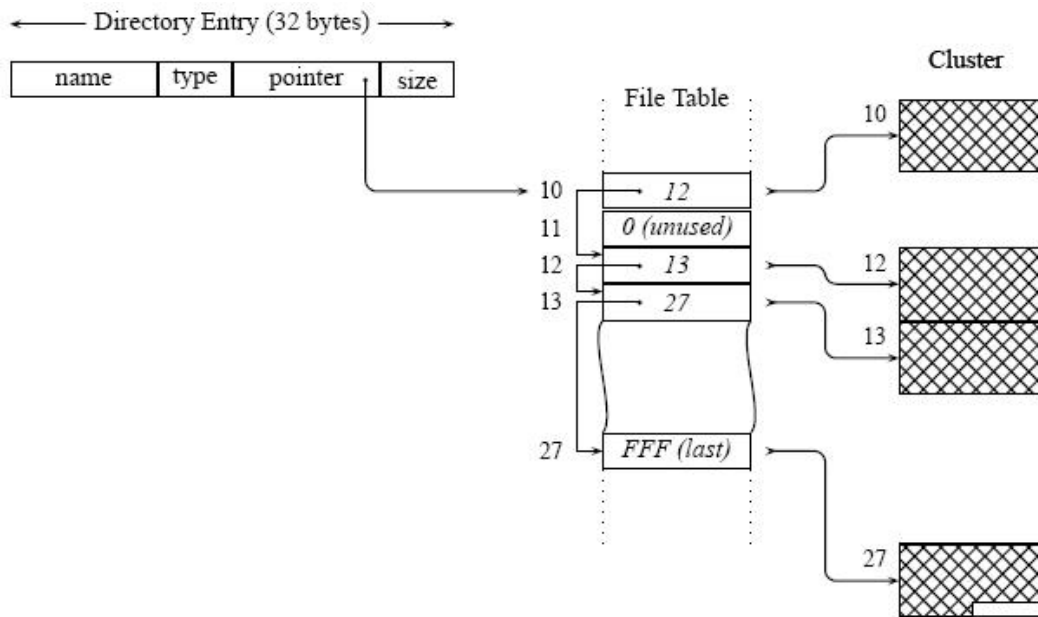


Figure 1: A graphical representation of a link list of clusters on a disk

3.1.3 Moving a File

Moving a file from directory A to directory B consists of marking the directory A entry as deleted (first character changed to 0xE5 hex or 229 ASCII [4]) and creating a new identical file in directory B. There will also be a change to the parent directory cluster within the directory entry of the file to match the new parent location ie marked as the first cluster of directory B. It should also be noted that this parent cluster change occurs within the original entry before the directory entry is copied. The last time accessed time will be updated in the deleted file in directory A and the new file in directory B will be given a creation time. Possible evidence left from this activity will be discussed in section 5.

3.2 NTFS

NTFS is used in the the NT family of Microsoft Windows, as with FAT there are several versions that act similarly. For the purpose of this report we consider v5.0+ that first shipped with Windows 2000.

On some levels NTFS works in the same way as FAT such as hierarchal directory structure (common to many other filesystems). Exact details regarding the implantation on NTFS are a closely held secret but a general outline can be made.

On an NTFS disk partition (alternatively referred to as a volume), the first 12.5% is dedicated to the MFT (Master File Table) zone. The remaining space for data. The following description is taken from a Microsoft article. [5]

“The MFT is a relational database that consists of rows of file records and columns of file attributes. It contains at least one entry for every file on an NTFS volume, including the MFT itself. Because the MFT stores information about itself, NTFS reserves the first 16 records of the MFT for metadata files (approximately 16KB), which are used to describe the MFT. Metadata files that begin with a dollar sign (\$) are described in the table Metadata Files Stored in the MFT.”

3.2.1 Attributes

NTFS regards files (and folder) as a list of attributes that are stored as a record in the MFT. The record also contains its position in the MFT. Noticable attributes are given below [5]

- Standard Information - Information such as access mode (read-only, read/write, and so forth) timestamp, and link count.
- Attribute List - Locations of all attribute records that do not fit in the MFT record. Used as pointer for linking
- File Name - File Name
- Data - File data. Each file typically has one unnamed data attribute. A file can also have one or more named data attributes.
- Index Root - Used to implement folders and other indexes.
- Index Allocation - Used to implement the B-tree structure for large folders and other large indexes.
- Bitmap - Used to implement the B-tree structure for large folders and other large indexes.

Each record has a size of 1KB. This gives rise to several possibilities:

- file/folder is < 1KB - is contained in a single file record in MFT. All attributes are 'resident'.
- file is > 1Kb - some attribute information has to be stored outside the MFT, these are 'nonresident' and stored on other clusters on the disk. Some of the nonresident attribute remains in the MFT and points to the external clusters. NTFS creates the Attribute List attribute to describe the location of all attribute records. Similar process to a FAT link-list.
- folder is > 1KB - some attribute information has to be stored outside the MFT, the directory is structured as a B-Tree (B+ specifically). Each record in the tree points to clusters containing folder entries not in the MFT.

The B-Tree directory structure in NTFS leads to faster access time of the files held within large directories.

3.2.2 Features

NTFS has a myriad of extra features in comparison to the FAT filesystem, detailing them all goes beyond the scope of this report, however, one feature relevant to evidence collection is the Change Journal. When this is enabled, all changes to files and folders on a volume are recorded. A record containing USN (Update Sequence Number), file name and information about the change is added to a log file. Note the file/folder itself is not logged.

NTFS also makes use of transaction. Similar to the Change Journal each time a change is made to files and folders it is recorded to a log . A transaction is occurring and as it progresses, NTFS records each of the changes it makes to any part of the volume. Once all of the changes are complete, the transaction is also complete, and a marking is placed in the activity log to indicate that the transaction was successful. This is called committing the transaction. If an error occurs during a transaction e.g. power failure then some entries in the transaction log will be complete but it will not have been committed, the filesystem knows to roll back all changes to maintain integrity[6]. It is not clear for how long the transaction log is held.

3.2.3 Moving a file

Moving a file from Folder A to Folder B is a different process compared to FAT filesystem. Simply the parent directory in the file record will be changed to point Folder B. The folder records of Folder A and B will be updated. This is true because under NTFS a folder does not hold files, just meta information. In the file record the time last accessed/modified will be updated, time of creation will remain the same. The transaction log is written as normal, if Change Journaling is active the changes will be logged down also.

4 Evidence

Under the two filesystems there will be different evidence left after a file has been moved, we assume no other changes have been made to the system.

In FAT, a file will have been created in one directory at the same/very close to the last access time of a deleted file in the same directory. It appears like this because parent cluster change occurs within the original entry before it is deleted. Therefore the deleted file appears to come from directory B. This naturally leads to the conclusion that there is evidence to show the file has been moved to a directory and when that occurred but not the originating source.

For NTFS, no file deletion occurs just an update of pointers in records. The changes will be reflected in the transaction log and change journal (if active). Therefore there is evidence of file movement; when and where.

5 Conclusion

Through the report we assumed that not many changes had been made to the system since a suspected file movement occurred. If the opposite is assumed true collecting evidence becomes more difficult. In the case of FAT, the deleted entry is marked as an unallocated and could be written over at any time by a new file. A disk defragment may also have a similar effect.

With respect to NTFS, if the size of the log files is finite it is reasonable to assume that after a period of time the system could start to overwrite from their start. This would result in the loss of information on the file movement. Interestingly, if a transaction log is not permanent and gets removed on commital, it offers the opportunity to track potential file movement in the case where a computer was turned off in the middle of a process and not booted again. It is good forensic practice not to boot a suspect system for such cases.

It has been shown that the collection of evidence on computers is a non-trivial task, requiring a good technical understanding of the operating system and underlying filesystem. The presence of different filesystems can also increase the analysis difficulty or offer better features to track activities.

References

- [1] DEW Associates Corporation *Hard Drive Clusters and File Allocation*. Web page in DEW Associates knowledge base. Not Dated. Downloaded 31 November 2005 from http://www.dewassoc.com/kbase/hard_drives/clusters.htm
- [2] Wikipedia *File Allocation Table*. Wikipedia Article. Accessed on 30 November 2005 from http://en.wikipedia.org/wiki/File_Allocation_Table
- [3] Microsoft Corporation *Microsoft Extensible Firmware Initiative FAT32 File System Specification, FAT: General Overview of On-Disk Format*. Microsoft Corporation. Version 1.03, December 6 2000.
- [4] NTFS.com *NTFS Recovery Concepts 2005*. A series of web pages regarding data recovery. Dated 2005. Accessed on 31 November 2005 from <http://www.ntfs.com/file-recovery-concepts.htm>
- [5] Microsoft Corporation *How NTFS Works*. Technical article produced by Microsoft. Last updated: March 28 2003 <http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/TechRef/8cc5891d-bf8e-4164-862d-dac5418c5948.mspx>
- [6] PC Guide *NTFS Directories (Folders)*. A series of web pages on the topic of directory structure. Dated April 17, 2001. Accessed on 30 November 2005 from <http://www.pcguides.com/ref/hdd/file/ntfs/filesDir-c.html>
- [7] Brian Carrier *File System Forensic Analysis*. Addison-Wesley, 2005

6 Source Evaluation

DEW Associates are a fairly well known source of information. The level is fairly basic which can mean that there is less chance of publishing error to their website. However the specific authors are unknown and could come from a biased background. I would say they were a fair to middling source for the report.

Wikipedia has several advantages and disadvantages compared to several sources. The dynamic and public changeable content means that errors are often quickly spotted and rectified often meaning that in general, the articles are very accurate and true - especially so in scientific/technical field. The nature of Wikipedia is a problem in respect of the fact that a malicious user could change the information to something utterly false. The constant updating to pages also means it is difficult to back reference and be sure that the information is that that was first used. I think Wikipedia was a good reference as it has a balanced/unbias view on the topic reflected by the breadth of the article.

The first Microsoft reference is a white paper. This is not a press release or similar but a technical article about FAT features. Microsoft first developed FAT so is the likely to be one of the best sources for information, the opportunity for bias is negligible. The paper is a little dated, however the filesystem specification is unchanging recently. I beleive this a good source.

The NTFS.com web page is highly commercial. The information they present is very technical but the ethos seems just as much about advertising as knowledge, damaging credibility. This was a fair source to use.

My opinion of the second Microsoft article is the same as the first.

The PC guide provided extensive information but as before there was a lot of advertising. This was a fair source to use.

The book was used as a general reference on the field of forensic computing. Being a printed piece lends a lot of credibility to the information it presents. This was a good source