



IDENTIFY

MATCH

Digital Archaeology

The Art and Science of Digital Forensics

SCAN

MICHAEL W. GRAVES

Digital Archaeology

Magic Numbers

Magic numbers are really nothing more than another method of structuring a header. Files incorporating magic numbers embed a file signature consisting of hexadecimal code into the first few bytes of the file to identify the file type. The term is derived from the Linux and UNIX (*nix) file system. The Linux Information Project (LIP 2006) defines the magic number as occupying the first six bytes of the file. Many programs use the magic number as the first step in identifying a file type. However, as with the file header, there are certain files, such as ASCII text files, HTML, and source code cabinets, that do not incorporate magic numbers.

Identifying a file by the magic number method does incorporate a small degree of latency (additional processing overhead required by an application to perform a specific set of tasks). Most Linux builds have defined lists of magic numbers in various directories. Among these are (Darwin 1999)

- /usr/share/file/magic.mgc—Compiled list of magic numbers
- /usr/share/file/magic—Default list of magic numbers
- /usr/share/file/magic.mime.mgc—Default compiled list that will display mime types when the -i trigger is used
- /usr/share/file/magic.mime—Default list that will display mime types when the -i trigger is used

In Linux, the `file(1)` command can be used to identify a file by its type. One of the command's first tests is to attempt to read a magic number and compare the number it finds to one or more of the magic number lists above. The digital forensic examiner can use a disk editor to view a file and examine the magic number directly in an effort to identify the file type.

UNDERSTANDING METADATA

The word *metadata* gets thrown around a lot and is used in more than one context. Earlier in this book, a loose definition of metadata was presented that simply defined it as data that describes data. However, metadata can exist in multiple forms. The operating system maintains information about files in various repositories. As discussed in the previous chapter, the NTFS file system makes use of a series of metadata files. Individual files can also contain information stored within the file that defines the file. Additionally, many applications, such as document management systems, maintain separate files containing metadata. All of these sources can be a gold mine of information for an investigator. *Aguilar v.*

Immigration and Customs Enforcement (2008) determined that the three types of metadata relevant to digital evidence include

- System metadata—Information generated by the file system or document management system
- Substantive metadata—Information that defines modifications to a document
- Embedded metadata—Information embedded by the application that creates or edits the file

Substantive metadata can fall within either of the other categories. Another form of metadata that exists that is important to the investigator is external metadata. Many document and image management software solutions maintain large amounts of information in the form of a database. Indexing, file modification, tracking, and auditing information is stored in separate files maintained by the application. Each of these types of metadata will be discussed over the next few pages.

SYSTEM METADATA

Chapter 8 introduced the concept of metadata usage by the OS. All file systems maintain vast amounts of information about the files and directories stored on the volumes they control. The fact is—the file system *is* the metadata that the operating system uses to manage files on the various media it controls. To be certain, there are physical aspects of the file system, such as the mapping of file allocation units on the drive itself, but that mapping is meaningless without the directions that tell the OS or the applications how to get there from here.

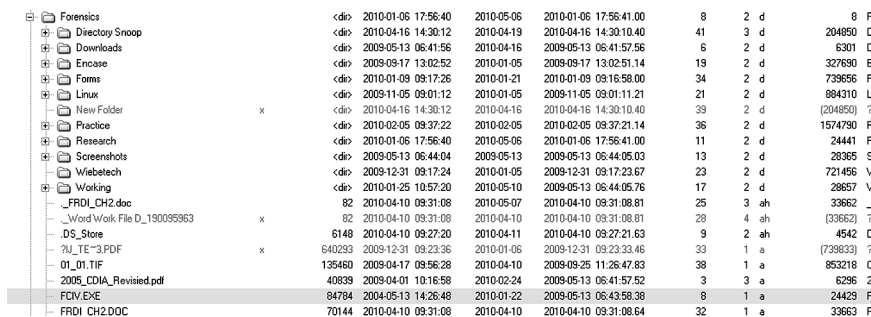
It isn't just hard disks that have volumes of metadata. CD-ROMs, DVDs, and even thumb drives need some form of file table that informs the system how and where files are store. Every computer running needs to be able to mount multiple file systems. The hard disk uses its system. As mentioned in the previous chapter, Linux systems might be formatted with the Ext2, Ext3, or Ext4 or perhaps the Reiser file system. Windows typically uses NTFS, although some legacy systems may use one of the several versions of FAT. Even an NTFS-based computer needs to be able to read FAT if that is how a USB flash drive was formatted. And in order to read CD-ROMs or DVDs, the ISO-9660 or the ECMA-167 file system must be mounted. Understanding how these files systems work is far beyond the scope of this book. However, a brief overview of how and where the system metadata is stored is essential for the digital investigator.

Value of OS Metadata

The useful aspect of OS metadata in the process of digital investigation is the ability to prove the existence of a deleted document and to research the timeline of a document. OS metadata does not help identify contents of files, aside from file type. A critical piece of information found here is the *modify/access/create* (MAC) data. Disks formatted with NTFS offer the additional attribute of entry modified (EM). EM notes the last time the MFT entry in the NTFS metafiles was modified. MAC information is valuable for creating a time line of events, as long as care is taken in analyzing and interpreting the data. It is important that the tools used by a forensic investigator are tested and verified to *not* alter MAC data.

All files stored on any file system are stamped with the time and date they were created, the last time they were accessed, and the last time they were modified. MAC data is easily viewed using a wide variety of commercial and shareware utilities. Figure 9.5 shows an example. Used in conjunction with other information found on the computer, it might be possible to identify what user was the last to access or modify a file and perhaps even who created it. A short discussion about each of the MAC attributes is in order here.

Create The create attribute on a file is generated the first time that the file is saved to the file system. Note that it is *not* necessarily the date that the file was originally saved. How can this be? Two things commonly affect the create date. If a user copies a file from one location to another, even though the two files are identical, each will have a different create date. The source file will show the date it was initially saved to that disk, while the new copy will have a create attribute that shows the time and date that it was first saved to the target drive.



File Name	Creation Date	Last Modified Date	File Size	Attributes
Forensics	<dir> 2010-01-06 17:56:40	2010-05-06	2010-01-06 17:56:41.00	8 2 d 8 F
Directory Snoop	<dir> 2010-04-16 14:30:12	2010-04-19	2010-04-16 14:30:10.40	41 3 d 204850 C
Downloads	<dir> 2009-05-13 06:41:56	2010-04-16	2009-05-13 06:41:57.56	6 2 d 6301 C
Encase	<dir> 2009-09-17 13:02:52	2010-01-05	2009-09-17 13:02:51.14	19 2 d 327690 E
Forms	<dir> 2010-01-09 09:17:26	2010-01-21	2010-01-09 09:16:58.00	34 2 d 739656 F
Linux	<dir> 2009-11-05 09:01:12	2010-01-05	2009-11-05 09:01:11.21	21 2 d 884310 L
New Folder	<dir> 2010-04-16 14:30:12	2010-04-16	2010-04-16 14:30:10.40	39 2 d (204850) ?
Practice	<dir> 2010-02-05 09:37:22	2010-02-05	2010-02-05 09:37:21.14	36 2 d 1574790 F
Research	<dir> 2010-01-06 17:56:40	2010-05-06	2010-01-06 17:56:41.00	11 2 d 24441 F
Screenshots	<dir> 2009-05-13 06:44:04	2009-05-13	2009-05-13 06:44:05.03	13 2 d 28365 S
Wiebetech	<dir> 2009-12-31 09:17:24	2010-01-05	2009-12-31 09:17:23.67	23 2 d 721456 V
Working	<dir> 2010-01-25 10:57:20	2010-05-10	2009-05-13 06:44:05.76	17 2 d 28657 V
_FRDI.CH2.doc	82 2010-04-10 09:31:08	2010-05-07	2010-04-10 09:31:08.81	25 3 ah 33662 _
_Word Vork File D_190095963	82 2010-04-10 09:31:08	2010-04-10	2010-04-10 09:31:08.81	28 4 ah (33662) ?
_DS_Store	6148 2010-04-10 09:27:20	2010-04-11	2010-04-10 09:27:21.63	9 2 ah 4542 C
_NJ_TE-3.PDF	640293 2009-12-31 09:23:36	2010-01-06	2009-12-31 09:23:33.46	33 1 a (739833) ?
01_01.TIF	135460 2009-04-17 09:56:28	2010-04-10	2009-09-25 11:26:47.83	38 1 a 853218 C
2005_CDIA_Revised.pdf	40839 2009-04-01 10:16:58	2010-02-24	2009-05-13 06:41:57.52	3 3 a 6296 Z
FDIV.EXE	84784 2004-05-13 14:26:48	2010-01-22	2009-05-13 06:43:58.38	8 1 a 24429 F
FRDI.CH2.DOC	70144 2010-04-10 09:31:08	2010-04-10	2010-04-10 09:31:08.64	32 1 a 33663 F

Figure 9.5 Several readily available utilities allow the user to view the currently active MAC data for a file.

The other way that create-attribute time stamps are modified is through a file system utility that allows a user to intentionally modify the attribute. There are several commercial and shareware applications that allow this. Therefore, by itself the create attribute doesn't prove much of anything. It serves only as supplemental evidence to support other findings. Most applications that are used to generate files also embed creation metadata within the file. If a comparison of the two values shows a difference, there is sufficient cause for the investigator to look more deeply.

Access The access attribute is the most volatile attribute of a file. Any time any user views, opens, copies, or backs up a file, this attribute is modified by the file system. Each time an executable is run, its access time is modified. Even the activity of antivirus scanning software has been known to alter the access time stamp. In fact, merely right-clicking on a file in Explorer and selecting Properties alters the access time stamp. There is no way for the investigator to accurately ascertain what action was invoked upon the file—only that one of them was. Many applications provide far more detail in their metadata concerning access information. For example, using the proper utilities, it is possible to identify the previous ten times that a document was accessed.

Modify The modify time stamp is arguably the most valuable of the time/date attributes contained within a file. This information tells when the contents of the file were last altered. Any change to the file content sufficient to alter its hash value (which is virtually any change at all) is sufficient to reset this value. Actions that change the access and create attributes do not impact modify times. The act of moving or copying a file has no impact. These actions, however, will likely impact the attributes of the folder containing the files. For example, if a user copies NOVEL.DOC from C:\Documents to C:\User\Documents, the attributes of NOVEL.DOC will change as follows:

- SOURCE FILE ENTITY — C:\Documents\NOVEL.DOC—Create time remains the same, access time is reset, modify time remains the same.
- SOURCE FILE CONTAINER — C:\Documents—Create time remains the same, access time is reset, modify time remains the same.
- DESTINATION FILE ENTITY — C:\User\Documents\NOVEL.DOC—Create time is reset, access time is reset, modify time remains the same.
- DESTINATION FILE CONTAINER — C:\User\Documents—Create time remains the same, access time is reset, modify time is reset.

Entry Modified The *entry modified* attribute (unique to NTFS) is modified each time any of the other three attributes is changed for any reason. It basically says that

something in the metadata that comprises the MFT entry for the file has changes. There is no indication of which attribute changed. By itself, this tells the investigator little, if anything. However, it does suggest that further examination is in order.

The MAC time stamps can all be easily viewed in Windows Explorer or in one of the Linux File browsers. Figure 9.6 shows these attributes displayed in the file properties of a file stored on a Windows machine. (The problem with this

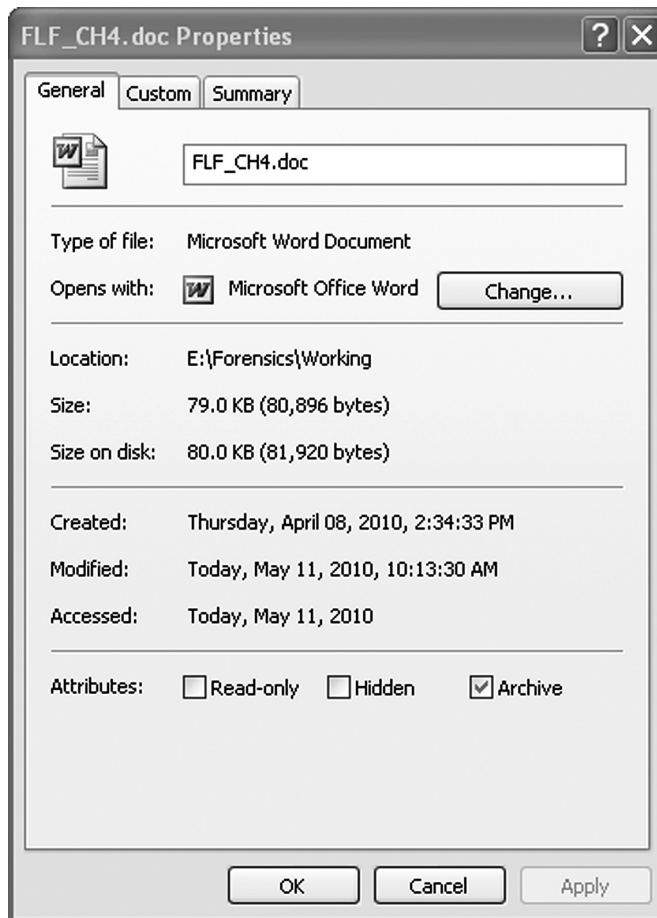


Figure 9.6 Windows Explorer is capable of displaying the Created, Accessed, and Modified file attributes.

approach is that merely viewing the file alters the accessed time stamp and is not acceptable procedure in the investigative process.) The entry modified attribute is not so easily viewed.

Using MAC

One of the first things an investigator does when approaching a new inquiry is to ask “Who did what, and when did they do it?” The who part is usually the more difficult question to answer, although the when can usually be narrowed down to a relatively narrow time frame. Once a specific time has been identified, it might be possible to identify the users who had access to the data or to begin the search for who might have gained access from beyond the network. Many applications feature filtering functions that assist in this task. Figure 9.7 illustrates a simple filter (a function of Directory Snoop) to locate all document files that were modified on a specific date. Figure 9.8 shows the results of that filter.

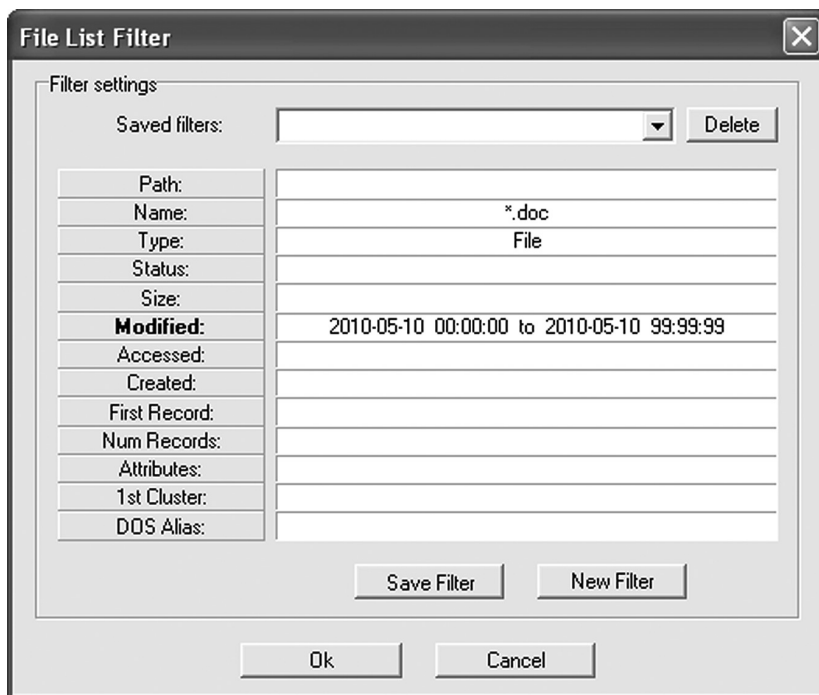


Figure 9.7 A simple filter to search for files modified on a certain date