Cloud Digital Investigations based on a Virtual Machine Computer History Model

Sean Thorpe[1], Indrajit Ray[2], Tyrone Grandison[3], Abbie Barbir[4]

[1] Faculty of Engineering & Computing, University of Technology,
Kingston, Jamaica
sthorpe@utech.edu.jm

[2] Department of Computer Science, Colorado State University,
Fort Collins, USA
indrajit@cs.colostate.edu

[3] IBM Research,
York Town Heights, New York, USA
tyroneg@us.ibm.com

[4] Bank of America,
abbie.barbir@bankofamerica.com

Abstract. In several traditional digital investigations, several forensic frameworks have been proposed. The selection of a suitable forensic framework for the cloud computing virtual environments further challenges the existing digital forensics space , as no conclusive generic framework exist that inclusively supports or can work for any Cloud Computing digital investigation. To solve this problem for the data cloud logical domains, this paper describes a model of using the computer's virtual machine history based on finite state machine (FSM) automata theory. The model can be used to define the theory of a Virtual Machine (VM) Cloud Computing digital investigation allowing one to set the stage for prescribed applications operating within these abstract domains. The paper summarizes the theoretical concept used by the virtual machine hypervisor kernel logs that map the ideal and inferred VM history to the set of corresponding low level primitive states and events of the VM hosted computer environment.

Keywords: logs, virtual, cloud, forensics, history, model.

1 Motivation

The basic concept that motivates the models defined in this work is that all objects, both physical and digital, have a history since the time they were created. The term VM object can refer to an entity, such as a physical thing or a bounded set of digital storage locations.

For physical objects, the history includes stimulation from physical senses: what the object "touched", "heard", or "saw". etc. Objects that are not alive do not interpret what they see, but the concept of sight is used to represent how film records the physical objects based on light. When a physical crime is considered, the investigators and forensic scientist analyze objects at the scene to determine their history. For example if someone or something touched the object. Materials may have been transferred between the objects when they come in contact and the evidence of the contact may exist when the investigation starts. A goal of a VM investigation is to learn as much as possible about the history of the VM objects from the log target database repository of potential evidence. This is similar to the traditional forensic investigator who learns from a crime scene and the objects that are suspected of being from that related crime scene. The VM investigator will never know the full story of a VM object, but he can define inferences about it from the collected kernel logs.

Histories occur at multiple levels. In the real world for physical objects, there are histories at the molecular level and macroscopic level. Not all investigations need to determine all levels of an object's history. For example, an object may need to be analyzed using visual techniques or it may need to be analyzed at lower levels of the system architecture if information is missing.

For a computer the history includes the states and events that occurred. A history includes the machine instructions that have been executed as well as complex events, such as user clicking on a button within a graphical interface. As in the physical world evidence from previous states and events may exist when a computer is investigated. The VM investigator hence must make inferences about the previous states and events based on a final, and possibly intermediate, states of the VM hosted computer system.

2 Introduction

Digital investigations have been conducted in various forms for many years, with several different methodological frameworks proposed. Many frameworks have been presented at past Digital Forensic Research Workshops (DFRWS). The Research Roadmap [6] from DFRWS 2001 outlined a five-phase framework and DFRWS 2004 had three presentations on the topic [2, 4]. To date, no conclusions have been made about which framework is ‘‘best’’ or even most ‘‘correct’’. The skew of correctness becomes even more farfetched as we migrate traditional forensic frameworks into one which is managed by the data clouds. As an argument it should be understood that Cloud Computing post dates the era of the mainframe and grid computing of the

1960's and the 1970's. The definition for Cloud Computing (CC) as argued by NIST [8], presents CC as a rentable service for the shared elastic pool of virtualized network resources. CC in turn provides measured of economies of scales based on this on demand use of resources. As cloud user communities proliferate, the prospects of criminal activities for these collocated logical services has multiplied the challenges of law enforcement who have to administer these virtualized infrastructures, platforms and software service application provisions.

To advance the field of digital investigations to one grounded in science requires the development of comprehensive models and theories that can be validated using scientific methods. This should serve to bring a formal basis to further research within a Cloud Computing Investigation, as well as to satisfy external requirements for scientific rigor (e.g. Daubert requirements).

In the past, the framework proposals have been high-level process models that outline phases of the process[1,2]. Unfortunately, there is typically no unifying process that must be followed when conducting a traditional investigation much less one in the "clouds". Multiple investigators may do different things at different times and all may reach the same conclusions. Further, there are different processes for existing categories of criminal investigations versus a computer intrusion investigation. Therefore, it has been difficult to show that one of the previously proposed frameworks is incorrect because its phases could be appropriate for some circumstances. The data clouds as a meta layer of abstraction to the physical hosted environments , compounds this prescribed difficulty for the digital investigator/system administrator who has to manage various distributed data centers. By assumption we can argue that the investigator for now has jurisdiction for the domains he investigates. This assumption is complicit with a private cloud digital investigation.

The goal of this work was to identify the common concepts of a Cloud digital investigation by focusing on the subject being investigated, which is a VM hosted physical computer. A computation model based on a finite state machine (FSM) and the history of a computer was used to define the VM digital evidence and digital investigation concepts.

3 General History Model of Virtual Machine Concepts

This section defines basic concepts that apply to the VM history model outlined in this paper. Our definitions are inspired by prior work [9, 7]. A synchronized VM digital system is defined as a connected set of digital VM log storage and VM log event devices. Digital storage devices are physical components that can store one or more values and a digital event device is a physical component that can change the state of a storage location. The state of a system is the discrete value of all VM log storage locations and a VM Log event is a meta occurrence that changes the state of the physical system.

This can be graphically seen from Figure 1, which shows that a VM Log event E_1 reading the values from the VM storage device locations R3 and R6 and writing to locations R3 and R4 . The history of the digital log VM system describes the sequence of log states and kernel events between two times.

The computer models represented in this paper makes assumptions about the VM system being investigated. The main assumption is that the system can be represented by a finite state machine(FSM) (Q, $\sum$, δ, s0, F). Modern computers are FSMs with a large number of states and complex transition functions. The FSM by default is only a generic computer to prevent us from being tied to the specific platform.

When considering the history of the modern computer, the FSM model is too simple and static because it does not directly support removable components. The removal or the insertion of components, such as external VM storage devices, coprocessors, and networks, will cause the storage and computing capabilities to change and therefore the Q, $\sum$, δ FSM sets and functions must also change.

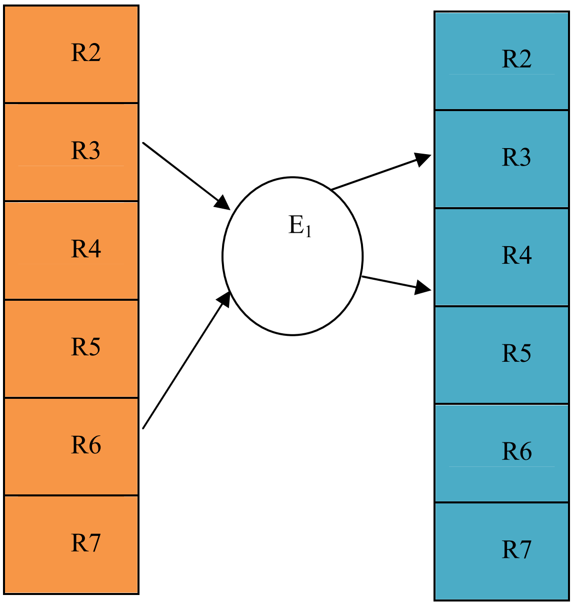

Fig. 1. VM Log Event Occurrence Model

To account for the changing system functions that map a time to the value of a log state, the FSM variables are defined as follows:

- $\sum(t)$ is the symbol alphabet of the VM at a time t $\in$ T.
- $Q(t)$ is set of all possible running kernel system and application state

processes within the VM as captured by the logs at time t $\in$ T.

- $\delta(t)$ (s,e) is the transition function of the VM at time t $\in$ T, for a log state s$\in Q(t)$ and event e $\in \sum(t)$

In addition, some systems have programmable devices where the logic for instructions can be changed and therefore the transition function δ can change. In most other contexts where FSMs are used, a new FSM can be defined when the system changes, but in this context the system changes must be included in the model because the possible states at a given time may need to be predetermined. To corroborate this formal approach we conducted hypothetical case studies using VM timeline log analysis examples [5, 10]. The study considered the use of modification, access, and creation (MAC) time scenarios that showed the impact on the changing states of the running VM sessions.

4 Conclusion and Future work

There is no single high-level process that every Cloud Computing digital investigation must follow. We have shown a model with theoretical foundations that may be adopted for use in existing frameworks. This work provides a formal basis for providing support on further theoretical research in this space. Our work also shows where the scientific method can arguably be used to support the logical components for a cloud digital investigation.

References

1. Baryamureeba, Venansius, Tushabe, Florence: The enhanced digital investigation process model. In: Proceedings of the 2004 digital forensic research workshop (DFRWS); 2004.
2. Beebe N. L., Clark J.G.: A hierarchical, objectives based framework for the digital investigation process. In: Proceedings of the 2004 digital forensic research workshop (DFRWS); 2004.
3. Carrier B.: Defining digital forensic examination and analysis tools using abstraction layers. International Journal of Digital Evidence (IJDE) winter 2003; 1(4).
4. Carrier B.D., Spafford E H.: An event-based digital forensic investigation framework. In: Proceedings of the 2004 digital forensic research workshop (DFRWS); 2004.
5. Thorpe, Sean, Ray, Indrajit: File Timestamps in a Cloud Digital Investigation. To Appear in the Journal of Information Assurance and Security (JIAS), Volume 7, March 2012.
6. Palmer Gary. A road map for digital forensic research. Technical Report DTR-T001–01, DFRWS; November 2001. Report from the first digital forensic research workshop
7. Thorpe, Sean, Ray Indrajit, Grandison, Tyrone: Towards a Formal Temporal Log Model for the Virtual Machine Kernel Synchronized Environment. Proceedings of the Journal of Information Assurance and Security (JIAS), Volume 6, March 2011.
8. Mell P. & Grance T.: NIST Cloud Computing Definitions, July 10, 2009. Retrieved from: http://csrc.nist.gov/groups/SNS/cloudcomputing.

9. Grandison T., Maximillen M., Thorpe S., Alba A.: Towards a Formal Definition of Cloud Computing. Proceedings of IEEE Services July, 2010.
10. Thorpe S., Ray I.: Detecting Temporal Inconsistency in Virtual Machine Activity Timelines. To Appear in the Journal of Information Assurance and Security(JIAS), Volume 7. March 2012.