Automatic Timeline Construction For Computer Forensics Purposes

Yoan Chabot*, Aurélie Bertaux*, Christophe Nicolle* and Tahar Kechadi†
*CheckSem Team, Laboratoire Le2i, UMR CNRS 6306
Faculté des sciences Mirande, Université de Bourgogne, BP47870,
21078 Dijon, France
Email: yoan.chabot@hotmail.fr
†School of Computer Science & Informatics
University College Dublin,
Belfield, Dublin 4, Ireland

***Abstract*—To determine the circumstances of an incident, investigators need to reconstruct events that occurred in the past. The large amount of data spread across the crime scene makes this task very tedious and complex. In particular, the analysis of the reconstructed timeline, due to the huge quantity of events that occurred on a digital system, is almost impossible and leads to cognitive overload. It becomes more and more necessary to develop automatic tools to help or even replace investigators in some parts of the investigation. This paper introduces a multi-layered architecture design to assist the investigative team in the extraction of information left in the crime scene, the construction of the timeline representing the incident and the interpretation of this latter.**

I. Introduction

Due to the advent of digital technologies, the field of computer forensics faces new challenges. The increasing number of digital devices own by each person in addition to the significant augmentation of their storage capacity results in the production of a large quantity of data that can potentially be used by investigators. Tools (EnCase, FTK, etc.) are available to help investigators to process this large amount of data. However, the scope of these tools is limited to the collection and a summary review of data of the crime scene. Therefore, it is necessary to fill the gap between the extraction of data found and the analysis of the timeline deduced from them.

In this paper, we present a multi-layered architecture (See Figure 1) to carry out automatically the reconstruction of events, from the extraction of data to the analysis of the timeline produced through the construction and the storage of the latter. The particularity of this architecture is the use of a knowledge representation model which allows to store rich semantic information about events such as the resources used by them or the participants involved in them. This knowledge is then used to provide to the investigators advanced analysis and visualization tools.

This paper is organised as follows. Section II reviews important issues of event reconstruction and the various approaches proposed so far to answer them. In Section III, we introduce the semantic-based approach SADFC (Semantic Analysis of Digital Forensic Cases) made to provide enhanced digital forensic timeline analysis capabilities. The architecture implementing this approach is proposed in Section IV.

II. Event Reconstruction Approaches

During a digital forensics investigation, and more particularly during the reconstruction of past events, investigators face many problems related to the new uses of technologies and constraints induced by the need to rigour in the field of computer forensics. First, from a technical point of view, the proliferation of digital devices and their more intensive use have resulted in the presence of large amounts of heterogeneous data in crime scene. Indeed, computers and others digital devices can themselves contain many different sources of information such as logs of software, web browsers histories, operating system registries, etc. To deal with these issues, approaches have to meet three main requirements which are:

- The use of automated techniques to mine data left on the crime scene and construct the timeline describing the events which occurred during the incident.
- The ability to handle heterogeneous sources of data. This prerequisite ensures that the approach is able to handle all the data of the crime scene.
- The availability of tools to assist investigators in the analysis and interpretation of the timeline. This requirement is motivated by the near impossibility of analysing all the data manually.

In a large part of existing approaches, solutions are proposed to meet the first and the second requirements. For this purpose, automatic extractors dedicated to each source of events are used to populate a central storage system (database [3], ontology [4], etc.). In the proposition of the ECF architecture [3], the use of a set of extractors to collect events and store them in a database forming the timeline is introduced. For its part, the FORE approach [4] uses an ontology to store a semantically richer representation of events. The use of an ontology provides several advantages that will be described in Section III. Regarding the analysis of timeline, existing approaches offer features to correlate events [4] or assist the investigators during the interpretation of the timeline by producing high-level events from low-level events extracted from raw data [5]. The FORE approach

introduces a system to identify correlations between events by connecting them with links of cause and effect. In [6], the authors carry out the event reconstruction by searching sequences of events satisfying the constraints imposed by the evidence in a finite state machine representing the behaviour of the system subject of the investigation. In [5], a system based on patterns is used to produce high-level events from a timeline containing low-level events. However, none of the approaches discussed offers a complete solution to assist investigators in the interpretation and analysis of chronologies.

Event reconstruction approaches must also meet legal requirements such as the credibility of the results produced, the integrity of data used and the reproducibility of the process of investigation [7]. In addition, [6] argue that a formalization of the problem of event reconstruction is necessary to better structure the reconstruction process, facilitate its automation and ensure the completeness of the reconstruction. The SADFC approach answers all these points by providing several mechanisms presented in [8].
In this paper, we focus on proposing new tools to assist the investigators during the analysis of the timeline. To achieve this objective, we propose a semantic-based approach using a rich knowledge representation of events. The use of knowledge about events in the analysis phase requires to intervene early in the process. The extraction tools, in particular, should be designed to enable the identification of knowledge and its extraction. This motivated us in creating a new approach covering the whole digital investigation process, from footprints extraction to timeline analysis.

III. Semantic-Based Approach for Event Reconstruction

With the SADFC approach, we introduce a new system able to automatically construct the timeline describing a computer forensic case, composed of events taking into account numerous semantic information. The main contribution of this approach is the introduction of new semantic dimensions to represent events. The use of a rich semantic representation of events (implemented in an ontology) provides two main advantages. First, the availability of rich semantics allows to represent events in a comprehensive and understandable way for both machines and humans. A second advantage is the possibility to use automatic processes to reason on knowledge thanks to the formal and explicit semantic. The proposed knowledge model contains entities representing a crime scene and events occurring during an incident in addition to operators allowing to acquire and manipulate this knowledge. The knowledge model and operators are formalized in [8] and are presented briefly below. Our event reconstruction process starts with the extraction of the footprints from the crime scene using extraction operators. According to [9], a *footprint* is the sign of a past activity and a piece of information allowing to reconstruct past events. A footprint may be a log entry or a web history for example as a log entry gives information about software activities and web histories provide information about user's behaviour on the Web. Footprint may be used to identify a user, get information about his past actions, the time at which each occurred, etc.

Extraction Operators aim to identify and extract relevant information contained in digital footprints from heterogeneous sources. *Mapping Operators* are then used to analyse data extracted from footprints to deduce and reconstruct events which occurred during the incident. Then, these operators store knowledge about events into the ontology.

Then it is possible to deduce new knowledge using *Inference Operators* on knowledge extracted from footprints. For example, few events carry information about the user who have launched them. Among these few events, login and logout events used by the user to connect to an OS session carry information such as a session identifier. Using temporal information about login and logout events, inference operators can deduce that events occurring between a login event and a logout event for a given session are initiated by the user identified by this session.

Finally, *Analysis Operators* are used to assist the investigators during the interpretation of the final timeline.

IV. Architecture

Based on theoretical elements presented in previous section, we introduce an architecture (See Figure 1) capable to automatise the reconstruction of events and allowing to assist the investigators during the analysis and the interpretation of the produced timeline. This architecture is made of four layers centred on an ontology implementing the proposed knowledge model.

A. Extraction Layer

The *Extraction Layer* aims to extract information from footprints contained in heterogeneous sources.

1) Extraction: During an investigation, many sources (which are all potential input data for the extraction layer) may be used to get information about user's activity. To deal with all these sources, the extraction layer is composed of several parsers (part A1 of Figure 1), each one dedicated to a unique source of footprints. The use of multiple and dedicated parsers allows to take into account specificity of each source while allowing to handle heterogeneous sources. It should be noted that some sources are more difficult to treat than others. Indeed, structured sources such as databases or XML files are easy to handle with appropriate parsers. In another hand, sources such as instant messaging histories or social networks require complex algorithms involving natural language processing or pictures require image processing algorithms to access and understand the content of them. A description of relevant information that can be extracted from sources is given below.

First, the activities of a user on a system can be studied using several sources. Operating systems record a lot of information about events occurring on a machine. From a machine running the Windows operating system for example, it is possible to get information about user's and software's activities using events logs (which 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.), system and software configuration

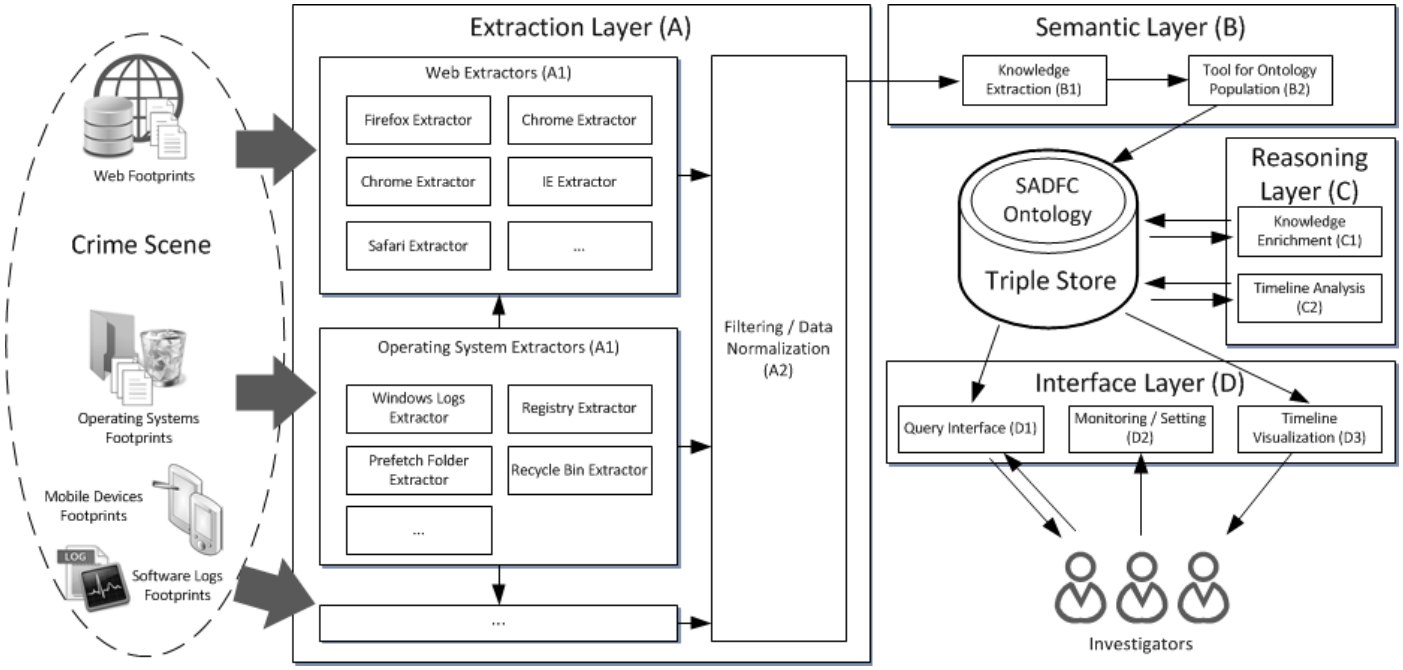

Fig. 1. Architecture

using the registry and software launched recently using the prefetch folder.

In complement to OS footprints, information about user's activities are also available in logs of software which are rich source of footprints. 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. In their part, files found on a crime scene allows to answer multiple purposes during an investigation by using their contents and metadata (which allow to know how and when a file was produced and who created it.

Third, the behaviour of a user on the web can be studied using information contained in files or databases used by browsers to work. First of all, browsers footprints can be used to know user's interests and activities by studying the user's browsing history (the user's visits and at the time of the visits), bookmarks and forms filled by the user (e.g. search field, registration form, etc.). Regarding contents of fields however, the highly dependent semantics of data makes its usage difficult (for example, data entered into the field of a search engine gives information about a user's interests, while a field of a registry form (e.g., to create a website account) may give private information about the user). Information about download activities of users can also be found in browser footprints. This information, for instance, can be used to identify a link between illegal applications and a remote site that provide these applications. The available information are usually (depending on the browser) the name of the downloaded file, the source URI, the URI destination (on the disk), the start date and the end date of the download, the download status (not started, in progress, completed, cancelled, failed...) or a reference to the application used once the download is complete. Browsers footprints also allow to quantify the importance of a webpage for a user. For this purpose, investigators can study bookmarks to determine which websites are important for the user. Browsers also generate footprints of user's browsing preferences (zoom used for navigation, character encoding, etc.). A website for which preferences are assigned can be considered as a significant website for the user. The preferences allow to dissociate the accidental visits ((e.g. the user has clicked on a link by accident) from intentional visits (this information may be valuable to determine if the suspect is responsible or not). The footprints left by login can also be used for this purpose. Indeed, information about all connection pages for which the user has requested to retain his user name and password are registered by browsers. Identifiers can be valuable information if successful decryption techniques are used.
As the operating system is the support of all others sources (browsers and others software are hosted by the operating system), it should be noted that the footprints sources related to the operating system can contained information about others sources of footprints (e.g. registry contained information about browsers, etc.) and should therefore be examined first. Thus, a sequential extraction process composed of two steps is used: footprints from OS are extracted first, and then footprints from all others sources are studied

2) Filtering and Normalization: The second objective of the extraction layer is to process extracted data to enable the semantic layer to handle them (part A2 of Figure 1). First, all data extracted from footprints are not relevant for an investigation, therefore, they need to be filtered in order to reduce the amount of data to be process by the upper layer. In conjunction, the normalization aims to solve heterogeneity problems by translating data produced by the extraction layer in the format used by the upper layer. Indeed, the extraction of footprints information from different sources lead to heterogeneity issues

due, for example, to different formats to store dates and times (granularity, time zone, etc.) or semantic problems (the same event can be interpreted in different ways). Each parser selects relevant data from sources (filtering) and then converts it into the appropriate format (data normalization).

B. Semantic Layer

The data extracted by the previous layer are raw data that need to be understood, interpreted and translated into knowledge. The *Semantic Layer* provides functionality to retrieve, store and manipulate the knowledge contained in data extracted by the extraction layer. The aim of the knowledge extraction module (part B1 of Figure 1) is to identify entities in footprints. An entity is a general concept which contains events (an action which occurred at a given time), objects (a resource used, created, modified or removed by an event) or a subject (a person or a process which is involved in an event). To convert footprints into entities, *mapping operators*, taking the form of rules composed of antecedents and consequences, are used. When elements satisfying antecedents are identified among the footprints, then, semantic entities are created accordingly to consequences. For example, a mapping rule can be created to identify footprints related to the creation of bookmark by web browsers. When such a footprint is identified (antecedents of the rule satisfied), then the rule orders the population tool to create a new object (representing the bookmark) and a new event representing the creation of this object as consequences. All the knowledge extracted must be federated into a unique knowledge model. The knowledge model used in our approach is briefly presented in Section III. This model is implemented in our architecture using an ontology which is itself stored in a triple store (a triple store is a database dedicated for storing knowledge in the form of triplets <subject, predicate, object>). The combination of the ontology and the triple store enables to provide a framework for knowledge representation (mapping rules populate semantic concepts whose meaning is defined in the ontology) and ensuring that the architecture is able to support the processing of large volumes of data. The population tool (part B2 of Figure 1) is used as an interface between the tool and the triple store. It contains all functions required to create instance of classes and properties according to instructions provided by the knowledge extraction module.

C. Reasoning Layer

The *Reasoning Layer* is designed to enhance and analyse the knowledge contained in the triple store. The extraction layer and the semantic layer extract knowledge from footprints. From this knowledge, it is then possible to deduce new knowledge which can not be identified directly in the crime scene (see example in Section III). Thus, the first goal of the reasoning layer is to enrich existing knowledge with new information using inference rules (part C1 of Figure 1). As mapping rules, inference rules are composed of antecedents and consequences. Then, when existing knowledge satisfied antecedents, the new knowledge defined in consequences is added to the ontology.

The second objective of the reasoning layer is to provide timeline analysis tools (part C2 of Figure 1). The aim of this tool is to carry out basic analysis tasks instead of the investigators to allow them to focus on other parts of the investigation where their skills and experience are the most needed. In our works, we introduce an operator allowing to quantify the correlation between two events based on criteria such as temporal proximity, the use of common resources, the processes, people involved in events and rules formulated by domain experts. Event correlation is an interesting tool to highlight chains of correlated events. These chains can then be used by the investigators to know the context of a given event (its causes and consequences for example).
For example, let an event representing the execution of a software and an event representing the download of this software by the same user. These two events are highly correlated because they use the same resource (the executable file) and they are created by the same user. The correlation of the two events allows to highlight information potentially useful for investigators by linking the execution of a program with the source where the program was obtained (download URL).

D. Interface Layer

The *Interface Layer* allows to interact with investigators. This layer is composed of three modules:

- *A Timeline Visualization Tool* (part D3 of Figure 1) graphically displaying a timeline containing all events stored in the triple store in addition to information about events (objects used, people and processes involved in it, correlations with others events etc.).
- *A Query Interface* (part D1 of Figure 1) which can be used by investigators to send SPARQL queries in order to access knowledge of the triple store.
- *A Settings Panel* (part D2 of Figure 1) allowing investigators to manage expert rules used by the correlation tool and adjust thresholds used by this tool.

V. Conclusion and Future Works

In this paper, we introduce an architecture to fill the gap between the extraction of footprints from a crime scene and the interpretation of the timeline by the investigators. To reach this objective, this architecture made of four layers provides functionalities allowing to extract, manage and reason on knowledge about a digital forensic case. This architecture is one of the components composing the semantic-based approach SADFC which allows to help investigators during the reconstruction and the analysis of digital forensics timeline while meeting legal requirements. The main contribution of this approach is the use of semantic tools such as ontology to provide a semantically rich representation of events and enhanced analysis capacities. Future works will concern the integration of new sources of footprints, the enrichment of the ontology with new concepts and the development of new operators for timeline analysis.

Acknowledgement

The above work is a part of a collaborative research project between the CheckSem team of Le2i laboratory (University of Burgundy) and the UCD School of Computer Science and Informatics and is supported by a grant from UCD and the Burgundy region (France).

REFERENCES

[1] G. Palmer, "A road map for digital forensic research," in *First Digital Forensic Research Workshop, Utica, New York*, 2001, pp. 27–30.

[2] B. D. Carrier and E. H. Spafford, "Defining event reconstruction of digital crime scenes." *Journal of Forensic Sciences*, vol. 49, no. 6, p. 1291, 2004.

[3] K. Chen, A. Clark, O. De Vel, and G. Mohay, "Ecf-event correlation for forensics," in *First Australian Computer Network and Information Forensics Conference*. Perth, Australia: Edith Cowan University, November 2003, pp. 1–10.

[4] B. Schatz, G. Mohay, and A. Clark, "Rich event representation for computer forensics'," *Proceedings of the Fifth Asia-Pacific Industrial Engineering and Management Systems Conference (APIEMS 2004)*, vol. 2, no. 12, pp. 1–16, 2004.

[5] C. Hargreaves and J. Patterson, "An automated timeline reconstruction approach for digital forensic investigations," *Digital Investigation*, vol. 9, pp. 69–79, 2012.

[6] P. Gladyshev and A. Patel, "Finite state machine approach to digital event reconstruction," *Digital Investigation*, vol. 1, no. 2, pp. 130–149, 2004.

[7] V. Baryamureeba and F. Tushabe, "The enhanced digital investigation process model," in *Proceedings of the Fourth Digital Forensic Research Workshop*. Citeseer, 2004.

[8] Y. Chabot, A. Bertaux, C. Nicolle, T. Kechadi *et al.*, "A complete formalized knowledge representation model for advanced digital forensics timeline analysis," *DFRWS 2014*, 2014.

[9] O. Ribaux, "Science forensique," 2013, http://www.criminologie.com/article/science-forensique.