Digital Investigation 11 (2014) 90–101

Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

Modelling and refinement of forensic data acquisition specifications

Benjamin Aziz*

School of Computing, University of Portsmouth, Portsmouth PO1 3HE, United Kingdom

ARTICLE INFO

Article history:
Received 7 October 2013
Received in revised form 5 April 2014
Accepted 8 April 2014
Available online 5 May 2014

Keywords:
Computer forensics
Disk data acquisition
Formal specifications
Event-B method
Formal refinement

ABSTRACT

This paper defines a model of a special type of digital forensics tools, known as data acquisition tools, using the formal refinement language Event-B. The complexity and criticality of many types of computer and Cyber crime nowadays combined with improper or incorrect use of digital forensic tools calls for more robust and reliable specifications of the functionality of digital forensics applications. As a minimum, the evidence produced by such tools must meet the minimum admissibility standards the legal system requires, in general implying that it must be generated from reliable and robust tools. Despite the fact that some research and effort has been spent on the validation of digital forensics tools by means of testing, the verification of such tools and the formal specification of their expected behaviour remains largely under-researched. The goal of this work is to provide a formal specification against which implementations of data acquisition procedures can be analysed.

Introduction

Digital forensics tools are becoming increasingly of a critical nature due to the complexity of attacks on digital assets and the sophisticated role that computer and Cyber systems play in modern day crime. As a result, there is continuous need in the law enforcement community to ensure the high quality of generated evidence and acceptable reliability levels for forensic tools used in digital crime investigations, particularly when such investigations are global and/or carry significant importance Friedberg (2012). As a result, it is important to understand properties of digital forensic tools, in particular, where correctness, accuracy and completeness of such tools is vital to the course of justice and the discovering of facts. This view is supported by research in recent years in the area of digital forensics modelling Carrier and Spafford (2004), Ciardhuáin (2004), Beebe and Clark (2005), Ieong (2006), Cohen (2009), and Casey and Rose (2010), where the need for the development of more robust and rigorous scientific methods is highlighted in the area of digital forensics in Garfinkel et al. (2009).

The National Institute of Standards and Technology (NIST) project on the Computer Forensic Tool Testing NIST (http://www.cftt.nist.gov/) aims at raising the assurance of computer forensic tools by providing informal definitions of the various computer forensic tools and the requirements underlying such tools. These requirements are then used for the development of functional specifications, test procedures, criteria, sets and hardware. In this paper, we take this assurance process to another level where the functional specifications and some of the properties of the computer forensic tools are formally defined and verified using a well-established framework based on the Event-B method Abrial (2010). According to Eoghan Casey Casey (2011), such formalisation "encourages a complete, rigorous investigation, en-sures proper evidence handling and reduces the chance of mistakes created by preconceived theories, time pressures and other potential pitfalls."

* Tel.: +44 02392842265; fax: +44 02392842525.
E-mail address: benjamin.aziz@port.ac.uk.

http://dx.doi.org/10.1016/j.diin.2014.04.001
1742-2876/

The Event-B method facilitates the modelling of system specifications based on a combination of set-theoretic and action semantics Mosses (1986); Watt (1987). The top-level abstract model is then refined by adding more detail and by following the rules of refinement Abrial et al. (2005) until the desirable level of refinement is reached. In this paper, the abstract model for a data acquisition tool is first defined and then refined by adding more detail that distinguishes between accessible and inaccessible data in the acquired source, and then by including constructs for preserving the integrity of the acquired data based on hash functions. Throughout this refinement, the focus of the work is on capturing some of the main requirements on data acquisition tools as stated by NIST (NIST, 2004), in particular requirements related to the accuracy and completeness of such tools. The result that the work shows is that though completeness is possible to express generally, accuracy is not. As a result, we conclude that any implementations of NIST's specification of a data acquisition tool must deal with accuracy in a delicate manner, paying attention to the accessibility property of the acquired data.

The rest of the paper is structured as follows. In Section 2 we discuss related work. In Section 3, we give a brief introduction to the Event-B method and language. In Section 4, we give an overview of NIST's main requirements for a data acquisition tool. In Section 5, we define the first abstract model of a data acquisition tool along with its completeness property. In Section 6, this model is refined by distinguishing between accessible, hidden and inaccessible data in the digital source. We show here that accuracy is possible to define. In Section 7, we further refine the specification to include the concept of hash functions and defined based on these the data integrity requirements of the tool. Finally, we conclude the paper in Section 8 and discuss future research directions.

Related work

The application of formal modelling and analysis techniques to digital forensics is by no means a new idea, though it has been under-researched in many aspects. In Gladyshev and Enbacka (2007), the B method Abrial (1996) is used for developing inconsistency checks and verifying the correctness of digital evidence. The B method has also been used to formally specify and refine write blocker systems in Enbacka and Laibinis (2005) and Enbacka (2007) based on NIST's informal definitions of these systems in NIST (2003) and provide formal definitions of the properties of these systems. Our work here follows on the footsteps of Enbacka and Laibinis (2005) by adopting similar approach for a different type of digital forensic tools.

In Leigland and Krings (2004), the authors propose a formal model for analysing and constructing digital forensic procedures. The model is based on set theory and incorporates attacks on systems. In Stephenson (2003), the author uses coloured Petri Nets to model root cause analyses of digital incidents (i.e. digital post mortems). In Gladyshev (2005), finite state machines are used as a defence tool to exploit weaknesses in claimed evidence in computer investigations. The approach is applied to a case of blackmail investigations, where finite state machines are used to demonstrate alternative scenarios to the claimed incident. More recently, James et al. (2009) compute the intersection of the various states in a finite automata to reconstruct events and evidence related to a specific crime incident. Earlier, in Carrier (2006), Carrier defines a model of hypothesis-based digital forensics based on finite state machines. The model captures the concept of *computer history* and consequently, formalises evidence based on this concept.

In Rekhis and Boudriga (2005, 2010), the authors developed a logic-based model, called *S-TLA*$^+$, capable of describing complex investigations and generate evidence under different levels of abstraction. The model is also capable of expressing anti-forensic attacks and provides the machinery to detect such attacks based on the analysis of their action traces. Recently, this model was extended in Rekhis and Boudriga (2012) to include a theory of hierarchical visibility providing better verification framework of anti-forensic attacks. In Mazza et al. (2011); Métayer et al. (2011), the authors propose a formal framework for specifying and reasoning about decentralised logs, and define an analysis that can generate both precise and approximate evidence of past events.

There are some frameworks and methodologies that propose a testing approach to the validation of digital forensics tools, including among others NIST (http://www.cftt.nist.gov/), Beckett and Slay (2007), Guo et al. (2009), and Shamala and Azizah (2012). Nonetheless, formal verification and analysis of such tools remains an area of research largely unexplored, towards which this paper aims to contribute.

Event-B

Event-B Abrial (2010) is an extension of Abrial's B method Abrial (1996) for modelling distributed systems. This section presents a brief overview of Event-B. Modularity is central to the Event-B method and this is achieved by structuring specifications and development into *machines*. Machines are essentially abstract data types with states, representing an abstract model of a system. An Evetn-B machine can be refined and implemented. The correctness of the machines and the refinements can be validated by proof obligations. Invariants and other predicates are given in first order predicate calculus and set theory. The underlying logic is untyped.

In Event-B, machines are defined in a context, which has a unique name and is identified by the keyword `CONTEXT`. It includes the following elements: `SETS` defines the sets to be used in the model; `CONSTANTS` declares the constants in the model; and finally, `AXIOMS` defines some restrictions for the sets and includes typing constraints for the constants in terms set membership. When a context is refined, it `EXTENDS` its related abstract context.

An Event-B machine is introduced by the `MACHINE` keyword, it has a unique name. A machine `SEES` a particular context, which means that it is able to access any sets or constants declared in that context. The machine also includes the following elements. `VARIABLES` represents the variables (state) of the model. `INVARIANT` describes the invariant properties of the variables defined in the