1.

Introduction

The term metadata is an ambiguous term which is used for two fundamentally different concepts or types. Although the expression "data about data" is often used, it does not apply to both in the same way. Structural metadata, the design and specification of data structures, cannot be about data, because at design time the application contains no data. In this case the correct description would be "data about the containers of data". Descriptive metadata, on the other hand, is about individual instances of application data, the data content. In this case, a useful description would be "data about data contents" or "content about content " thus metacontent.

Metadata (metacontent) is traditionally found in the card catalogs of libraries. As information has becoming increasingly digital, metadata is also used to describe digital data using metadata standards specific to a particular discipline. By describing the contents and context of data files, the quality of the original data/files is greatly increased. For example, a webpage may include metadata specifying what language it's written in, what tools were used to create it, and where to go for more on the subject, allowing browsers to automatically improve the experience of users. An example of a metadata can include following information within it:

- Your name
- Your initials

- Non-visible portions of embedded OLE objects
- The names of previous document authors
- Document revisions
- Document versions
- Template information
- Hidden text
- Comments

1.1 Types of Metadata

From a forensic analyzer's point of view, metadata can aid forensic investigator with file name searches, timeline analysis, report generation, and decreasing the number of files that needs to be subject to thorough analysis. Metadata found on a hard drive can also be used to associate possession of the hard drive to a probable individual owner, and metadata within a file can be used to support information about the file itself. Two classes of metadata that are attention-grabbing to computer forensics investigator are file system metadata and digital media metadata.

1.1.1. Metadata In Media Files

Media files are well-known resources of metadata. It deals with various file formats like mpeg, gif, mp3, jpeg, aac, tiff etc. Metadata related with media files is significant for forensics because this type of metadata gives investigators with potentially significant proof such as camera manufacturer, serial numbers, and also the file owners. Let's take a look at the metadata mechanism drawn in in the popular media files.

1.1.1.1. GIF

Graphics Interchange Format (GIF) defines a set of rules intended for the on-line transfer and exchange of raster graphic data that does not depend of the hardware used in their creation or display. GIF is described in terms blocks and sub-blocks. The blocks and sub-blocks include parameters and data consumed in the production of a graphic. A GIF data stream is a series of set of rules blocks and sub-blocks representing a group of graphics. Blocks can be classified into three classes: Control, Graphics-Rendering, and Special Purpose. Extensions can be found within the Special Purpose class. This format supports up to 8 bits per pixel therefore permitting a single image to reference a palette of up to 256 distinct colors. These colors are selected from the 24-bit RGB color space. This format also supports animations and provides a separate palette of 256 colors for each frame

GIF was created by CompuServe in the year 1980 and it was the first extensively-used compressed image file format on the Internet. Therefore, support for GIF has been in web browsers from the beginning of the World Wide Web and still remains a very accepted file format on the Internet today[1].

Unfortunately, GIF files have some degree of metadata. However, one of the most worthwhile metadata within a GIF file is the comment extension, an optional special purpose block that contains textual information not integrated as part of the actual graphics within a data stream. Comment extensions in GIF are normally used for credits, descriptions, or other kinds of non-control and non-graphical data. The suggested positioning of the comment extension within a data stream is at the starting or at the end of the stream [1]. A second important place for metadata within a GIF file is the application extension. Similar to the comment extension, the application extension is an optional special purpose block within the data stream. The application

extension contains application related information for particular programs to act upon. The extension can be for any use, which is why only special programs will identify and operate upon the information.

Once mined, the comment extension may give insight to a forensic investigator regarding the source or possibly the purpose of the image. Identifying the depth of the comments, the file owner or an associate of the owner could be identified.

Likewise, because the application extension discloses what application used the file, forensic investigators may be able to create a tighter connection between the image, and the drive where the image was originated. Metadata extraction tool, libextractor offers a means for extracting metadata from GIF files [2].

1.1.1.2. JPEG

JPEG is the most well-known file format today for the illustration of digital photographs. Nearly all if not all digital cameras support the jpeg format. Jpegs are extensively distributed on the World Wide Web, and images of jpeg format are rich in metadata. Metadata standard of jpeg is the Exchangeable Image File (EXIF) specification, formed by the Japan Electronics and Information Technology Industries Association (JEITA).

Metadata for jpeg characterized by the International Press Telecommunications Council (IPTC) and XMP from Adobe Systems are also popular [3]. The metadata within a jpeg file can consist of

• Make, model and serial number of the digital camera that took the photograph

• Date and time picture was taken

• Distance setting for the camera's focus

• Location information where the picture was taken (from a GPS)

• Thumbnail image of the picture [3]

Identifying the types of information included within the metadata of a jpeg file is significant for many forensic applications. For instance, many digital photograph software applications do not alter the thumbnail image within the metadata when the primary picture is edited. If any picture is taken of a subject in a rather compromising position, the subject may not be alert that although the compromising pose has been cropped out, the thumbnail image of the pose is likely still intact [3]. If forensic investigators get a camera from a crime, the location detail from where a picture was taken can help out forensic investigators that are solving a criminal puzzle. If forensic investigators get a camera from a crime scene, and the images contain additional evidence of a crime, being able to attach the images to a location can provide the investigators with additional leads. In addition, with information on the distance setting of the camera's focus, given the location of the photographed subject, it may be feasible to pin point the exact location of the photographer [3]. More, adding the date and time information into the group of evidence can tighten the timeline for the forensic investigators.

1.1.1.3. Music - MP3 and AAC

Advanced Audio Coding (AAC) and Moving Picture Expert Group – 1 (MPEG-1), audio layer 3 (MP3) characterize lossy, compressed, perceptual coding method formats that are division of the MPEG set of standards for music encoding. AAC was improved for the MPEG-4 standard as

MPEG-4 AAC. MPEG-4 audio supports a group of applications that vary from speech to high-quality multi-channel audio, and from natural to synthesized sounds. The most important advantage of MPEG-4 AAC over MP3 audio is that the clarity of MPEG-4 is just about twice that of MP3 for similar bit rates, and files half the size for the same supposed quality. Both AAC and MP3 formats comprise metadata specifications; however, separate MP3 files do not include a standard method of storing the metadata. In 1996 Eric Kemp created a simple method for adding a small chunk of data to the end of the audio file. The standard for containing this metadata was called ID3v1 for "IDentify an MP3". The ID3v1 tag takes up the last 128 bytes of an MP3 file and starts with the string "TAG". The tag allots 30 bytes for the title, artist, album, and a "comment," 4 bytes for the year, and 1 byte as a predefined genre identifier located at the end of the file. ID3v1 tags were measured too short to contain sufficient meaningful metadata, so in 1998 the ID3v2 standard was launched. Each ID3v2 tag holds one or more frames, which can be up to 16 MB in size for a total of 256 MB per tag. Each frame can have any type of metadata information such as lyrics, album, title, artist, equalizer presets, website, copyright information, media type, and pictures. Basically ID3v2 is another container specification. ID3v2 also gives some flexibility for providing user added metadata [4].

Even though AAC files also comprise metadata, the information is not stored within an ID3 tag [4]. Apple in its place uses a proprietary audio file format known as the Apple Core Audio Format for audio data. The files inside the Core Audio Format are chunk-based and contain AAC data formats. In particular, Apple uses protected AAC to encode copy-protected music titles acquired from the iTunes Music Store [5].

The files acquired from the iTunes Music Store consists of the following metadata.

• Name

• Email address of purchaser

• Year

• Album

• Grouping

• Comments

• Genre

• Lyrics

• Artwork [5]

The metadata is helpful for various reasons. First, the iTunes and iPod user interface is made from the metadata. Then, the metadata improves the effectiveness of browsing and searching for files. At last, the metadata helps to support and reinforce the content. From a forensics point of view, the metadata can be helpful in discovering the owner or relating the files to a potential owner. The insertion of the email address gives one link, and user added comments or pictures could give another link. There are a small number of tools on hand for extracting metadata from the ID3 tagged MP3 files and the AAC files. One tool is the MP3::Tag module, which is one of the Perl ID3 tag parsers that are easily available on CPAN (Comprehensive Perl Archive Network). MP3:: Perl Tag reads both ID3v1 and ID3v2 tags. In addition, MP3::Tag provides the user to edit the content of an MP3 file tag or create a new tag. One more tool that is capable of extracting metadata from MP3 files is libextractor [2]. In support of AAC files, the MPEG4IP application provides many tools for working with audio and visual files including files in AAC

format. Particularly, the mp4dump is a service that can extract mp4 file metadata into a text form. The MPEG4IP application also includes an mp4info service that gives mp4 file summary information [6].

1.1.1.4 Tagged Image File Format (TIFF)

Tagged Image File Format (TIFF) files are container files that can contain metadata and data in a various formats. Metadata information that can be contained in a TIFF version 6.0 file includes:

• The date and time that the image was created.

• An image description

• Make and model of the equipment that produced the image

• Software version that produced the image

• Creator of the image (artist)

• Copyright holder

1.1.2 File System Metadata

File system metadata generally contains metadata such as the specific data units' allotted to a file, access date and time stamps for the file, the size of the file. The exact metadata preserved depends on the file system (e.g., ext2/ext3, FAT, NTFS), and the data structure employed by that file system. Let's look at the metadata method involved in the well known file system NTFS and ext3.

1.1.2.1 NTFS

NTFS (New Technology File System) is the standard file system of Windows NT, including its later versions Windows 2000, Windows XP, Windows Server 2003, Windows Server 2008, Windows Vista, and Windows 7. NTFS supersedes the FAT file system as the preferred file system for Microsoft’s Windows operating systems. NTFS has several improvements over FAT which is improved support for metadata

The NTFS file system stores virtually all data, both user data and internal management data, in the form of files. The most important of these are a set of special system files, which are also called metadata files. The prefix "meta-" generally refers to something "transcendent" or "beyond"--or merely self-referring. So "metadata files" are files that contain data about data. And that's exactly what these files do. They contain internal information (data) about the "real" data stored on the NTFS volume.

These metadata files are created automatically by the system when an NTFS volume is formatted, and are placed at the beginning of the partition..Understanding their location in an NTFS volume requires another key structure, the NTFS Master File Table (MFT). The MFT is actually one of these metadata files, but it also contains descriptions of the other metadata files, and in some cases entire other metadata files. The MFT contains a record describing every file and directory in an NTFS volume, and if the file is small enough, its actual contents may be stored in the MFT itself. Since the metadata files are just "files" to NTFS (albeit special ones), they too have records in the MFT. In fact, the first 16 records of the MFT are reserved for metadata files.

1.1.2.2 ext3

The ext3 or third extended filesystem is a journaled file system that is commonly used by the Linux kernel. It is the default file system for many popular Linux distributions. The ext3 file system is a journaled file system because keeps it track of the changes that will be made in a journal (usually a circular log in a dedicated area of the file system) before committing them to the main file system.

There are three levels of journaling available in the Linux implementation of ext3:

- Journal (lowest risk) : Both metadata and file contents are written to the journal before being committed to the main file system. Because the journal is relatively continuous on disk, this can improve performance in some circumstances. In other cases, performance gets worse because the data must be written twice - once to the journal, and once to the main part of the filesystem.

- Ordered (medium risk) :Only metadata is journaled; file contents are not, but it's guaranteed that file contents are written to disk before associated metadata is marked as committed in the journal. This is the default on many Linux distributions. If there is a power outage or kernel panic while a file is being written or appended to, the journal will indicate the new file or appended data has not been "committed", so it will be purged by the cleanup process. (Thus appends and new files have the same level of integrity protection as the "journaled" level.) However, files being overwritten can be corrupted because the original version of the file is not stored. Thus it's possible to end up with a file in an intermediate state between new and old, without enough information to restore either one or the other (the new data never made it to disk completely, and the old data is not stored anywhere). Even worse, the intermediate state might intersperse old and new data, because the order of the write is left up to the disk's hardware. XFS uses this form of journaling.

- Writeback (highest risk) : Only metadata is journaled; file contents are not. The contents might be written before or after the journal is updated. As a result, files modified right before a crash can become corrupted. For example, a file being appended to may be marked in the journal as being larger than it actually is, causing garbage at the end. Older versions of files could also appear unexpectedly after a journal recovery. The lack of synchronization between data and journal is faster in many cases. JFS uses this level of journaling, but ensures that any "garbage" due to unwritten data is zeroed out on reboot.

In all three modes, internal structure of file system is assured to be consistent even after a crash. Risk only affects user data content and users metadata of files. In any cases only data content of files or directories which was modified when system crashed will be affected, rest will be intact after recovery.

1.1.3 Metadata in Document Files

Metadata in document files is used generally by Microsoft Office products,OpenOffice.org applications, and portable document format (PDF) documents to contain information such as the, date of creation, document's creator, date and last time it was saved, the location of the document where the document was saved, as well as contributors to the document. Metadata is very helpful at various occasions and helps to facilitate the cooperation amongst several people. It provides the application, to keep track of changes and comments made to a document by corresponding reviewers. The document author can choose to allow or refuse the reviewer suggested document changes and view any comments presented.

1.1.3.1 Microsoft Office

Microsoft Office document is one of the most well known user generated documents in existence today. Some general examples of Microsoft Office documents comprise Microsoft Word, Excel, and Power Point documents. For Microsoft Office document versions 1995 – 2003, Microsoft used Object Linking and Embedding (OLE) technology to handle its file format. OLE is technology that allows applications to generate compound documents from several sources [7]. With Microsoft Office 2007, Microsoft began using the Office Open XML format. Microsoft Office documents hold a range of metadata. By simply opening a Microsoft Word document and selecting properties from the File menu, metadata such as company, author, title, subject, comments, modified, accessed, and created times, keywords, and file location can easily be found. In addition, in some documents, information can be found regarding document reviewers and savers.

Microsoft Office documents also hold hidden information, data that is not accessible through the Office application's interfaces, or that can be hidden from view using application settings. Examples of hidden data comprise track changes and comments, which can be hidden from view as a default setting, and author history and fast save data, which are not reachable from the native application interfaces. Numerous stories have surfaced concerning hidden information in Microsoft Word documents. One of the most famous involved a UK Government issued document regarding Iraq's concealment of weapons of mass destruction. The Word version of the document was placed on the Internet, and when the document was analyzed, the contents were discovered to be largely based on a journal article written by a postgraduate student in the United States. A deeper look into the document's revision log revealed the names of four of the people, who prepared the document for publication along with the offices where some of them worked.

Computer forensics experts can also get valuable information or case leads from metadata and hidden data included in Microsoft Office documents. The BTK killer was caught easily after police were able to extract the metadata information involving his name and the name of the church of which he was president, housing the computer he used to create a Microsoft Word document he sent to the police. There are some tools available for extracting metadata from Microsoft OLE formatted files. One is the wv library. The wv library can load and parse Microsoft Word 2003, 2000, 97, 95 and 6 file formats. Included with wv is an application called wvWare, a command-line application with helper scripts–one of which is used for extracting metadata information from files. WvSummary is a helpful script that prints out the metadata from Microsoft Office documents [8].

1.1.3.2 Office Open XML Format (Microsoft Office 2007)

From the release of Microsoft Office 2007, Microsoft started using its Office Open XML format. Microsoft competes that the new formats improve file and data management, interoperability and, data recovery with line-of-business systems. Under this format, any application that supports XML can access and work with data in the new format [9]. This section gives an introductory look at the new Office Open format. Table 1 below presents a list of the different Office Open XML file types and their extensions [9].

Table 1. Office Open XML file types

Document Format	File Extension

Word 2007 XML Document	*.docx
Word 2007 XML Macro-Enabled Document	*.docm
Word 2007 XML Template	*.dotx
Word 2007 XML Macro-Enabled Template	*.dotm
Excel 2007 XML Workbook	*.xlsx
Excel 2007 XML Macro-Enabled Workbook	*.xlsm
Excel 2007 XML Template	*.xltx
Excel 2007 XML Macro-Enabled Template	*.xltm
Excel 2007 XML Binary Workbook	*.xlsb
Excel 2007 XML Macro-Enabled Add-In	*.xlam
PowerPoint 2007 Macro-Enabled XML Presentation	*.pptm
PowerPoint 2007 XML Template	*.potx
PowerPoint 2007 Macro-Enabled XML Template	*.potm
PowerPoint 2007 Macro-Enabled Add-In	*.ppam
PowerPoint 2007 XML Show	*.ppsx
PowerPoint 2007 Macro-Enabled XML Show	*.ppsm

The new file format of Microsoft Office 2007 container is based on the ZIP compressed file format specification (Figure 2). The document elements are stored in the file container and are

mostly consisted of of XML files that explain document properties, application data, metadata, and customer data [9].

There are various exciting files and folders that include the various component divisions of an Office Open XML document. A brief explanation of each is presented below.

• Application Folder – It contains the application particular document component files for instance Excel, Word, Power Point.

• Media Folder – It contains the image.jpeg files that were implanted within a document as picture content controls.

• Audio File – It contains any audio files such as .mid, .mp3, or .wav. These files potentially store their own metadata and could be attractive to a forensic investigator on their own accord.

• rels Folder - It contains a .rels file that describes the root relationship inside the package. The .rels file inside this folder contains a unique relationship Id.

• document.xml file – It is an XML file that stores the main text and body of the document [9]

Additionally to the ordinary metadata such as author, creation date, modified date, thumbnail image, revision number, filename found in the files and folders listed above, users can put in their own data and content by generating custom-defined xml and placing it in the file as other part [9].

Inside a .docx document, there is a Revision Identifier for a paragraph (rsidR) whose values are used to recognize the editing session in which a paragraph was included to a main document. An rsidR value should be unique, and examples with the same value within a single document point

to that the regions were customized during the same editing session. Connected with an rsidR attribute is the rsidRDefault attribute.

An rsidRDefault attribute is particularly used for all runs inside a paragraph in which an rsidR is not described. Overall, Office 2007 documents store a considerable amount of information that can be extracted from a .docx package ahead of the standard author, creation/modification dates, that were obtained from previous versions of Office, and the Office Open XML format gives forensic investigators with a simpler environment in which they can extract preferred metadata from the xml files. For example, a simple method could be written to generate a list of documents written by a particular user, or a list of comments extracted from the xml files [9]. Additionally, the addition of a thumbnail.jpeg file or other embedded image inside the package provides forensic investigators with extra metadata which can be obtained by running the Exif program on the file. The metadata can then be examined by forensic investigators. At last by examining rsid values within the XML, investigators can recognize documents that were generated during the same editing session, but later isolated and modified separately.

1.1.3.3 Open Office

A similar file structure to the Office Open XML format is the Open Document Format Alliance (ODF) which was created to facilitate sharing of information between different word processing applications. Like Office Open XML files, an ODF file is basically a zipped archive comprised of several XML files. The exact files and directories contained within an ODF file will differ depending on the systems the document was created on and the type of information stored within the ODF file itself. Table 2 provides a list of different ODF file types and their extensions. When an ODF text file is unzipped, the archive will have some of the following files:

• mimetype

• content.xml

• styles.xml

• meta.xml

• settings.xml

• META-INF/manifest.xml [28]

Table 2. ODF File Types

Document Format	File Extension
OpenDocument Text	*.odt
OpenDocument Text Template	*.ott
OpenDocument Master Document	*.odm
HTML Document	*.html
HTML Document Template	*.oth

OpenDocument Spreadsheet	*.ods
OpenDocument Spreadsheet Template	*.ots
OpenDocument Drawing	*.odg
OpenDocument Presentation	*.odp
OpenDocument Presentation Template	*.otp
OpenDocument Formula	*.odf
OpenDocument Database	*.odb

Of these files, mimetype, meta.xml, META-INF/manifest.xml, and content.xml offer the most potential for extracting meaningful metadata. The mimetype file is a single line file with the mimetype of the content file. META-INF/manifest.xml contains a listing of all the files in the zipped archive, and meta.xml holds the metadata such as the author and creation dates. Although not necessarily a source of metadata, the content.xml file should be identified because this file contains the data for the document itself. The potential uses of metadata associated with Open Office documents follow the same path as the uses of metadata with Microsoft Office 2007 documents. Investigators can readily search for all files created by the same author, extract files based on creation or modification dates to build timelines, and they can generate associations between files with the same authors appearing on different drives. Additionally, the manifest.xml file provides a resource for investigators to use when accounting for all files in the archive. If an

investigator notices that a file is listed in the manifest.xml file, but is not included in the archive directory structure, the investigator now has a lead to search for a potentially interesting file that has been deleted and should be recovered.

1.1.3.4 Portable Document Format (PDF)

Adobe PDF is the organic file format used by the Adobe family of products. The primary goal of the PDF is to allow users to easily exchange and view unmodifiable electronic documents. A PDF file may contain metadata such as title, author, creation date, and modification date. Metadata within a PDF file can be stored in one of two ways:

• In a document information dictionary

• In a metadata stream.

Within the trailer of a PDF file, there is an optional Info entry that holds the document information dictionary containing metadata for the document. The contents of the document information dictionary are as follows:

• Title - document's title (optional)

• Author - name of the person who created the document (optional)

• Subject - subject of the document (optional)

• Keywords - keywords associated with the document (optional)

• Creator – name of application that was used to originally create the document if document was converted to PDF (optional)

• Producer – name of application used to convert document to PDF from another format if a conversion took place (optional)

• CreationDate - date and time the document was created (optional)

The second way to store metadata within a PDF file is in a metadata stream. Metadata streams hold two primary advantages over document information dictionaries.

First, metadata-bearing artwork is frequently embedded as a component within larger documents by PDF-based workflows, so metadata streams standardize the preservation of these components for future examination.

Second, because PDF documents are usually available on the Internet or other environment, the documents should be easily examined, catalogued, and classified by the many tools used to perform these functions. The tools should be able to comprehend the self-contained description of the document even if the tools do not understand PDF [10].

XML is used to represent the contents of a metadata stream. The XML is visible as plain text only if the tools are PDF aware, or if the tools are not PDF aware if the metadata stream is both unfiltered and unencrypted. The specific format of XML used is defined as part of the Extensible Markup Platform framework [10]. The primary function of metadata within the PDF document is to facilitate cataloguing and searching for documents in external databases [10]. The metadata retrieved from PDF files offers investigators many of the same benefits as metadata retrieved from Microsoft Office and Open Office documents. One point of interest is that many users will

“convert” their Microsoft Office documents to a PDF format to eliminate the possibility of disclosing hidden information. The information available depends on the authoring software used to create the document. Libextractor is one of the tools currently available for extracting metadata from PDF files [2].

1.2 Applications Of Metadata

Metadata are widely used in various applications. Metadata has use for data developers, data managers, data users, and organizations. Standardized metadata documentation is searchable, allowing data developers and users to search for existing data and avoid data duplication. It provides a venue for sharing and publicizing data production efforts. These both reduce workloads and increase efficiency. It allows searching for specific geographic locations and gives information on data acquisition and transfer. In an organization, metadata increases and protects the value of its investment in data. Data productions and planned acquisition can be managed through metadata. Quality control, data restrictions and uses can be applied to the entire data in holdings. Metadata documentation transcends people and time. Staff turnover and balancing of multiple projects can be mitigated with metadata, providing data permanence and the documentation of institutional knowledge. It is also used in search engine to facilitate quick and effective search. Typical application areas of metadata include following[11]:

1.2.1 Data Virtualization

Data Virtualization has emerged as the new software technology to complete the virtualization stack in the enterprise. Metadata is used in Data Virtualization servers which are enterprise infrastructure components, alongside with Database and Application servers. Metadata in these servers is saved as persistent repository and describes business objects in various enterprise systems and applications[11].

1.2.2 Statistics and census services

Standardization work has had a large impact on efforts to build metadata systems in the statistical community. Several metadata standards are described, and their importance to statistical agencies is discussed. Applications of the standards at the Census Bureau, Environmental Protection Agency, Bureau of Labor Statistics, Statistics Canada, and many others are described. Emphasis is on the impact a metadata registry can have in a statistical agency[11].

1.2.3 Library and information science

Libraries employ metadata in library catalogues, most commonly as part of an Integrated Library Management System. Metadata is obtained by cataloguing resources such as books, periodicals, DVDs, web pages or digital images. This data is stored in the integrated library management system, ILMS, using the MARC metadata standard. The purpose is to direct patrons to the

physical or electronic location of items or areas they seek as well as to provide a description of the item/s in question[11].

More recent and specialized instances of library metadata include the establishment of digital libraries including e-print repositories and digital image libraries. While often based on library principles the focus on non-librarian use, especially in providing metadata means they do not follow traditional or common cataloging approaches. Given the custom nature of included materials metadata fields are often specially created e.g. taxonomic classification fields, location fields, keywords or copyright statement. Standard file information such as file size and format are usually automatically included.

Standardization for library operation has been a key topic in international standardization (ISO) for decades. Standards for metadata in digital libraries include Dublin Core, METS, MODS, DDI, ISO standard Digital Object Identifier (DOI), ISO standard Uniform Resource Name (URN), PREMIS schema, Ecological Metadata Language, and OAI-PMH. Leading libraries in the world give hints on their metadata standards strategies[11].

1.2.4 Metadata and the law and forensics

Problems involving metadata in litigation in the United States are becoming widespread Courts have looked at various questions involving metadata, including the discoverability of metadata by parties. Although the Federal Rules of Civil Procedure have only specified rules about electronic documents, subsequent case law has elaborated on the requirement of parties to reveal

metadata. In October 2009, the Arizona Supreme Court has ruled that metadata records are public record.

Document Metadata has proven particularly important in legal environments in which litigation has requested metadata, which can include sensitive information detrimental to a party in court.Using metadata removal tools to "clean" documents can mitigate the risks of unwittingly sending sensitive data. This process partially (see Data remanence) protects law firms from potentially damaging leaking of sensitive data through Electronic Discovery.

Metadata plays a number of important roles in computer forensics:

- It can provide corroborating information about the document data itself.
- It can reveal information that someone tried to hide, delete, or obscure.
- It can be used to automatically correlate documents from different sources.

Since metadata is fundamentally data, it suffers all of the data quality and pedigre issues as any other form of data. Nevertheless, because metadata isn't generally visible unless you use a special tool, more skill is required to alter or otherwise manipulate it[11].

1.2.5 Metadata in healthcare

Australian researches in medicine started a lot of metadata definition for applications in health care. That approach offers the first recognised attempt to adhere to international standards in medical sciences instead of defining a proprietary standard under the WHO umbrella first.
The medical community yet did not approve the need to follow metadata standards despite respective research[11].

1.2.6 Metadata and data warehousing

Data warehouse (DW) is a repository of an organization's electronically stored data. Data warehouses are designed to manage and store the data whereas the Business Intelligence (BI) focuses on the usage of data to facilitate reporting and analysis.

The purpose of a data warehouse is to house standardized, structured, consistent, integrated, correct, cleansed and timely data, extracted from various operational systems in an organization. The extracted data is integrated in the data warehouse environment in order to provide an enterprise wide perspective, one version of the truth. Data is structured in a way to specifically address the reporting and analytic requirements.

An essential component of a data warehouse/business intelligence system is the metadata and tools to manage and retrieve metadata. Ralph Kimball describes metadata as the DNA of the data warehouse as metadata defines the elements of the data warehouse and how they work together.

Kimball refers to three main categories of metadata: Technical metadata, business metadata and process metadata. Technical metadata is primarily definitional while business metadata and process metadata are primarily descriptive. Keep in mind that the categories sometimes overlap.

- **Technical metadata** defines the objects and processes in a DW/BI system, as seen from a technical point of view. The technical metadata includes the system metadata which defines the data structures such as: Tables, fields, data types, indexes and partitions in the relational engine, and databases, dimensions, measures, and data mining models. Technical metadata defines the data model and the way it is displayed for the users, with the reports, schedules, distribution lists and user security rights.

- **Business metadata** is content from the data warehouse described in more user friendly terms. The business metadata tells you what data you have, where it comes from, what it means and what its relationship is to other data in the data warehouse. Business metadata may also serves as documentation for the DW/BI system. Users who browse the data warehouse are primarily viewing the business metadata.
- **Process metadata** is used to describe the results of various operations in the data warehouse. Within the ETL process all key data from tasks are logged on execution. This includes start time, end time, CPU seconds used, disk reads, disk writes and rows processed. When troubleshooting the ETL or query process, this sort of data becomes valuable. Process metadata is the fact measurement when building and using a DW/BI system. Some organizations make a living out of collecting and selling this sort of data to companies - in that case the process metadata becomes the business metadata for the fact and dimension tables. Process metadata is in interest of business people who can use the data to identify the users of their products, which products they are using and what level of service they are receiving[11].

1.2.7 Metadata on the Internet

The HTML format used to define web pages allows for the inclusion of a variety of types of metadata, from basic descriptive text, dates and keywords to further advance metadata schemes such as the Dublin Core, e-GMS, and AGLS standards. Pages can also be geotagged with coordinates. Metadata may be included in the page's header or in a separate file. Micro formats

allow metadata to be added to on-page data in a way that users do not see, but computers can readily access.

Interestingly, many search engines are cautious about using metadata in their ranking algorithms due to exploitation of metadata and the practice of search engine optimization, SEO, to improve rankings. See Meta element article for further discussion[11].

1.2.8 Metadata on the broadcast industry

In broadcast industry, metadata are linked to audio and video Broadcast media to:

- identify the media: clip or playlist names, duration, timecode, etc.
- describe the content: notes regarding the quality of video content, rating, description (for example, during a sport event, keywords like goal, red card will be associated to some clips)
- classify media: metadata allow to sort the media or to easily and quickly find a video content (a TV news could urgently need some archive content for a subject).

These metadata can be linked to the video media thanks to the video servers. All last broadcasted sport events like FIFA World Cup or Olympic Games use these metadata to distribute their video content to TV stations through keywords. It's often the host broadcaster who is in charge of organizing metadata through its International Broadcast Centre and its video servers. Those metadata are recorded with the images and are entered by metadata operators (loggers) who associate in live metadata available in metadata grids through software (such as Multicam(LSM) or IPDirector used during FIFA World Cup or Olympic Games)[11].

1.2.9 Geospatial metadata

Metadata that describe geographic objects (such as datasets, maps, features, or simply documents with a geospatial component) have a history dating back to at least 1994 (refer MIT Library page on FGDC Metadata). This class of metadata is described more fully on the Geospatial metadata page[11].

1.2.10 Ecological & environmental metadata

Ecological and environmental metadata are intended to document the who, what, when, where, why, and how of data collection for a particular study. Metadata should be generated in a format commonly used by the most relevant science community, such as Darwin Core, Ecological Metadata Language, or Dublin Core. Metadata editing tools exist to facilitate metadata generation (e.g. Metavist, Mercury: Metadata Search System, Morpho). Metadata should describe provenance of the data (where it originated, as well as any transformations the data underwent) and how to give credit for (cite) the data products.

1.2.11 Metadata on CDs and DVDs

CDs such as recordings of music will carry a layer of metadata about the recordings such as dates, artist, genre, copyright owner, etc. The metadata, not normally displayed by CD players, can be accessed and displayed by specialized music playback and/or editing applications.

1.2.12 Cloud applications

With the availability of Cloud applications, which include those to add metadata to content, metadata is increasingly available over the Internet.

1.2.13 Metadata storage

Metadata can be stored either internally, in the same file as the data, or externally, in a separate file. Metadata that is embedded with content is called embedded metadata. A data repository typically stores the metadata *detached* from the data. Both ways have advantages and disadvantages:

- Internal storage allows transferring metadata together with the data it describes; thus, metadata is always at hand and can be manipulated easily. This method creates high redundancy and does not allow holding metadata together.
- External storage allows bundling metadata, for example in a database, for more efficient searching. There is no redundancy and metadata can be transferred simultaneously when using streaming. However, as most formats use URIs for that purpose, the method of how the metadata is linked to its data should be treated with care. What if a resource does not have a URI (resources on a local hard disk or web pages that are created on-the-fly using a content management system)? What if metadata can only be evaluated if there is a connection to the Web, especially when using RDF? How to realize that a resource is replaced by another with the same name but different content?

Moreover, there is the question of data format: storing metadata in a human-readable format such as XML can be useful because users can understand and edit it without specialized tools. On the

other hand, these formats are not optimized for storage capacity; it may be useful to store metadata in a binary, non-human-readable format instead to speed up transfer and save memory.

1.2.14 Database management

Each relational database system has its own mechanisms for storing metadata. Examples of relational-database metadata include:

- Tables of all tables in a database, their names, sizes and number of rows in each table.
- Tables of columns in each database, what tables they are used in, and the type of data stored in each column.

In database terminology, this set of metadata is referred to as the catalog. The SQL standard specifies a uniform means to access the catalog, called the INFORMATION_SCHEMA, but not all databases implement it, even if they implement other aspects of the SQL standard. For an example of database-specific metadata access methods, see Oracle metadata. Programmatic access to metadata is possible using APIs such as JDBC, or SchemaCrawler.

Some kinds of metadata those are interesting in computer forensics:

- File system metadata (e.g. MAC times, access control lists, etc.)
- Digital image metadata. Although information such as the image size and number of colors are technically metadata, JPEG and other file formats store additional data about the photo or the device that acquired it.
- Document metadata, such as the creator of a document, it's last print time, etc[11].

1.3 Metadata Management

Meta-data management (also known as metadata management, without the hyphen) involves storing information about other information. With different types of media being used, references to the location of the data can allow management of diverse repositories[12].

URLs, images, video etc. may be referenced from a triples table of object, attribute and value. With specific knowledge domains, the boundaries of the metadata for each must be managed, since a general ontology is not useful to experts in one field whose language is knowledge-domain specific.

If anyone in the process of making a knowledge management solution, managing the metadata is very important. In such a project, one would typically appoint a metadata manager. This is a person who will be responsible for the metadata strategy, and for the implementation of this. A metadata manager does not need to know about and be involved with everything concerning the solution, but rather, will have overall responsibility. Managing the metadata in a knowledge management solution is an important step of a metadata strategy. It is part of the strategy to make sure that at any given point in time, the metadata are complete, current and correct. And it is about making sure that users of the solution are aware of the possibilities and how to use these possibilities. It is very important to monitor the metadata, constantly making sure that the knowledge management solution provides data that corresponds with organizational requirements.

Though it's difficult to find an exact analogy for Metadata, here is one effort. Meta-Data can also be considered as an equivalent of Amazon book store. If we consider each data element as a book, the meta-data will contain name of the book, summary of the book, assessments about the book, the date of publication, high level description of what it contains, who are the publishers,

how you can find the book, author of the book, whether the book is available OR not. This information helps you to: [12]

- Search for the book
- Access the book
- Understand about the book before you access OR buy it.

Metadata Management is not a pure business intelligence (BI) subject, though it has many applications around BI components like Data Warehouse and OLAP. Metadata Management (as obvious from the definition) serves every possible stakeholder within an organization. Technically speaking, it can be used even by people who have never seen a computer. Enterprise Resource Planning (ERP) systems are as much a stakeholder for Meta-data project as a data Warehouse.

Metadata (like other data management and BI initiatives) is a business owned initiative. Business provides sponsorship and also most of the components of metadata. Metadata in its extremely restricted form (whereby, only the data lying in the IT systems), may be thought of as an IT initiative. However, even for the data lying in IT systems have a major Business stake-holding. It's true that a well-functioning metadata repository will require IT platforms, but that part comes in the 'support' domain and not in 'ownership domain'

Metadata management (like other data management initiatives like data quality), comes under the ownership and accountability of a data steward (which is a business role). Refer Maximizing Effectiveness of Data Steward and Business Intelligence Organization- Roles & Skills for a better perspective on this role. In metadata, a data steward[14] is a person that is responsible for maintaining a data element in a metadata registry. Systematic data stewardship can foster: consistent

use of data management resources, easy mapping of data between computer systems and exchange documents, lower costs associated with migration to (for example) Service Oriented Architecture (SOA).

1.3.1 Metadata as a Component of Data Management

1.3.1.1 Metadata Repository

Metadata repository contains all details on an organization management environment. These details are placed in a central repository OR in well-connected synchronized repositories (for example Data Warehouse could be having its own repository and an ERP could be having its own). An ideal situation (for which we are not aware of a case-study) is to have a single repository. When there are multiple repositories, one does run into a challenge of making sure that they are well-synchronized and integrated. There have been issues around creating a single metadata standard. Typically organizations rely on XML to integrated and exchange metadata information.

1.3.1.2 Meta-Data Model

A metadata repository has its own data model. For example there will be a data-model for storing database tables. A typical data model for database table metadata will contain fields like name of the table, location of the table, data of table creation, systems accessing the table etc.

1.3.1.3 Usage

Whenever one wants to have any usage of a given set of metadata, one can log into a data repository and get information on that meta-data. For example, if you are looking for Standards of Business Conduct, you may log into the metadata repository and find out on its intranet address OR the person who has the custody from where you can ask for a copy. The systems also use meta-data to conduct their operations. For example, before every extraction done by a data warehouse, it will check the meta-data on the location of the database tables of the source systems (this is called the 'operational use' of metadata).

1.3.1.4 Metadata Management

Metadata Repository is one of the outcomes of metadata management process. Metadata management is an end-to-end process for creating, enhancing and maintaining meta-data repository and associated processes. Metadata management includes establishing processes, mind-sets, organization and capabilities to build a metadata environment. Like BI and Master Data Management, bigger challenge on meta-data management is related business process discipline and culture.

1.3.2 REASONS FOR METADATA MANAGEMENT

1. 3.2.1 Data Quality

Data quality is driven by a common set (and common understanding) of data standards, domain standards, business rules etc. If the systems follow the common standards (creating same checks,

controls, table structure, field definitions...) there can be a big gain on data quality. Metadata repository:

- Provides the details on the data standards to follow
- Enforces the adherence to the standards as defined in the repository.

1.3.2.2 IT systems productivity

Given that data standards, business rules, and models etc. exist in the metadata, one builds productivity on following counts:

- Automatic creation of the tables and models**:** Systems can pick-up the details from the metadata repository and build the components. This will save time and effort to firstly creating the models and then build them.
- Avoid cost of mistakes and iteration**:** One may not have to go through the pains of change controls, if your design is built from common standards

1.3.2.3 Avoiding duplication

If data is available in the system, one does not have to re-create it. If you are looking for 'sales productivity MIS', you may find it (OR something close to it) by searching through metadata. This gives a boost to the business effectiveness as other-wise, they will need to wait for their turn in the queue. This also helps in focusing the resources on fulfilling new business requirement, instead of re-creating the old ones.

1.3.2.4 Avoiding information conflict issues

By using metadata repositories and enforcing common standards and calculation formulae, the reports and dashboards will have a greater probability of reflecting same figures. This will avoid board room time waste on find which are the correct figures.

1.3.2.5 Regulatory compliance

With all the above benefits, one can expect that business will be able to produce correct reports faster and cheaper.

1.3.2.6 Business Process Management and its cascading impacts

With every change in business processes, one can find the cascading impact on various components like policies, business process documentation, business rules, configuration and set-up changes in IT systems . For example if a new business process allows a sales manager to manage more than one outlet, it will have a cascading impact on the set-ups, software changes, ETL and dimensional models.

1.3.2.7 Handling any kind of change management

Whenever anything changes with-in an organization environment, metadata repository helps you to understand the impact. For example, if you want to change ‘maker checker’ control policy, Metadata repository will be able to tell you on which all systems, database, business processes you have to change.

1.3.2.8 Better estimations and business case management

With metadata repository telling you the impact of a requirement and also providing some efficiency gains, one can do a better estimate of the cost of making a change.

1.3.2.9 Making scalable and extensible models

This is not a direct benefit of Metadata repository, but it supports it. Smart modelers (with solid business knowledge), can help create models (for example Foundation Dimensions and Facts in Dimensional Model of a data warehouse) which can quickly respond to the changes. A metadata helps you to manage this modeling.

1.3.2.10 Reduce redundancy

With all the data elements maps stored in the metadata repository, one can identify the redundant data and processes, and work on their reduction OR elimination.

From the above-said benefits, one can understand that Metadata management is core to building intelligent and high-performing enterprises. It benefits all facets of an organization including business process management, BI, IT management, performance management and so on. There is a cascading impact on better business performance, employee satisfaction and customer satisfaction.

1.4 METADATA ARCHITECTURE

A sound meta data architecture incorporates five general characteristics:

- Integrated
- Scalable
- Robust
- Customizable
- Open[13]

Integrated

Anyone who has worked on a decision support project understands that the biggest challenge in building a data warehouse is integrating all of the disparate sources of data and transforming the data into meaningful information. The same is true for a meta data repository. A meta data repository typically needs to be able to integrate a variety of types and sources of metadata and turn the resulting stew into meaningful, accessible business and technical meta data. For example, a company may have a meta data require-ment to show its business users the business definition of a field that appears on a data warehouse report. The company probably used a data modeling tool to construct the physical data models to store the data presented in the report's field. Let's say the business definition for the field originates from an outside source (i.e., it is external meta data) that arrives in a spreadsheet report. The meta data integration process must create a link from the meta data on the table's field in the report to the business definition for that field in the spreadsheet. When we look at the process in this way, it's easy to see why integration is no easy feat. (Just consider creating the necessary links to all of the various types and sources of data and the myriad delivery forms that they involve.) In fact, integrating the data is probably the most complex task in the meta data repository implementation effort.

Scalable

If integration is the most difficult of the meta data architecture characteristics to achieve, scalability is the most important characteristic. A meta data repository that is not built to grow, and grow substantially over time, will soon become obsolete. Three factors are driving the current proliferation of meta data repositories:

- Continuing growth of decision support systems
- Recognition of the value of enterprise-wide meta data
- Increasing reliance on knowledge management

Robust

As with any system, a meta data repository must have sufficient functionality and performance to meet the needs of the organization that it serves. The repository's architecture must be able to support both business and technical user reports and views of the meta data, as well as providing acceptable user access to these views. Some of the other functionality required from the meta data architecture includes: Ability to handle time- or activity-generated events, Import/export capability, Support for data lineage, Security setup and authorization facilities, Archival and backup facilities, Ability to produce business and technical reports

Customizable

If the meta data processes are home-grown (i.e., built without the use of meta data integration or access tools), then customization is not a problem since the entire application is tailored for the specific business environment. If, however, a company uses meta data tools to implement the repository architecture (as most do), the tools need to be customized to meet the specific current and future needs of the meta data initiative. Customization is a major issue for companies that

purchase prepackaged meta data solutions from software vendors. These solutions are generally so rigid in their architecture that they cannot fill the specific needs of any company. In the case of a meta data solution, one size definitely does not fit all! To be truly effective, these prepackaged solutions require a significant amount of customization to tailor them for each business environment.

Open

The technology used for the meta data integration and access processes must be open and flexible. For example, the database used to store the meta data is generally relational, but the meta data architecture should be sufficiently flexible to allow a company to switch from one relational database to another without massive architectural changes. Also, an open meta data repository enables a company to share meta data externally, and most important, make it accessible to all users. If, for exam-ple, a company decides to Web-enable all of its meta data reports, the processes for providing access to these reports should be able to use any standard Web browser.Essentially, there are two basic approaches to meta data repository architecture:

1) Centralized 2) Decentralized

A meta data repository is the logical place for uniformly retaining and managing corporate knowledge within or across different organizations. For most small to medium-sized organizations, a single meta data repository (centralized approach) is sufficient for handling all of the meta data required by the various groups in the organization. This architecture, in turn, offers a single and centralized approach for administering and sharing meta data by various teams. However, in most large enterprises that deploy multiple information management applications (e.g., for data warehousing and decision support), several meta data repositories

(decentralized approach) are often necessary to handle all of the company's various types of meta data content and applications[13].

1.4.1 Centralized Meta Data Repository Architecture

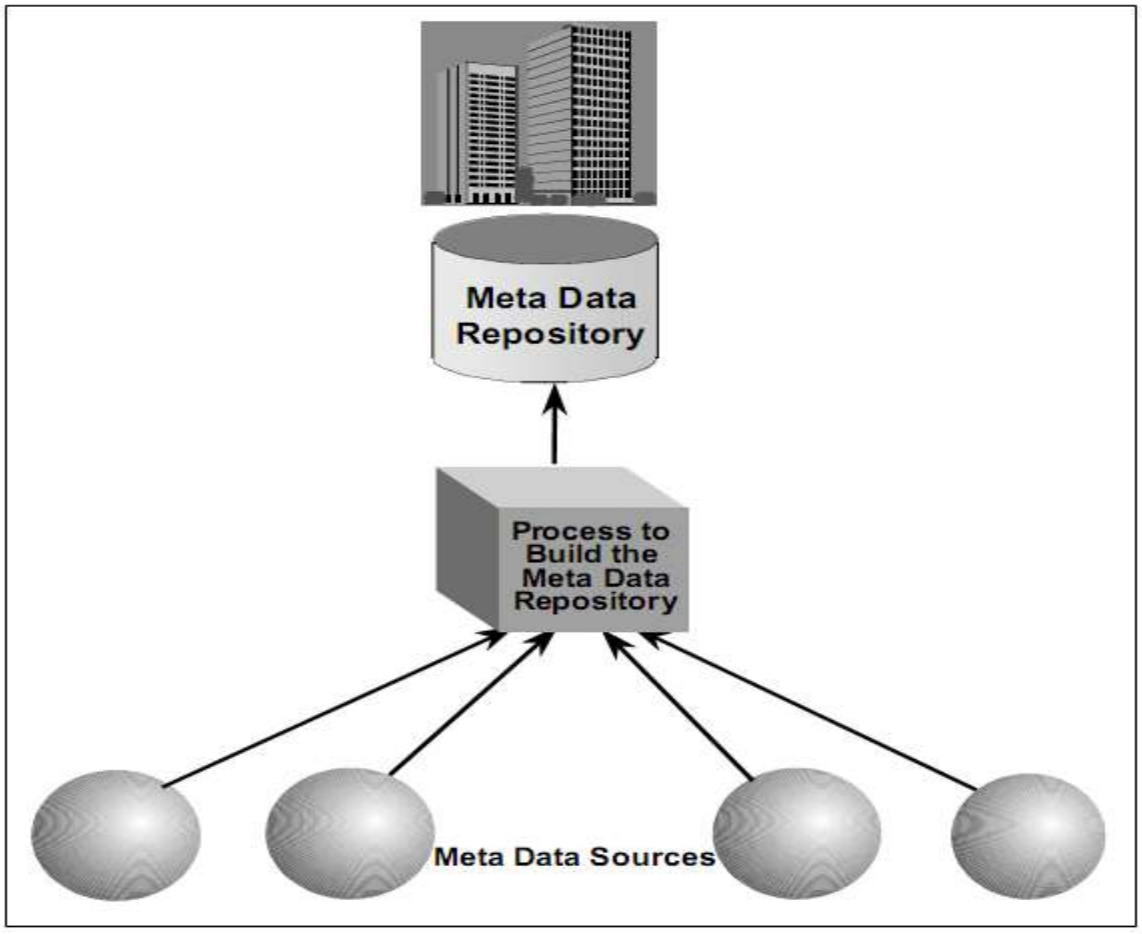

Figure 1.1 Centralized Meta Data Repository Architecture

The underlying concept of a centralized meta data architecture (like the one illustrated in Figure 1.1) is a uniform and consistent meta model that mandates that the schema for defining and organizing the various meta data be stored in a global meta data repository, along with the meta data itself.

In a typical repository installation, the meta data repository shares a hardware platform (e.g., mainframe, AS400, UNIX, etc.) with the DSS or some other application(s). This is because the repository database usually requires only about 5 gigabytes (GB) to 15 GB of raw, physical database storage, with perhaps another 5 to 15 GB for data staging areas, indexes, and so forth[13].

1.4.2 Decentralized Meta Data Repository Architecture

The objective of a decentralized architecture, like the one illustrated in Figure 1.2, is to create a uniform and consistent meta model that mandates the schema for defining and organizing the various meta data be stored in a global meta data repository and in the shared meta data elements that appear in the local repositories.

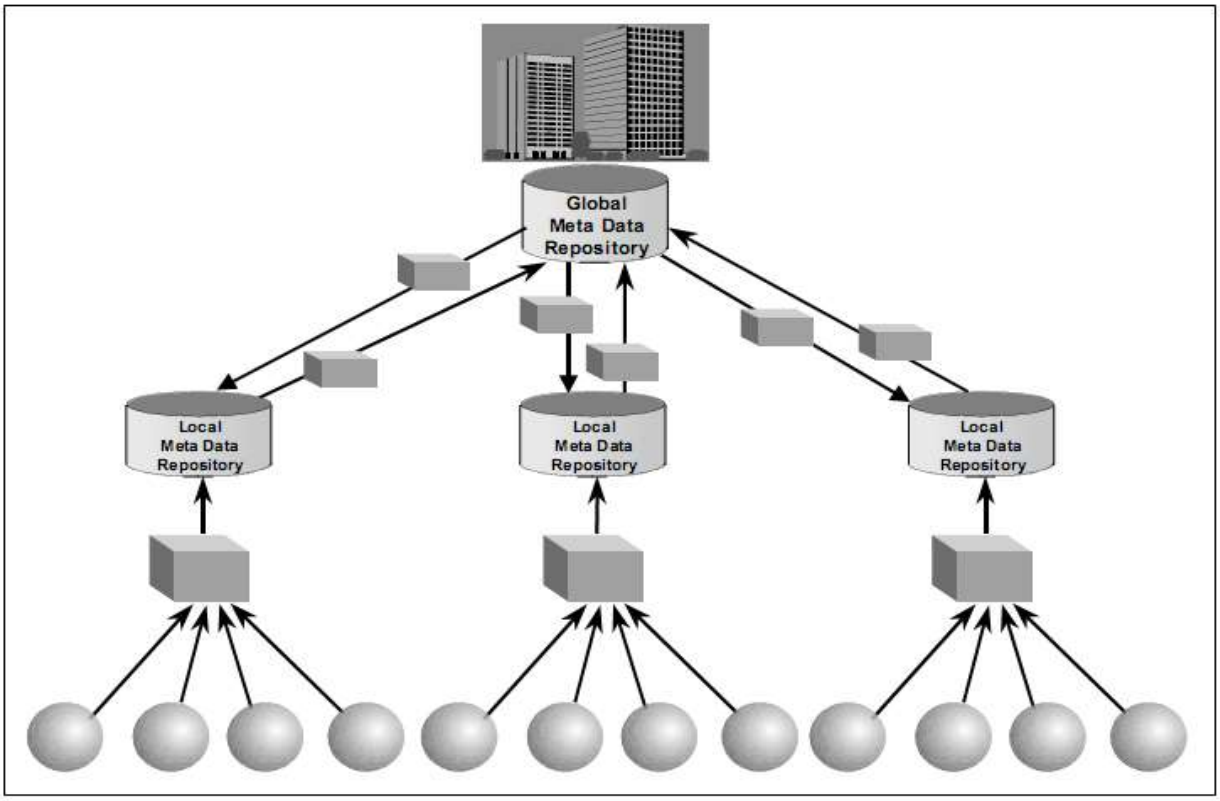

Figure 1.2 Decentralized Meta Data Repository Architecture[arc]

All meta data that is shared and reused among the various repositories must first go through the central global repository, but sharing and access to the local meta data are independent of the central repository.Keep in mind that the global repository is a subset of the meta data stored in the local repositories. The reason for this is that if all meta data was stored globally, there wouldn't be the need to have local repositories. This architecture is highly desirable for those companies that have very distinct and nonrelated lines of business.

While this architecture provides the means for centrally managing the administration and sharing of meta data across multiple meta data repositories, it also allows each local repository to be autonomous for its own content and administration requirements. This architecture is similar to a federated management in that its central governing architecture provides the guidelines that are common to all of its members, and each of its members can also create localized guidelines for their specific needs[13].

1.4.3 Bidirectional Meta Data

A bidirectional meta data architecture, like the one illustrated in Figure 1.3, allows meta data to be changed in the repository, then fed back from the repository into the original source. For example, if a user goes through the repository and changes the name of an attribute for one of the decision support system's data marts, if the repository has a bidirectional architecture, the change is fed back into the data modeling tool to update the physical model for that specific data mart.

Bidirectional architecture is highly desirable for two key reasons. First, it allows tools to share meta data, which is particularly desirable in the data warehousing market. Because most companies that built decision support systems did so with best-of-breed tools rather than integrated tool sets, the tools are not integrated with one another and do not communicate easily.

Bidirectional meta data resolves this lack of integration and communication by letting the tools share meta data. Second, because bidirectional meta data enables companies to sweep meta data changes throughout the enterprise, it is extremely attractive for organizations that want to implement a meta data repository on an enterprise-wide level. This would allow a corporation to make global changes in the meta data repository and have them sweep throughout the enterprise.

There are three obvious challenges to implementing bidirectional meta data: (1) it forces the meta data repository to contain the latest version of the meta data source that it will feed back into; (2) changes need to be systematically trapped and resolved because one user may be changing the meta data in the repository at the same time that another user is changing the same meta data at its source; and (3) additional sets of process interfaces need to be built to tie the meta data repository back to the meta data source[13].

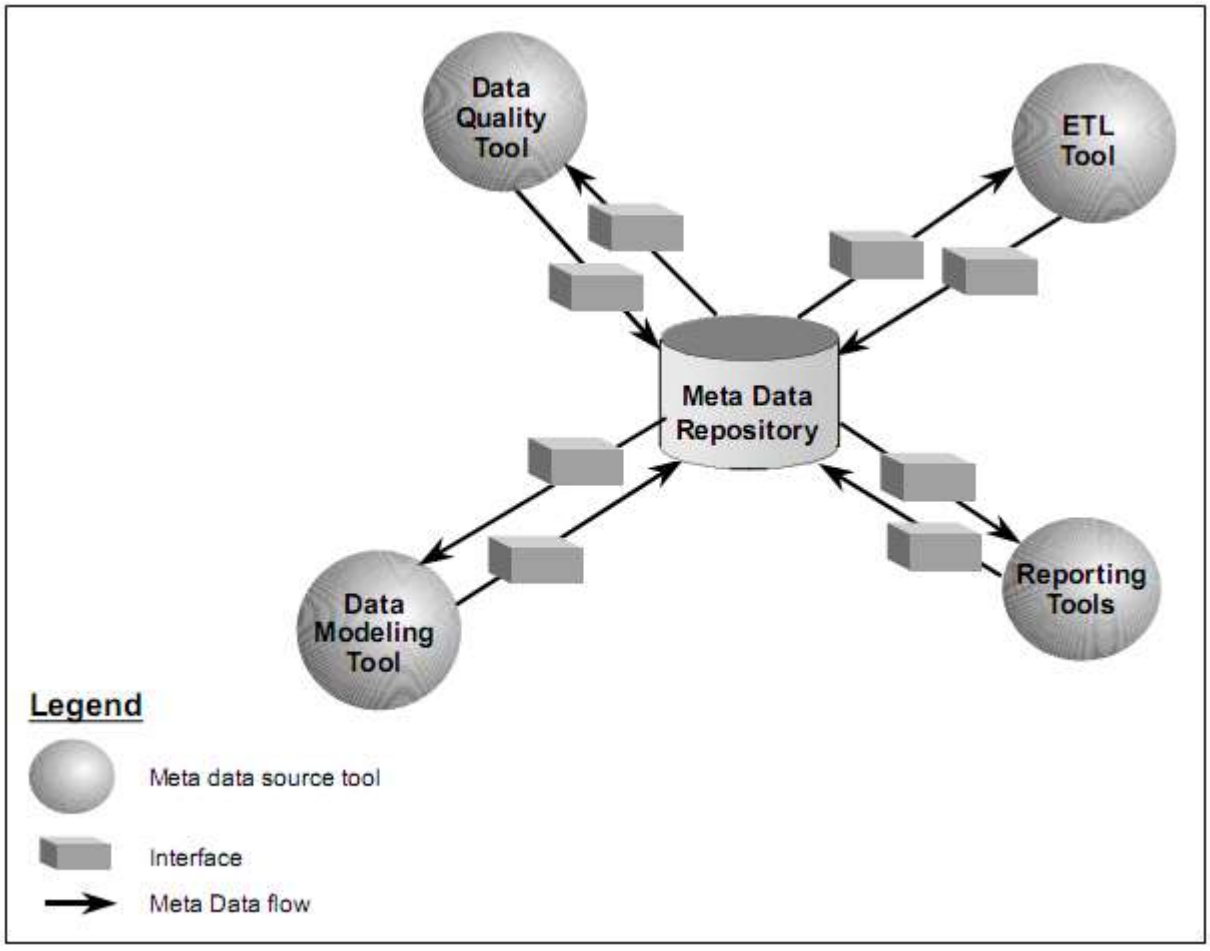

Figure 1.3 Bidirectional Meta Data Architecture

1.4.4 Closed-Loop Meta Data

A closed-loop meta data architecture allows the repository to feed its meta data back into a company's operational systems. (Figure 1.3 illustrates this type of architecture.) This concept is similar to bidirectional meta data architecture, but in this case the meta data repository is feeding its information into operational systems rather than into other applications. Closed loop meta data architecture is gaining popularity among organizations that want to implement an enterprise-wide data repository because it allows them to make global changes in the meta data repository and have those changes sweep throughout the operational systems of the enterprise.

Closed-loop meta data architecture adds some of the same complexities to the meta data repository initiative as does bidirectional meta data architecture. If the meta data that will be fed

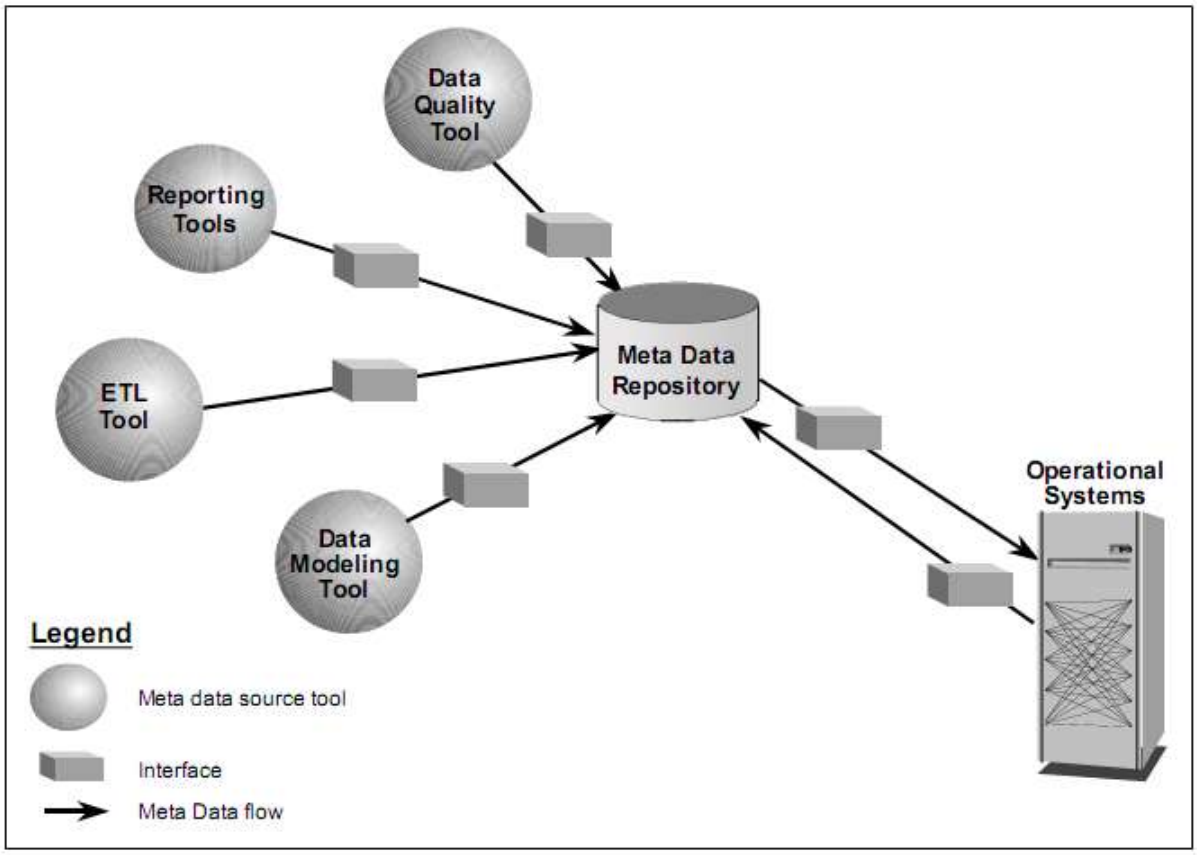

Figure 1.4 Closed loop Meta Data Architecture

from the repository to the operational system can also be maintained in the operational system, the meta data repository must contain the latest version of that meta data. If the repository does not contain the latest version, there is no assurance that the repository user is updating the latest copy of the meta data. Also, one user may make changes to the meta data in the repository at the same time that another user is changing the operational system. These conflicts must be systematically trapped and program interfaces built to tie the meta data repository back to

operational systems. Although relatively few companies are using closed-loop architecture at this time, it is a natural progression in the architecture of meta data repositories[13].

1.4 Computer Forensic

1.4.1 What is Computer Forensic?

At a basic level, computer forensics is the analysis of information contained within and created with computer systems and computing devices, typically in the interest of figuring out what happened, when it happened, how it happened, and who was involved.

This can be for the purpose of performing a root cause analysis of a computer system that had failed or is not operating properly, or to find out who is responsible for misuse of computer systems, or perhaps who committed a crime using a computer system or against a computer system. This being said, computer forensic techniques and methodologies are commonly used for conducting computing investigations - again, in the interest of figuring out what happened, when it happened, how it happened, and who was involved.

Think about a murder case or a case of financial fraud. What do the investigators involved in these cases need to ascertain? What happened, when did it happen, how did it happen, and who was involved.

In many cases, information is gathered during a computer forensics investigation that is not typically available or viewable by the average computer user, such as deleted files and fragments of data that can be found in the space allocated for existing files - known by computer forensic practitioners as slack space. Special skills and tools are needed to obtain this type of information

or evidence. Think of a case where the specific firearm that fired a bullet needs to be identified. This information could not be readily ascertained by just any member of law enforcement, so ballistics professional with special skills and tools is needed.

Computer forensics or forensic computing in the vein of computer crime or computer misuse is defined as follows:

“The preservation, identification, extraction, interpretation, and documentation of computer evidence, to include the rules of evidence, legal processes, integrity of evidence, factual reporting of the information found, and providing expert opinion in a court of law or other legal and/or administrative proceeding as to what was found”[15].

1.4.1.1 Preservation

When performing a computer forensics analysis, we must do everything possible to preserve the original media and data. Typically this involves making a forensic image or forensic copy of the original media, and conducting our analysis on the copy versus the original.

1.4.1.2 Identification

In the initial phase, this has to do with identifying the possible containers of computer related evidence, such as hard drives, floppy disks, and log files to name a few. Understand that a computer or hard drive itself is not evidence - it is a possible container of evidence.

1.4.1.3 Extraction

Any evidence found relevant to the situation at hand will need to be extracted from the working copy media and then typically saved to another form of media as well as printed out.

1.4.1.4 Interpretation

This is a biggie. Understand that just about anyone can perform a computer forensics "analysis." Some of the GUI tools available make it extremely easy. Being able to find evidence is one thing, the ability to properly interpret it is another story. Entire books could be written citing examples of when computer forensics experts misinterpreted their results of a forensic analysis.

1.4.1.5 Documentation

Documentation needs to be kept from beginning to end, as soon as you become involved in a case. This includes what is commonly referred to as a chain of custody form, as well as documentation pertinent to what you do during your analysis. We cannot overemphasize the importance of documentation.

1.4.1.6 Rules of Evidence

There are various tests that courts can apply to the methodology and testimony of an expert in order to determine admissibility, reliability, and relevancy. The particular test(s) used will vary from state to state and even from court to court within the same state.

1.4.1.7 Legal Processes

This has to do with the processes and procedures for search warrants, depositions, hearings, trials, and discovery just to name a few.

1.4.1.8 Integrity of Evidence

This has to do with keeping control over everything related to the case or situation. We are talking about establishing and keeping a chain of custody, as well as making sure that you do not alter or change the original media. As well, you cannot talk to other people about the case or situation specifics that are not involved.

1.4.1.9 Factual Reporting of the Information Found

Your findings and reports need to be based on proven techniques and methodology, and you as well as any other competent forensic examiner should be able to duplicate and reproduce the results.

1.4.1.10 Providing Expert Opinion

One may have to testify or relate his findings and opinions about his findings in a court of law or other type of legal or administrative proceeding[15].

1.4.2 Computer Forensics Basics

The field of computer forensics is relatively young. In the early days of computing, courts considered evidence from computers to be no different from any other kind of evidence. As computers became more advanced and sophisticated, opinion shifted -- the courts learned that computer evidence was easy to corrupt, destroy or change.

Investigators realized that there was a need to develop specific tools and processes to search computers for evidence without affecting the information itself. Detectives partnered with

computer scientists to discuss the appropriate procedures and tools they'd need to use to retrieve evidence from a computer. Gradually, they developed the procedures that now make up the field of computer forensics.

Usually, detectives have to secure a warrant to search a suspect's computer for evidence. The warrant must include where detectives can search and what sort of evidence they can look for. In other words, a detective can't just serve a warrant and look wherever he or she likes for anything suspicious. In addition, the warrant's terms can't be too general. Most judges require detectives to be as specific as possible when requesting a warrant.

For this reason, it's important for detectives to research the suspect as much as possible before requesting a warrant. Consider this example: A detective secures a warrant to search a suspect's laptop computer. The detective arrives at the suspect's home and serves the warrant. While at the suspect's home, the detective sees a desktop PC. The detective can't legally search the PC because it wasn't included in the original warrant.

Every computer investigation is somewhat unique. Some investigations might only require a week to complete, but others could take months. Here are some factors that can impact the length of an investigation:

- The expertise of the detectives
- The number of computers being searched
- The amount of storage detectives must sort through (hard drives, CDs, DVDs and thumb drives)
- Whether the suspect attempted to hide or delete information
- The presence of encrypted files or files that are protected by passwords[16]

1.4.3 Phases of a Computer Forensics Investigation

Judd Robbins, a computer scientist and leading expert in computer forensics, lists the following steps investigators should follow to retrieve computer evidence [16]:

1. Secure the computer system to ensure that the equipment and data are safe. This means the detectives must make sure that no unauthorized individual can access the computers or storage devices involved in the search. If the computer system connects to the Internet, detectives must sever the connection.
2. Find every file on the computer system, including files that are encrypted, protected by passwords, hidden or deleted, but not yet overwritten. Investigators should make a copy of all the files on the system. This includes files on the computer's hard drive or in other storage devices. Since accessing a file can alter it, it's important that investigators only work from copies of files while searching for evidence. The original system should remain preserved and intact.
3. Recover as much deleted information as possible using applications that can detect and retrieve deleted data.
4. Reveal the contents of all hidden files with programs designed to detect the presence of hidden data.
5. Decrypt and access protected files.
6. Analyze special areas of the computer's disks, including parts that are normally inaccessible. (In computer terms, unused space on a computer's drive is called unallocated space. That space could contain files or parts of files that are relevant to the case.)

7. Document every step of the procedure. It's important for detectives to provide proof that their investigations preserved all the information on the computer system without changing or damaging it. Years can pass between an investigation and a trial, and without proper documentation, evidence may not be admissible. Robbins says that the documentation should include not only all the files and data recovered from the system, but also a report on the system's physical layout and whether any files had encryption or were otherwise hidden.
8. Be prepared to testify in court as an expert witness in computer forensics. Even when an investigation is complete, the detectives' job may not be done. They may still need to provide testimony in court.

1.4.4 Anti-forensics

Anti-forensics can be a computer investigator's worst nightmare. Programmers design anti-forensic tools to make it hard or impossible to retrieve information during an investigation. Essentially, anti-forensics refers to any technique, gadget or software designed to hamper a computer investigation.

There are dozens of ways people can hide information. Some programs can fool computers by changing the information in files' headers. A file header is normally invisible to humans, but it's extremely important -- it tells the computer what kind of file the header is attached to. If you were to rename an mp3 file so that it had a .gif extension, the computer would still know the file was really an mp3 because of the information in the header. Some programs let you change the information in the header so that the computer thinks it's a different kind of file. Detectives

looking for a specific file format could skip over important evidence because it looked like it wasn't relevant.

Other programs can divide files up into small sections and hide each section at the end of other files. Files often have unused space called slack space. With the right program, you can hide files by taking advantage of this slack space. It's very challenging to retrieve and reassemble the hidden information.

It's also possible to hide one file inside another. Executable files -- files that computers recognize as programs -- are particularly problematic. Programs called packers can insert executable files into other kinds of files, while tools called binders can bind multiple executable files together.

Encryption is another way to hide data. When you encrypt data, you use a complex set of rules called an algorithm to make the data unreadable. For example, the algorithm might change a text file into a seemingly meaningless collection of numbers and symbols. A person wanting to read the data would need the encryption's key, which reverses the encryption process so that the numbers and symbols would become text. Without the key, detectives have to use computer programs designed to crack the encryption algorithm. The more sophisticated the algorithm, the longer it will take to decrypt it without a key.

Other anti-forensic tools can change the metadata attached to files. Metadata includes information like when a file was created or last altered. Normally you can't change this information, but there are programs that can let a person alter the metadata attached to files. Imagine examining a file's metadata and discovering that it says the file won't exist for another three years and was last accessed a century ago. If the metadata is compromised, it makes it more difficult to present the evidence as reliable.

Some computer applications will erase data if an unauthorized user tries to access the system. Some programmers have examined how computer forensics programs work and have tried to create applications that either block or attack the programs themselves. If computer forensics specialists come up against such a criminal, they have to use caution and ingenuity to retrieve data[16].

1.4.5 Usage Of Computer Forensic

Computer forensics is a field of study concerned with the digital extraction and analysis of latent information. While a relatively new science, computer forensics has gained a reputation for being able to uncover evidence that would not have been recoverable otherwise, such as emails, text messages and document access. Although many people do not realize it, their computers are recording every keystroke, file access, website, email or password. While this does present a threat from "hackers," it is this latent information that is being used in an increasing number of ways.

1.4.5.1 Criminal

Computer forensics is popularly used in criminal cases. Computer forensics analysis may provide evidence that a crime has been committed, whether that crime involved computers directly or not. Evidence may be in the form of a document, an email, an instant message, a chat room or a photograph. This is seen frequently in narcotics cases, stalking, sexual harassment, sexual exploitation, extortion, kidnapping and even murder cases.

1.4.5.2 Domestic

Computer forensics also frequently plays a role in domestic cases and is generally centered on proof of infidelity. Examples include recovered emails, chat room transcripts, instant messaging and photographs.

1.4.5.3 Security

The Center for Computer Forensics reports that 92 percent of all business documents and records are stored digitally and that although hackers are commonly seen as a threat to security, in reality greater risks are found within a company. Examples include theft of intellectual property (such as customer lists, new designs, company financials or trade secrets) and embezzlement. The fact is that if a person is alone with a computer for less than five minutes, it is enough time to copy a hard drive on a removable storage device.

1.4.5.4 Internal

There are many uses of computer forensics that exist within companies to monitor computer usage. While what is being monitored may not be illegal itself, it is tracked because doing so is "illegal" within the confines of the company. For example, many companies have "acceptable use policies," meaning policies prohibiting personal use of the computers. Common examples of acceptable use violations include online shopping, Internet surfing, online gambling, personal emails and instant messaging or chats.

1.4.5.5 Marketing

Computer forensics is also used in marketing. Examples of this can be seen on Amazon.com when recommendations are provided, or "Just for you" from the iTunes Store. When a person visits a website, a memory of that website is placed in the computer's memory. Each site has different meta-tags embedded in it; meta-tags are one or two word descriptions of the site content. The advertisements that person experiences are tailored to the meta-tags of the sites visited, similar to a target demographic.

1.5 Metadata Extraction Tools Used in Computer Forensic

Metadata tools are generally used to extract metadata from file system and individual files. These forensic tools are designed to recover and analyze data that has been intentionally or unintentionally deleted or otherwise hidden. Originally, this class of tools was associated with activities in the law enforcement sector and was used exclusively to discover legal evidence. Recently, they have been seen widely used in business and corporate environments. Forensic tools are capable of bypassing the limits imposed by the operating system and can find file content that has been deleted (i.e., no longer available to the operating system) or has been stored in a place not typically accessible to a user, such as RAM or the registry. They can display files stored in a variety of formats, allowing a knowledgeable user to find information that would otherwise appear to be inaccessible. Many tools are used in computer forensic. Some of these tools are listed below: [17]

1.5.1 WINDOWS BASED TOOLS

1. Blackthorn GPS Forensics

   Blackthorn2 is the premier GPS forensics acquisition, examination and analysis platform.

2. Forensic Toolkit (FTK) by AccessData

   Forensic Toolkit (FTK) is recognized around the world as the standard in computer forensics software. This court-validated digital investigations platform delivers cutting-edge computer forensic analysis, decryption and password cracking software.

3. RecycleReader by Live-Forensics

   A command line tool that outputs the contents of the recycle bin on XP, Vista and windows7.

4. Belkasoft Evidence Center by Belkasoft

   This product makes it easy for an investigator to search, analyze and store digital evidence found in Instant Messenger histories, Internet Browser histories and Outlook mailboxes.

5. DateDecoder by Live-Forensics

   A command line tool that decodes most encoded time/date stamps found on a windows system, and outputs the time/date in a human readable format.

6. CD/DVD Inspector by InfinaDyne

This is the only forensic-qualified tool for examinination of optical media. It has been around since 1999 and is in use by law enforcement, government and data recovery companies worldwide.

7. EMail Detective - Forensic Software Tool by Hot Pepper Technology, Inc

"E-Mail Detective - Forensic Software Tool" is used by law enforcement agencies in the U.S. and has become an invaluable tool in numerous investigations and data recovery by forensic examiners. The EMD application is an exceptionally quick and easy application to use. Within minutes, any AOL email that has been cached or saved on a user's disk drive is extracted, complete with all embedded pictures.

8. HashUtil by Live-Forensics

HashUtil.exe will calculate MD5, SHA1, SHA256 and SHA512 hashes. It has an option that will attempt to match the hash against the NIST/ISC MD5 hash databases.

9. X-Ways Forensics by X-Ways AG

X-Ways Forensics is an advanced work environment for computer forensic examiners and our flagship product. It runs under Windows 2000/XP/2003/Vista/7, 32 Bit/64 Bit. Compared to its competitors, X-Ways Forensics is more efficient to use after a while, often runs faster, is not as resource-hungry, finds deleted files and search hits.

1.5.2 LINUX BASED TOOLS

1. LINReS by NII Consulting Pvt. Ltd.

LINReS is a Live Response script designed to run on suspect/compromised Linux systems system with a minimal impact on the system to satisfy various forensic standards requirements. This script has been tested successfully on RedHat Enterprise Linux systems.

2. SMART by ASR Data

SMART for Linux is packaged as a completely self-installing Linux program. It provides Integrated acquisition, authentication and analysis in an intuitive GUI.

1.5.3 OPEN SOURCE TOOLS

1. Autopsy

The Autopsy Forensic Browser is a graphical interface to the command line digital investigation analysis tools in The Sleuth Kit. Together, they can analyze Windows and UNIX disks and file systems (NTFS, FAT, UFS1/2, Ext2/3).

2. Sleuthkit

The Sleuth Kit (TSK) is a C library and a collection of command line digital investigation tool(a.k.a. digital forensic tool) that run on Windows and Unix systems (such as Linux, OS X, Cygwin, FreeBSD, OpenBSD, and Solaris)

3. fiwalk

fiwalk is a program that processes a disk image using the SleuthKit library and outputs its results in Digital Forensics XML, the Attribute Relationship File Format (ARFF) format used by the Weka data mining toolkit, or an easy-to-read textual format.

2.

Literature Review

2.1 History Of Metadata in File Systems

Most computing systems have some type of prolonged data storage that may be examined for evidence. Even though it need not be the case for every system, the standard organization of this storage is composed of files, directories, and metadata. As metadata we express all the data in the file system that describes the layout and attributes of the regular files and directories This includes attributes such as timestamps, access control information, file size, but also information on how to place and assemble a file or directory in the file system. This latter information contains pointers to data blocks, or even complete blocks used as internal nodes of lookup data structures such as B-trees.

File system metadata was not originally designed to be used for the purpose of reconstructing events that occurred on the system. Anderson was the first to utilize such data for threat monitoring. He proposed to utilize System Management Facilities (SMF) records, which were kept by mainframe servers, such as those running IBM's OS/360. In the 1960s most computing tasks were performed on mainframe computers, with OS/360 one of the dominating operating systems. Information stored on the servers' disks described the entire batch job of a user. The

3.

Problem Formulation

Computer forensics is the specialized practice of investigating computer media for the purpose of discovering and analyzing available, deleted, or "hidden" information that may serve as useful evidence in a legal matter. Computer forensics can be used to uncover potential evidence in many types of cases including, for example: Copyright infringement, Industrial espionage, money laundering, piracy, sexual harassment, theft of intellectual property, unauthorized access to confidential information , blackmail, corruption, decryption, destruction of information, fraud illegal duplication of software, unauthorized use of a computer, child pornography

Fiwalk is widely popular among the computer forensics to extract metadata from files and disks. In the fiwalk we have a plug-in architecture allowing new metadata extractors to be readily incorporated. This feature can be assessed to detect the differences of output obtained by using specific plug-ins (new metadata extractors) and the default library extractor. So that we can determine whether to incorporate specific plug-ins are feasible or not. Also we can incorporate our plug-in to extract metadata. Since word and jpeg both have their specific metadata extractors to extract metadata from their respective files, we can focus on other file type and it is PDF(Portable Document Format) .

Also when using this type of tool in forensic investigation we need to be sure of the result (metadata) which we are getting is accurate or not. For this we have to compare the output of this tool to the original file or document to determine its accuracy. It will also compare the results(metadata) of similar file types to determine which file format is more useful from the forensic expert's point of view.

Also when using this type of tool in forensic investigation we need to be sure of the result (metadata) which we are getting is accurate or not. For this we have to compare the output of this tool to the original file or document to determine its accuracy. It will also compare the results(metadata) of similar file types to determine which file format is more useful from the forensic expert's point of view.

This research will eventually provide us the feasibility of combining metadata extraction plug-ins and also the accuracy or deficiency if any in the metadata tool.

Also we need to find out which file format among the similar type file formats are more useful. That is we want to determine which format is more useful to investigator. That is we wants to know which file from similar file format provides better capability. The file with greater metadata extraction will be more useful for the forensic investigator.

3.1 Objectives Of Research

The objective of our research is to study and analyze the fiwalk tool so that we can measure the feasibility of incorporating plug-ins and combining the extractors, and also to determine the deficiency if any in the existing tool.

Major Research Questions

Q.1 what is the feasibility of combining/ incorporating plug-ins in fiwalk to extract metadata?

Q.2 Do the existing open source metadata extraction tool provide accurate results?

Also we need to find out which format out of similar file formats are useful for forensic expert regarding the metadata they are generating.

4.

Experimental Observation

This thesis presents a new technique for automatically extracting metadata from files and file contents. The technique is embedded in a program called fiwalk that has a plug-in architecture allowing custom and new metadata extractors to be readily incorporated in it. Fiwalk is a program used to retrieve information from disk partitions found on disk images. Fiwalk was chosen for use in this thesis because the fiwalk have above mentioned plug-in capability and also the output of fiwalk can be directly put into the data mining tool weka because it can also give its output in ARFF (Attribute Relationship File Format).

This Chapter of experimental observation is further divided into three sections,the first section deals with the evaluation of different extractors involved in the fiwalk to find out what metadata is extracted from the plug-ins, second section deals with the metadata extractor I have created for this thesis to extract metadata from pdf files. Next section deals with the extraction of metadata from similar file types. In this thesis we have chosen Office Open XML format and Open Document Format (ODF) for comparison, since both the formats are used to perform same type of operations on documents which include creating spreadsheet, text document and presentation. The comparison is carried out to find out which format gives better results as a metadata to forensic investigator.

4.1 Evaluation of Plug-ins Involved in Fiwalk

Fiwalk has plug-in architecture which uses different type of plug-ins to extract metadata from file. Plug-ins are called according to the file type encountered during the processing of files. Firstly, fiwalk gathers the file system and file metadata by using the The Sleuth Kit’s

programmer's interface. After that, fiwalk obtains the list of all the partitions and for each and every partition, fiwalk obtains the list of all the files. For each file corresponding metadata and file sytem is obtained from The Sleuth Kit's Library and the plug-ins are called according to the file type.

DGI is the process through which fiwalk plug-ins communicate with fiwalk program. Fiwalk puts the data into a file in the file system; the plug-in reads the file and returns the data as a sequence of name:value pairs. several name:value pairs come from TSK while others come from the fiwalk's plug-ins. The task of the fiwalk is to collect all of the name:value pairs from each and every file, as provided by TSK, enhance them with name:value pairs from the plug-ins, and place the result into a single ARFF /XML file.

4.1.1 JPEG PLUGIN

JPEG stands for Joint Photographic Experts Group. It is a standard method of compressing photographic images. We also call JPEG the file format which employs this compression. The file extensions for this format are .JPEG, .JFIF, .JPG, OR .JPE although .JPG is the most common on all platforms. It is the most popular and widely used format for representing digital images. Extracting metadata from images is very important for forensic investigators. One of the best metadata extraction program to extract metadata from jpeg files is exif program. The fiwalk program comes with a plug-in called jpeg_extract that uses the exif program to extract metadata from files. Whenever a jpeg file is encountered jpeg_extract plugin is called, which in- turn calls the exif program to extract metadata related to the jpeg file. the desired metadata is returned to jpeg_extract plug-in by exif program and placed into dgi format for further processing.

Figure 4.1 below shows the sample run of jpeg_extract when a .jpeg file is encountered in fiwalk.

Manufacturer: Canon

Model: Canon EOS 1000D

Orientation: top - left

x_Resolution: 72.00

y_Resolution: 72.00

Resolution_Unit: Inch

Date_and_Time: 2011-01-20 17:18:07

YCbCr_Positioning: co-sited

Compression: JPEG compression

x_Resolution: 72.00

y_Resolution: 72.00

Resolution_Unit: Inch

Exposure_Time: 1/200 sec.

FNumber: f/9.0

Exposure_Program: Normal program

ISO_Speed_Ratings: 400

Exif_Version: Exif Version 2.21

Date_and_Time__original_: 2011-01-20 17:18:07

Date_and_Time__digitized_: 2011-01-20 17:18:07

Components_Configuration: Y Cb Cr -

Shutter_speed: 7.62 EV (APEX: 14, 1/197 sec.)

Aperture: 6.38 EV (f/9.1)
Exposure_Bias: 0.00 EV
Metering_Mode: Pattern
Flash: Flash fired, compulsory flash mode.
Focal_Length: 55.0 mm
Maker_Note: 8552 bytes undefined data
User_Comment:
SubsecTime: 02
SubSecTimeOriginal: 02
SubSecTimeDigitized: 02
FlashPixVersion: FlashPix Version 1.0
Color_Space: sRGB
PixelXDimension: 2816
PixelYDimension: 1880
Focal_Plane_x_Resolution: 3214.61
Focal_Plane_y_Resolution: 3224.70
Focal_Plane_Resolution_Unit: Inch
Custom_Rendered: Normal process
Exposure_Mode: Auto exposure
White_Balance: Auto white balance
Scene_Capture_Type: Standard
InteroperabilityIndex: R98
InteroperabilityVersion: 0100
ThumbnailSize: 6567

Figure 4.1 Sample run of jpeg_extract in fiwalk

4.1.2 Miocrosoft Office Document Extractor

Two types of metadata extractors are employed in fiwalk to extract metadata from Microsoft office documents. The first plug-in make use of wvSummary script to extract metadata from Microsoft office documents. The next plug-in is an lengthened XML parser written in Python. This particular plug-in is used to extract metadata from Office Open XML files, the local format of Microsoft Office 2007.

4.1.2.1 WORD_EXTRACT

If Microsoft Office documents other than that of Microsoft Office 2007/2010 are encountered than fiwalk calls the plugin word_extract to extract the file metadata, which in turn calls the wvSummary to extract metadata. wvSummary extracts the metadata and translates its output in dgi format. Figure 4.2 shows the sample run of a Microsoft Word document after being processed by wvSummary.

```
# plugin_process
Filename: /tmp/AashishResumBE.doc
```

Template: Elegant Resume
Security_Level: 0
Created: 2010-06-14T17:34:00Z
Last_Saved_by: Prakash
Revision: 6
Last_Printed: 2003-10-05T09:01:00Z
Keywords:
Subject:
Generator: Microsoft Office Word
Thumbnail: ((GsfClipData*) 0x91f7940)
Number_of_Characters: 1750
Last_Modified: 2010-07-06T06:58:00Z
Creator: Mayank
Number_of_Pages: 2
msole_codepage: 1252
Number_of_Words: 306
Editing_Duration: 2009-04-22T19:45:48Z
Title: Elegant Resume
Links_Dirty: FALSE
Number_of_Lines: 14
UseDefaultLanguage: TRUE
Version: 99022200
LCID: 1033
Document_Parts: [(0, Elegant Resume)]

Scale: FALSE
Number_of_Paragraphs: 4
Unknown_: FALSE
Unknown_: 786432
Company:
Document_Pairs: [(0, Title), (1, 1)]
Unknown_: 2052
Unknown_: FALSE
msole_codepage: 1252

Figure 4.2. Sample run of Microsoft word document by word_extract(wvSummary)

The same extractor(wvSummary) is used to extract metadata from Microsoft PowerPoint and Microsoft Excel. The sample runs for both type of files is given below in Figure 4.3, Figure 4.4:

Filename: /tmp/SeminarFinal.ppt
Template: Fading Grid
Created: 2010-11-16T16:11:19Z
Last_Saved_by: Guest
Revision: 28
Generator: Microsoft PowerPoint
Thumbnail: ((GsfClipData*) 0x8135930)
Last_Modified: 2010-11-15T21:10:32Z

Creator: Guest

Number_of_Words: 365

msole_codepage: 1252

Editing_Duration: 2009-04-22T23:38:09Z

Title: A SEMINAR ON DESIGN TECHNIQUE OF VOICE BROWSER

Links_Dirty: FALSE

Document_Parts: [(0, Arial), (1, Times New Roman), (2, Wingdings), (3, Fading Grid), (4, A SEMINAR ON DESIGN TECHNIQUE FOR VOICE BROWSER), (5, CONTENTS), (6, VOICE BROWSER?), (7, Voice Browser (Cont.)), (8, Speech Recognition), (9, Speech Synthesis), (10, VoiceXML), (11, VoiceXML Example (Cont.)), (12, VoiceXML Tags), (13, Voice browser requirements), (14, Architecture), (15, Summary), (16, Bibliography), (17, Queries??)]

Number_of_Paragraphs: 66

Scale: FALSE

Number_of_Bytes_in_the_Document: 90829

_Clips: 0

Unknown_: 657985

Number_of_Slides: 14

Company:

Document_Pairs: [(0, Fonts Used), (1, 3), (2, Design Template), (3, 1), (4, Slide Titles), (5, 14)]

Unknown_: FALSE

msole_codepage: 1252

Number_of_Notes: 0

Unknown_: FALSE
Number_of_Hidden_Slides: 0

Figure 4.3. Sample run of Microsoft PowerPoint document by word_extract(wvSummary)

Filename: /tmp/akpNEWCONTACTS.xls
Last_Modified: 2011-07-11T06:13:29Z
Last_Saved_by: akp
Security_Level: 0
msole_codepage: 1252
Created: 2011-07-11T06:13:29Z
Links_Dirty: FALSE
Document_Parts: [(0, akp)]
Scale: FALSE
Unknown_: FALSE
Unknown_: 786432
Document_Pairs: [(0, Worksheets), (1, 1)]
Unknown_: FALSE
msole_codepage: 1252

Figure 4.4. Sample run of Microsoft Excel document by word_extract(wvSummary)

4.1.2.2 DOCX_EXTRACTOR (Microsoft Office 2007 Documents)

Fiwalk executes the docx_extractor if a Microsoft Office 2007/2010 file is encountered to extract metadata from Microsoft Word, Microsoft Excel or Microsoft PowerPoint file. At first docx, xlsx, pptx archives are unzipped by the extractor. After that the docx_extractor evaluates the XML to find values related with alias and tag, id attributes of the structured document tag <w:sdt>. These values are also important as they can provide additional information that can be used to discover content controls such as textboxes, image files that have been added to the document. The extractor also parses the XML to recover data store ids and GUIDs, which attach the custom pieces to the right data within the data store.

At last the extractor focuses on the more traditional metadata. The extractor receives metadata encapsulated in following tags:

- Creator
- Last Modified
- Created date
- Modified date
- Title
- Subject
- Description

- Application
- Company
- Number of characters, words, lines, paragraphs, pages, etc.
- Template
- Revision number

Figure 4.5 shows the sample output of a Microsoft Word 2007 document after being processed by the docx_extractor, Figure 4.6 shows the sample output of a Microsoft PowerPoint 2007 document after being processed by the docx_extractor and the Figure 4.7 shows the sample output of a Microsoft Excel 2007 document after being processed by the docx_extractor.

<Archive_File_>docProps/app.xml</Archive_File_>

<Archive_File_>docProps/core.xml</Archive_File_>

<Archive_File_>word/document.xml</Archive_File_>

<Archive_File_>webSettings.xml</Archive_File_>

<Archive_File_>settings.xml</Archive_File_>

<Archive_File_>styles.xml</Archive_File_>

<Archive_File_>theme/theme1.xml</Archive_File_>

<Archive_File_>fontTable.xml</Archive_File_>

<Paragraph_Revision_ID_>007E6E1B</Paragraph_Revision_ID_>

<Paragraph_Revision_ID_Default_>007E6E1B</Paragraph_Revision_ID_Default_>

<Generator>Microsoft Office Word</Generator>

<Template>Normal</Template>

<Number_of_Pages>7</Number_of_Pages>

<Number_of_Lines>22</Number_of_Lines>

<Number_of_Paragraphs>6</Number_of_Paragraphs>

<Number_of_Words>472</Number_of_Words>

<Number_of_Characters>2691</Number_of_Characters>

<Created>2010-11-02T16:22:00Z</Created>

<Last_Modified>2010-11-02T16:24:00Z</Last_Modified>

<Creator>Prakash</Creator>

<Revision>2</Revision>

<LastSavedBy>Prakash</LastSavedBy>

Figure 4.5. Sample output of a Microsoft Word 2007 document from fiwalk

Archive_File_: docProps/core.xml

Archive_File_: docProps/thumbnail.jpeg

Archive_File_: ppt/presentation.xml

Archive_File_: docProps/app.xml

Archive_File_: ../slideLayouts/slideLayout2.xml

Archive_File_: ../slideLayouts/slideLayout1.xml

Archive_File_: slides/slide7.xml

Archive_File_: slides/slide12.xml

Archive_File_: slides/slide2.xml

Archive_File__: slides/slide6.xml

Archive_File__: slides/slide11.xml

Archive_File__: tableStyles.xml

Archive_File__: slides/slide1.xml

Archive_File__: theme/theme1.xml

Archive_File__: slideMasters/slideMaster1.xml

Archive_File__: slides/slide5.xml

Archive_File__: slides/slide10.xml

Archive_File__: slides/slide4.xml

Archive_File__: viewProps.xml

Archive_File__: slides/slide9.xml

Archive_File__: slides/slide3.xml

Archive_File__: slides/slide8.xml

Archive_File__: presProps.xml

Archive_File__: ../slideMasters/slideMaster1.xml

Archive_File__: ../slideLayouts/slideLayout8.xml

Archive_File__: ../slideLayouts/slideLayout3.xml

Archive_File__: ../slideLayouts/slideLayout7.xml

Archive_File__: ../theme/theme1.xml

Archive_File__: ../slideLayouts/slideLayout6.xml

Archive_File__: ../slideLayouts/slideLayout11.xml

Archive_File__: ../slideLayouts/slideLayout5.xml

Archive_File__: ../slideLayouts/slideLayout10.xml

Archive_File__: ../slideLayouts/slideLayout4.xml

Archive_File__: ../slideLayouts/slideLayout9.xml

Archive_File__: ../media/image1.jpeg

Generator: Microsoft Office PowerPoint

Template: Apex

Number_of_Paragraphs: 82

Number_of_Words: 543

Number_of_Slides: 12

Number_of_Hidden_Slides: 0

Number_of_Notes: 0

_Clips: 0

Presentation_Format: On-screen Show (4:3)

Created: 2011-03-14T19:30:46Z

Last_Modified: 2011-07-06T15:39:51Z

Creator: PUROHIT

Title:

Figure 4.6 Sample output of a Microsoft PowerPoint 2007 document from fiwalk

filename: akp.xlsx

partition: 1

id: 6

name_type: r

filesize: 8387

alloc: 1

used: 1

inode: 4

meta_type: 1

mode: 0

nlink: 1

uid: 0

gid: 0

crtime: 1310611047

crtime_txt: 2011-07-14 02:37:27

MD5: f90342d392389f4bef5ceae26dd826ee

SHA1: bd0f7974ea8917b35718d0fa04ee0bd977e4abda

plugin_process

Archive_File_: docProps/app.xml

Archive_File_: docProps/core.xml

Archive_File_: xl/workbook.xml

Archive_File_: worksheets/sheet3.xml

Archive_File_: worksheets/sheet2.xml

Archive_File_: worksheets/sheet1.xml

Archive_File_: sharedStrings.xml

Archive_File_: styles.xml

Archive_File_: theme/theme1.xml

Created: 2006-09-16T00:00:00Z

Last_Modified: 2011-07-13T13:07:27Z

Generator: Microsoft Excel

Figure 4.7 Sample output of a Microsoft Excel 2007 document from fiwalk

4.1.3 LIBEXTRACT_PLUG-IN (Default Plug-in OF Fwalk)

The default plug-in uses libextractor, a tool capable of reading metadata from a extensive range of file formats[2]. It is quite similar to the UNIX file program, which attempts to classify a given file based on the contents of the file as opposed to making a determination based solely on the file extension. Libextractor can recognize additional information such as the name of the software used to produce the file, the author, descriptions, album titles, image dimensions, or the length of a movie.

Libextractor retrieves the metadata by executing a parser for the different file formats. The current types of file formats supported under libextractor include MP3, Ogg, Real Media, MPEG, RIFF (avi), GIF, JPEG, PNG, TIFF, HTML, PDF, PostScript, Zip, OpenOffice.org, StarOffice, Microsoft Office, tar, DVI, man, Deb, elf, RPM, and asf[2]. In general liextractor can handle a wide range of document types but does not cover any of them as deeply as specially written plug-ins. For this reason, fiwalk plug-in for libextractor was created to be called when a more specific plug-in is not available.

Figure 4.8 shows the output of .jpeg file after being processed by libextractor plug-in, that is the default plug-in.

white balance - Auto

image quality - Fine

macro mode - (0)

metering mode - Pattern

exposure mode - Auto

iso speed - 400

focal length - 55.0 mm

flash bias - 0 EV

flash - Flash fired

exposure bias - 0.00 EV

aperture – F 9.1

exposure - 1/180 s

date - 2011-01-20 17:18:07

orientation - top - left

camera model - Canon EOS 1000D

camera make - Canon

size – 32816 x 1880

mimetype - image/jpeg

Figure 4.8 Sample output of .jpeg file after being processed by libextractor

Now, Figure 4.9 shows the output of .mp3 file after being processed by libextractor plug-in, that is the default plug-in.

duration: 0m54

format: MPEG-1 Layer III audio, 320 kbps (CBR), 44100 Hz, stereo, no copyright, copy

resource_type: MPEG-1

mimetype: audio/mpeg

description: jashane: baharan (Jodhaa Akbar *2008* (Pre-Rele)

genre: Other

album: Jodhaa Akbar *2008* (Pre-Rele

artist: jashane

title: baharan

year: 2008

album: Jodhaa Akbar *2008* (Pre-Release)

content_type: (12)

title: jashane baharan_by_MaFiAdOn

Figure 4.9 Sample output of .mp3 file after being processed by libextractor plug-in.

4.2 Default Plugin (LIBEXTRACTOR) VS Specific Written Plug-in

The default plug-in uses libextractor, a tool capable of reading metadata from a wide array of file formats. Libextractor is similar to the Unix file program, which attempts to classify a given file based on the contents of the file as opposed to making a determination based solely on the file extension. However, libextractor is different from file system tool in that libextractor attempts to derive more than just the mime type. For instance, libextractor can identify additional information such as the name of the software used to create the file, the author, descriptions, album titles, image dimensions, or the length of a movie.

libextractor can handle a wider range of document types but does not cover any of them as deeply as specially written plug-ins. For this reason, a fiwalk plug-in for libextractor was created to be called when a more specific plug-in is not available. Following output of a jpeg file proves this point.

For example, we have a disk image which consists of a jpeg file. At first we will extract the metadata using libextractor plugin to find out the metadata involved in this file.

white balance - Auto

image quality - Fine

macro mode - (0)

metering mode - Pattern

exposure mode - Auto

iso speed - 400

focal length - 55.0 mm

flash bias - 0 EV

flash - Flash fired

exposure bias - 0.00 EV

aperture – F 9.1

exposure - 1/180 s

date - 2011-01-20 17:18:07

orientation - top - left

camera model - Canon EOS 1000D

camera make - Canon

size – 32816 x 1880

mimetype - image/jpeg

Figure 4.10 Metadata extracted by plug-in libextract from jpeg file.

Now we will extract metadata extract the same jpeg file by the specific written metadata extractor which is jpeg_extractor. It is clear from the figure 4.11 that specific written metadata plug-ins are better than that of the libextractor.

Manufacturer: Canon

Model: Canon EOS 1000D

Orientation: top - left

x_Resolution: 72.00

y_Resolution: 72.00

Resolution_Unit: Inch

Date_and_Time: 2011-01-20 17:18:07

YCbCr_Positioning: co-sited

Compression: JPEG compression

x_Resolution: 72.00

y_Resolution: 72.00

Resolution_Unit: Inch

Exposure_Time: 1/200 sec.

FNumber: f/9.0

Exposure_Program: Normal program

ISO_Speed_Ratings: 400

Exif_Version: Exif Version 2.21

Date_and_Time__original_: 2011-01-20 17:18:07

Date_and_Time__digitized_: 2011-01-20 17:18:07

Components_Configuration: Y Cb Cr -

Shutter_speed: 7.62 EV (APEX: 14, 1/197 sec.)

Aperture: 6.38 EV (f/9.1)

Exposure_Bias: 0.00 EV

Metering_Mode: Pattern

Flash: Flash fired, compulsory flash mode.

Focal_Length: 55.0 mm

Maker_Note: 8552 bytes undefined data
User_Comment:
SubsecTime: 02
SubSecTimeOriginal: 02
SubSecTimeDigitized: 02
FlashPixVersion: FlashPix Version 1.0
Color_Space: sRGB
PixelXDimension: 2816
PixelYDimension: 1880
Focal_Plane_x_Resolution: 3214.61
Focal_Plane_y_Resolution: 3224.70
Focal_Plane_Resolution_Unit: Inch
Custom_Rendered: Normal process
Exposure_Mode: Auto exposure
White_Balance: Auto white balance
Scene_Capture_Type: Standard
InteroperabilityIndex: R98
InteroperabilityVersion: 0100
ThumbnailSize: 6567

Figure 4.11 Metadata extracted by plug-in jpeg_extract from jpeg file.

By comparing the above result we can identify that specific written plug-ins extract more metadata than the libextractor which is the default extractor. Figure 4.10 derives total of 18

metadata name:value pairs while the Figure 4.11 derives a total of 45 metadata name:value pairs.From the above comparison it motivates to have a specific written plug-in for other files to derive more metadata, that is more name:value pairs which can be valuable for a forensic investigator. Randomly I have chosen pdf file to create a specific written plug-in for pdf files so that I can improve the performance of this open source tool to extract more metadata from pdf files.libextractor can produce following metadata if a pdf file is encountered in it. The results of the metadata is given below in Figure 4.12

filename: callletter_allahabad bank.pdf

partition: 1

id: 8

name_type: r

filesize: 308471

alloc: 1

used: 1

inode: 6

meta_type: 1

mode: 0

nlink: 1

uid: 0

gid: 0

crtime: 1291059332

crtime_txt: 2010-11-29 19:35:32

MD5: b16c8165f03257af430222c6d801557f

SHA1: 760314fae3b2107cb5e543af48ab618f9c08d649

plugin_process

format: PDF 1.7

mimetype: application/pdf

Figure 4.12 metadata extraction from a pdf file using libextractor (default) plug-in

That is the plug-in individually extracts only two name:value pairs which includes following:

format: PDF 1.7

mimetype: application/pdf

To create a specific written pdf extractor first we need to know how metadata is organized in pdf files. The organization of metadata in pdf is carried out in following ways:

A. **Info dictionary** :

The info dictionary has been a part of PDF since PDF version 1.0. This area belongs to the document itself and contains a collection of name, value pairs. Predefined pairs include Title, Author, Subject, Keywords, and others. You can also add your own values

B: **XMP (Extensible Metadata Platform)**

XMP is a more recent development. It was introduced with PDF 1.4 (Acrobat 5). XMP is based on RDF (Resource Definition Framework). RDF is a W3C standard for XMP-based metadata Adobe's Extensible Metadata Platform (XMP)[99] is a labeling technology based on XML that allows you to embed data about a file, known as metadata, into the file itself. With XMP, desktop applications and back-end publishing systems gain a common method for capturing, sharing, and leveraging this valuable metadata — opening the door for more efficient job processing, workflow automation, and rights management, among many other possibilities. With XMP, Adobe has taken the "heavy lifting" out of metadata integration, offering content creators an easy way to embed meaningful information about their projects and providing industry partners with standards-based building blocks to develop optimized workflow solutions.

I have used java language to build this pdf extractor. The pdf extractor employs the itext library. iText is a library that allows you to create and manipulate PDF documents. It enables developers looking to enhance web- and other applications with dynamic PDF document generation and/or manipulation[21]. So iText library need to be imported to use the functions involved in this library and with this library function it is quite easy to extract metadata from PDF documents.

Developers can use iText to:

- Serve PDF to a browser
- Generate dynamic documents from XML files or databases
- Use PDF's many interactive features

- Add bookmarks, page numbers, watermarks, etc.
- Split, concatenate, and manipulate PDF pages
- Automate filling out of PDF forms
- Add digital signatures to a PDF file

The algorithm for creating pdf_extractor is given below:

1) Create object of PDFReader for the file under investigation
2) Invoke getmetadata function to capture metadata in a string.
3) Print String to standard output or to the xml file.

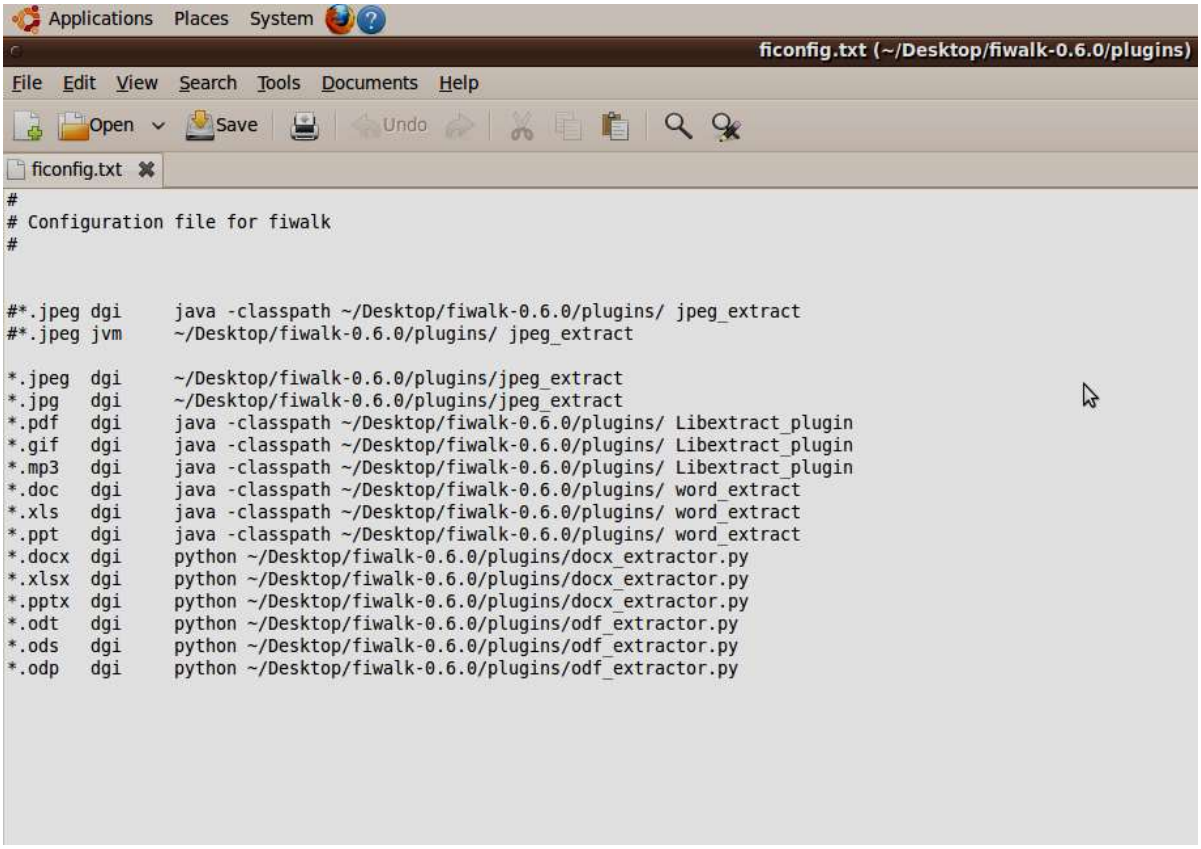

Figure 4.13 Contents of configuration file of fiwalk before using pdf_extractor

After creating this pdf_extractor in java. We have to copy this file in the fiwalk's plugin folder. Now to use this plugin we need to edit the configuration file which also exist in the same plugin folder of fiwalk. Configuration file of fiwalk is named as ficonfig.txt which is used to configure the plugins used in fiwalk. It derives the rules to call plugins involved in fiwalk. A sample ficonfig.txt is shown in Figure 4.13 which is prior to using fiwalk with pdf_extractor.

From the above contents it is clear that whenever a file of type *.pdf is encountered then libextract_plugin will be called through dgi.

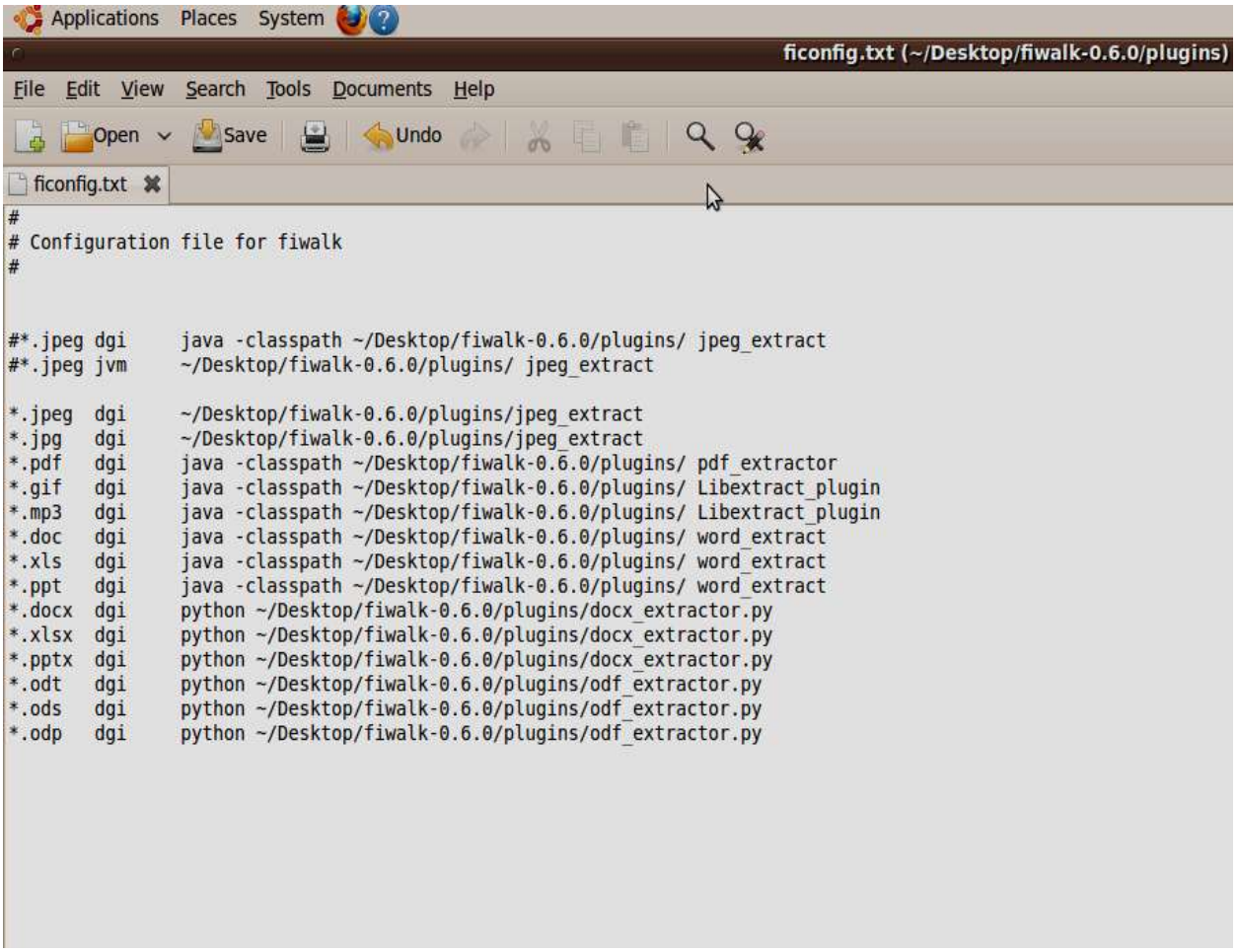

Figure 4.14 Contents of configuration file of fiwalk after using pdf_extractor

So to use specific written pdf_extractor we need to change the configuration file contents to be able to use the extractor. So we need to only replace the Libextract_plugin with pdf_extractor. After that metadata from pdf file will be extracted by this custom written pdf_extractor.Figure 4.14 shows the contents of ficonfig.txt after the pdf_extractor has been put into the plug-ins folder of fiwalk.

After using the pdf_extractor plug- in we get the following metadata for the same file shown in Figure 4.15 which we used for extraction with libextract_plugin earlier.

filename: callletter_allahabad bank.pdf

partition: 1

id: 8

name_type: r

filesize: 308471

alloc: 1

used: 1

inode: 6

meta_type: 1

mode: 0

nlink: 1

uid: 0

gid: 0

crtime: 1291059332

crtime_txt: 2010-11-29 19:35:32

MD5: b16c8165f03257af430222c6d801557f

SHA1: 760314fae3b2107cb5e543af48ab618f9c08d649

plugin_process

ModifyDdate -20101129193532-08'00'

CreationDate - 20101129193532-08'00'

format- PDF 1.4

title-

creator – TCPDF

subject –

producer - TCPDF 5.3.006 (http://www.tcpdf.org) (TCPDF)

keywords- TCPDF, PDF, example, test, guide TCPDF

mimetype - application/pdf

Figure 4.15 metadata extracted by pdf_extractor for the same file used earlier

It is clear after comparing Figure 4.12 and Figure 4.15 that pdf_extractor extracts more metadata than that extracted by the libextractor. After analyzing this output we can say that we have improved the metadata extraction feature of fiwalk. It is done by creating the specific written plug-in for pdf files which is extracting more than eight name:value pairs.

4.3 Comparing Metadata Of Similar File Formats

This section provides an initial look at the both Microsoft Office 2007 and Open Office file formats. This chapter constructs upon the Office 2007 introduction and provides a comparative metadata analysis of the two document file types by examining timestamps, encryption, and thumbnails associated with each type.

4.3.1 Examination Of Microsoft Office 2007 Documents

4.3.1.1 document.xml file

The document.xml file contains the main text and body of the document. Some of the important XML elements within this file include the following:

- <w:document> - root element; required to start defining document
- <w:body> - child element of <w:document>; text that comprises document

is stored here

• <w:p> - paragraph within <w:body>, basic unit of document

• <w:r> - region of text with a common set of properties; paragraphs can be

split into multiple runs, but runs must be contained within a paragraph, and can contain run properties, revision ids, and run content

• <w:sdt> - structured document tag; node is created when content control is added to a document; two sections within structured document node are properties and content sections.

• <w:t> - text element; document can have multiple <w:t> within a <w:r>

4.3.1.2 Content Controls

Users can add various content controls (bounded and potentially labeled regions within a document that serve as containers for specific types of content) to their .docx files. Items such as dates, lists, and paragraphs of formatted text can be contained within individual content controls. Adding content controls to other content controls is also possible. The controls provide the capability to create rich, structured blocks of content. They also build on the custom XML support introduced in Microsoft Office Word 2003 [22].

Additionally, content controls can be used to prompt users to provide data or to bind values to controls within the document.xml file from separate .xml files in the .docx package [23]. By inspection of Microsoft Word 2007, the types of content controls include the following:

• Rich text

• Plain text

• Picture content

• Combo box

• Drop down list

• Date picker

• Building block gallery

• Legacy controls such as Active X, text fields and check boxes

4.3.1.3. Identifiers

Office 2007 documents contain many types of identifiers. The paragraph revision id was examined in Chapter 1. In this section, Relationship Ids and store item ids are discussed along with an explanation of how data and content are bound within a document. The Relationship Id of two separate .docx documents created on the same computer were examined. The same relationship Id, <Relationship Id="rId2"> was found in each. Uniqueness thus appears to mean unique within the document. Rice confirms this deduction, by stating that the id can be any string as long as it is unique within the .rels file [24].

With the new file formats, users can add their own data and content by creating their own custom-defined xml and placing it in the file as another part [24]. The value of <w:customXml> is that the user can markup elements within the document.xml file, which is useful for adding metadata/business semantics or for adding levels of granularity for search purposes. The

customized xml can be used with a tool or application that is capable of accessing and reading Office Open XML formats [24].

Data store items are used to distinguish the content pieces and to ensure the correct data is bound. The identifier is called the store item id and is attached to customXml by using a properties file[8].

The properties file defines the ID of the customXml part and the XML schema for the part. The property files are placed inside the customXml folder within the .docx package[23].

Each customXml part is related to its properties part through entries in customXml/_rels within the package. Assuming there are two custom parts - item1.xml and item2.xml, then the entries within customXml/_rels would be item1.xml.rels and item2.xml.rels.

The remaining piece that must be referenced is the link to the custom parts from the document.xml file. For this reference to hold, there will be a <w:databinding> element within the structured document properties [23].

StoreItem ids (introduced above) are assigned to pieces and referenced in the document.xml file. If the content is edited and saved again within Office 2007, new ids will be assigned. For instance, if the pieces are named piece1.xml and piece2.xml, they will be renamed to item1.xml and item2.xml once the document is edited and saved within Office 2007. All of the package references will be updated as well [23].

To an investigator, the various identifiers contained within an Office 2007 document provide a possible means of tracking user added content and data within a document. Of course, remembering that identifiers are only 32 bits long and that there is no way of guaranteeing that

they are unique is important. There exists a 1:232 chance that two specific documents will match by chance, and a much higher chance that at least two documents in a corpus will match – a direct result of the birthday paradox.

4.3.2 TIMESTAMPS

Time is frequently of critical importance in forensic investigations. Both Open Office and Office Open XML contain numerous internal timestamps indicating the time that documents were created or modified. Timestamps are there in the ZIP archive itself, (Figures 4.16, 4.17) in the embedded XML files, and potentially in other embedded objects (for example, in the EXIF headers of embedded JPEGs). Unfortunately, not all of the timestamps are precise. As demonstrated in Figure 4.17, OpenOffice.org sets the timestamps of the ODF file's ZIP archive to be the same as the system clock. The times are expressed in GMT, without a local time zone correction. Microsoft Word and Excel, on the other hand, set the timestamp on the ZIP archive to be January 1, 1980, the Epoch of the Microsoft FAT file system.

In addition to these ZIP directory timestamps, there are many different timestamps embedded within various XML sections, including:

• Word 2007, Excel and PowerPoint put the document's creation date in the tag of the core.xml file. The modified date is coded in the dcterms:modified element of the same file.

• PowerPoint 2007 additionally coded the document's creation date in the a:fld XML tag of each slideLayout.xml file.

• When change tracking was enabled in Word 2007, the XML file was annotated with multiple w:ins tags. Each tag included a w:author attribute with the editor's name, a w:date attribute with the date of the modification, and a w:id attribute with the ID number of the modification.

• A GOOGLE.COM webpage pasted from a web browser into a Microsoft Word document and then saved as a docx file contained timestamps embedded in the GOOGLE.COM URLs.

• OpenOffice.Org encoded the document's creation date in the meta:creation-date tag of the meta.xml section contained within blank ODP, ODS and ODT files.

• OpenOffice.Org likewise embedded a thumbnail.pdf file inside blank ODP, ODS and ODT files. This PDF file included comments for a CreationDate.

• OpenOffice.Org embedded the date in a text:date tag within the styles.xml file of a blank presentation.

Length	Date	Time	Name
1312	01-01-80	00:00	[Content_Types].xml
590	01-01-80	00:00	_rels/.rels
817	01-01-80	00:00	word/_rels/document.xml.rels
985	01-01-80	00:00	word/document.xml
6992	01-01-80	00:00	word/theme/theme1.xml

```
  1532        01-01-80 00:00   word/settings.xml
  1031        01-01-80 00:00   word/fontTable.xml
   260        01-01-80 00:00   word/webSettings.xml
   712        01-01-80 00:00   docProps/app.xml
   753        01-01-80 00:00   docProps/core.xml
 14840        01-01-80 00:00   word/styles.xml
--------                      --------
 29824                        11 files
```

Figure 4.16. Contents of an empty Microsoft Word 2007 document (Windows environment). Note that none of the timestamps have been properly set.

These timestamps might be significant in a forensic examination. For example, the timestamps potentially show when an ODF or Office Open XML file was edited with an ODF/Office Open XML-aware application. The timestamps might indicate multiple editing sessions. Alternatively, they might indicate tampering of a document. The timestamps might even be used in a file carver to determine which recovered pieces of a file match with other pieces.

```
Length      Date    Time    Name
--------    ----    -----   -----
```

```
      39  07-13-11 18:26   mimetype
       0  07-13-11 18:26   Configurations2/statusbar/
       0  07-13-11 18:26   Configurations2/accelerator/current.xml
       0  07-13-11 18:26   Configurations2/floater/
       0  07-13-11 18:26   Configurations2/popupmenu/
       0  07-13-11 18:26   Configurations2/progressbar/
       0  07-13-11 18:26   Configurations2/menubar/
       0  07-13-11 18:26   Configurations2/toolbar/
       0  07-13-11 18:26   Configurations2/images/Bitmaps/
    2756  07-13-11 18:26   content.xml
    8678  07-13-11 18:26   styles.xml
    1004  07-13-11 18:26   meta.xml
     729  07-13-11 18:26   Thumbnails/thumbnail.png
    1043  07-13-11 18:26   Thumbnails/thumbnail.pdf
    7476  07-13-11 18:26   settings.xml
    1959  07-13-11 18:26   META-INF/manifest.xml
--------                   -------
   23684                   16 files
```

Figure 4.17. ZIP directory for a OpenOffice ODT Word Processing file

4.3.3. ENCRYPTION

File encryption is another area that impacts an investigator's ability to extract metadata from files. During this research effort, only ODF and Microsoft Office 2007 documents were encrypted to determine the effect encryption had on the available metadata.

Open Office allows users to save a file with a password. The analysis of ODF files indicates that some sections are encrypted when a password is provided, but others are not. To test the encryption, an Open Office file with a single line was created and saved with a password. Nine files were stored in the resulting archive. Of those files, the following were not encrypted:

• META-INF/manifest.xml

• meta.xml

• mimetype

The following files were encrypted:

• Configurations2/accelerator/current.xml

• content.xml

• settings.xml

• styles.xml

• Thumbnails/thumbnail.pdf

• Thumbnails/thumbnail.png

Revealing the contents of the encrypted file would require cracking the encryption algorithm or guessing the encryption password, but forensic investigators may find the information in the unencrypted sections useful nevertheless. In the test document created with OpenOffice.Org, the following XML tags might be potentially relevant to an investigation:

• meta:generator---The specific build of the specific application that created the document.

• meta: creation-date---The document's creation date in local time.

• dc:language---The document's primary language

• meta:editing-cycles---The number of times the document had been edited

• meta:user-defined---User-definable metadata

• meta:document-statistics---Including the number of tables, images, objects, page count, paragraph count, word count, and character count.

Microsoft Office 2007 allows users to password-protect documents as well. Four different Microsoft Word 2007 documents were created in a Windows environment. Each document was encrypted using a password. The documents created included one with just text, another with text and an embedded image file; and a third with text, a text content control object, and an image content control object. The fourth document contained text and manually entered metadata such as subject, keywords, and comments. The intent was to determine which files within the archive would be encrypted and which would not for a comparison to the ODF encryption results.

The examination revealed that encrypting the documents protected the archive. Neither Filzip nor ZIP software recognized the encrypted document as a ZIP archive. However, the following files were extracted using 7Zip software:

• EncryptionInfo

• EncryptedPackage

• WordDocument

• [6]Dataspaces/DataSpaceMap

• [6]Dataspaces/Version

• [6]Dataspaces/DataSpaceInfo/StrongEncryptionInfo

• [6]Dataspaces/DataSpaceInfo/TransformInfo/StrongEncryptionTransform/

• [6]Primary

The contents of the above files did not reveal any meaningful information. However, the EncryptionInfo section contained the following readable text:

Microsoft Enhanced RSA and \AES Cryptographic Provider (Prototype).

The Primary file contained the following readable text:

FF9A3F03-56EF-4613-BDD5-5A41C1D07246

Microsoft.Container.EncryptionTransform AES 128.

Based on the analysis, encrypted Microsoft Office 2007 files appear to leak less information than ODF files, and therefore, would present move of a challenge for investigators needing to extract the metadata from these files.

4.3.4. THUMBNAILS

```
Archive:  pp1-save1-1.pptx
  Length      Date    Time    Name
 --------    ----    ----    ----
     3142  01-01-80 00:00   [Content_Types].xml
      738  01-01-80 00:00   _rels/.rels
      311  01-01-80 00:00   ppt/slides/_rels/slide1.xml.rels
      976  01-01-80 00:00   ppt/_rels/presentation.xml.rels
     3228  01-01-80 00:00   ppt/presentation.xml
     1072  01-01-80 00:00   ppt/slides/slide1.xml
      311  01-01-80 00:00   ppt/slideLayouts/_rels/slideLayout7.xml.rels
      311  01-01-80 00:00   ppt/slideLayouts/_rels/slideLayout8.xml.rels
      311  01-01-80 00:00   ppt/slideLayouts/_rels/slideLayout10.xml.rels
      311  01-01-80 00:00   ppt/slideLayouts/_rels/slideLayout11.xml.rels
      311  01-01-80 00:00   ppt/slideLayouts/_rels/slideLayout9.xml.rels
      311  01-01-80 00:00   ppt/slideLayouts/_rels/slideLayout1.xml.rels
      311  01-01-80 00:00   ppt/slideLayouts/_rels/slideLayout2.xml.rels
      311  01-01-80 00:00   ppt/slideLayouts/_rels/slideLayout3.xml.rels
      311  01-01-80 00:00   ppt/slideLayouts/_rels/slideLayout4.xml.rels
      311  01-01-80 00:00   ppt/slideLayouts/_rels/slideLayout5.xml.rels
     1991  01-01-80 00:00   ppt/slideMasters/_rels/slideMaster1.xml.rels
     3063  01-01-80 00:00   ppt/slideLayouts/slideLayout11.xml
     2839  01-01-80 00:00   ppt/slideLayouts/slideLayout10.xml
     4259  01-01-80 00:00   ppt/slideLayouts/slideLayout3.xml
     2784  01-01-80 00:00   ppt/slideLayouts/slideLayout2.xml
     4175  01-01-80 00:00   ppt/slideLayouts/slideLayout1.xml
    12065  01-01-80 00:00   ppt/slideMasters/slideMaster1.xml
     4530  01-01-80 00:00   ppt/slideLayouts/slideLayout4.xml
     7041  01-01-80 00:00   ppt/slideLayouts/slideLayout5.xml
     2051  01-01-80 00:00   ppt/slideLayouts/slideLayout6.xml
     1713  01-01-80 00:00   ppt/slideLayouts/slideLayout7.xml
     4620  01-01-80 00:00   ppt/slideLayouts/slideLayout8.xml
     4475  01-01-80 00:00   ppt/slideLayouts/slideLayout9.xml
      311  01-01-80 00:00   ppt/slideLayouts/_rels/slideLayout6.xml.rels
     7004  01-01-80 00:00   ppt/theme/theme1.xml
     2048  01-01-80 00:00   docProps/thumbnail.jpeg
      287  01-01-80 00:00   ppt/presProps.xml
      182  01-01-80 00:00   ppt/tableStyles.xml
      862  01-01-80 00:00   ppt/viewProps.xml
      675  01-01-80 00:00   docProps/core.xml
     1111  01-01-80 00:00   docProps/app.xml
 --------                   -------
    80663                   37 files
```

.

.

Figure 4.18 Thumbnails found in the .pptx created by PowerPoint 2007 on Windows XP

Embedded “thumbnail” images were found in the Microsoft Office files created by Open Office. Thumbnails were also found in the .pptx created by PowerPoint 2007 on Windows (Figure 4.18). No thumbnail images created by Word 2007 or Excel 2007 were encounteredThe thumbnails were, without exception, images of the document’s first page when rendered with the program that created the document file. The thumbnail images were examined for metadata themselves. The JPEG thumbnails only contained metadata for the image size and resolution (Figure 4.19)

Tag |Value

---

x-Resolution |72.00

y-Resolution |72.00

Resolution Unit |Inch

PixelXDimension |256

PixelYDimension |149

Figure 4.19 JPEG thumbnails containing metadata for image size and resolution

5.

Results and Discussions

Through the analysis of different file formats and the evaluation of existing metadata extraction tools and specific written extractors we have got following results regarding the questions raised under this research. The results are divided into subsections in this chapter to correspond the research question. These results are derived from the experiments and evaluation of the fiwalk under various files, file formats and plug-ins.

5.1 Feasibility of combining plug-ins into fiwalk

Through the analysis of various file formats and the evaluation of existing metadata extraction tools such as wvSummary, libextractor, docx_extractor and the specially written pdf_extractor, I determined that by integrating the plug-ins into the application fiwalk, metadata extraction tools could be combined for the automated processing of disk images.

Also I have found that by incorporating specific written plug-in pdf_extractor the metadata extraction capability is improved in this tool. This can be shown by the results achieved during the evaluation of pdf file before using pdf_extractor and after using the pdf_extractor.

The metadata extracted using the default plug-in libextractor is shown in Figure 5.1, which derives only two name:value pair and the metadata extracted from the same pdf file using the specially constructed plug-in pdf_extractor, which derives nine name: value pairs. It is shown in Figure 5.2. Clearly pdf_extractor extracts more metadata from pdf files.

format: PDF 1.7

mimetype: application/pdf

Figure 5.1 Metadata extracted by the default extractor (libextractor)

ModifyDdate: 20101129193532-08'00'

CreationDate :20101129193532-08'00'

format: PDF 1.4

title:

creator : TCPDF

subject :

producer: TCPDF 5.3.006 (http://www.tcpdf.org) (TCPDF)

keywords: TCPDF, PDF, example, test, guide TCPDF

mimetype:- application/pdf

Figure 5.2 Metadata extracted by pdf_extractor

In digital forensics, intelligence, law enforcement, and military organizations need a means to rapidly process and analyze captured information for evidence, and being able to use metadata within the data mining and data warehouse. Any additional information can become very important to these organizations. So capability improvement of existing metadata extractors can be very helpful to these organizations.

5.2 Comparison of similar type file formats

Metadata of similar file formats is also conducted to determine which file format gives better results to forensic analysts considering the metadata extracted from these files. To perform this task we considered Microsoft Office Open XML documents and OpenOffice.org documents. Because determining a clear cut metadata winner is not an easy task, comparing similar attributes between Open Document Format (ODF – Open Office) and Office Open XML files presents a clearer picture of the metadata potentially available for forensic investigations. As discussed in Chapter 4, three areas for direct comparison are timestamps, encryption, and thumbnails. These three points are discussed below to find out the winner between the Microsoft Office Open XML documents and OpenOffice.Org documents:

5.2.1 Timestamps Comparison

Timestamps are very crucial regarding any document because it provides valuable information to the forensic analyst. It can provide the information regarding when the file was created, when the file was last accessed or when the file was last time modified.

Length	Date	Time	Name
1312	01-01-80	00:00	[Content_Types].xml
590	01-01-80	00:00	_rels/.rels
817	01-01-80	00:00	word/_rels/document.xml.rels
985	01-01-80	00:00	word/document.xml
6992	01-01-80	00:00	word/theme/theme1.xml
1532	01-01-80	00:00	word/settings.xml
1031	01-01-80	00:00	word/fontTable.xml
260	01-01-80	00:00	word/webSettings.xml
712	01-01-80	00:00	docProps/app.xml
753	01-01-80	00:00	docProps/core.xml
14840	01-01-80	00:00	word/styles.xml
29824			11 files

Figure 5.3 Contents of an empty Microsoft Word 2007 document (Windows environment).

Length	Date	Time	Name
39	07-13-11	18:26	mimetype
0	07-13-11	18:26	Configurations2/statusbar/
0	07-13-11	18:26	Configurations2/accelerator/current.xml
0	07-13-11	18:26	Configurations2/floater/
0	07-13-11	18:26	Configurations2/popupmenu/
0	07-13-11	18:26	Configurations2/progressbar/
0	07-13-11	18:26	Configurations2/menubar/
0	07-13-11	18:26	Configurations2/toolbar/
0	07-13-11	18:26	Configurations2/images/Bitmaps/
2756	07-13-11	18:26	content.xml
8678	07-13-11	18:26	styles.xml
1004	07-13-11	18:26	meta.xml
729	07-13-11	18:26	Thumbnails/thumbnail.png

```
    1043          07-13-11 18:26   Thumbnails/thumbnail.pdf
    7476          07-13-11 18:26   settings.xml
    1959          07-13-11 18:26   META-INF/manifest.xml
--------                           -------
   23684                           16 files
```

Figure 5.4 ZIP directory for a OpenOffice ODT Word Processing file

Clearly by analyzing these results we can say that Microsoft Word and Excel, set the timestamp on the ZIP archive to be January 1, 1980, the Epoch of the Microsoft FAT file system. OpenOffice.org sets the timestamps of the ODF file's ZIP archive to be the same as the system clock. The times are expressed in GMT, without a local time zone correction. Therefore clearly both the softwares fails to provide accurate results.

5.2.2 Encryption

An Open Office file with a single line was created and saved with a password. Nine files were stored in the resulting archive. Of those files, the following were not encrypted:

- META-INF/manifest.xml
- meta.xml
- mimetype

Taking forensic analysis under consideration, following XML tags might be potentially relevant and useful for an investigator:

• meta:generator---The specific build of the specific application that created the document.

• meta: creation-date---The document's creation date in local time.

• dc:language---The document's primary language

• meta:editing-cycles---The number of times the document had been edited

• meta:user-defined---User-definable metadata

• meta:document-statistics---Including the number of tables, images, objects, page count, paragraph count, word count, and character count.

Similar type of file was created and saved with the password using Microsoft Word 2007 and the following files were extracted using 7Zip software:

• EncryptionInfo

• EncryptedPackage

• WordDocument

• [6]Dataspaces/DataSpaceMap

• [6]Dataspaces/Version

• [6]Dataspaces/DataSpaceInfo/StrongEncryptionInfo

• [6]Dataspaces/DataSpaceInfo/TransformInfo/StrongEncryptionTransform/

• [6]Primary

The contents of the above files did not reveal any meaningful information. However, the EncryptionInfo section contained the following readable text.

Thus by examining both type of files I found that encrypted Microsoft Office 2007 files appear to leak less information than ODF files, and therefore, would present more of a challenge for investigators needing to extract the metadata from these files.

5.2.3 THUMBNAILS

Embedded “thumbnail” images were found in the Open Office files created by OpenOffice.org Thumbnails were also found in the .pptx created by PowerPoint 2007 on Windows. No thumbnail images created by Word 2007 or Excel 2007 were encountered. The presence of thumbnails was an additional area for consideration when comparing the metadata of the document file formats. ODF files contain both .jpeg and .pdf thumbnails. In contrast, the Windows version of Microsoft Office 2007, only contributes a thumbnail image in the PowerPoint 2007 document archives.

5.3 Deficiencies in Existing Metadata Extraction Tools

Another aim of this thesis was to determine whether or not existing open source metadata extraction tools generate accurate results. During the research attempt, several shortcomings with the metadata extraction tools were encountered.

Understanding that these deficiencies exist is important for forensic investigators because some result may require additional analysis or research. A discussion of the deficiencies is discussed below.

WvSummary believed to be very useful in extracting most metadata from pre – 2007 Microsoft Office documents. However, some inconsistencies and inaccuracies were found after looking at the output of wvSummary. The output of wvSummary given in the Figure 5.5 states that total number of pages in this document is two but in the original document the total number of pages are three. This inaccuracy was encountered in other test files as well. However, the number of worksheets and slides, including hidden slides, was accurately reported by wvSummary.

Filename: /tmp/AashishResumBE.doc

Template: Elegant Resume

Security_Level: 0

Created: 2010-06-14T17:34:00Z

Last_Saved_by: Prakash

Revision: 6

Last_Printed: 2003-10-05T09:01:00Z

Keywords:

Subject:

Generator: Microsoft Office Word

Thumbnail: ((GsfClipData*) 0x91f7940)

Number_of_Characters: 1750

Last_Modified: 2010-07-06T06:58:00Z

Creator: Mayank

Number_of_Pages: 2

msole_codepage: 1252

Number_of_Words: 306

Editing_Duration: 2009-04-22T19:45:48Z

Title: Elegant Resume

Links_Dirty: FALSE

Number_of_Lines: 14

UseDefaultLanguage: TRUE

Version: 99022200

LCID: 1033

Document_Parts: [(0, Elegant Resume)]

Scale: FALSE

Number_of_Paragraphs: 4

Unknown_: FALSE

Unknown_: 786432

Company:

Document_Pairs: [(0, Title), (1, 1)]

Unknown_: 2052

Unknown_: FALSE

msole_codepage: 1252

Figure 5.5 Output of wvSummary under Microsoft word 2003 file

Also, within the document analyzed in Figure 5.5, comments were manually entered into the file, but comments were not one of the extracted pieces of metadata retrieved by wvSummary. The metadata added to the Microsoft Excel and Microsoft PowerPoint files was also captured by wvSummary.

Both post and pre Microsoft Word 2007 applications allow other Word documents to be embedded within a Word Document using the Insert/Object menu command. To test the efficiency of the metadata extraction tools, a Word document was embedded within a .doc and a .docx document. Figure 5.6 provides the ZIP archive of a .docx document with another Word file embedded within it. In the case of the .doc file, wvSummary and its sister tool wvText failed to identify the embedded file. Similarly, docx_extractor failed to identify the embedded document within the .docx archive. This test highlighted a bug within docx_extractor that needs to be fixed. Of note, contrary to the results observed from the open source wvSummary.

Issues were also encountered when using libextractor. In the initial trials, limited metadata was retrieved. The test files included .gif, .jpg, .pdf, and .html files. The metadata obtained included filename, file size, and mime type. Libextractor claims to provide an extensive list of keywords that can be matched within the metadata, but initial trials did not retrieve metadata such as title or comments.

Length Name

```
-------          ----
   1527          [Content_Types].xml
    735          _rels/.rels
   1107          word/_rels/document.xml.rels
   4780          word/document.xml
   6613          word/media/image1.png
   7559          word/theme/theme1.xml
  39832          docProps/thumbnail.jpeg
  25316          word/embeddings/Microsoft_Word_Document1.docx
   2036          word/settings.xml
    276          word/webSettings.xml
    734          docProps/app.xml
    726          docProps/core.xml
  15019           word/styles.xml
   1521          word/fontTable.xml
-------          -------
 107781          14 files
```

Figure 5.6 ZIP archive of a .docx document with another Word file embedded within it

Unfortunately, the Exif program utilized in jpeg_extract also demonstrated shortcomings. Exif processed digital images taken with a Canon EOS Rebel camera without any issues, but digital pictures taken with the Olympus FE-280 and Olympus Stylus 400 cameras were not recognized as being EXIF format.

By analyzing these shortcomings it is confirmed that the existing metadata extraction tools does have some shortcomings. These shortcomings can be rectified in future with more research and analysis.

6.

Summary & Conclusion

In this section I will list out the summary of the work carried out during this thesis and also the conclusions drawn upon this thesis.

6.1 Summary

Firstly we evaluated different type of extractors. These extractors were evaluated by entering different types of file under scan which generated different type of metadata. This study helped me to identify the basic metadata involved in different type of media and document files.

At next I constructed the specific metadata extractor for pdf files to extract the capability of metadata extractor tool fiwalk. Its result proved the feasibility of adding specific written metadata extractor in fiwalk.

After that comparison of metadata among similar file types was carried out. It was carried out to find out which file type gives more support to digital forensics. In this thesis we have chosen Office Open XML format and Open Document Format (ODF) for comparison, since both the formats are used to perform same type of operations on documents which include creating.

creating spreadsheet, text document and presentation. The comparison is carried out to find out which format gives better results as a metadata to forensic investigator.

After that I carried out some random experiments on fiwalk and compared the result of extracted metadata with the original file and attempted to find out whether existing open source metadata extraction tools used in forensic provides the accurate result.

6.2 Conclusion

By the analysis of various file formats and the examination of existing metadata extraction tools such as exif, wvSummary, libextractor, and the specially created pdf_extractor, it is concluded that it is feasible to add or integrate metadata plug-ins to improve the capability of metadata

extraction tool. Also by integrating the plug-ins into the application fiwalk, metadata extraction tools could be combined for the automated processing of disk images.

Because finding out a clear cut metadata winner is not easily done, comparing similar attributes between Open Document Format (ODF – Open Office) and Office Open XML files presents a better picture of the metadata potentially available for forensic investigations. As discussed in Chapter 4, three areas for direct comparison are timestamps, encryption, and thumbnails. During the research, timestamps were found in Open Office and Office Open XML documents. However, the time stamps were not always accurate.

Encryption was another area available for direct comparison between the document file types. During the research effort, only ODF and Microsoft Office 2007 documents were encrypted to determine the effect encryption had on the available metadata. Based on the analysis, encrypted Microsoft Office 2007 files appear to leak less information than ODF files, and therefore, would present more of a challenge for investigators needing to extract the metadata from these files.

The presence of thumbnails was an additional area for consideration when comparing the metadata of the document file formats. ODF files contain both .jpeg and .pdf thumbnails. In contrast, the Windows version of Microsoft Office 2007, only contributes a thumbnail image in the PowerPoint 2007 document archives.

Another aim of this thesis was to determine whether or not existing open source metadata extraction tools generated correct results. During the research effort, several shortcomings with the metadata extraction tools were encountered. Understanding that these deficiencies exist is important for forensic investigators because some results may require additional analysis or research.

Through the research effort, document files (pre – 2007 Microsoft Office, Microsoft Office 2007, and Open Office) were found to contain metadata that would be interesting from a computer forensic perspective. While media files also contained metadata, the information available would

be less useful for an investigator when compared to metadata available in document files. For instance, the metadata associated with an .mp3 file (artist, title, year, genre), which when compared to the metadata associated with a document file (file creator, revision number, modifier, description) is not as pertinent to a forensics investigation. However, some media files such as jpegs potentially contain metadata such as camera manufacturer and serial number, which an investigator would be interested in obtaining.

As previously discussed, Office 2007 documents provide a significant amount of information that can be extracted from a package beyond the standard author, creation/modification dates, that were obtained from prior versions of Office. The Office Open XML format presents the forensic investigator and forensic tools developer with a simpler environment in which to extract metadata. The presence of a thumbnail file or other embedded image within the package provides investigators with additional metadata that can be extracted and analyzed. By examining rsid values within the XML, investigators can identify documents that were created during the same editing session but were later dispersed and modified separately.

7.

Future Work

Given more time, I would integrate additional functionality into fiwalk. For example, adding automated feature extraction would provide a positive supplement to the metadata extraction capabilities provided by the plug-ins. Additionally, developing more metadata extraction plug-ins and modifying fiwalk to process all content with the plug-ins as opposed to just named files would increase the range of fiwalk.

Currently, a plug-in for extracting metadata from Portable Document Format documents has been written, but the plug-in needs additional work to ensure that metadata buried deep in the XML trees is recovered. Once the plug-in is completed and tested, it should be integrated within fiwalk.

Under the existing framework, metadata is not recursively extracted from every container encountered. By including this feature, metadata from files such as ZIP and tar archives as well as metadata from embedded .jpeg and document files inside .docx archives would be available for forensic investigators to perform more in-depth analysis.

Although fiwalk provides the option to produce the output in multiple formats such as ARFF and XML, being able to process the output as SQL would enable the user to automatically populate a

database with the metadata extracted from the files on the disk image for further analysis and faster data retrieval.

One limitation of working with metadata is that all metadata is susceptible to tampering. For instance, Office 2007 provides a means for programmers to search and remove metadata and

content from a .docx document. The effect on an investigator is that information that may have been previously extracted from an Office document left behind by a naive user may no longer be present. As another example, some file timestamps can be manipulated by individuals trying to distort or modify the timeline history of a file. A useful tool or plug-in for investigators would be one that detects and reports instances of tampering of metadata and if possible recovers the original metadata.

8.

Literature Cited

[1] CompuServe Incorporated. (1990). GIF graphics interchange format, version 89a. Retrieved 6/15/2011, from http://www.w3.org/Graphics/GIF/spec-gif89a.txt

[2] C. Grothoff. (2005). Reading file metadata with extract and libextractor. Retrieved 6/15/2011, from http://www.linuxjournal.com/article/7552.

[3] Hidden data in JPEG files. Retrieved 6/15/2011, from http://www.treatingyourself.com/vbulletin/showthread.php?t=23635.

[4] ID3v2Easy - ID3.org. Retrieved 1/25/2008, from http://www.id3.org/ID3v2Easy.

[5] Apple Education Creating content for iPod + iTunes. Retrieved 6/19/2011, from http://images.apple.com/support/itunes_u/docs/iTunes_U_Creating_Content.pdf.

[6] MPEG4IP - open streaming video and audio. Retrieved 6/19/2011, from http://mpeg4ip.sourceforge.net/documentation/index.php.

[7] OLE concepts and requirements overview. Retrieved 6/19/2011, from http://support.microsoft.com/kb/86008/en-us.

[8] wvWare, library for converting word documents. Retrieved 6/19/2011, from http://wvware.sourceforge.net/

[9] Introducing the Office (2007) Open XML File Formats , Retrieved 6/20/2011, from http://msdn.microsoft.com/en-us/library/aa338205.aspx

[10] PDF Reference and Adobe Extensions to the PDF Specification, Retrieved 6/20/2011, from http://www.adobe.com/devnet/pdf/pdf_reference.html

[11] Metadata - Wikipedia, Retrieved 6/21/2011, from en.wikipedia.org/wiki/Metadata

[12] Meta-data management – Wikipedia retrieved 6/21/2011, from en.wikipedia.org/Meta-data_management

[13] Constructing a metadata architecture, Retrieved 6/23/2011, from http://media.wiley.com/product_data/excerpt/32/04713552/0471355232.pdf

[14] Data Steward Wikipedia Retrieved 6/23/2011, from http://en.wikipedia.org/wiki/Data_steward

[15] Cyber Security Institute- What is Computer Forensic, retrieved 6/23/2011, from http://www.cybersecurityinstitute.biz/forensics.htm

[16]HowStuffWorks “How Computer Forensic Works”, retrieved 6/23/2011, from http://computer.howstuffworks.com/computer-forensic.htm

[17]Tools-Forensics Wiki, retrieved 6/23/2011 from http://www.forensicswiki.org/wiki/Tools

[19] On the role of metadata in digital forensics , retrieved 6/24/2011, from homes.cerias.purdue.edu/~florian/publications/metadata_jdi.pdf

[18]Use of computer forensics technology crime investigation by hyechin Blakeslee ,retrieved 6/24/2011 from, http://acsupport.europe.umuc.edu/~sdean/ProfPaps/Bowie/S09/Blakeslee.pdf
[20]Automating Disk Forensic Processing with SleuthKit, XML and Python by Simson Garfinkel, retrieved 3/25/2011 from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.149.5362&rep=rep1&type=pdf
[21] iText - Free / Open Source PDF Library for Java and C# retrieved 6/28/2011 from http://itextpdf.com
[22] Building word 2007 document templates using content controls. Retrieved 6/28/2011, from http://msdn.microsoft.com/en-us/library/bb264571.aspx?ref=carstuning.biz
[23] P. Aven. (2008). A final word. Part 6 in a series on MarkLogic server and office 2007. retrieved, 6/30/2011, from http://xqzone.marklogic.com/blog/smallchanges/2008-01-22?hl=%20final%20word.%20Part%206%20in%20a%20series%20on%20MarkLogic%20server%20and%20office%202007
[24] Frank Rice. (2006). Introducing the office (2007) open XML file formats. Retrieved 6/30/2011 from http://msdn2.microsoft.com/en-us/library/aa338205.aspx