A Study of Forensic & Analysis Tools

Sriram Raghavan
Secure Cyber Space (*www.securecyberspace.org*)
Email: sriram.raghavan@securecyberspace.org

S V Raghavan
Department of Computer Science & Engg., IIT Madras
Chennai, INDIA
Email: svr@cs.iitm.ernet.in

Abstract— ***There are a wide range of forensic and analysis tools to examine digital evidence in existence today. Traditional tool design examines each source of digital evidence as a BLOB (binary large object) and it is up to the examiner to identify the relevant items from evidence. In the face of rapid technological advancements we are increasingly confronted with a diverse set of digital evidence and being able to identify a particular tool for conducting a specific analysis is an essential task.***

In this paper, we present a systematic study of contemporary forensic and analysis tools using a hypothesis based review to identify the different functionalities supported by these tools. We highlight the limitations of the forensic tools in regards to evidence corroboration and develop a case for building evidence correlation functionalities into these tools.

Keywords— Digital evidence, Binary abstraction, File system and schema support, Metadata, Evidence composition

I. Introduction

The challenge we face today is that the forensic tools currently in existence are highly specialized for certain types of digital evidence. Primarily, Encase and FTK only support computer based digital evidence, such as hard disks and memory dumps. Besides, these toolkits are highly monolithic in design and hence functional integration with complementary forensic tools remains a challenge. The highly specialized nature of these tools has also precluded their inter-operability, barring a few exceptions[1]. This paper develops a hypothesis based approach to identify the different functionalities supported by forensic and analysis tools to examine and analyze digital evidence.

A. Forensic Tools

There are several computer forensic tools both in the commercial and the open domain. The commonly used forensic toolkits for analyzing file systems are Encase, FTK, X-Ways, Nuix, TCT, Sleuthkit, DFF, OCFA, Snorkel and LibForensics. Of these, Encase[2], FTK[3] and X-Ways[4] are commercial toolkits while TCT, Sleuthkit [2], DFF, OCFA, Snorkel and LibForensics are in the open domain[5]. Among these tools, most commercial varieties also support the examination of memory dumps and mobile device flash memories.

All forensic tools take a forensic image of the "source" as input and provide binary abstractions to raw data. This allows the entire source to be read as a binary stream of data. In our work, we refer to this functionality as the *binary abstraction*. The tools also distinguish the different files and their application formats on the file systems using standard file signatures [3]. A notable feature of this technology is the development of the *known file filter* (KFF) to omit system files during evidence examination. We refer to the functionality of recognizing files and automatically associating them with their application to parse the file as *file system support*. These two functionalities address the complexity problem in digital evidence [1]. The tools extract file system metadata associated with each file including the location of the file, MAC timestamps, file ownership, file size and so on. Typically forensic tools do not rely on application metadata and consequently do not extract or parse them. To provide that functionality, one may resort to analysis tools.

B. Analysis Tools

Analysis tools are usually specialized; there are analysis tools for examining files, memory dumps, log files, network packet captures and so on. Some examples of such analysis tools are Volatility [16] for memory dumps, PyFlag [5] for log files and network packet captures, GrokEvt, libevt and Event Log Parser for Windows event logs, AWStats for web browser logs, RegRipper, python-registry, Forensic Registry EDitor and Win32Registry for Windows Registry and Wireshark [6] and tcpdump [14] for network packet captures. It is notable that PyFlag has since integrated the Sleuthkit and Volatility and in that way allows the examination of forensic disk images and the analysis of memory dumps. Although these tools may not operate on forensic images, they guarantee read-only access which is a requirement for maintaining the integrity of the digital evidence.

The analysis tools directly access the "source" and parse the contents as independent records. Each record can contain several attributes, including timestamps, which are parsed for analysis. This functionality can also be broadly classified under the *schema support*, as part of the *file system support* layer. We term the ability to parse or extract metadata, including file system metadata, application metadata and all related attributes of an artifact in a non-intrusive manner as *metadata parsing*. Log analyzers such as PyFlag, GrokEvt, libevt, Event Log Parser parse the respective logs and their attributes. Typically, the attributes in such logs contain an

[1] Pyflag is an example
[2] http://www.guidancesoftware.com/encase-forensic.htm
[3] http://www.accessdata.com/products/digital-forensics/ftk#.UeTux6q6aM8
[4] http://www.x-ways.net/forensics/index-m.html
[5] http://www2.opensourceforensics.org/tools

Sponsored by IEEE Louisville Chapter.
978-1-4799-4061-5/13/$31.00 ©2013 IEEE

event description, username associated with the event, event timestamp and so on. Wireshark and tcpdump parse corresponding attributes from network packet captures. Network packet attributes can include a packet sequence number, protocol for communication, source and destination IP addresses, hosts' MAC addresses, hosts' operating systems and browser applications and so on. A series of keyword based searching and filtering is used to conduct the actual analysis.

C. Classification & Grouping of Artifacts

Typically, forensic and analysis tools can classify the artifacts using the file metadata or log or network attributes parsed, one attribute at a time. File owner, username, last modified or event timestamp, protocol, source or destination IP address, are some attributes that are commonly used during analysis. However, when deep analysis is required, the artifacts are often classified multiple times using different attributes. In some cases, this can be a laborious task, particularly, when the attribute or combination of attributes which hold the answers are yet unknown. This disparity is more pronounced when the sources of digital evidence span different file and log formats or source types.

II. Hypothesis based review

Our approach to conducting this review of forensic and analysis tools was grounded on hypothesis testing. Based on the tool testing capabilities [10, 11] identified at the National Institute of Standards and Technology (NIST), we developed hypotheses to determine the support extended by the different tools. In our review, the hypotheses concerning the different tools are listed as follows: A forensic or analysis tool can …

1. *successfully load a source of digital evidence*
2. *read binary data on a source of digital evidence*
3. *interpret binary data on a source of digital evidence*
4. *recognize different file systems on a source of digital evidence*
5. *identify the individual digital artifacts on a source of digital evidence*
6. *extract/parse the metadata from individual digital artifacts*
7. *combine/group multiple digital artifacts based on metadata*
8. *combine/group multiple digital artifacts using metadata in an unconstrained manner*
9. *interpret the semantics of the metadata linked to a digital artifact*

In order to test these hypotheses, we conducted the following experiment. The experiment is discussed in general and was applied to each forensic or analysis tool in turn to determine the outcome. We created a forensic (raw) image of a volume partition containing a FAT32 file system and an NTFS file system. The file system contained files created with *normal user behavior under corporate setup* and contained different word processing files (Word documents, MS PowerPoint, MS Excel, Rich Text Format files, Adobe PDF files, Text files, digital images. The files contained both file system and application metadata and recorded events related to activity on the files. The files mainly contained blank content or random content that weren't intended for use during the experiments. We also created a Windows XP SP2 raw memory dump, a browser history log from a web login session; and a network packet capture from the web login session on the same system used to create the files. Our experiment involved the following tasks:

1. *Load the sources on a tool check completion status for each source*
2. *Read the first 10 and last 10 bytes of the image and print it to user interface*
3. *Display the displayed content into Hexadecimal and into text*
4. *Identify and list the file systems on the image*
5. *Identify and list all the digital artifacts on the image. Digital artifacts corresponds to:*
   a. *files on a file system*
   b. *process control blocks on memory dumps*
   c. *log records on a log file*
   d. *network packets on packet captures*
6. *On each digital artifact, parse/extract metadata from the system as well as the application. This corresponds to:*
   a. *File system and application metadata on files*
   b. *Operating system memory map and process attributes on memory dumps*
   c. *File system metadata of the log file and log record attributes on log files*
   d. *File system metadata of the packet capture and network packet attributes on network packet captures*
7. *Identify methods to group two or more artifacts on the image using metadata*
8. *Identify combinations of metadata that are supported by the tool to group 2 or more artifacts*
9. *Modify a file metadata and replace the values for Author with "788755 bytes" and file size with "Jeffrey". Recreate the forensic image with new file and load it into the tool. Now extract the metadata and note observations.*

The tasks corresponded with their respective hypotheses. In each case, a success corresponded to accepting the proposed hypothesis and a failure corresponded to rejecting the hypothesis. If a tool was able to read and print the content, it supported binary data. In addition if the tool was also able to translate the content into hexadecimal and text, it supported interpretation. Listing the file system was applicable only to the forensic disk image, while on the other sources, success was implied by listing the digital artifacts (log records or network packets) on the source. For each digital artifact that was successfully traversed, the metadata was parsed for extraction.

In literature, keyword filtering and classification are the common methods to group artifacts on a source. We conducted such filtering using metadata both explicitly and implicitly. When we used metadata explicitly, we used the values assumed by the metadata as specific keywords to conduct searches while in the implicit method, we programmed to use the metadata label. We conducted multiple sequences of grouping and regroupings to determine which combinations of metadata were permitted by each tool. In regard to metadata semantics, if a tool flagged an error for replacing expected values on the metadata, then we interpreted the outcome as a YES. On the other hand, if the tool did not raise a flag, then the tool was syntactic by design. Our findings are summarized in

Table I. The symbol '√' denotes the presence of a particular functionality while the symbol '×' denotes its absence.

TABLE I. TABULATING THE RESPECTIVE FUNCTIONALITIES OF VARIOUS FORENSIC AND ANALYSIS TOOLS

	Digital Evidence access	**Digital Artifact Traversal & Examination**					**Metadata Parsing & Extraction**	**Evidence Composition using metadata**	
	Binary abstraction to DE[6]	*File system examination*	*Memory dump examination*	*Log examination*	*Packet capture examination*	*Text indexing and Search*		*Multiple sources of DE (examination and analysis)*	*Identify correlations*
Encase Forensic[7]	√	√	√	×	×	√	Only FS[8] metadata	Can group artifacts using FS metadata, one at a time	×
FTK[2]	√	√	√	×	×	√	Only FS metadata	Can group artifacts using FS metadata, one at a time	×
X-Ways Forensics[2]	√	√	√	×	×	√	Only FS metadata	Can group artifacts using FS metadata, one at a time	×
Nuix Investigator[2]	√	√	√	×	×	√	Only FS metadata	Can group artifacts using FS metadata, and keywords, configurable	Can correlate from specific keywords across content
Sleuthkit	√	√	×	×	×	√	Only FS metadata	Can group artifacts using FS metadata, one at a time	×
PyFlag	√	√	√	√	√	√	Only FS metadata	Can group artifacts using FS metadata, one at a time, can classify using by combining timestamps across sources	×
OCFA	√	√	×	×	×	√	Only FS metadata	Can group artifacts using FS metadata, one at a time	×
DFF	√	√	×	×	×	√	Only FS metadata	Can group artifacts using FS metadata, one at a time	×
Snorkel	√	√	×	×	×	√	Only FS metadata	Can group artifacts using FS metadata, programmable prioritization	×
Nirsoft log analysis	×	×	×	√	×	√	Log attributes	Can classify using one attribute at a time	×
GrokEvt	×	×	×	√	×	√	Log	Can classify	×

[6] DE = digital evidence
[7] These are the respective commercial product names
[8] FS = file system

							attributes	using one attribute at a time	
Libevt	×	×	×	√	×	√	Log attributes	Can classify using one attribute at a time	×
RegRipper	×	×	×	√	×	√	Log attributes	Can classify using one attribute at a time	×
Volatility	√	×	√	×	×	√	Memory attributes	Can classify using one attribute at a time	×
Log2timeline	×	×	×	√	×	×	Log attributes	Multiple timestamps can be combined for time-lining	×
Wireshark	√	×	×	×	√	√	Network packet attributes	Can filter using multiple attributes; classify using one at a time	×

III. DISCUSSION

From this review, we elicit that all forensic tools provide binary abstraction to forensic images to handle forensic images of hard disk drives or memory dumps. While the commercial toolkits may support both file system images as well as memory dumps, most open source forensic tools predominantly handle only file system images, albeit in different image formats. File system contain metadata associated with file activity which is independent of file content and forensic tools extract these metadata to identify the owner, MAC timestamps, access privileges and so on. However, these tools do not, under normal circumstances, extract or use application metadata from files, which also contain valuable information. All forensic tools support text indexing and searching on an image and classify the artifacts on the image according to the file system metadata. While these tools support multiple forensic images, they do not seem to correlate the different metadata across files and alert an examiner when related metadata are discovered. Besides, log files which can also be found on many file systems are processed as files by these tools which have to be exported for analysis.

Most analysis tools, with the possible exception of Volatility or Wireshark, do not provide binary abstraction. These tools interpret the contents and process the data as independent entries while parsing the respective attributes for reporting. The analysis tools usually process one source at a time and occasionally support indexing and searching. The analysis tools also support classification of the log entries based on the parsed attributes, however, the functionality to combine multiple attributes to derive semantic relationships is also necessary.

Both forensic and analysis tools group their respective contents using two techniques, keyword filtering and attribute classification. Typically one may need to filter the contents based on different keywords or classify based on different attributes during analysis to determine a pattern. Evidently, these techniques are keyed by a human and unless the right combinations of keywords and attributes are specified, the pattern being sought is likely to be missed. Some attributes can also be combined during classification, even if sequentially. The most common type of combining attributes for classification as reported in literature [9, 17] involved combining timestamps with owner for forensic images, *username* for log files and *IP address* for network packet captures. This leaves the remaining metadata and attributes largely underutilized.

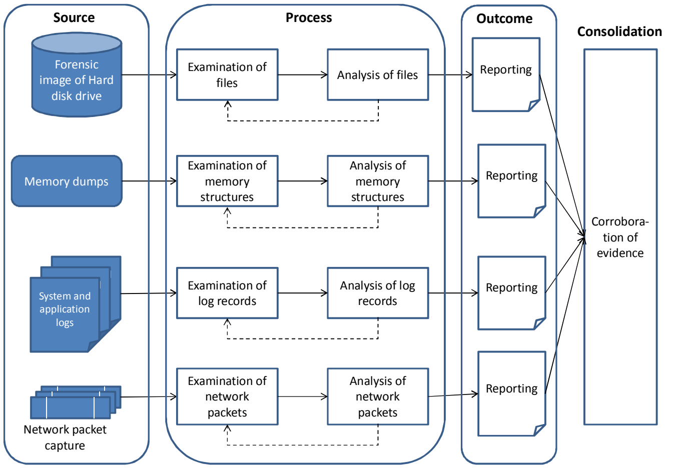
Source
Process
Outcome
Consolidation
Forensic image of Hard disk drive
Examination of files
Analysis of files
Reporting
Memory dumps
Examination of memory structures
Analysis of memory structures
Reporting
System and application logs
Examination of log records
Analysis of log records
Reporting
Network packet capture
Examination of network packets
Analysis of network packets
Reporting
Corrobora-tion of evidence

Fig. 1 Traditional method for conducting forensic analysis on different sources

Traditionally in computing environments, hard disk drives were the dominant source of digital evidence and as a result analysis was predominantly conducted on files. Today, however, in addition to hard disks, data is also found on volatile memory, log files and network packets, that too in different formats. As a consequence, system and application logs, volatile memory images and network packet traces have become equally important. When heterogeneous sources of evidence are analyzed using traditional tools, redundancy in processing the evidence becomes unavoidable, as illustrated in Fig 1. In fact, even among multiple sources of the same type, redundancy results. Processing digital evidence in the traditional manner contains four parts; *source*, *process* (examination and analysis), *outcome*, and *consolidation*. For each one of the sources that require processing, the artifacts need to be examined and analyzed individually for generating relevant reports that are corroborated in the final step. The workflow described hitherto underlines the need for a cohesive approach to analyze diverse sources of digital evidence to arrive at a consolidated outcome.

Based on the understanding gained from the work reported in this paper, the authors have developed AssocGEN [13], an engine based on FIA architecture [12] for integrating forensic disk images, file systems, system and application logs and network packet captures by integrating heterogeneous digital artifacts based on metadata based associations. This architecture utilizes existing tools to build the lower layer support and focuses on grouping related digital artifacts to aid in forensic analysis.

IV. Summary & Future

In this paper, we presented a systematic study of contemporary forensic and analysis tools to examine and analyze digital evidence. We presented a system of hypotheses using which a specific tool can be studied to identify the different capabilities that it provides for examining one or more sources of digital evidence. This study highlighted the significance of metadata and has drawn the need to use metadata across heterogeneous sources of digital evidence for corroboration and analysis.

Garfinkel [8] has observed that present-day forensic tools are designed to find new pieces of digital evidence but the analysis continues to remain largely manual. There is a need to consolidate the research findings into a more unified form of forensic examination providing a seamless transition to analysis, especially with multiple sources of digital evidence. There is a need for forensic tools which can identify metadata based associations, such as those proposed in the FACE [4] and FIA architectures, in an unconstrained manner both within a single source like files on a forensic disk image and across multiple heterogeneous sources.

In the future, the authors intend to extend their AssocGEN architecture in identifying and grouping related artifacts for conducting forensic analysis. There is a lot of scope to develop efficient algorithms for identifying metadata based associations in digital evidence and grouping the related artifacts to aid in an investigation.

References

[1] Carrier, B. D., (2003). Defining Digital Forensic Examination and Analysis Tools Using Abstraction Layers. *International Journal of Digital Evidence (IJDE),* Vol. 1(4), pp. 1-12.

[2] Carrier B D., (2003), Sleuthkit, http://www.sleuthkit.org/sleuthkit/, last retrieved on July 12, 2011

[3] Carrier, B. D., & Spafford, E. H. (2004). An Event-based Digital Forensic Investigation Framework, *Paper presented at the 4th Annual Digital Forensic Research Workshop (DFRWS '04).*

[4] Case A, Cristina A, Marziale L, Richard G G and Roussev V. (2008). FACE: Automated Digital Evidence Discovery and Correlation, *Digital Investigations, Proceedings of the 8th Annual Digital Forensic Research Workshop (DFRWS '08),* 5(Supplement 1), pp. S65-S75.

[5] Cohen M I. (2008). PyFlag – An Advanced Network Forensic Framework, Digital Investigations, *Proceedings of the 8th Annual Digital Forensic Research Workshop (DFRWS '08)*, 5(Supplement 1), pp. S112-S120.

[6] Combs G. (1998). Wireshark – Network Protocol Analyzer, http://www.wireshark.org/about.html, last retrieved on July 12, 2011

[7] Garfinkel S L. (2009). Digital Forensic Research: The next 10 years, Digital Investigations, *In Proceedings of the 10th Annual Conference on Digital Forensic Research Workshop (DFRWS '10)*, Vol. 7(2010), pp. S64-S73.

[8] Garfinkel S., (2009), Automating Disk Forensic Processing with Sleuthkit, XML and Python, *In Proceedings of the 2009 Fourth International IEEE Workshop on Systemmatic Approaches to Digital Forensic Engineering* (SADFE 2009), Berkeley, California, ISBN: 978-0-7695-3792-4, pp. 73-84.

[9] Minack E., Paiu R., Costache S., Demartini G., Gaugaz J., Ioannou E., Chirita P-A, and Nejdl W., (2010), Leveraging personal metadata for Desktop Search: The Beagle ++ System, *Journal of Web Semantics: Science, Services, and Agents on the WWW, Elsevier Science Publications,* ISSN: 1570-8268, Vol. 8(1), pp. 37-54.

[10] NIST. (Nov 2001). General Test Methodology for Computer Forensic Tools. *www.cftt.nist.gov/Test Methodology.doc pp. 1-8, 2001.*

[11] NIST. (Jul 2010). Computer Forensic Tool Testing handbook, *http://www.nw3c.org/docs/wccrc/cftt-booklet-071910.pdf?sfvrsn=1*, pp. 1-127, 2010.

[12] Raghavan S., Clark A J., and Mohay G. (2009). FIA: An Open Forensic Integration Architecture for Composing Digital Evidence., *Forensics in Telecommunications, Information and Multimedia, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 2009*, Volume 8(1), pp. 83-94, DOI: 10.1007/978-3-642-02312-5_10

[13] Raghavan S. and Raghavan S. V. (2013). AssocGEN: Engine for Analyzing Metadata Based Associations in Digital Evidence, *In Proceedings of the 2013 8th International Workshop on Systematic Approaches to Digital Forensics Engineering (SADFE)*, IEEE 978-1-4799-4061-5, Hong Kong, China, Nov 21-22, 2013.

[14] TCPDUMP, Command-line packet analyzer, http://www.tcpdump.org/, last retrieved on July 12, 2011

[15] Timeline analysis part 3, log2timeline, http://thedigitalstandard.blogspot.com/2010/03/timeline-analysis-part-3-log2timeline.html, last retrieved on July 12, 2011

[16] Volatility – Volatile memory artifact extraction utility framework, Volatile Systems, https://www.volatilesystems.com/default/volatility, last retrieved on July 12, 2011

[17] Zander S., Nguyen T. and Armitage G. (2005)., Automated Traffic Classification and Application Identification using Machine Learning, *In Proceedings of the IEEE Conference on Local Computer Networks,* IEEE LCN 2005, Sydney, Australia, ISBN: 0-7695-2421-4, pp. 250-257.