Event Reconstruction: A state of the art

ABSTRACT

Event reconstruction is one of the most important step in digital forensic investigations. It allows investigators to have a clear view of the events that have occurred over time. Event reconstruction is a complex task which requires exploration of a large amount of events due to the pervasiveness of new technologies nowadays. Any evidence produced at the end of the investigative process must also meet the requirements of the courts, such as reproducibility, verifiability, validation, etc. After defining the most important concepts of event reconstruction, a survey of the challenges of this field and solutions proposed so far is given in this chapter.

INTRODUCTION

Cybercrime and digital forensics have become increasingly commonplace in today's world. Crimes committed with the aid of or against digital systems are being reported almost daily. Internet fraud, cyber-bullying, cyber-terrorism, systems intrusion perpetrated against both individuals and corporations are costing businesses and governments billions in lost revenue and security updates (Anderson, et al., 2012). Due to all these issues, digital forensics has become an important research area in the last few years. Digital forensics is defined by (Palmer, 2001) as a set of methods based on proven scientific theories which aim to enable the reconstruction of past events related to an incident and the detection of criminal acts. To reach these objectives, each digital investigation is conducted according to a rigorous process (Palmer, 2001) starting with the identification of an incident and ending with the final decision of the court of justice. This process includes steps allowing to preserve the integrity of evidence, seize sources of footprints from the crime scene, examine these sources to find relevant information and finally analyse this information to be able to make assumptions about the incident.

Several tools are available to help investigators during the first steps of this process. For example, EnCase or FTK can help investigative agents during the collection and the examination of digital objects while preserving their integrity. However, these tools are limited regarding the analysis step, which allows to fully understand what happened during the incident. Collecting evidence and studying its properties is an important part of the investigative process. However, to extract acceptable evidence, it is also necessary to infer new knowledge such as the causes of the current state of the evidence (Carrier & Spafford, 2004). For example, a file illegally modified may be identified during the first steps of an investigation. Although the identification of such an object is interesting, only the analysis phase can help investigators to understand the causes of this modification. Among all the techniques used during the analysis phase, event reconstruction enables investigators to have a global overview of the events occurring before, during and after a given incident. The story produced as output of this process can answer many questions such as « What happened?" and "Why did these events took place?".

This chapter aims to present different aspects of the field of event reconstruction and outlines the various approaches proposed so far. The next section of this chapter presents several notions extensively used in this field (e.g. footprint, event, etc.). Challenges encountered during the conception of event reconstruction approaches are reviewed in section 2. The different approaches used to perform event reconstruction are introduced and assessed in section 3. For each of them, a description of the method used and a synthesis of strengths and limitations in relation to the challenges of the field are given. Finally, future directions for research are given in the last section.

DEFINITIONS

In this section, the terminology used in the rest of the chapter is explained. Event reconstruction is "the process of identifying the underlying conditions and reconstructing the sequence of events that led to a security incident" (Jeyaraman & Atallah, 2006). There are several types of event reconstruction depending on the nature of the incident. This chapter focuses on prosecutorial forensic analysis which is used to solve digital crime.

First, the crime scene is a space where a crime or an incident takes place. In (Carrier, Spafford, & others, 2003), a physical crime scene is defined as "a physical environment containing physical evidence related to an incident". The physical environment in which happen the incident is called the primary physical crime scene. Because of network connections for example, the crime scene can be extended to other places (e.g. if one of the protagonists has communicated with a remote party or download a file from a remote server, it can be necessary to seize the remote machines). The crime scene is not necessarily limited to a single building or environment and the subsequent scenes are called secondary physical crime scene. Then, a digital crime scene is defined as a component of a physical crime scene. A digital crime scene is defined as "a virtual environment created by hardware and software and containing digital evidence related to an incident". A physical crime scene may contain several digital crime scene (a computer, a cell phone or other electronic device).

After the incident and the arrival of the officers in charge of the investigation, the crime scene becomes a protected space where the state of resources is preserved. After ensuring the protection of the crime scene, investigators begin the collection phase. The purpose of this latter is to collect objects which carries footprints that can be relevant in respect of the objectives defined at the beginning of the investigation. The objects collected during an investigation carry digital footprints and may themselves contain many digital footprint sources (e.g. a computer may contain digital footprint source such as Firefox logs, Apache logs, etc.). According to (Ribaux, 2013), a footprint (or a trace) is the sign of a past event. A footprint is the only available information to define the past events (e.g. a fingerprint indicates that an object was grasped by a person, information extracted from Firefox logs may indicate that the user has visited a given webpage, etc.). Footprints carry information about the events that have produced them and thus, they can be used by investigators to reconstruct the events which happened during an incident.

In a digital context, a footprint may be a piece of information about web browser activity, a document or a file left in the bin. There is a large number of footprints sources (Forensics Wiki, 2007) (Gudhjonsson, 2010):

- First, web browsing and emails can be used to get information about the user behaviour on the web. Each web browser stores in files or in databases a large number of potentially useful information for investigators. Regarding the web browser Mozilla Firefox for example, it is possible to obtain information about the webpages visited, the content entered into form fields, the bookmarks and downloads performed by a given user. Footprints extracted from web browsers allows to know the user's interests (based on query to search engines, bookmark and visits, etc.), to identify potential accomplices (malicious file downloaded from a remote server, etc. Contents and headers of emails are also a source of relevant information for an investigation.
- Social networks allow people to share information, location and other multimedia contents with private or business contacts. Information left by browsers, temporary files or data stored in the memory of mobile devices offering social applications may be useful for investigators to obtain information about user contacts as well as his activity (Al

Mutawa, Baggili, & Marrington, 2012) (e.g. sending date of a tweet on Twitter (Morrissey, 2010)).

- Operating systems record a lot of information about events occurring on a machine. In the operating system Windows for example, footprints can be collected from several locations:
    - Windows event log EVT and EVTX record information about various kind of events such as session login, start/stop service or software, error occurred during the execution of a program, installation of a new software, etc.).
    - The registry is a database containing a large amount of data stored as keys. It stores information about system configuration, devices or software configuration.
    - Prefetch and superfetch folder are used to speed up the loading of applications which are used on a regular basis. For each start of software, a file containing information about the software (loaded data, locations used, etc.) is created in the prefetch folder. These files are a potential source of information for the investigator as each of this file allows to see the name of the executable, the name of files used by this latter, the number of uses of the software and the date of the last launch.
    - The recycle bin and the restore points allow to discover deleted files which are potentially interesting for the investigation.
- Logs of software are also a rich source of information. For example, antivirus logs contain information about exploits and malicious software detected on the computer. Server logs such as Apache logs or Microsoft IIS logs can be used to get information about query sent to the server.
- Content of files can be used for multiple purposes during an investigation. In addition, metadata associated to each file allows to know how and when a file was produced and who created it. For example, image metadata contains information about the user (location), the camera used, the date on which the image was taken ,etc. XMP metadata used for PDF files allows to know the title, the author or the creation date of a document. Another example is the MAC times used by file systems. These allows to know when a file has been modified, accessed, created or last modified (in Linux).

When the extraction of footprints is completed, investigators need to convert them into events and build a timeline containing all the events related to the incident. This timeline allows investigators to have a global overview of the case and to know for example what machines was used, what applications were running or what files have been modified at a given time. An event is a single action occurring at a given time and for a certain duration. An event may be the drafting of a document, the reading of a webpage or a chat conversation with somebody. Each event carries temporal information allowing to know when the event occurred. This information takes the form of a time interval to define the beginning and the end of the event and implicitly, its duration. Besides the duration, the use of a time interval rather than an instant allows to represent the notion of uncertainty (Liebig, Cilia, & Buchmann, 1999). When the time at which the action occurs cannot be determined accurately, the use of an approximation by the use of an interval is an adequate solution.

CHALLENGES

Event reconstruction has many issues which are directly related to the size of the data, digital forensics process complexity, and IT infrastructures challenges. While some of these challenges have been a focus of many researchers and developers for the last decade, the size of data volumes (Richard III

& Roussev, 2006) and data heterogeneity are still very challenging. The first (large data sizes) introduced many challenges at every phase of the digital forensic process; from the data collection to the interpretation of the results. The evolution of new technologies (high increase of storage capacity, ubiquitous devices, etc.) leads to the necessity to handle very large volumes of data during an investigation. Thus, investigators are often confronted with the problem of cognitive overload during the interpretation of data. The second (data heterogeneity) is usually due to multiple footprint sources such as log files, information contained in file systems, etc. We can classify events heterogeneity into three categories:

- Format: The information encoding is not the same among sources due to the formatting or other issues. So, depending on the source, the footprint data may be different.
- Temporal: The use of sources from different machines may have timing problems (e.g., unsynchronised clocks, different time zones, etc.).
- Semantic: The same event can be interpreted or represented in different ways. For example, an event may appear in different forms in different sources.

In order to gather all the events found in footprint sources in a single timeline, a good handling of all these forms of heterogeneity is required. This leads to the development of an automated information processing approach that is able to extract knowledge from these heterogeneous sources. In addition, once extracted, this knowledge should be federated within the same model so as to facilitate their interpretation and future analysis.

In addition, all approaches have to satisfy some key requirements such as credibility, integrity, and reproducibility of the digital evidence (Baryamureeba & Tushabe, 2004). In recent years, the protagonists of digital forensics moved away from investigative techniques that are based on the investigators experience and intuition, to techniques based on proven theories. It is also necessary to provide clear explanation about the reasoning used to reach each conclusion of the investigation. These explanations allow to give support to the conclusions and enable the justice to fully understand and reproduce the reasoning process. In addition, one has to ensure that the tools used do not modify the data collected on the crime scene. Thus, it is necessary to develop tools that extract excellent quality of the evidence, while preserving the integrity of data.

EVALUATION OF EXISTING APPROACHES

In this section, the most significant approaches to carry out event reconstruction are presented and discussed. For each of them, an overview of the architecture used and the functionalities proposed is given. We then study limits and strengths of each approach by focusing on a number of criterion.

Classification of approaches

Event reconstruction approaches can be classified depending on the sources used and the time at which the tool is used:

- Event reconstruction tool can be based on a unique source (e.g. timestamp from file system) or based on multiple sources (e.g. logs files, file system, operating system information) (inglot2012framework). In the first approach, the timeline does not fully represent what happened on the machine and therefore the investigators may miss important information. In the second approach (also called super-timeline approach), the timeline is more accurate than in the first approach but the produced timeline is large and therefore difficult to analyse.
- Tools can used ex post evidence or ex ante logging (Jeyaraman & Atallah, 2006). In the first case, the tool starts working after the incident happened and tries to identify and retrieve evidence to

construct the timeline. In the second case, the tool starts working before the incident by recording all events occurring on the machine. When an incident occurs, the recorded information can be used to understand what happened.

In this study, we focus only on approaches which can cope with a large number of situations. Thus, we review only polyvalent approaches that are able to work without prior knowledge of the systems studied during the investigation (ex post evidence approach). In addition, we restrict this study to approaches able to fully complete the reconstruction of events (e.g. tools providing timeline visualization functionalities only are not taken into account).

Criteria

Several criteria are assessed in this state of the art to evaluate the capacity of approaches to meet the needs of investigators and give solutions to challenges described above. First, the ability of approaches to solve problems related to information processing is assessed using the following criteria:

- The approach provides automated tools to extract the events from footprint sources and build the timeline associated.
- The approach is able to process multiple and various footprint sources and to federate information collected in a model in a coherent and structured way.
- Tools are proposed to assist investigators in the tasks of timeline analysis.

To study the capacity of the approaches to fulfil the justice requirements, we enrich our criteria with the following elements:

- A theoretical model is used to support the proposed approach and the ability to explain the reasoning performed.
- The approach is able to cope with problems related to the preservation of the integrity of data used during the investigation.

ECF: Event Correlation for Forensics purposes

(Chen, Clark, De Vel, & Mohay, 2003) argue that it is possible to correlate the information contained in computers (log files, etc.) despite the heterogeneous nature of data. The ECF architecture proposed in this work is made of a storage element containing events extracted during an investigation in addition to tools able to settle and query this database. ECF is composed of events parsers, a database and a user interface. An overview of the ECF architecture is given in *Figure 1*.

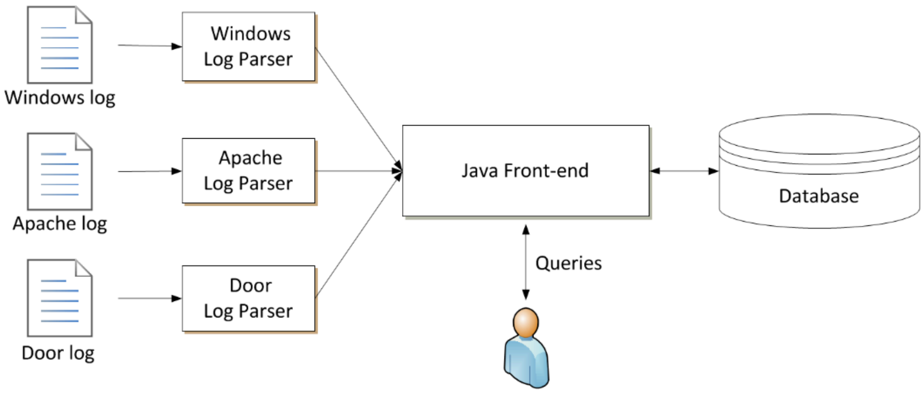

Figure 1 ECF architecture

In this proposal, a canonical representation of events is used to standardize the representation of events extracted from heterogeneous sources. The events are stored in a table that has eleven attributes including an identifier for the event, the date and time at which it occurs, information about the actor who caused the event (e.g. IP address), information about the object affected by the event (URL if the object is a webpage for example), the action represented by the event, the result of the event (success, failure, unknown) and information about the source used to identify the event. A second table is used to store specific information about events depending on the source from which they are extracted.

The system described offers five main functionalities:

- Event extraction: this function allows to parse event sources, format events and populate the database. ECF proposes parsers to handle sources such as Apache logs, Windows 2000 logs or door logs.
- Dynamic queries: this interface allows the investigator to query the database. Queries are built by assembling constraints on one or more fields of the event table using Boolean operators. For example, the investigator may look for events occurring between two dates or search for all events caused by a given person.
- Custom queries: this interface allows to execute directly SQL queries. Therefore, this interface provides more flexibility to the user.
- Hypotheses testing: this tool allows the user to create new events and test the validity of these assumptions.

  The main contribution of this work is the introduction of an architecture able to gather events from heterogeneous sources into a single structure. The proposed system uses a set of automatic parsers and a canonical form to represent the events extracted during an investigation. This idea has been widely adopted in subsequent approaches. However, the approach does not propose any functionality to assist the investigator during the analysis of the events. Thus, a large part of the investigation have to be carry out by investigators.

Auto-ECF

(Abbott, Bell, Clark, De Vel, & Mohay, 2006) proposed Auto-ECF which is an evolution of the previous approach. Auto-ECF was designed to address several shortcomings of ECF. In this work, a new canonical form consisting of four required attributes is proposed to represent events:

- A unique identifier.
- The date and time at which the event occurred.
- The type of event (e.g. session login, file creation, etc.).
- The result of the event (success, failure or unknown).

Each event can also carry a number of additional attributes to allow the storage of more specific information. The main purpose of this approach is to provide automatic mechanisms to convert events extracted from heterogeneous sources to high-level events which are easier to understand for an investigator. To reach this objective, several concepts are introduced by the authors:

- Raw event: event contained in the event sources such as log files.
- Logical event: event stored in canonical form in the database.
- Simple event: logical event resulting from the conversion of a raw event.
- Composite event: logical event resulting from the aggregation of several logical events.

To convert the raw events in logical events and to construct composite events, Event Logical Patterns (LEP) are used. After extracting the events from sources, a dedicated algorithm is used to search for occurrences of the patterns (stored in a XML file) and to create new events associated to each pattern. This process is illustrated in *Figure 2*.

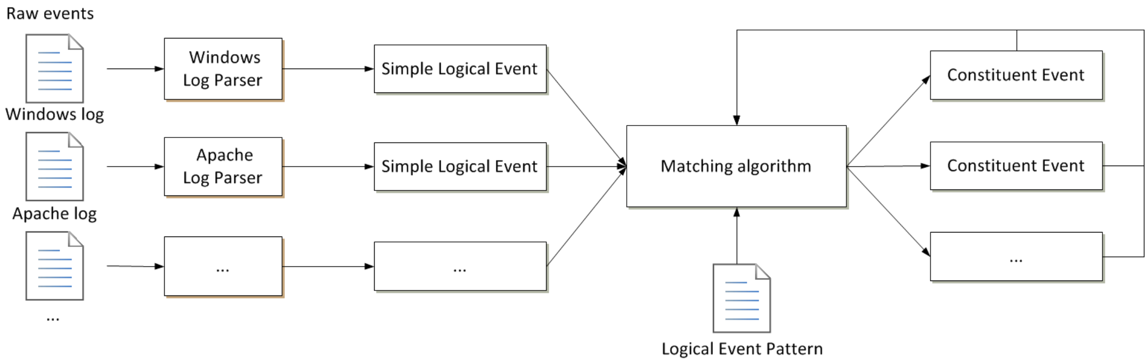

Figure 2 Auto-ECF process

This approach allows to convert raw events in events that are more readable for humans. Even if this functionality is useful for investigators, it represents only a small part of the analysis and, thus, investigators have to carry out the rest of the analysis. For example, this approach does not allow to find relationships between events.

FORE: Forensics of Rich Events

The FORE approach proposed by (Schatz, Mohay, & Clark, 2004) allows to carry out an investigation using heterogeneous sources of events. The aim of this work is to propose a solution to deal with the large amount of data to be processed during an investigation and the difficulties encountered by investigators to interpret these data. To serve this purpose, an ontology-centric architecture is introduce (*Figure 3*). The ontology is used to store events and is rooted by two classes which are the Entity class (representing objects of the world) and the Event class (representing the state changes of an object over time). It should be noted that all the expressiveness of the ontological language used (OWL) is not used to described the ontology. Indeed, to model the knowledge on events, Schatz only uses a hierarchy of classes in addition to properties and individuals (no constraints on classes and properties). The proposed ontology also includes temporal information on events in addition to causal relationships between events.

The FORE architecture is composed of three parts which are the extraction module, the ontology and the analysis tools. In the extraction module, a set of parsers is used to extract knowledge from various sources such as Apache server logs, Windows 2000 logs, etc. The knowledge extracted is then used to populate the ontology with new instances of events. Each parser is dedicated to a specific type of source allowing to take into account the specificity of each source. Then, automated tools are proposed to process this knowledge. A correlation tool based on rules is used to identify causal relationships between events ("if event A is the cause of event B then event A has to occur to allow event B to occur"). To express rules, a rule language called FR3 has been created. A rule expressed with the language FR3 is composed of antecedents and consequences. An inference engine is used to browse the knowledge base to find elements appearing in the antecedents of a rule. If all elements composing antecedents of a rule occur in the ontology, the rule is satisfied and elements appearing in the consequences of the rule are then added to the ontology. Finally, an interface is also provided to allow the user to visualize the knowledge contained in the ontology.

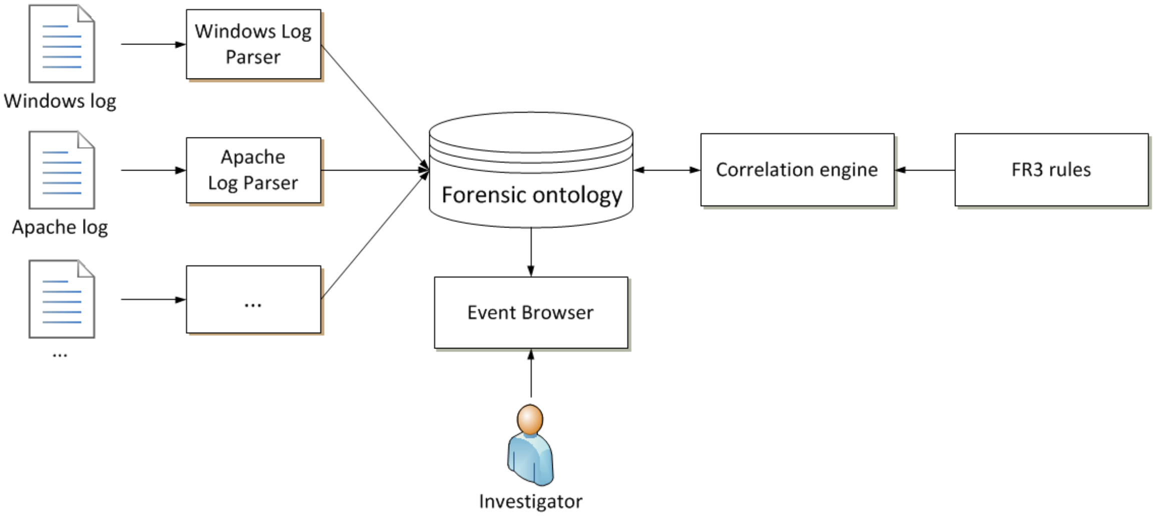

Figure 3 FORE Architecture

The use of an ontology is an efficient way to deal with heterogeneity of event sources. Regarding the analysis of the timeline, the proposed tool help investigators by highlighting non-explicit relationships between events (causality). However, the use of a rule-based approach makes the use of this tool tedious and time-consuming (need to create and manage a large set of rules).

Finite state machine

In this work, (Gladyshev & Patel, 2004) argue that a formalization of the event reconstruction problem is needed to better structure the reconstruction process, facilitate its automation and ensure the completeness of the reconstruction. To address these problems, an approach based on finite state machine is proposed. In this latter, the behaviour of the system under investigation is represented using a state machine. Subsequently, some scenarios are removed using evidence collected by the investigator. Once the number of potential scenarios has been reduced, a backtracking algorithm is used from the final state (the state observed at the beginning of the investigation) to the initial state of the system. In this approach, the event reconstruction can be seen as a process finding the sequence of transitions that satisfies the constraints imposed by evidence.

One of the main strengths of this approach is that it allows to conduct forensic investigations with the support of a theory widely recognized in the scientific community to explain the conclusions of the investigation. However, the finite state machine approach has several limitations. In particular, this approach cannot be used to conduct complex investigations. Indeed, the use of finite state machine to represent systems often results in combinatorial explosions. Thus, this approach seems inadequate for real forensic cases. For example, the investigation of a single computer may involve several processes such as web browsers, file system, instant messaging software, etc. Thus, the representation of such a system with a finite state machine seems not possible.

In (James, Gladyshev, Abdullah, & Zhu, 2010), improvements of the previous approach are proposed. In this paper, authors highlight the main problem of the original approach: the exponential growth of the size of the state machine and therefore, the number of possible scenarios to examine during the backtracking phase. The solution carried by this paper is to convert the finite state machine into a deterministic finite state machine. However, even if this solution allows to reduce the size of the state machine, experiments show that the approach still not usable on real forensic cases.

Zeitline

(Buchholz & Falk, 2005) introduced a timeline editor named Zeitline allowing the investigator to create scenarios from multiple sources of events. The proposed tool also provides functions to group and hierarchically organize events. To store events, the authors argue that a data structure able to withstanding the scalability is necessary to handle the knowledge used during an investigation. In addition, this data structure must allow to quickly sort and query knowledge. The authors chose to use a variant of balanced binary search tree. This approach distinguishes two types of events: atomic events which are extracted from event sources and complex events containing several atomic or complex events. Each of these types is implemented using a Java class inheriting from the class TimeEvent representing events. Whether they are complex or atomic, each event has a number of attributes including the date and time at which the event occurred, the name of the event, a description and a pointer to the "parent" event. In addition to these attributes, instances of AtomicEvent and ComplexEvent classes carry specific attributes such as the source of the event (for atomic event) and pointers to « children » for complex events.

In addition to the possibility to extract events from various sources, users can create their own extractors allowing them to easily extend the number of event sources supported by Zeitline. The tool also offers to investigators an interface allowing him to add new events to the timeline, aggregate several events to build a complex event or search for specific events using a query tool based on keywords. Finally, Zeitline is restricted by a number of rules that aim to prevent the alteration of evidence. These restrictions allow to take into account a part of the requirements of justice. To prevent the modification of evidence, a system of views is also used to avoid the removal of information contained in evidence. When the investigator deletes an event from the timeline, the event is removed from a view but still physically preserved. This special attention given to the preservation of the integrity of the information is one of the main contribution of this tool.

A framework for post-event timeline reconstruction using neural networks

On the basis that most of the existing methods cannot efficiently handle large volumes of data, (Khan, Chatwin, & Young, 2007) introduced a new approach using a neural network to show the ability of machine learning techniques to quickly process large amounts of data. The use of machine learning techniques also allows to explicit the reasoning made to produce a conclusion which is one of the justice requirements. However, the authors indicate that this feature is not applicable to neural networks. Indeed, during the learning phase, some parameters used remain unknown.

The proposed approach uses traces left by user activities in the system to detect the activity of software. The proposed tool is composed of three parts:

- The parsers allowing to extract traces found in various types of sources (log files, registry, etc.).
- The preprocessor used to convert data extracted by parsers to make it usable by the neural network.
- The neural network used to identify launched applications using input data.

As admitted by the author, the performance of the proposed tool is low. In addition, the training of the neural network and the need to use the neural network several times to get a complete scenario make this tool very time-consuming.

FACE: Forensics Automated Correlation Engine

In (Case, Cristina, Marziale, Richard, & Roussev, 2008), the authors points out that the consultation of data produced during an investigation is a tedious work. As the current forensic tools are limited to the extraction and presentation of information extracted from sources, there is an important need to develop tools able to assist investigators during the interpretation and the analysis of the data.

In this work, an approach called FACE is introduced to collect and analyse data (event correlation) from various sources. FACE is able to handle five different data sources which are memory dumps, network activities, disk images, log files and user configuration files. Once data is extracted, the correlation tool allows to discover logical relationships between events and between objects and events (e.g. a file). The output of the proposed tool is a report describing the activities of the user. This report is composed of activities linked by hyperlinks to facilitate the consultation of the timeline.

One of the main contribution of this approach is the introduction of a tool allowing to correlate events and thus, carry out a part of the analysis. The second contribution lies in the presentation of data. The proposed tool offers different views on events and objects in addition to hyperlinks to make the reading of the timeline easier and more intuitive for investigators.

CFTL: Cyber-Forensic TimeLab

In this proposal, (Olsson & Boldt, 2009) discuss the need for a system to view and navigate the data related to an investigation in an intuitive way to discover evidence. To reach this objective, the tool Cyber-Forensic TimeLab described in this work extracts timestamps found in a machine or a group of machines, builds the timeline and then provides a graphical view of all the events. The investigator can then browse the events and identify relevant information more easily.

The proposed tool is composed of two parts: a scanner and an event viewer. The scanner is used to extract timestamps from sources (file system, Windows or Unix logs, JPEG files) and store them in a XML file. Each evidence has three required attributes (name, type and an identifier) and several optional attributes. Once timestamps are extracted, the event viewer reads the XML file, orders events and then display them in a graphical timeline. The main added value of this approach is the improvement of the ergonomics of the interface between the timeline and the investigator.

Log2Timeline

(Gudhjonsson, 2010) proposes a system allowing to construct automatically a super-timeline using a large number of sources. The author highlights several limitations of current event reconstruction approaches such as the limited number of sources used. This makes the truthfulness of the timeline vulnerable to anti-forensics techniques (e.g. alteration of timestamp). In addition, the quality of the timeline also suffers from the small number of sources. For example, some contextual events may not appear in the timeline. The proposed solution is to increase the number of event sources to enhance the quality of the timeline and to minimize the impact of anti-forensics techniques. The architecture presented in this work is composed of a module used to extract events and a module able to display the timeline produced or serialize it in a CSV file. Each extracted event carries several attributes including a timestamp, a description of the event, an identifier and the type of the source used to determine the event. Log2timeline used a large number of sources: web browsers histories, log files of antivirus software, operating system logs, information extracted from the bin, etc.

One of the main limitation of this approach is the lack of functionalities to help the investigator during the analysis of the super-timeline. The use of a large number of sources lead to the creation of huge timeline which are very difficult to interpret by the investigator.

Automated timeline reconstruction approach

In (Hargreaves & Patterson, 2012), a system able to automatically reconstruct high-level events using large amount of low-level events extracted by log2timeline or Zeitline is proposed. The authors highlight that the amount of data and the number of events make the visualisation and the analysis of a timeline difficult, especially with the super-timeline approach. The aim of this work is to facilitate the reading of the timeline by introducing a mechanism allowing to create high-level events (which are easier to understand for investigator) from low-level events (events extracted from sources). This process can be compared to the production of a summary of the timeline. The author also tries to meet the needs of justice by storing traceability information during the process of summarization. For each high-level event, the investigator has therefore the possibility to know the low-level events used to create it.

The proposed solution implements a two-step process: the extraction of low-level events and the construction of high-level events. A system composed of parsers and bridges is used to carry out the low-level event extraction. Parsers are used to process the content of sources. Two types of sources are used: the file system and the information contained in the files themselves. Then, bridges convert the extracted data into the format used for low-level events. To represent events in memory, a standard format consisting of nine attributes is used (an identifier, a date of start and a date of end, the source used to identify the event, the information used to construct the event, the parser used for the extraction, the event type, etc.).

One of the specificities of the proposed format compared to the format used in tools like Zeitline or log2timeline is the use of interval to define dates. Indeed, the authors make the assumption that the dates may be inaccurate and thus, an interval is more suitable to represent them. Once the timeline containing low-level events is built, a process is used to produce high-level events. The timeline is browsed to search for specific patterns of one or several low-level events. When a pattern is found, the corresponding high-level event is added to the timeline. Each high-level event consists of fourteen attributes including information about the reasoning used to identify it (pattern used) and a description of the event. The summarization of the timeline is a useful functionality for investigators as it allows them to save time during the reading and the interpretation of the timeline. However, this functionality represents only a part of the analysis process.

FUTURE RESEARCH DIRECTIONS

A large majority of the proposed approaches provide solutions to extract events which are spread across different types of sources and to build the associated timeline. However, the extraction of events from a large number of sources (approach super-timeline) lead to the creation of huge timeline which are very difficult to read and interpret for humans. Few solutions are provided to assist the investigator during this phase of the investigation. The solution provided in (Gladyshev & Patel, 2004) is able to identify relevant scenarios for a given incident but due to performance reasons and lack of automation, the approach is unusable for real cases. Thus, there is a strong need to develop an approach providing a complete set of advanced techniques of timeline analysis:

- Processes to reduce the amount of data that investigators have to read by filtering data or summarize the timeline as proposed in (Abbott, Bell, Clark, De Vel, & Mohay, 2006) and (Hargreaves & Patterson, 2012).
- Operators to deduce new knowledge about events using the knowledge extracted from sources. In (Schatz, Mohay, & Clark, 2004) and (Case, Cristina, Marziale, Richard, & Roussev, 2008), correlation tools are proposed to identify implicit relationships between events.
- Tools able to highlight the most relevant information of a timeline to solve the case.

Regarding legal aspects, only few approaches are supported by theories. The use of the finite state machine theory allows (Gladyshev & Patel, 2004) and (James, Gladyshev, Abdullah, & Zhu, 2010) to provide an approach based on a proven theory. Regarding the preservation of the integrity of the information, the approach describes in (Buchholz & Falk, 2005) uses a set of restrictions to prevent the modification of evidence.

CONCLUSION

In this chapter, we introduced the problem of event reconstruction which is a crucial step of a digital investigation. This phase allows investigators to understand what happened during an incident using footprints left on a crime scene. Several approaches have been proposed to carry out the event reconstruction. However, none of them is able to assist investigators during the whole investigative process (from the extraction of events to the analysis of the timeline) while meeting the constraints imposed by Justice.

REFERENCES

Abbott, J., Bell, J., Clark, A., De Vel, O., & Mohay, G. (2006). Automated recognition of event scenarios for digital forensics. *Proceedings of the 2006 ACM symposium on Applied computing*, (pp. 293--300).

Al Mutawa, N., Baggili, I., & Marrington, A. (2012). Forensic analysis of social networking applications on mobile devices. *Digital Investigation, 9*, S24--S33.

Allen, J. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM, 26*(11), 832--843.

Anderson, R., Barton, C., Böhme, R., Clayton, R., van Eeten, M., Levi, M., . . . Savage, S. (2012). Measuring the cost of cybercrime. *11th Workshop on the Economics of Information Security (June 2012).*

Baryamureeba, V., & Tushabe, F. (2004). The enhanced digital investigation process model. *Proceedings of the Fourth Digital Forensic Research Workshop.*

Buchholz, F., & Falk, C. (2005). Design and implementation of Zeitline: a forensic timeline editor. *Digital forensic research workshop.*

Carrier, B., & Spafford, E. (2004). Defining event reconstruction of digital crime scenes. *Journal of Forensic Sciences, 49*(6), 1291.

Carrier, B., Spafford, E., & others, . (2003). Getting physical with the digital investigation process. *International Journal of Digital Evidence, 2*(2), 1--20.

Case, A., Cristina, A., Marziale, L., Richard, G., & Roussev, V. (2008). FACE: Automated digital evidence discovery and correlation. *digital investigation, 5*, S65--S75.

Chandrawanshi, R., & Gupta, H. (2013). Implementation Of An Automated Server Timeline Analysis Tool For Web Forensics. *International Journal of Engineering, 2*(4).

Chen, K., Clark, A., De Vel, O., & Mohay, G. (2003, November). ECF-event correlation for forensics. *First Australian Computer Network and Information Forensics Conference* (pp. 1--10). Perth, Australia: Edith Cowan University.

Cloppert, M. (2008). Ex-tip: an extensible timeline analysis framework in perl. *Bethesda, MD: SANS Institute*.

Forensics Wiki. (2007, Juin). *File formats*. Récupéré sur Forensics Wiki: http://www.forensicswiki.org/wiki/Category:File_Formats

Gladyshev, P., & Patel, A. (2004). Finite state machine approach to digital event reconstruction. *Digital Investigation, 1*(2), 130--149.

Gudhjonsson, K. (2010). Mastering the super timeline with log2timeline. *SANS Reading Room*.

Hargreaves, C., & Patterson, J. (2012). An automated timeline reconstruction approach for digital forensic investigations. *Digital Investigation, 9*, 69--79.

Inglot, B., Liu, L., & Antonopoulos, N. (2012). A Framework for Enhanced Timeline Analysis in Digital Forensics. *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on*, (pp. 253--256).

Jeyaraman, S., & Atallah, M. (2006). An empirical study of automatic event reconstruction systems. *digital investigation, 3*, 108--115.

Khan, M., Chatwin, C., & Young, R. (2007). A framework for post-event timeline reconstruction using neural networks. *digital investigation, 4*(3), 146--157.

Liebig, C., Cilia, M., & Buchmann, A. (1999). Event composition in time-dependent distributed systems. *Proceedings of the Fourth IECIS International Conference on Cooperative Information Systems* (pp. 70--78). Washington, DC, USA: IEEE Computer Society.

Morrissey, S. (2010). *iOS Forensic Analysis: for iPhone, iPad, and iPod touch .* Apress.

Olsson, J., & Boldt, M. (2009, September). Computer forensic timeline visualization tool. *Digital Investigation, 6*, 78--87.

Palmer, G. (2001). A road map for digital forensic research. *First Digital Forensic Research Workshop, Utica, New York*, (pp. 27--30).

Ribaux, O. (2013). Science forensique.

Richard III, G., & Roussev, V. (2006). Digital forensics tools: the next generation. *Digital Crime and Forensic Science in Cyberspace. Idea Group Publishing*, 75--90.

Schatz, B., Mohay, G., & Clark, A. (2004). Rich Event Representation for Computer Forensics'. *Proceedings of the Fifth Asia-Pacific Industrial Engineering and Management Systems Conference (APIEMS 2004), 2*(12), 1--16.

ADDITIONAL READING SECTION

Arasteh, A., Debbabi, M., Sakha, A., & Saleh, M. (2007). Analyzing multiple logs for forensic evidence. *digital investigation, 4*, 82--91.

Casey, E. (2002). Error, uncertainty, and loss in digital evidence. *International Journal of Digital Evidence, 1*(2), 1--45.

Cohen, M., Garfinkel, S., & Schatz, B. (2009). Extending the advanced forensic format to accommodate multiple data sources, logical evidence, arbitrary information and forensic workflow. *digital investigation, 6*, S57--S68.

Eiland, E. (2006). Time Line Analysis in Digital Forensics. *New México: sn.*

Garfinkel, S. (2010). Digital forensics research: The next 10 years. *Digital Investigation, 7*, S64--S73.

Gladyshev, P., & PATEL, A. (2005). Finite state machine analysis of a blackmail investigation. *International Journal of Digital Evidence, 4*(1), 1--13.

Herrerias, J., & Gomez, R. (2007). A log correlation model to support the evidence search process in a forensic investigation. *Systematic Approaches to Digital Forensic Engineering, 2007. SADFE 2007. Second International Workshop on*, (pp. 31--42).

Jeyaraman, S. (2011). Practical automatic determination of causal relationships in software execution traces.

Marrington, A., Mohay, G., Clark, A., & Morarji, H. (2007). Event-based computer profiling for the forensic reconstruction of computer activity.

Schatz, B., Mohay, G., & Clark, A. (2004). Generalising event forensics across multiple domains. *2nd Australian Computer Networks Information and Forensics Conference* (pp.

136--144). Perth, Australia: School of Computer Networks Information and Forensics Conference, Edith Cowan University.

Schatz, B., Mohay, G., & Clark, A. (2006). A correlation method for establishing provenance of timestamps in digital evidence. *digital investigation, 3*, 98--107.

Sebastian, M., & Chandran, P. (2011). Towards designing a tool for event reconstruction using Gladyshev Approach. *Proceedings of the 2011 ACM Symposium on Applied Computing*, (pp. 193--194).

Undercoffer, J., Joshi, A., & Pinkston, J. (2003). Modeling computer attacks: An ontology for intrusion detection. *RAID*, (pp. 113--135).

Undercoffer, J., Pinkston, J., Joshi, A., & Finin, T. (2004). A target-centric ontology for intrusion detection. *18th International Joint Conference on Artificial Intelligence*, (pp. 9--15).

KEY TERMS & DEFINITIONS

Digital forensics: Use of computer science to help investigators to solve cybercriminal cases.

Event reconstruction: Process allowing to describe exhaustively an incident using information left on a crime scene.

Evidence: Entity used to affirm or refute an assertion.

Legal requirements: To be admissible in a court, each evidence must meet several legal requirements such as reproducibility of the process used, credibility and integrity of data.

Timeline: Structure containing events chronologically ordered. A timeline allows investigators to have a global overview of the case and to know for example what machines was used, what applications were running or what files have been modified at a given time.

Crime scene: The crime scene is a space where a crime or an incident takes place.

Footprint: Trace of a past activity. In a digital context, a footprint may be a piece of information about web browser activity, a document or a file left in the bin

Event: An event is a single action occurring at a given time and for a certain duration. An event may be the drafting of a document, the reading of a webpage or a chat conversation with somebody.