An Architecture for the Forensic Analysis of Windows System Artifacts

Noor Hashim and Iain Sutherland

Faculty of Advanced Technology, University of Glamorgan, United Kingdom
{nhashim,isutherl}@glam.ac.uk

Abstract. We propose an architecture to enable the forensic investigator to analyze and visualise a range of system generated artifacts with known and unknown data structures. The architecture is intended to facilitate the extraction and analysis of operating system artifacts while being extensible, flexible and reusable. The examples selected for the paper are the Windows Event Logs and Swap Files. Event logs can reveal evidence regarding logons, authentication, accounts and privileged use and can address questions relating to which user accounts were being used and which machines were accessed. The Swap file may contain fragments of data, remnants or entire documents, e-mail messages or the results of internet browsing which may reveal past user activities. Issues relating to understanding and visualising artifacts data structures are discussed and possible solutions are explored. We outline a proposed solution; an extraction component responsible for extracting data and preparing the data for visualisation, a storage subsystem consisting of a database that holds all of the extracted data and the interface, an integrated set of visualization tools.

Keywords: Forensics, Visualisation, Open platform.

1 Introduction

In searching for evidence as part of the forensic process, considerable effort is focused on exploring the contents of the file system and any deleted material that may reside on the media. This will often involve keywords or pattern matching techniques to examine data based on names, content or metadata possibly relating to temporal information, such as the last accessed or written time to be listed [4]. The results therefore can be file content, data fragments and metadata. The investigator can follow a forensic process model to aid the investigation. A forensic process model can be described as follows: for each file, perform a number of type-specific operations such as indexing, keyword searches and thumbnail generation. Thus, the model applies to evidence such as deleted files, file slack, registries, directories and other operating system structures that includes system artifacts. The challenge in digital forensics is to find and discover forensically interesting, suspicious or useful patterns within often very large data sets [2].

I. Baggili (Ed.): ICDF2C 2010, LNICST 53, pp. 120–128, 2011.

2 Forensic Analysis of Windows System Artifacts

A digital forensic investigation of a hard drive can involve analyzing a large volume of evidence derived from numerous files, directories, unallocated space and file systems [13]. Therefore the forensic analysis of Windows system generated artifacts can be one of many different activities undertaken during a digital forensic investigation. Previous authors [2] have commented on digital forensics' unique requirements and these have to be considered when analyzing Windows system artifacts. These include; the relationship between instances of data, data sources and the issue of false negatives when executing a search over a large volume of data.

3 Windows System Artifacts

There are a number of system generated files of potential evidential value; hidden files, web artifacts, temporary and system files. System files are created as a routine function of the operating system and often without reference to the user. These artifacts are therefore important for digital investigators as they capture a user's activities, but are often overlooked by users if they attempt to conceal or remove evidence of their activities. System files are normally obscured from the average user and require specific knowledge to find and in some cases are only visible or accessible if specialized tools are used. Therefore there should be an element of the forensic process that is focused on capturing and analyzing the information contained in these files.

Table 1. Windows System Artifacts

Event Logs - Event log files record information about which users have been accessing specific files, successfully logging on to a system, unsuccessfully to log on to a system, track usage of specific applications, track alterations to the audit policy, and track changes to user permissions [10].
Swap File - A swap file is a disk-based file under the exclusive control of the Memory Manager [16].
Registry - A central hierarchal database that maintains configuration settings for applications, hardware devices and users. [5].
Recycle Bin – Part of the file system that contains files no longer required by the user. A user may then retrieve a file that has been deleted by mistake, providing the Recycle Bin has not be emptied (placing the file in unallocated space) [6].
Web Cache - Web browsers e.g: Internet Explorer cache the content of visited web pages and cookies within system files. In the case of IE named index.dat [8].
Prefetch - Prefetch caches take information from the boot process and from Scheduled Tasks to speed up boot and application launch time [7].

Based on existing studies [1], [5], [6], [8], [10], [12], analysis of system artifacts play significant role in informing a digital investigation [1], [5]. The ease with which these systems artifacts can be accessed and interpreted depends upon the degree of structure and the form of encoding used in that particular artifact. In some cases the information is stored in plain text, in a highly structured human readable fashion. In other cases, as these files are not intended to be access by the user the system files,

they may be encoded and the structure may be unclear without a degree of processing and interpretation. Table 1 includes a brief description of six artifacts represent example of Windows system artifacts.

3.1 Event Logs Evidentiary Values, Features, Tools and Related Issues

Event logs records contain a significant degree of information concerning the activities that occur on a system. They are used to diagnose and troubleshoot issues on a system as they record information about hardware and software problems. According to [10], by reviewing Event logs, a variety of information of evidentiary value can be obtained: they may record successful and unsuccessful logon attempts, user access to specific files, track usage of specific applications, track alterations to the audit policy and track changes to user permissions.

In one example relating to access across a network, Date, Time, IP addresses and Computer Names can be used to determine which computer was used to perform a specific action [6]. Therefore event logs can play an important role in addressing intrusion cases relating for example to the misuse of remote desktop connections. The Event ID column contains a number that indicates the type of event that has occurred. The Event ID is most commonly associated with logon and authentication activity. The Event ID can also be useful in identifying the name and IP address of the computer where the connection originated.

Windows systems record the event that occur on a system into one of three log files: AppEvent.Evt, SecEvent.Evt and SysEvent.Evt. These three files record within many facets of a systems behavior.

Table 2. Event Logs Organisation

File Name	File Location For Windows NT/XP, 2000, Vista
Application Event Log AppEvent.Evt	Contains a log of application usage and logged messages from the operating system and programs. %SYSTEMROOT%\system32\config\
Security Event Log SecEvent.Evt	Records activities that have security implications such as logins. %SYSTEMROOT%\WINNT\config\
System Event Log SysEvent.Evt	Notes system events such as shutdowns. %SystemRoot%\system32\winevt\Logs

Table 2 illustrates the different locations that the various versions of Windows store the .Evt file. According to [1], in order to facilitate the examination of the contents of an event log, the event log header and event records have some structure, values and information that can assist an investigator in recognizing and interpreting the files. These values include date, time, user, computer, event ID, source, type and category [11]. It is maintained as a circular buffer where older event records are overwritten once the file reaches a specified size and when a new event record is added to the file. At the same time, there is correlation between the event logs, registry and many of message files (DLL) on a system [5].

Information, warning and error entries are stored in the Application and System logs, while success and failure are recorded in the Security log. This type of event is frequently used in attempts to troubleshoot system anomalies and used with other column fields to determine evidence of a breach, or attempted breach of the computer system. In Windows, each of the different versions of the operating systems used logon type to indicate different kind of logon event and these nine logon types are values to indicate the way in which the account logged on to the system.

Since these logs are stored in a proprietary binary format and not in a Human readable format, appropriate tools are required to access and interpret the data. Event Viewer by Microsoft is one example. This depicts the event logs in two different panes. One pane shows the list of the available log files and the other provides a list of each different event entry. Other tools that rely on the Windows API is 'Log Parser'. For Log Parser the processing of event logs is done by three engines: its input engine, its SQL engine (which uses SQL queries to parse, filter and analyze logs) and its output engine. LANguard Security Event Log Monitor (LANSELM, from GFI) is a network-wide tool that retrieves events from NT/2000 servers and workstations and alert the administrator of possible intrusions.

One important element of forensic value is the timestamp. Event logs record the date the entry was made and the time entry was written in the log. Windows stores timestamps in FILETIME format and in GMT but does not contain any information concerning the time zone. Therefore, when comparing a timestamp with another system, in addition to compensating for any clock variation, differences in time zones may have to be considered.

Another issue is the data loss that occurs when an event ID has been updated on a newer version of the operating system, and an older version of the operating system is used to interpret the event log files. This problem also arises in relation to SIDs when a log from a different computer is analyzed instead of local machine. The evolution of Windows operating system also creates problems as it has resulted in changes to the way that logs are generated, the evidence found would therefore be a consequence of the version of the operating system that a victim is using.

3.2 Swap File Evidentiary Values, Features, Tools and Related Issues

The swap file is a portion of disk storage used for memory pages belonging to various processes [9] and threads [15] and also stores CrashDump data when a "blue screen of death" occurs [14]. This swap file can provide a great deal of information, specifically passwords [6], [9], user IDs and information that the user did not intend to save to the disk. The later could include chat information, credit card numbers, URLs, print spooling and numerous other user activities [13].

When a computer's RAM is full, the operating system allocates memory for an application, Windows creates swap files on the root folder of the system drive to make more RAM available. The default swap file location is as shown in Table 3. The Swap file is generated at each boot session and is closed on system shutdown. However, the shutdown period increases when the swap file is configured to clear out by setting the registry value to 1 for: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SessionManager\MemoryManagement\ClearPageFileAtShutdown.

The swap file is locked by the kernel when Windows is running. However, it is possible to access the swap file either by using a specially written driver or by accessing the swap file by removing the files from a 'cold' system, by forensically imaging the hard drive. The swap file size depends on the volume of RAM present on the system and how much virtual memory space is required by a particular workload. If the minimum and maximum swap file size is not the same, fragmentation happens and this can lead to performance degradation.

In this project various computer forensic filters were designed to automatically identify string based information present in a Windows swap files.

Table 3. Swap File Default Name and Location

File Name	Windows Version	Location
Win386.swp	Windows 95/98/ME	%SystemRoot%\
pagefile.sys	Windows NT/2000/XP/Vista	

4 Architecture and System Requirements

The aim of the project is to develop an extensible open source architecture for system generated artifacts. The previous sections highlight the evidential value of these files. The current state of the art forensic tools (EnCase and FTK) are capable of capturing the system-generated files present on a drive. These commercial tools also provide standard features to search and pattern match the contents of these files. However the coverage of system generated artifacts varies; facilities to process and explore the registry tend to be feature rich, whereas those for analyzing swap files and event logs are less well developed. The information on the artifacts above and the functionality provided by the commercial tools provides a baseline for the development of a series of minimum requirements for the proposed architecture.

4.1 Functional Requirements

The following functional requirements have been indentified: The need to automate the analysis of system artifacts (extraction, processing and presentation) to help the investigator to explore and understand the content of the files. There is also a requirement to facilitate the analysis and visualization of forensic data from various types of file format and data with different degrees of complexity. The system will need to be able to understand the variation between different versions of the operating system, to analyze evidence items collected from different platforms. The system will also need to present and summaries the information of interest for reporting purposes.

4.2 Security, Software Quality and Other Requirements

In terms of security and other requirements the following have been identified. As a forensics tool, data integrity is essential and the original copies of the data / media must remain unchanged and this should be verifiable. Some form of hashing function (MD5, SHA-1 or similar) is required to prove that the evidence has not been modifed.

A further particularly essential requirement for a forensics tool is accuracy. The accuracy of the generated information output from the analyzed files and shown using textual visualization technique is of paramount importance.

The system should be both reliable with stable, repeatable performance and scalable to add visualization techniques capability to additional evidence items as the need arises. In term of extensibility the following two requirements were identified:

- Extensibility-1: It is easy to add support for new types of data sources regarding analysis.
- Extensibility-2: Additional functionality can be added through plugins or modules, as well as scripting capability via an extensive and usable API.

The extensibility should also support dynamic reconfiguration, modifications and enhancements must be possible without taking the whole system down and implementation should be independent of underlying software.

Confidentiality is important and methods shall be considered to protect data within the evidence item from being disclosed to unintended parties. Finally there should be some form of integrity assurance. To provide audit record about timestamp or actions taken, or results returned from running utilities.

5 System Architecture

This research concentrates on the examination of the system artifacts' data structures and transforming the data into structured form, thereby helping the investigator by automating the time consuming aspect of low-level analysis of the system file format and related data complexity. The basic approach consists of these four steps:

The first step is to prepare the artifacts to be processed. The artifact to be processed comes in one of these formats: exported from imaged hard drive using tools such as Mount Image Pro or FTK, acquired from Digital Evidence Bag container of digital evidence obtained from disparate sources, stored using the Advanced Forensics Format (AFF) to indicate imaged disk storage and compressed data used to store digital evidence.

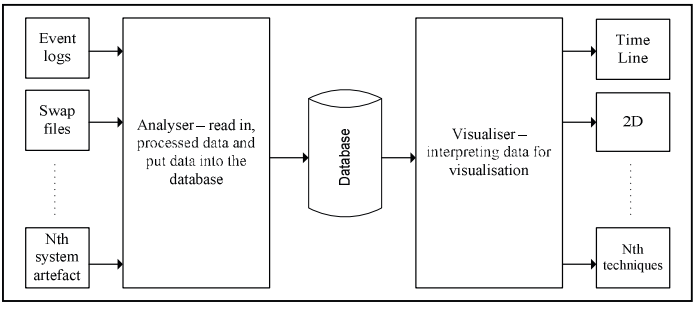

Fig. 1. High Level Description of Overall Architecture

The second step will be to invoke analyzer to identify and extract data in each of the artifacts being processed. The analyzer will need to be easily reconfigurable to interpret the variety of artifacts.

The third step is to invoke the database as a data store that enables data to be queried and retrieved.

The fourth step in the process is to transform the data into an easily readable format. It is a mechanism where data is queried from the database, data representation and further analysis. This is used to structure the data into a narrative construct.

The top-level design of the Windows System Artifacts Analysis system is shown in Fig. 1.

5.1 Analyzer, Database and Visualizer

The key features of the proposed system are therefore the analyzer, the database and the visualization system. The analyzer component firstly reads in the evidence item and translates it into usable structure. This performs the automated analysis. Once processed, the data is sent to the database to allow for further future analysis by the operator. The main process of the analyzer consists of five processes:

1. Locate: Locate the fields of the data structure (the units of information that comprise the data structure).
2. Extract: Extract the fields of the data structure from the raw stream of bytes.
3. Decode: Further extraction is necessary, specifically the bit fields. Examples of bit fields are: flags, attributes, date field, time field, etc.
4. Interpretation: Takes the output of the decoding phase or the extraction step and performs various computations using the information. Examples: the value for the years of date field and second of time field need to be interpreted.
5. Reconstruction: Information from the previous step is used to reconstruct a usable representation of the data structure or at least the relevant fields.

Part of the initial system configuration for processing a particular artifact, in addition to providing the analyzer with information on the structure of the artifact, will be configuring the database. The database must be available to store data and this data can be rapidly retrieved and queried. It must be extensible in term of accepting data without any knowledge of the structure and extraction of the data for display or report. This database can be regarded as an interface between the two components to communicate in a common language that overcomes any specifics related to the syntax and semantics of the data set.

The proposed design of the visualizer module further analyzes, interprest and understands the data generated from the analyzer component and to output the data for the user to analyze by examining the data found in more detail. This provides interfaces to the database, to display the artifact data in selective views and thus focus on particular aspects of the data or logical flow. The visualizer consists of eight processes:

1. Acquire: Obtain the data, whether from a file on a disk, or a source over a network.
2. Parse: Provide some structure for the data's meaning.

3. Filter: Remove any irrelevant data.
4. Mine: Apply methods from statistics or data mining as a way to discern patterns or place the data in mathematical context.
5. Represent: Determines the basic form that a set of data will take, such as a graph.
6. Refine: Design methods are used to improve the basic representation to make it clearer and more visually engaging.
7. Interact: Add methods for manipulating the data or controlling what features are visible.
8. Output: The information file is translated into required output format and output to the screen.

6 Conclusion and Future Work

This is a research-in-progress. The contribution lies not on the ability to manipulate and visualize the event logs and swap files, but rather the development of a flexible and extensible architecture to process various artifacts based on a clear understanding of the priority of the requirements.

Forensic analysis has a number of unique requirements that directly impact the design of the architecture. For example, the need to interact with multiple disparate data types suggests the development of plugins and flexibility. The architecture should be able to process other system artifacts (e.g. Registry, Internet Explorer Activity Files, Prefetch file) besides event logs and swap files. Our ongoing work includes the implementation of the architecture to enable it to visualize known and unknown data structure of files. This will enable the investigator to easily determine what data of interest is available within these system areas of Windows.

References

1. Anson, S., Bunting, S.: Mastering Windows Network Forensics and Investigation, Indiana (2007)
2. Brown, R., Palm, B., de Vel, O.: Design of a Digital Forensics Image Mining System (2005), http://www.springerlink.com/content/3a7t7cxk3mdrajb0/
3. Caloyannides, M.A.: Computer Forensics and Privacy, Boston (2001)
4. Carrier, B.: File System Forensic Analysis, Indiana (2005)
5. Carvey, H.: Windows Forensic Analysis DVD Toolkit, Burlington (2007)
6. Casey, E.: Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet, Boston (2004)
7. Hay, S.A.: Windows File Analyzer Guidance (2005), http://www.mitec.cz/Downloads/WFA%20Guidance.pdf
8. Jones, K.J.: Forensic Analysis of Internet Explorer Activity Files, Forensic Analysis of Microsoft Windows Recycle Bin Records (2003)
9. Lee, S., Savoldi, A., Lee, S., Lim, J.: Windows Pagefile Collection and Analysis for a Live Forensics Context. J. Future Gen. Comm. and Net. 2, December 6-8 (2007)
10. Mandia, K., Prosise, C., Pepe, M.: Incident Response & Computer Forensics, New York (2003)

11. Microsoft TechNet: Fundamental Computer Investigation Guide For Windows: Overview (2007)
12. Murphey, R.: Automated Windows event log forensics. J. Digital Investigation 4S, S92–S100 (2007)
13. Nelson, B., Phillips, A., Enfinger, F., Steuart, C.: Guide to Computer Forensics and Investigations (2008)
14. Ruff, N.: Windows Memory Forensics. J. Computer Virology 4S, S92-S100. The British Library (2007)
15. Schuster, A.: Searching For Processes And Threats In Microsoft Windows Memory Dump. J. Digital Investigation 3S, S10–S16 (2006)
16. The NT Insider: Windows NT Virtual Memory. Open System Resources. V. 5, I. 2 (1998), http://www.osronline.com/custom.cfm?name=articlePrint.cfm&id=60