A Formal Model for Forensic Storage Media Preparation Tools

Benjamin Aziz[1], Philippe Massonet[2] and Christophe Ponsard[2]
[1]*School of Computing, University of Portsmouth, Portsmouth, U.K.*
[2]*Centre dExcellence en Technologies de lInformation et de la Communication (CETIC), Charleroi, Belgium*

Keywords: Computer Forensics, Digital Media Preparation Tools, Event-B Language, Refinement Methodology.

Abstract: This paper defines a model of a special type of digital forensics tools, known as digital media preparation forensic 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 the evidence produced by such tools to be able to 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 media preparation forensic tools by means of testing (e.g. within NIST), 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 the implementations of such tools can be analysed and tested in the future.

1 INTRODUCTION

Computer forensics tools are becoming increasingly of a critical nature due to the complexity of attacks on digital assets and the sophisticated roles that computers 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). Furthermore, 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; Casey and Rose, 2010), where the need for the development of more robust and rigorous scientific methods is highlighted in this area by (Garfinkel et al., 2009).

The term *computer forensics tools* refers to all software and hardware tools used in a forensically sound manner to identify, preserve, recover, analyse and present facts and opinions about information recovered from computers involved in criminal and illegal cases. The National Institute of Standards and Technology (NIST) project on the Computer Forensic Tool Testing (CFTT) (NIST, tgov) 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. We take this assurance process here to another level where the functional specifications and some of the properties of the computer forensic tools are formally defined and verified using the well-established refinement framework of the Event-B method (Abrial, 2010). According to Casey (Casey, 2011), such formalisation "encourages a complete, rigorous investigation, en-sures proper evidence handling and reduces the chance of mistakes created by pre-conceived theories, time pressures and other potential pitfalls."

This paper presents a specification of one class of digital forensic tools, known as *forensic storage media preparation tools* (NIST, 2009), in Event-B (Abrial, 2010). The aim behind this specification is to provide the tool implementations a robust basis in reasoning about their behaviour and to provide more formal grounds for future generation of test cases. More importantly, the significance of such work is that it provides first steps for a new research direction exploring the much-needed use of well-established, industrial-scale formal modelling and analysis frameworks in the critical field of computer and digital forensics.

Aziz B., Massonet P. and Ponsard C..
A Formal Model for Forensic Storage Media Preparation Tools.
DOI: 10.5220/0004996001650170
In *Proceedings of the 11th International Conference on Security and Cryptography* (SECRYPT-2014), pages 165-170
ISBN: 978-989-758-045-1

The rest of the paper is organised as follows. In Section 2, we give an overview of NIST's definition of forensic storage media preparation tools and some of their requirements, both core and optional. In Section 3, we define our abstract machine specifying how such tools should behave at the highest level of specification. In Section 4, we refine the abstract model by including the concept of hidden data sectors in the device being prepared. Finally, we conclude the paper in Section 5 and outline future research directions.

2 FORENSIC STORAGE MEDIA PREPARATION

Forensic storage media preparation tools is a term often referred to any storage devices (hard disks, CDs, solid-state devices etc.) that are used in digital forensic investigations by the investigators. Often, there is a need to re-use such devices from one investigation to another, however, it is mandatory from a legal point of for such devices to be absolutely sanitised in order to prevent data from earlier investigations corrupting evidence in new ones. The sanitisation process, called *media preparation*, involves overwriting all user data on these devices with some agreed-upon form of benign data.

NIST's CFTT programme defines informally the requirements, both core and optional, for forensic storage media preparation tools (NIST, 2009). In addition to this informal specification, NIST defines the test assertions and plan (NIST, 2005) that are expected to be used for setting-up and executing tests and measuring their results. We give an overview next of only the core and optional requirements that we have focused on in our formal specification presented later:

- *Core Requirements*:

  FMP-CR-01. All visible sectors shall be overwritten.

- *Optional Requirements*:

  FMP-RO-01. If the tool supports overwriting hidden sectors, then all sectors contained in a hidden area shall be overwritten.

  FMP-RO-02. If a hidden area exists on the storage device the tool may optionally remove the hidden area from the storage device.

  FMP-RO-03. If the tool supports selection of a command for overwriting and the selected storage device supports an ERASE command for overwriting, then the tool shall allow selection of the ERASE command.

3 THE ABSTRACT MODEL: VISIBLE SECTORS

Our first abstract model of a forensic storage media preparation tool captures only the concept of a visible data sector on the digital medium being prepared. The specification of the abstract context is shown in Figure 1.

CONTEXT AbstractContext
SETS

$Data$ The set of all the data in the world

CONSTANTS

$benignDataElement$ Some benign data element

AXIOMS

axm3 : $Data \neq \varnothing$

Data is not an empty set

axm2 : $benignDataElement \in Data$

The benignDataElement is also an element of Data

END

Figure 1: The Abstract Context of the Media Preparation Tool.

The context introduces the set of all data, *Data*, and a single *benignDataElement* that we use to overwrite the prepared digital storage medium. This benign data according to the NIST document (NIST, 2009) can either be a fixed value, e.g. 0, a fixed pattern of binary data or random data. Based on this context, the abstract machine of Figure 2 defines two events: *Preparation* and *Termination*.

Once the preparation of the forensic storage medium has ended, the *Termination* event terminates the machine by changing the value of the *Terminated* machine state variable from False to True. The preparation of the medium takes the form of replacing a complete visible sector of the medium with a data set, *benignDataSet*, which contains only the *benignDataElement* value. The preparation is deemed to have ended once data in the *ForensicStorageMedium* have been replaced by the benign data.

The completeness property of the preparation of the forensic storage medium is expressed simply as the following invariant:

$$Terminated = TRUE \Rightarrow \forall x \cdot (x \in ForensicStorageMedium) \Rightarrow (x = benignDataElement)$$

This invariant supports the single core requirement **FMP-CR-01** outlined in (NIST, 2009). There is no explicit accuracy requirement in the NIST spec-

MACHINE Abstract Machine
SEES AbstractContext
VARIABLES

ForensicStorageMedium The forensic storage medium variable
Terminated A variable representing whether the machine has terminated or not

INVARIANTS

inv1 : $ForensicStorageMedium \subseteq Data$ Type of ForensicStorageMedium
inv2 : $Terminated \in BOOL$ Type of Terminated
inv3 : $Terminated = TRUE \Rightarrow \forall x \cdot (x \in ForensicStorageMedium) \Rightarrow (x = benignDataElement)$ Completeness Invariant

EVENTS
Initialisation
begin

act1 : $ForensicStorageMedium :| ((ForensicStorageMedium' \neq \varnothing) \wedge (ForensicStorageMedium' \cap \{benignDataElement\} = \varnothing))$
The ForensicStorageMedium must be initially non-empty and not prepared
act2 : $Terminated := FALSE$ We start in a non-terminated state

end
Event *Preparation* $\widehat{=}$
any

visibleSector Pick a visible sector of the medium
benignDataSet Pick a set of benign data elements

where

grd5 : $Terminated = FALSE$ Machine must be non-terminated
grd1 : $visibleSector \subseteq ForensicStorageMedium$ Type of visibleSector
grd4 : $\forall x \cdot (x \in visibleSector) \Rightarrow (x \neq benignDataElement)$ visibleSector currently not prepared
grd6 : $visibleSector \neq \varnothing$ visibleSector is not empty
grd3 : $\forall x \cdot (x \in benignDataSet) \Rightarrow (x = benignDataElement)$ Type of benignDataSet
grd2 : $card(benignDataSet) = card(visibleSector)$ Cardinality of benignDataSet

then

act1 : $ForensicStorageMedium := ((ForensicStorageMedium \setminus visibleSector) \cup benignDataSet)$
Replace the visibleSector with the benignDataSet in the ForensicStorageMedium

end
Event *Termination* $\widehat{=}$
when

grd2 : $Terminated = FALSE$ Machine must currently be non-terminated
grd1 : $\forall x \cdot (x \in ForensicStorageMedium) \Rightarrow (x = benignDataElement)$
Only terminate if ForensicStorageMedium has been completely prepared

then

act1 : $Terminated := TRUE$ Terminate machine operation

end
END

Figure 2: The Abstract Machine for the Media Preparation Tool.

ification since the mode of overwriting the storage medium can vary according to the value selected for the benign data.

4 HIDDEN SECTORS

The abstract machine of the previous section was only dealing with visible sectors on storage media. In this part, we increase with level of detail by refinining the machine to be able to deal with hidden as well as visible sectors. The extended context specification is shown in Figure 3.

The refined machine shown in Figure 4 introduces four new events: *VisibleSectorPreparation*, *HiddenSectorPreparation*, *RemoveHiddenAreas* and *OverwritingHiddenAreasSelection*. This is in addition to the *Termination* event extended from the abstract machine. The first event *VisibleSectorPreparation* extends the original *Preparation* event without adding any new functionality. It is still intended (as its abstract parent) to replace only visible sectors on a forensic storage medium.

Along with this event, we define the new event *HiddenSectorPreparation*, which replaces a hidden sector with a benign data set. It does this on a special sector area defined as the set *ForensicStorageMediumHiddenAreas*, which represents all the hidden sectors on the storage medium. This event implements

CONTEXT FirstExtendedContext
EXTENDS AbstractContext
CONSTANTS

$InaccessibleVisibleData, VisibleData, HiddenData, benignData, benignfill$

AXIOMS

axm3 : $InaccessibleVisibleData \subseteq DigitalSource$
axm4 : $VisibleData \subseteq DigitalSource$
axm5 : $HiddenData \subseteq DigitalSource$
axm6 : $InaccessibleVisibleData \cap VisibleData = \varnothing$
axm7 : $HiddenData \cap VisibleData = \varnothing$
axm8 : $InaccessibleVisibleData \cap HiddenData = \varnothing$
axm9 : $DigitalSource = HiddenData \cup VisibleData \cup InaccessibleVisibleData$
axm14 : $VisibleData \neq \varnothing$
axm15 : $InaccessibleVisibleData \neq \varnothing$
axm16 : $HiddenData \neq \varnothing$
axm10 : $benignData \subseteq Data$
axm11 : $benignData \cap DigitalSource = \varnothing$
axm12 : $benignfill \in DigitalSource \rightarrow benignData$
axm17 : $null \notin benignData$

END

Figure 3: First Extension of the Context.

the first optional requirement **FMP-RO-01** (NIST, 2009) for the overwriting of hidden sectors. Since this requirement is stated in an optional sense, the event *OverwritingHiddenAreasSelection* is provided to allow the user interacting with the tool to either switch on or off this functionality by setting a machine variable called *overwriteHiddenData*.

The final new event that we introduce in this refinement is the *RemoveHiddenAreas* event. This, as its name suggests, is intended to implement the optional requirement **FMP-RO-02**, which gives the tool user the option of removing the hidden areas in the storage device by: first joining the existing hidden area to the visible areas, and second, setting the machine variable *ForensicStorageMediumHiddenAreas* to the empty set. The NIST specification of this requirement is quite ambiguous and omits several details. For example, it does not specify what the semantics of the "removal" action of hidden sectors is, and the assumption we make here is that removal means turning hidden areas into visible ones. However, alternatively, this could have been taken to mean the deletion of these areas. Also, it was not clear whether this removal of the hidden areas happens before or after the overwriting of visible sectors as captured by requirement **FMP-CR-01**. If this happens after the overwriting, then these hidden areas (turned visible) will not be overwritten.

We now strengthen our completeness invariant as follows:

MACHINE Refined Machine
REFINES Abstract Machine
SEES AbstractContext
VARIABLES

ForensicStorageMedium The forensic storage medium variable
Terminated A variable representing whether the machine has terminated or not
ForensicStorageMediumHiddenAreas A variable to represent the hidden areas in the medium
overwriteHiddenData A variable to indicate whether or not the tool supports overwriting hidden areas

INVARIANTS

inv1 : $ForensicStorageMediumHiddenAreas \subseteq Data$
inv2 : $overwriteHiddenData \in BOOL$
inv3 : $Terminated = TRUE \Rightarrow (\forall x \cdot (x \in ForensicStorageMedium) \Rightarrow (x = benignDataElement)) \wedge ((ForensicStorageMediumHiddenAreas \neq \varnothing) \Rightarrow (\forall x \cdot (x \in ForensicStorageMediumHiddenAreas) \Rightarrow (x = benignDataElement)))$
New Completeness Invariant

EVENTS
Initialisation
extended
begin

act1 : $ForensicStorageMedium :| ((ForensicStorageMedium' \neq \varnothing) \wedge (ForensicStorageMedium' \cap \{benignDataElement\} = \varnothing))$
The ForensicStorageMedium must be initially non-empty and not prepared
act2 : $Terminated := FALSE$
act3 : $ForensicStorageMediumHiddenAreas :| ((ForensicStorageMediumHiddenAreas' \neq \varnothing) \wedge (ForensicStorageMediumHiddenAreas' \cap \{benignDataElement\} = \varnothing))$
act4 : $overwriteHiddenData := FALSE$
Initialise overwriteHiddenData

end
Event *VisibleSectorPreparation* $\widehat{=}$
extends *Preparation*
any

visibleSector Pick a visible sector of the medium
benignDataSet Pick a set of benign data elements

where

grd5 : $Terminated = FALSE$
grd1 : $visibleSector \subseteq ForensicStorageMedium$
Type of visibleSector
grd4 : $\forall x \cdot (x \in visibleSector) \Rightarrow (x \neq benignDataElement)$
visibleSector currently not prepared
grd6 : $visibleSector \neq \varnothing$ visibleSector is not empty
grd3 : $\forall x \cdot (x \in benignDataSet) \Rightarrow (x = benignDataElement)$
Type of benignDataSet
grd2 : $card(benignDataSet) = card(visibleSector)$
Cardinality of benignDataSet

then

act1 : $ForensicStorageMedium := ((ForensicStorageMedium \setminus visibleSector) \cup benignDataSet)$
Replace the visibleSector with the benignDataSet in the ForensicStorageMedium

end

Figure 4: First Refinement of the Forensic Media Preparation Tool Machine.

EVENTS
Event *HiddenSectorPreparation* $\widehat{=}$
any
hiddenSector
benignDataSet
where
grd12 : $Terminated = FALSE$
grd11 : $hiddenSector \subseteq ForensicStorageMediumHiddenAreas$
grd13 : $\forall x \cdot (x \in hiddenSector) \Rightarrow (x \neq benignDataElement)$
grd14 : $hiddenSector \neq \varnothing$
grd15 : $\forall x \cdot (x \in benignDataSet) \Rightarrow (x = benignDataElement)$
grd16 : $card(benignDataSet) = card(hiddenSector)$
grd17 : $overwriteHiddenData = TRUE$
then
act11 : $ForensicStorageMediumHiddenAreas := ((ForensicStorageMediumHiddenAreas \setminus hiddenSector) \cup benignDataSet)$
end
Event *Termination* $\widehat{=}$
extends *Termination*
when
grd2 : $Terminated = FALSE$
Machine must currently be non-terminated
grd1 : $\forall x \cdot (x \in ForensicStorageMedium) \Rightarrow (x = benignDataElement)$
Only terminate if ForensicStorageMedium has been prepared
grd3 : $\forall x \cdot (x \in ForensicStorageMediumHiddenAreas) \Rightarrow (x = benignDataElement)$
then
act1 : $Terminated := TRUE$
end
Event *RemoveHiddenAreas* $\widehat{=}$
when
grd1 : $ForensicStorageMediumHiddenAreas \neq \varnothing$
then
act1 : $ForensicStorageMedium := ForensicStorageMedium \cup ForensicStorageMediumHiddenAreas$
act2 : $ForensicStorageMediumHiddenAreas := \varnothing$
end
Event *OverwritingHiddenAreasSelection* $\widehat{=}$
when
grd1 : $overwriteHiddenData = FALSE$
then
act1 : $overwriteHiddenData := TRUE$
end
END

Figure 4: First Refinement of the Forensic Media Preparation Tool Machine (Cont.).

$$Terminated = TRUE \Rightarrow (\forall x \cdot (x \in ForensicStorageMedium) \Rightarrow (x = benignDataElement)) \land ((ForensicStorageMediumHiddenAreas \neq \varnothing) \Rightarrow (\forall x \cdot (x \in ForensicStorageMediumHiddenAreas) \Rightarrow (x = benignDataElement)))$$

where the second part of the invariant expresses the case where hidden sectors have not been removed, and are therefore also overwritten by the benign data element.

5 DISCUSSION AND CONCLUSION

This paper presented a formal specification of forensic digital media preparation tools based on Event-B. The specification defined three levels of abstraction; the first abstract level does not distinguish in terms of the accessibility of visibility of the sectors in the prepared device, the second includes more detail by distinguishing between accessible, non-accessible and hidden data sectors. Finally, the third level also allows for the possibility of selecting the *Erase* hardware command in the prepared device. The discharging of the proof obligations for the accuracy and completeness properties helped reveal that accuracy, unlike completeness, is not a general property that can be specified, reasoned on or even talked about in a uniform manner. The validity of the accuracy property is closely coupled with the accessibility property of the prepared device sectors.

The application of formal modelling and analysis techniques to digital forensics is by no means a new idea, although it has been massively under-researched in many aspects within the field of digial forensics. In (Gladyshev and Enbacka, 2007), the B method (Abrial, 1996) was used for developing incosistency checks and verifying the correctness of digital evidence. The B method has also been used to formally specify and refine write blocker systems in (Linas and Laibinis, 2005; 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 (Linas and Laibinis, 2005) by adopting similar approach for a different type of digital forensic tools.

The refinement methodology (where Event-B is an example of) allows detail to be included in the model to as much precision as needed by the system and its context. Therefore the above three levels of abstraction are by no means an exhaustive definition of how forensic media preparation is performed with real tools. Further refinement machines could take this additional detail into consideration. There are several other directions for future research based on the results of this paper. Finally, we plan to consider other digital forensics tools such as deleted file recovery and data acquisition.

REFERENCES

Abrial, J.-R. (1996). *The B Book*. Cambridge University Press.

Abrial, J.-R. (2010). *Modeling in Event-B: System and Software Design*. Cambridge University Press.

Beebe, N. and Clark, J. G. (2005). A hierarchical, objectives-based framework for the digital investigations process. *Digital Investigation*, 2(2):147–167.

Carrier, B. D. and Spafford, E. H. (2004). An event-based digital forensic investigation framework. In *Proc. of the 4th Digital Forensic Research Workshop*, DFRWS'04.

Casey, E. (2011). *Digital Evidence and Computer Crime Forensic Science, Computers and the Internet 3rd Ed.* Elsevier.

Casey, E. and Rose, C. (2010). *Forensic Discovery: Handbook of Digital Forensics and Investigation*. Academic Press.

Ciardhuáin, S. O. (2004). An extended model of cybercrime investigations. *IJDE*, 3(1).

Cohen, F. (2009). *Digital Forensic Evidence Examination*. Fred Cohen & Associates.

Enbacka, A. (2007). Formal methods based approaches to digital forensics. Master's thesis, Åbo Akademi University.

Friedberg, S. (2012). Report of Digital Forensic Analysis in: Paul D. Ceglia v. Mark Elliot Zuckerberg, Individually, and Facebook, Inc. Technical Report Civil Action No: 1:10-cv-00569-RJA.

Garfinkel, S., Farrell, P., Roussev, V., and Dinolt, G. (2009). Bringing science to digital forensics with standardized forensic corpora. *Digital Investigation*, 6:2–11.

Gladyshev, P. and Enbacka, A. (2007). Rigorous Development of Automated Inconsistency Checks for Digital Evidence Using the B Method. *IJDE*, 6(2).

Ieong, R. S. C. (2006). Forza - digital forensics investigation framework that incorporate legal issues. *Digital Investigation*, 3(Supplement-1):29–36.

Linas, A. E. and Laibinis (2005). Formal Specification and Refinement of a Write Blocker System for Digital Forensics. Technical Report 718.

NIST (2003). Software write block tool specification and test plan (v3.0). Technical report, NIST.

NIST (2005). Forensic media preparation tool test assertions and test plan (v1.0). Technical report, NIST.

NIST (2009). Forensic storage media preparation tool specification (v1.0). Technical report, NIST.

NIST (http://www.cftt.nist.gov/). Computer forensics tool testing (cftt) project web site.