VISUAL CORRELATION IN THE CONTEXT OF POST-MORTEM ANALYSIS

Michael Hayoz and Ulrich Ultes-Nitsche

Research group on telecommunications, networks & security
Department of Informatics, University of Fribourg, Switzerland

{michael.hayoz | uun}@unifr.ch, Bd de Pérolles 90, CH-1700 Fribourg

ABSTRACT

One of the biggest challenges in the field of digital forensics lies in the ability to bring all potential evidence into a chronological correlation, and to draw appropriate conclusions to allow plausible and reproducible chains of activity. The ever-growing size of storage devices results in a considerable amount of information to process. Analysing forensic data, as a general rule, is a time-critical process. The output of current forensic tools mostly has the form of exhaustive lists and tables and can be difficult to manage and interpret. These constraints leave forensic specialists with a need for improvement in the way they handle huge amounts of suspect data. The present paper introduces an attempt to optimize the post-mortem analysis by means of visualization. The approach uses output data of current forensic tools and allows investigators to visually build correlations, with the aim of getting hints as to where it would make sense to start looking for evidence. Formerly unrelated primitive objects are visually classified and aggregated to more complex objects through attributes. As a result, disk-images can be searched for occurrences of patterns similar or close to the ones specified. Search results can be of different type; possible examples are graphs of statistical distributions or even self-organizing maps.

KEY WORDS

digital forensics, post-mortem analysis, visual correlation

VISUAL CORRELATION IN THE CONTEXT OF POST-MORTEM ANALYSIS

1 INTRODUCTION

The field of digital forensics is considered as a branch of common forensic science, and has increasingly detached itself from the broader area of computer security to become a self-contained forensic discipline over the past ten years. Digital forensics are defined in several ways. (Computer Legal Experts, 2007) states it as being *"*[...] *the application of computer investigation and analysis techniques in the interests of determining potential legal evidence".*

The process of forensic analysis aims at answering questions about former system states and events by reproducing chains of digital activity. These chains of activity are the result of bringing potential legal evidence into a chronological progression and represent one of the most difficult tasks for an expert. It is fairly easy to collect information from a system; the complexity lies in the ability to correlate bits and pieces into a reproducible sequence of past events. Several tools and toolkits have been developed to assist forensic experts and security specialists in their daily work, and have proven their reliability during the process of forensic analysis. There are both open and commercial products available, *The Sleuth Kit* (Carrier, The Sleuth Kit) and *EnCase* (EnCase Forensic, 2008) being two prominent and widely used examples. The authors emphasize that the approach at hand will focus on open source tools only, for the time being.

1.1 Motivation

A forensic investigation is a time critical process. In most cases, external circumstances determine the time available to experts to find supportive evidence. Efficiency and an intuitive handling of large sets of data are of prime importance to the process of forensic analysis. Most security incidents implicate more than just one storage medium. Discussions with Swiss security and forensic experts (Bundeskriminalpolizei, 2008) have shown that whenever a set of storage media has to be examined, it is done sequentially. Either, because there is not enough equipment at hand or simply, because qualified human resources are low.

The data that eventually represent evidence, are but a small fraction of the data stored on a disk. Furthermore, the capacities of storage media keep increasing, which makes it even more difficult for specialists to know where it makes sense to start looking for evidence among the data to examine. Many of the forensic tools currently in use generate their results as lists or tables, whose length depends on the number of matchings found after applying one or several filters these tools provide. This leads to the fact, that the process of forensic analysis becomes more and more complex in terms of getting a quick overview of the data, and the efficiency of the way that data is being processed.

Studies (Miller, 1956) show that the ability of the human brain to understand complex structures and the relations they induce can be significantly increased through visual stimuli. Hence the approach presented in this paper builds upon the assumption that there is a need for a simplified, assistive means, which allows for a coherent view on the structures and relations of data of different type through the use of abstract visualization. The focus of this work is set to the so-called *post-mortem* analysis, which will be discussed in more detail in Section 3.

The next section will give a brief overview of the process of forensic analysis and describe the different phases it consists of. Section 3 will outline the post-mortem analysis to introduce the context of the presented approach. Section 4 gives an insight into the basic features of common open source forensic tools. The fifth section will present the approach the authors suggest, and the last section will conclude with a brief summary and outline future work.

2 THE PROCESS OF FORENSIC ANALYSIS

The process of forensic analysis defines a sequence of actions to be taken in the event of IT security incidents, e.g. where one or several computers have either been used as a target, or as a means to commit a crime. As a general rule, the authors divide this process into the following 4 phases:

1. Coverage of the crime scene – covering a crime scene goes beyond the seizure of suspect hardware. The surroundings have to be given just as much attention. For detailed information on crime scene investigation, refer to (Fisher, 2000).

2. Data acquisition – if the suspect system is still running, all volatile data (this concerns all data held in RAM at runtime and temporary files on the hard drive) is to be recovered, if possible without changing the system's actual state. For further details on *live acquisition*, refer to (Carrier, 2005). The second step during data acquisition is called *forensic duplication* and consists of creating exact copies (images) of all hard drives and related media like USB sticks, CD-ROMs etc. to a clean hard drive. This process can be performed locally or over a secured network channel. In depth information on disk imaging can be found in (Carrier, 2005).

3. Post-Mortem analysis – all acquired data images are examined and searched for supportive evidence within a secure environment. This analysis is always performed on copies, never on the original data. Post-mortem analysis will be discussed in more detail in the next section.

4. Consolidation of the investigation's results – all potential evidence is put in chronological correlation, which allows for investigators to draw appropriate conclusions, in order to rebuild plausible and reproducible chains of activity for further use before court.

3 POST-MORTEM ANALYSIS

A post-mortem or *dead analysis* is performed on copies of duplicates gathered during data acquisition. Forensic experts can work without the pressure of a live system, since there is always a backup of the original image available, if necessary. Data acquisition, as a general rule, is done at the disk-level. Loss of possible evidence has to be avoided; this is why disk images should not be created at the volume, file or application levels. For example, if the data would be acquired at the file level, non-allocated space would not be copied and hence make a recovery of deleted files impossible. Potential evidence is lost at every level of abstraction; therefore data, as a rule of thumb, should be acquired at the disk level in order to save every byte that may contain evidence. However, there are situations, in which an investigator might decide to duplicate data at a higher level. It is up to the expert in charge to decide, where evidence is most likely to be found. This decision mostly depends on the expected type of attack, and the experience of the specialists assigned to the case.

A post-mortem analysis examines all the data gathered from a suspect system for potential leads and evidence. This analysis is done on all possible levels, spanning from the application level down to the disk level, if applicable. Points of interest are unallocated space on hard drives (including slack space), MAC-times (last modification, access, change), swap space, hidden files, deleted files, the structure and content of unknown binaries, log files and operating system related information (kernel version, loaded modules, registry information on Windows etc.), to name a few. It is highly recommended to start with recovering deleted information when conducting a post-mortem analysis (Jones, Bejtlich, & Rose, 2005). Most perpetrators make sure to delete all information relevant to an investigation, before leaving a system. Furthermore, experience shows that a set of suspect data can be reviewed more efficiently, if it is previously reduced to what is relevant to the process of finding evidence. There is much more to say about post-mortem analysis, but doing so would be out of the scope of this paper. Suffice it to say, that all the steps of such an analysis can be performed with the aid of current forensic tools. Detailed information on how to conduct a post-mortem analysis can be found in (Farmer & Venema, 2005).

As aforementioned, the phase of post-mortem analysis sets the foundation for the approach suggested in this paper. The next section will briefly discuss the information one can extract with most of the current open source forensic tools.

4 OPEN SOURCE FORENSIC TOOLS

Ever since digital forensics became important to criminal investigation, people have been working on tools to assist specialists and simplify the task of finding relevant information on corrupted systems. Most of these tools significantly improved the process of forensic analysis, mostly by providing scripts to automate or partly automate the acquisition, recovery and analysis of suspect data. Authorities work with both commercial and open source toolkits. Discussions and experience reports show, that one of the most recurrent drawbacks of current forensic tools is the graphical user interface (GUI) or lack thereof. If available, these GUIs are often complex in their usability and make it difficult to get a fast overview of relevant information. However, efforts have been made to address the issue: Brian Carrier's *Autopsy* (Carrier, Autopsy Forensic Browser, 2008) tool is a notable example to account for these efforts.

Most toolkits are available as a live CD and assemble a collection of useful tools for live and dead data acquisition, as well as tools for forensic analysis for both Unix-based (The UNIX system, 2008) and Windows operating systems. Many suppliers rely on Linux distributions with good hardware detection capabilities, such as debian (debian, 2008) or KNOPPIX (KNOPPIX, 2008), which builds upon debian. These distributions are very convenient in that they allow for an immediate forensic analysis environment to be set up. The live CD can be mounted on a still running system, commonly referred to as a *smoking gun*, and an incident response can be performed out of the box. Statically pre-compiled binaries are used in order to avoid the execution of any system binaries, which might have been tampered with, root kits being a current example. An investigator can mount a system's partitions in read-only mode; execution of system binaries is prevented as well. Most live CDs provide a host of utilities to extract valuable runtime information, such as RAM content, process information, network information (open sockets etc.) and other temporary data, which would be lost after a system shutdown.

Tools for data acquisition are indispensable for any forensic toolkit. As a general rule, a "good" toolkit will allow experts to duplicate both dynamic and static data to any clean hard drive or over an encrypted network channel. As for the analysis of acquired disk images, the possibilities are far-reaching. Data recovery on different disk levels, timeline analysis (through data timestamps), analysis of unknown binaries and meta-data analysis, are but a few of the possibilities offered to specialists. EnCase has become the state of the art solution for digital forensics among all available commercial products on the market. Its feature set is impressive indeed, but many of the available products developed in the open source community can hold their ground and offer a huge potential for both research and development alike.

The next section discusses a new approach to assist forensic specialists in their work during a post-mortem analysis.

5 VISUAL CORRELATION

This section introduces an approach to optimize the post-mortem analysis by means of visualization. Based on the facts stated throughout this paper, the authors suggest to make use of the advantages brought into play by visual and interactive assistance to simplify the process of rebuilding past chains of activity. These chains eventually result from correlating initially unrelated data and the appropriate interpretation of an investigator, which relies on past experience, to obtain legal evidence. The present research is in its early stages; a prototype implementation is not yet available but will be established in the near future.

The authors propose a GUI, which uses any set of results from current forensic tools as input. This makes sense because all suspect data has already been reduced to a subset of relevant data at this point. As a first step, the input data needs to be pre-processed in order to be graphically displayed. This is done through a logical interface, which recognizes the type of information contained in the input set and abstracts it to classes of graphical entities. For example, if the input data contains information on i-nodes, access times, log files and system processes, the pre-processing will find out about four different types of information. From a graphical point of view, the interface will display these four entities, each of which stands as a representative for its respective type of information (e.g. i-NODE, MAC, LOG and PROC). This abstraction is needed, because displaying every single entity of the input set would result in an unreadable and hence unusable mix-up.

Each decision one makes depends on former action. The same holds for any event on a computer system. Getting back to the suggested approach, the process of correlating entities can be performed the very same way. Before getting into any further detail, another assumption needs to be made. Every data structure is described through a set of properties. These can be meta-data, file names, file extensions, file content, network class, process information, to name just a few. So each of the representative classes can be assigned a set of possible attributes to describe them. An investigator can now use this information to visually build correlations between different classes. Back in the GUI, a selection of attributes can be made for each of the initial graphical entities. The next step consists of aggregating several representatives to form a possible correlation. Consider the following example to clarify this process. The GUI initially displays 3 representative classes, according to the content found in the input set. These entities are IP, @ and LOG. The investigator would like to find out, if a specific e-mail address can be put in relation with one or several IP addresses. So he/she first selects the e-mail address from the @ class' *sender / recipient* attribute and proceeds the same way to select a range of IP addresses suggested by the IP class. Both graphical entities are then visually correlated, e.g. graphically connected to each other, to form what the authors call a *visual pattern* or *interrogator*. This process of visual correlation is illustrated in Figure 1.

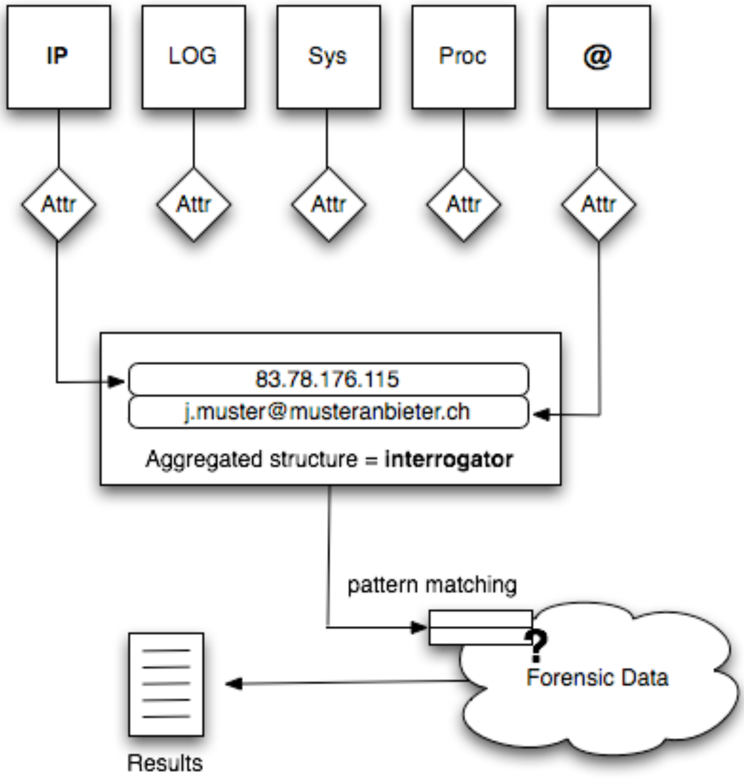

Fig. 1 The process of visual correlation

On a logical level, an interrogator is nothing else than a pattern, which is eventually applied to the input data set. The forensic data is being searched for matching or similar patterns. In the above example, the result might contain one or several entries from a log file, which states that J. Muster has sent or received e-mail on the machine with IP 83.78.176.115. This new information might give the investigator a new lead and will determine the next steps, which might either consist of building a whole new pattern or reusing the previous one to correlate even further, e.g. to refine or restructure the actual interrogator using additional attributes and classes. Another focus will be set on reusability. Most attacks follow specific patterns and stand out through unusual activity. Interrogators can be saved for reuse, or even previously specified and applied to a certain category of forensic data with similar structure.

In a sense, this concept allows to ask questions in a top-down manner, and to reformulate these questions, should the answer be unsatisfying for the examiner. The authors believe that this approach will increase the efficiency with which investigators correlate suspect data and interpret it to retrace past events and system states. The concept presented above provides a simplified view on structures and relations of suspect data, and removes a part of the complexity when it comes to evaluating long lists of results.

6 CONCLUSION

The authors have introduced a new approach to the process of post-mortem analysis, more precisely, to the correlation of initially unrelated data. They propose a visual concept to assist investigators in the process of reproducing chains of previous system activity. The complexity and effort, needed to process large sets of data, is reduced through abstraction. Content of input data sets is classified and displayed by means of graphical class representatives. Visual correlation allows forensic experts to easily specify search patterns, which can be applied to forensic data.

The suggested GUI is comparable to the evidence finding process in common forensic science. Potential evidence of different type (fingerprints, pictures, textile fragments, DNA etc.) is collected and correlated, to verify if there is any relation between them that might be used to incriminate a suspect.

6.1 Future Work

As aforementioned, this work is current research in its early stages. A first prototype has to be developed to deliver a proof of concept. This will allow verifying the use and the applicability of the authors' assumptions. The logical interface, which pre-processes and classifies input data sets as graphical representatives, is about to be specified. Each of the possible representatives with their respective set of attributes will be specified with XML Schema (Vlist, 2002). The prototype will be implemented with the Java programming language (Flanagan, 2005) to allow for maximum portability on different platforms.

Different possibilities to report results are also being considered. First of all, it has not yet been decided how results are being processed and displayed to the user. It might be of interest to offer the ability to choose from different representations. One might include a plug-in mechanism to provide a flexible means to add new reporting types at a later point. Possible types could be graphs and diagrams to visualize the number and relations of particular matchings. Statistical distributions and even self-organizing maps (Kohonen, 2007) might be taken into account. The proposed approach leaves room for discussion, but first meetings with researchers and professionals alike have shown that there clearly is a need for visual concepts in the field of digital forensics. The authors currently seek feedback from law enforcement agencies to test the applicability of their approach to real digital forensic investigations.

7 REFERENCES

(2008). From Bundeskriminalpolizei: http://www.fedpol.admin.ch/fedpol/de/home/fedpol/organisation/bundeskriminalpolizei.html

Carrier, B. (2008). *Autopsy Forensic Browser*. From Autopsy Forensic Browser: http://www.sleuthkit.org/autopsy/index.php

Carrier, B. (2005). *File System Forensic Analysis.* Amsterdam: Addison-Wesley Longman.

Carrier, B. (n.d.). *The Sleuth Kit*. Retrieved 2007 from The Sleuth Kit: http://www.sleuthkit.org/sleuthkit/

Computer Legal Experts. (2007). From Computer Legal Experts: http://www.computerlegalexperts.com/

debian. (2008). From debian: http://www.debian.org/

EnCase Forensic. (2008). From EnCase Forensic: http://www.guidancesoftware.com/products/ef_index.asp

Farmer, D., & Venema, W. (2005). *Forensic Discovery.* Amsterdam: Addison-Wesley Longman.

Fisher, B. A. (2000). *Techniques of Crime Scene Investigation.* CRC Press.

Flanagan, D. (2005). *Java in a Nutshell - A Desktop Quick Reference.* O'Reilly Media.

Geschonneck, A. (2006). *Computer-Forensik.* Dpunkt Verlag.

Jones, K. J., Bejtlich, R., & Rose, C. W. (2005). *Real Digital Forensics.* Amsterdam: Addison-Wesley Longman.

Miller, G. A., (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63, 81-97. Retrieved 2007 from Psychology Department of University of Toronto.

KNOPPIX. (2008). From KNOPPIX: http://www.knoppix.org/

Kohonen, T. (2007). *Self-Organizing Maps.* Berlin: Springer.

The UNIX system. (2008). From The UNIX system: http://www.unix.org/unix03.html

Vlist, E. v. (2002). *XML Schema.* O'Reilly Media.