DATA REDUCTION - REFINING THE SIEVE

Gord K. Hama, P. Eng.
Royal Canadian Mounted Police

Mark M. Pollitt
Federal Bureau of Investigation

ABSTRACT:

In times of ever increasing electronic storage capacity and devices; efficient and effective methods of identifying information of investigative interest must be developed.

This paper will address various techniques for reducing the volume of data needing to be reviewed when conducting investigations. The strengths and weaknesses of each method will be evaluated. Most examinations will likely require that a combination of tools & strategies be used depending on the type and volume of data being examined.

INTRODUCTION:

Any casual review of current computer literature clearly demonstrates that most home computer systems have hard drives with one or more gigabytes of storage capacity. Readily available tape drives currently store more at least a couple of gigabytes of data. Recordable CD-ROMs are becoming more and more common place both in the office and the home.

There is a continuing explosion of inexpensive electronic storage of huge and expanding capacity which is a boon to both users and software vendors. However, it is also becoming huge and expanding problem for law enforcement whose job it is to search electronic media for evidence of crimes.

The tremendous increase in storage can be likened to the proverbial needle in a haystack where the needle stays the same size and the haystack continues to grow! When storage capacity was relatively small it was possible to examine every last byte. Newer media are capable of containing thousands and tens of thousands of files. Time and resource requirements are no longer adequate to examine a typical system using traditional tools and methods. It is no longer physically possible to view each and every file, not to mention all the non-file space.

If one cannot view everything, then one must do the next best thing - view parts of it. But, how does one decide or select which parts to view? If one knew where to search or what to look at there wouldn't be any problems. However, it is usually easier to decide what NOT to look for.

This presentation discusses strategies for "**Data Reduction**" or filtering, by technical methods of non relevant information. Data Reduction thereby permits the isolation of information requiring further manual review for evidentiary value.

As each examination is always unique, this presentation does not suggest any protocol. A variety of tools and techniques may need to be employed for specific purposes and different types of examination.

THE EXAMINATION PROCESS:

It is worthwhile to review the examination process as it relates to electronic evidence. Specific techniques and methods may need to be designed for each and every examination. Typically they proceed through a three-step process:

1)The physical evidence is first examined to determine the origin, type, manufacturer, capacity, format and likely operating system. (This level of examination allows the Examiner to authenticate the evidence in court and also forms the foundation of the following steps).

2)The evidence is then examined logically, in the context of an operating system. The structure and organization is documented. (At this point it is also likely that a directory listing will begin to give the examiner an indication of what may be contained on the media).

3)Finally, the examination of the data in context can proceed ie: files, blocks, allocation units, etc. (This is the meat of the examination and is also where the problems arise). This vast landscape of data is what we must survey in hopes of finding pertinent information.

The examination process parallels information theory. At the logical level we have data, which is defined as raw intelligence stored in an organized fashion. When this information is viewed in context, such as in a Word Perfect file, it becomes information. If this Word Perfect file is then reviewed and found to contain information of probative value; it can then be submitted as evidence in court.

It is that middle step, determining what is pertinent and what is not, that is the focus of this paper.

APPROACHES TO DATA REDUCTION:

If one were assigned a research paper on a given topic, one might go to the library. Even in these days of electronic catalogues, it would not be very efficient to look at every book in the library to determine its pertinence to the project. When we go to the library we design our search to target areas which have a strong likelihood of having useful information. Conversely, we ignore areas not likely to contain useful information. This is the basis of data reduction.

For our purposes, we will take two different approaches in selecting where to search for potential evidence:

1)First, one looks at the basic unit of information, the file. Files are data that has been organized into a single unit. It is reasonable to believe that there is an underlying purpose to each file. It is that premise that is exploited.

2)Secondly, in most operating systems, there is also data that is not contained within files - the residue. This data may consist of deleted files, file slack, cache files, etc. It may also be worthwhile to examine these areas for potential evidence as well! (But, it takes a different methodology to examine these)!

It is important to understand that the nature of "Data Reduction" is to reduce or eliminate the amount of material to review. Other than by reviewing each bit of data, there is no way of insuring that pertinent information will not be overlooked. The Examiner must consider the costs and benefits of both techniques. There are many cases which will not require an exhaustive examination. The ultimate goal is to make the application of computing and human resources more cost effective and efficient.

FILE-BASED TECHNIQUES:

Examination of Directory Listings

An old adage says that "one cannot tell a book from its cover". While this analogy is even truer in the electronic context, reviewing directory listings is still a valuable technique. Although it is very simple to alter directory information in most operating systems and disguise the nature of the information; most people do not do not view their electronic files as potential evidence against themselves.

People, especially those in a business environment, rely on their files to conduct business. They don't usually bother to complicate matters for themselves by disguising the nature of their business on their electronic filing system.

Data and executable programs are stored as logically accessible entities called files. As a general rule, executables consist of static information, whereas data files tend to be dynamic in nature and are generally application-oriented.

Application data files may contain information of varying types, for example: encrypted data, textual, graphic, compressed, digitized sound, executable, etc. according to specific installed applications. Some examples of application data files are word processing and spreadsheet files.

It may be beneficial to target only data files. A review of the file directory can reveal, with a moderate level of confidence, the applications utilized by the computer. The question, "Could this application store information of investigative value"? assists in deciding whether to further review the contents of the application directories. Accounting data, for example, is unlikely to be contained in a subdirectory labelled and containing "Games".

File names within directories are also useful for determining relevance. Application data files can often be visually identified by their extensions and/or naming convention. Examples of this are *.EXE, *.DBF, *.DLL and *.c files. File names can be altered but unless there are indications of deception, they can often be used to eliminate entire classes of files, such as executables, or library files, or source code etc.

Dates and times may also be used, especially if the system clock is reasonably accurate and the date and time stamps of files are consistent. It is common to specify a search period when obtaining a search warrant. Obviously, this technique should not be the sole basis for eliminating files from review, but can seriously assist in reducing the number of files to be reviewed.

By combining the static nature of executables along with common practice (such as all application software files having a common file creation date and time); one can infer that files with application consistent names, common file dates and times are likely not relevant.

Although these techniques are imprecise they have their uses. If we know what we are looking for these techniques may suffice. For example: consider the demonstration of possession and transmission of child pornography via an online service during a specific time frame. A review of the directory may reveal a directory with the name of the online service containing a large number of JPG or GIF files. The investigator can then match directory information, if not file name, to tentatively identify a downloaded file. This is clearly more efficient than loading and viewing each and every graphic file.

File Headers:

This method identifies file types such as executables and application specific data files. File type identification would be made by comparing internal structures (file headers) with known file characteristics. This methodology would require a database of known file type characteristics for comparison.

File types can be mislabelled, but many files are internally identifiable by having application signatures imbedded in the file. Many types of files have unique headers. Often, these headers must be present for the application to make use of the file.

Headers can be used to identify those files which are inconsistent with their file extensions or type. This will alert the Examiner that extra care must be taken in examining this file system and that the subject may be attempting to conceal data.

File headers could be used either in conjunction with, or in place of some of the directory search techniques discussed above.

File Authentication

Computers rely on a huge quantity of software to provide the functionality that is required by users today. Operating systems, applications, and hardware drivers can add up to hundreds of megabytes of files. An experienced Examiner will learn to visually identify many of these files. However, no one can recognize all of them and there is always the possibility that the file has been altered to conceal data. What is needed is a way in which to authenticate a given file or group of files.

It is suggested that some form of "Authentication Vector", or AV, could be used. Known files could be checksummed with the resulting AVs then loaded into a database. After which, unknown files could be subjected to the same mathematical processes ie: checksummed. The resulting AV could then be compared to those of known files. These could then be used to filter out files that are known to not be of investigative importance.

This method would positively identify known commercial products by file authentication techniques and would require use of a database of file information and pre-established AV's (similar to checksum or CRC). Product AV's would have to be initially loaded into a database, and would have to be kept current to include new applications and product revisions/updates.

Perhaps, with the use of surveys, the most popular products could be determined. The top 100 or so most commonly used applications could be entered. This methodology would be somewhat costly in CPU time and logistics support because each suspect file would have to undergo an AV computation and comparison. (To match the AV known samples). This activity could be conducted in a laboratory environment. This method would be somewhat time consuming.

By including common products with numerous static components ex: Windows-3.1/NT/95, MS-DOS and Novell; this method could easily identify any known components. (This particular task would require a large database due to the numerous known versions).

At the present time, the cyclical redundancy check (CRC-32) is an option. It has been determined that this particular algorithm is not fool-proof. If a greater level of certainty is needed a more sophisticated algorithm could be developed.

String Search

String searching is also an acceptable and still widely used method of finding character strings in files and residue. There are pitfalls however.

String searching looks for specified strings. (It can however, produce reams of irrelevant information depending on the uniqueness and quality of the search strings). Based on the ASCII character set, textual/numeric information is easily isolated, such as credit card (cc) and phone numbers, names, addresses and various other commonly sought information. For example numbers which match patterns can be easily detected by implementing a numeric filter template.

String searching also implies a language based search, fortunately, the ASCII character set is widely used in North America. As the world shrinks due to the spread of technology and global commerce, there will certainly be an increased demand for non ASCII based language keyword searches. Several examples have already surfaced, ie: a foreign language version of a word processing package was used to create documents. Files stored in a graphical format are not amenable to string searches.

Over the years, enhancements have been made to string searching, such as the use of the "SOUNDEX" algorithm that includes searches for phonetically close keywords. To our knowledge this methodology has yet to be implemented in PC based text searching.

String searching is an intensive linear process, where files are sequentially searched for specified strings, consuming time. Risks can include missing intelligence in files/residue, a dependency on specified search string(s), and the fact that information being sought may not be stored as identifiable ASCII text string. Text searching alone may be too specific. As all examiners know, the potential for missing good intelligence and evidentiary material is always present.

Non-file Space:

Information that occupies unused storage space is referred to as residue, and may contain intelligible information from previous use. In MS-DOS filesystems, residue is normally found in "free", "reserved" and "lost" clusters, and in "FileSlack". Lost and reserved clusters are the least common. "Lost" clusters represent file data that is not assigned to any file. "Reserved" clusters are those deemed to be unusable by the initial disk formatting process. Residue may consist of a mix of textual, encrypted, graphic, executable, digitized sound, compressed and other types of information. The uniqueness of residue is that it can contain complete files, file fragments, memory contents or just plain old "junk".

File contents become residue through the file creation/deletion process. Because of its origin, residue usually consists of a mix of everything left by whichever application have been used. For performance reasons most systems do not actually overwrite files upon file deletion, but rather, the file is marked as deleted and the space occupied by the file is freed for re-use (in the event that it is needed). If newer information is not written over the previous information, the deleted file's contents, remain intact and is usually recoverable.

Although not always 100% accurate, a logical approach to analyzing residue is to "undelete" or, reclaim residue back to its file form. Once recovered, deleted files can then be identified by AV, header or by content and analyzed accordingly.

Another form of residue, known as FileSlack, is written to disk with the last cluster of file information. Since the last cluster is "owned" by the file, FileSlack remains untouched until overwritten, perhaps by another file. Fileslack contents are unpredictable. Usually they consist of the contents of memory buffers at the time that the file was written to disk or created. Depending on the amount written, the last portion of FileSlack may remain unchanged because disk is physically accessed by sectors.

A popular method of analyzing residue is to filter out non ASCII characters and view or search the resulting text. The increasing media sizes dictate that alternate strategies be used. Perhaps for ASCII text searches, the filtering process could apply basic language rules to further filter out residue. Even with ASCII filtering in place the yield is massive.

At present, residue analysis has not progressed much beyond filtering out all but ASCII characters and repeating patterns.

This level of filtering is starting to produce unmanageable amounts information.

Future Directions:

Computer forensic examiners will face an ever increasing volume of work. The number and diversity of systems will increase. The capabilities and storage capacities will grow. Manually examining systems has become impractical in most instances. The examination tools will have to evolve and become more sophisticated by incorporating different methodologies to accommodate future forensic needs. Techniques and methods to automate the process and reduce the work load must be developed. Data reduction should include filtering functions for ASCII, Numerics, and redundant patterns as well as an optional language based filter.

The ability to efficiently examine systems will depend on the ability to automatically perform the functions which do not require the skill and judgement of the examiner. Perhaps, a combination of authentication vector, header information and content analysis could be used to identify useful, irrelevant and potentially pertinent information.

Perhaps a low risk, automated approach might combine file identification by AV to eliminate the most commonly encountered commercial software and file header analysis, followed by language based text searches of the remaining files for intelligible textual content.

By using automated tools to reduce data to manageable quantities, the risk of missing valuable information remains. Examiners will have to remain cognisant of the fact that there are risks involved in all these processes. The level of risk could be kept manageable by utilizing tools incorporating various up-to-date methodologies.