

The FAT File System

Sleuth Kit Implementation Notes (SKINs)

www.sleuthkit.org

Brian Carrier

Last Updated: June 2003

Introduction

This document contains information on the implementation of the FAT file system in [The Sleuth Kit](#). The Sleuth Kit is based on the original designs of The Coroner's Toolkit (TCT), which was designed only for UNIX file systems. The FAT file system and UNIX file systems are very different and this document will identify how those differences were handled. A basic understanding of FAT is assumed.

The major design "decisions" that had to be made are related to:

- Disk unit addressing
 - Meta-data addressing
-

Disk Unit Addressing

FAT saves file content in clusters. A cluster is a grouping of consecutive sectors (512-bytes each). When a file is described by the directory entries and File Allocation Table, the cluster numbers are used as addresses. The problem, is that cluster 0 is not at the beginning of the partition. Cluster 0 is in the Data Area, which is after the super block and File Allocation Tables and can be hundreds of sectors into the partition. This creates a problem because if The Sleuth Kit were to use clusters as the addressable units, then there would be no way to identify the non-"data area" sectors.

This problem was solved by making the sector as the addressable unit, instead of the cluster. When a file is described (using 'istat' for example), the sector addresses are given. In the output of 'fsstat', the File Allocation Table contents are displayed in sectors and when using 'dls -l', the sector status is given.

This actually makes manual data recovery easier because one can use 'dd' to carve out data using the

sector addresses. If clusters were given, the user would have to translate the Data Area address to sectors before carving out data.

Meta-Data Addressing

FAT describes its files in a directory entry structure, which is contained in the sectors allocated by the parent directory. The directory entry structures have a fixed size of 32-bytes, not addressed, and can exist anywhere in the partition. The Sleuth Kit requires some type of addressing method for meta data structures, so this became a problem. Also, the root directory does not have a directory entry. In other words, there is no descriptive information for the root directory.

The solution to this problem was to use the same method that is used in many UNIX implementations. Each sector in the data area is treated as though it could be full of directory entries. As each sector is 512-bytes and each directory entry is 32-bytes, each sector could contain 16 entries. To keep things similar to UNIX, the root directory is given the value of 2 (and its meta-data is set to 0). The first 32-bytes of the first sector in the data area are addressed as 3, the second 32-bytes of the sector are 4 etc. The Sleuth Kit will scan through the sectors and identify which ones actually contain directory entries.

This method will produce large gaps of addresses between used address values and places a limit on the size of the partition that can be analyzed. The limit is:

$$2^{32} / 16 = 2^{28} \text{ sectors}$$

Therefore, we can handle partitions of size 137,438,953,472 bytes. It is unlikely that FAT file systems will be over 128GB in size.

Notes on Timezones

FAT does not store the file times in the delta format that UNIX does. Instead of saving the difference in time from GMT, FAT simply saves the raw hour, minute, and second values. The Sleuth Kit stores all times in the UNIX GMT offset format and will translate the FAT time to the UNIX offset. This uses the current timezone value when identify the GMT offset.

If the tool displays the time in a nice ASCII format, the same timezone will be used to translate the offset value into a date. Therefore, you can use any timezone value and the time will not change (just the timezone name). On the other hand, if you use a tool such as 'ils' or 'fls -m', which display the time in the offset format, then it will have the offset of the current timezone or the one specified with '-z'. Therefore,

ensure that the same '-z' argument is used with 'mactime' to display the correct time in the timeline.

General Notes on Time

Each file in FAT can store up to three times (last accessed, written, and created). The last written time is the only 'required' time and is accurate to a second. The create time is optional and is accurate to the tenth of a second (Note that I have seen several system directories in Windows that have a create time of 0). The last access time is also optional and is only accurate to the day (so the times are 00:00:00 in The Sleuth Kit).

Notes on ASCII and UNICODE

FAT32 allows names to be written in UNICODE, but The Sleuth Kit will convert the value (no matter what) to ASCII. Future versions may support UNICODE.

The FAT spec can be found at:

<http://www.microsoft.com/whdc/system/platform/firmware/fatgen.mspx>

The Sleuth Kit can be found at:

www.sleuthkit.org

Copyright © 2002-2003 by Brian Carrier. All Rights Reserved