Selection and Ranking of Remote Hosts for Digital Forensic Investigation in a Cloud Environment

George Sibiya, Thomas Fogwill
Meraka Institute
Council for Scientific and Industrial Research, CSIR
Pretoria, South Africa
{gsibiya, tfogwill}@csir.co.za

H. S. Venter
Department of Computer Science
University of Pretoria
Pretoria, South Africa
hventer@cs.up.ac.za

Abstract— **Cloud computing is a new computing paradigm which presents challenges for digital forensic investigators. Digital forensics is a branch of computer security that makes use of electronic evidence to build up a criminal case or for troubleshooting purposes. Advances have been made since the advent of Cloud computing in addressing issues that came with the Cloud including that of security. However, not all aspects of security are advancing. Developments in digital forensics still leave a lot to be desired in terms of standards and appropriate digital forensic tools that are applicable in the Cloud. To achieve that, standards as well as standard tools are required for successful evidence collection, preservation, analysis and conviction in case of a criminal case. This paper contributes towards addressing issues in digital forensics by presenting an algorithm that can be used in the evidence identification phase of a digital forensic process. Data in Cloud environments exist in the Internet or in networked environments and data is always accessed remotely. There is therefore at least one connection to a host that exists in a Cloud environment. In a case of a computer system that hosts a Cloud service, the number of connections from clients can be very large. In such a scenario it is very hard to identify an attacker from both active and recently disconnected connections to a host. This may require an investigator to probe all individual IP addresses connected to the host which can be time consuming and costly. There is therefore a need for a mechanism that can identify and rank remote hosts that are connected to a victim host and that may be associated with a malicious activity. In this paper we present an algorithm that uses probabilities to identify and rank suspicious remote hosts connected to a victim host. This algorithm helps minimize the effort required of investigators to probe each IP address that is connected to a victim as connected IP addresses will be prioritized according to their rank.**

Keywords-component; Digital Forensics; Digital Forensic Processes; Cloud Computing; Intrusion Detection

I. INTRODUCTION

Cloud Computing is a relatively new computing paradigm. It still presents research issues in the field of computing. Those research issues mainly comprise issues of security. The aspect of computer security that this paper focuses on is that of digital forensics. Digital forensics (DF) is challenging in Cloud computing because of the distributed nature of the Cloud. Digital forensic investigations that involve the Cloud may be abandoned before perpetrators are successfully prosecuted. The reason for abandonment is that enterprises prefer to carry out an investigation if and only if it is cost-effective. That is, the costs of an investigation always need to be lower that the loss or the value of the assets that are vulnerable to the threat being investigated. Abandoning the case however is not helpful as the perpetrator may continue to commit the same crime. We address this issue of the costs involved when conducting an investigation in a Cloud environment. Our solution will contribute in improving the cost-effectiveness of a digital forensic investigation process.

In Section II we present a brief background on Cloud computing and digital forensics. In section III we present our model that aims to minimize costs associated with conducting a digital forensic investigation in a Cloud environment. In section IV we present an example that shows how the model can be applied in practice. In section V we conclude the paper.

II. BACKGROUND

Cloud computing is built upon virtualization technologies. Hardware, platforms and software that were traditionally installed in the vicinity of the user are now offered as services by a third party [1], [2]. These services include storage and processing hardware, development platforms such as Java Virtual Machines (JVM), and software platforms such as human resource management systems. A third party may be another company within national boarders or a company outside national borders. In all of these scenarios the effort that would be required in carrying out digital forensic investigation differs. The costs, for example, will differ when a need to collaborate with international law enforcement agencies arises versus when collaboration is not required. This is one of the challenges faced by digital forensic investigators in a Cloud environment.

Digital forensic readiness is another approach that is used to minimize efforts required and hence minimize costs when an investigation has to be carried out [3][4]**.** Digital forensics readiness makes data that may be used as evidence readily available throughout the lifetime of a live system or ICT infrastructure. This approach minimizes the effort needed to conduct the investigation as evidence is readily available. The investigation can quickly move to the advanced phases of an investigation process. Human resources required during an investigation are minimized and this also reduces the costs.

Although there are benefits in incorporating digital forensic readiness into an infrastructure, not all infrastructures will do so. When an environment that is without forensic readiness mechanisms is compromised, a cost-effective investigation needs to be conducted as well. Shin in [5] attempts to reduce the costs of a digital forensic investigation by proposing a forensic procedure model. Another model that takes costs into account in fraud detection is the one proposed by Stolfo, Fan, Lee, Prodromidis and Chan in [6]. In contrast to most Intrusion Detection models that concentrate on model accuracy, Stolfo et al take into consideration the costs implications that can result from an undetected fraudulent activity.

Most of the research that address issues of digital forensics either focus on intrusion detection [6–8] which corresponds to an incident detection phase of a DF process or on the latter phases of the process, namely evidence collection and evidence analysis. This is a shortcoming of traditional digital forensic as data in the Cloud is huge, distributed and hence, very hard to collect. The evidence identification phase, which is not given much attention by researchers, can play a role in reducing the scope of data that needs to be collected as evidence.

Cost of conducting an investigation in the Cloud is also increased by the lack of standards and tools. Traditional tools are not suitable for the Cloud due to the large amount of data hosted in it. One of the phases of the digital forensic investigation process presented in the draft ISO/IEC standard in [9] is the evidence identification phase. The standard presents twelve phases of an investigation which are incident detection; first response; planning; preparation; incident scene documentation; potential evidence identification; potential evidence collection; potential evidence transportation; potential evidence storage; potential evidence analysis; presentation and conclusion. Potential evidence identification can play a major role in reducing the costs of a digital investigation in a Cloud environment. Data is distributed in the Cloud and this phase can optimize the identification and selection of locations from where evidence can be obtained. The model presented in this paper therefore contributes to improving this phase.

In section III we present a mathematical modeling of the incident scene as well as our algorithm.

III. Host Selection model

In this section we present a formal representation of an incident scene after it has been reported. We further present an algorithm that determines connected hosts that need to be investigated based on the presented model.

A. *Incident scene modeling*

Consider a live system host that hosts Cloud services in a Cloud environment on which an incident has been reported. As the host is residing in the Cloud, a large number of connections from remote hosts that consume the hosted Cloud service are expected. We represent a set of such remote hosts both connected and recently disconnected as follows:

$$H = \{h_i | h_i \textit{ is a remote host}, \qquad i \in \mathbb{N}\} \tag{1}$$

From the set of hosts that are connected or were connected to a victim, we need select and prioritize remote hosts for a cost effective investigation. A remote host can be associated with at least one active or inactive connection in the victim host. We refer to the set of hosts with active connections as H_A and the set of hosts with inactive connections to the victim host as H_D. H_A and H_D are covering subsets of H. i.e.

$$H = H_A \cup H_D \tag{2}$$

Incident types that can be detected in a computer environment are from a finite set defined in the computer security domain. In this paper we represent a set I of incident types as follows:

$$I = \{i_k | i_k, \textit{is a type of an incident}, \qquad k \in \mathbb{N}\} \tag{3}$$

Each incident type can be associated with a subset of network connection attributes. Network connection attributes include the source and destination ports, among others. We represent the set of attributes, A as follows:

$$A = \{a_i | a_i \ , \textit{a connection attribute}\} \tag{4}$$

Each attribute from set A can take any value from a set of values. The values can either be discrete or continuous. A union of the sets of attribute values is represented by Equation (5).

$$V = \bigcup_i^n V_i = \{x | x \in V_i, \qquad i \in \mathbb{N}\} \tag{5}$$

We take as an example $V_1 \in V$ to represent a distance between hosts. If we chose to represent distance between hosts as a number of hops, V_1would be a set of natural numbers.

The last set that we use in modeling the incident scene is the set of connections, C. We represent the set in Equation (6):

$$C = \{c_i | c_i, \textit{is a network connection and } \ i \in \mathbb{N}\} \tag{6}$$

Each connection will have a subset of attributes from the set A in Equation (4). Following the formal representation of the incident scene through the equations (1) through (6) we can now present the composition of functions that compute a set of prioritized hosts that need to be investigated. These functions are presented in the next section.

B. *Algorithm*

The algorithm starts with a function that takes the incident type as an input and provides a subset of attributes relevant to the scene.

$$f : I \to A \tag{7}$$

Usually when an incident is detected and reported, during the preliminary examination of the scene (such as the incident response phase), the incident type is also identified. The incident type helps in searching for relevant attributes in the victim host and this is the role of the function f in Equation (7). For example, if the type of attack that is being investigated is the denial of service performed through the classic ping of death [10], an attribute that would be searched in the system would be that of protocol type from set A, which would have a symbolic value of Internet Control Messaging Protocol (ICMP) from set V. Other attributes may also be associated with a ping of death and such attributes would also be considered and included in set A. Building the attributes set is the step that follows immediately after the incident type has been determined.

Next, to associate the attributes set A, in Equation (4) and the connections set C in Equation (6), only relations can be used and not functions. The reason being that a single connection can be associated with multiple attributes in the attributes set A. Similarly, an attribute and its value may be associated with multiple connections in the connections set C. i.e., $f(a) = x$ and $f(a) = y$ with $x \neq y$. Thus, relations are not functional. We therefore represent the association with a symmetric relation R, a subset of the Cartesian product of sets A and C as follows:

$$R \subseteq A \times C \tag{8}$$

We then define two functions that move from the relation R to produce the set of hosts to be investigated, H. These functions are g and h. The function g has the relation R defined in Equation (8) as its domain and its range is C. i.e,

$$g: R \to C \tag{9}$$

And function h is a function with domain C and the range, H. i.e,

$$h: C \to H \tag{10}$$

It is worth noting that, $\forall c \in C \exists! h \in H$. This means that there is no connection that exits without source or destination host. The two functions form a composite function:

$$h^{\circ}g = h(g(R)) \tag{11}$$

The functions, h and g utilize the characteristic functions k and l in Equations (12) and (13) in building the subsets. The characteristic functions are defined as follows:

Let $C_j \subseteq C$. And also let H be a set of remote hosts, $H_j \subseteq H$ and $j \in \mathbb{N}$. Characteristic functions that determine if a host or a connection is an element of the subsets C_j and H_j respectively can be applied. These functions are denoted by k_{C_j} and l_{C_j}.

$$k_{C_j}(c) = \begin{cases} 1, & if\ c \in C_j \\ 0, & if\ c \notin C_j \end{cases} \tag{12}$$

Similarly,

$$l_{H_j}(h) = \begin{cases} 1, & if\ h \in H_j \\ 0, & if\ h \notin H_j \end{cases} \tag{13}$$

Finally, there is a need to assign weights on each host based on factors such as the distance of the remote host away from the incident scene (locality). A remote host can be from within an organization, inter-organizational, from within the same country or from a foreign country. We assign values 0 through 1 to each remote host represented in set H. These weights reflect the effort (or cost) required to investigate a host. i.e.

$$D = \{d | 0 \leq d \leq 1, d \in \mathbb{Z}\} \tag{14}$$

These weights are used to further reduce the set of remote hosts produced by the composite function in Equation (11).

With the incident scene and the algorithm as presented as above, the model can be implemented. To summarize the algorithm we make use of the flow chart represented in Figure 1.

The most critical part of the algorithm is the final stage where weights are assigned to hosts based on their location and number of connections. It is more costly to investigate hosts that are too far according to Equation (14) than it is to investigate a host that is closer. On the other hand, a remote host that has more multiple connections to a victim host has a higher probability to be an attacker.

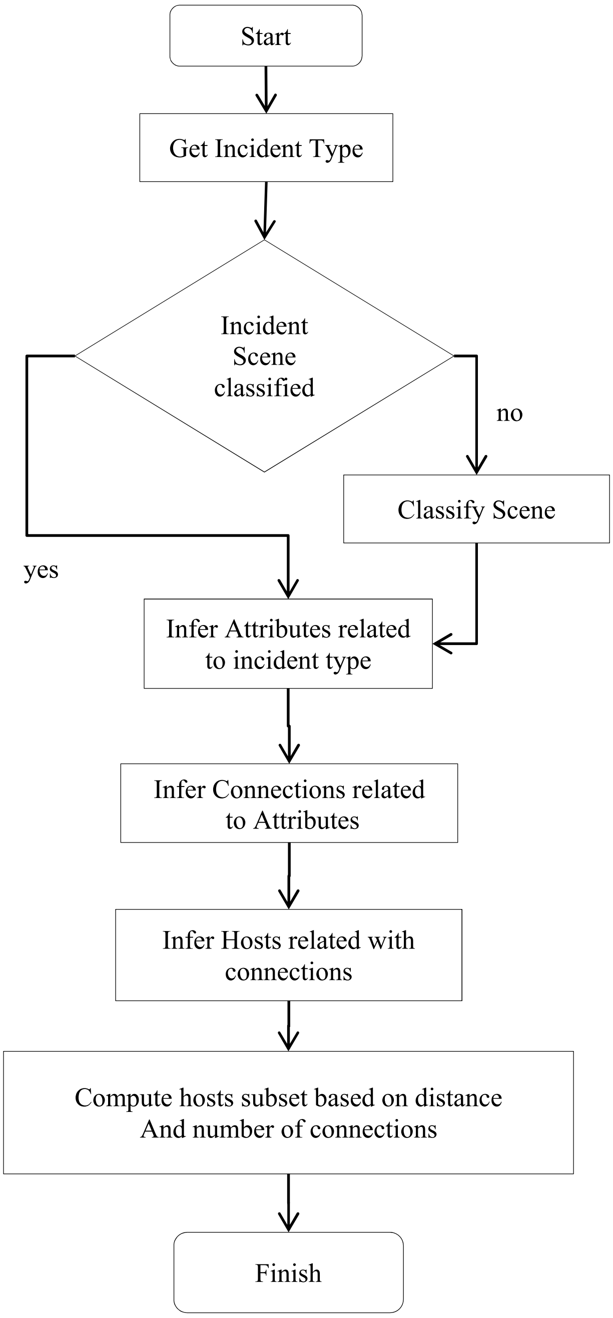
Start
Get Incident Type
Incident Scene classified
no
yes
Classify Scene
Infer Attributes related to incident type
Infer Connections related to Attributes
Infer Hosts related with connections
Compute hosts subset based on distance And number of connections
Finish

Figure 1: Hosts selection algorithm

In the selection process of the host the algorithm needs to find a balance between distance and the number of connections from a remote host.

In the following section we demonstrate the applicability of our model to an incident scene and show how it can be used to select hosts for a cost effective investigation.

IV. Example

In this section we present as an example the representation of an incident scene using the model presented in section III. Typical data that can be found in a TCP dump are data such as the KDD99 data set described in [11], [12]. The KDD99 dump data comprises twenty-two attack types and these attacks are classified into four categories i.e., denial of service (DoS), unauthorized access from a remote machine (R2L), unauthorized access to local super user (root) privileges and surveillance and probing (probing) [13]. The data set has up to forty-one attributes that are associated with each network connection to the host. Much [6], [11], [12], [14–16] has focused on analyzing and classifying attacks in the data set. In this paper we assume classification has already been performed by the intrusion detection system that reported the incident. Our task is to identify hosts that can be prioritized for a cost-effective digital forensic investigation.

We start by mapping the information represented in the KDD99 data set into it. Since the data set has forty-one attributes, Equation (4) becomes:

$$A = \{a_i | a_i \ , 1 \leq i \leq 41, i \in \mathbb{N}\} \tag{15}$$

Connections correspond to instances in the data set. With 409021 records each corresponding to a connection, Equation (6) is as follows:

$$C = \{c_i | c_i, 1 \leq i \leq 409021, i \in \mathbb{N}\} \tag{16}$$

The initial set of hosts denoted as H_{init}, comprises both source and destination attribute values from all the connections in the victim host. i.e., let I_s be a set of source IP addresses and I_d be the set of destination IP addresses. Therefore:

$$H_{init} = \{I_s \cup I_d | I_s \cap I_d = \emptyset\} \tag{17}$$

Next, duplicate entries and local host IP address are removed from the set H_{init} to obtain H in Equation (1). Hence, the size of the set H will therefore be:

$$|H| = \begin{cases} n - 1 - \sum |h_i|, & if\ \exists h_k \in H \ni h_i = h_k \\ n - 1, & if\ \nexists h_k \in H \ni h_i = h_k \end{cases} \tag{18}$$

Where n is the size of the combination of the sets of source and destination IP addresses, Equation (17). And $h_i, h_k \in H_{init}$. If we assume that the 409021 connections in the KDD99 data set did not have duplicate entries the size of set H ($|H|$) would be 409020.

With the data set mapped into our model, the simple algorithm presented in Figure 1 can be applied. The first step in the algorithm is to obtain the incident type as classified by an intrusion detection system or by any other means through which the incident was detected and reported. From the specified class or attack type, a subset of attributes from set A in Equation (15) can be determined. In this example we refer to results obtained using entropy to determine a subset of attributes that best describe or isolate a class. Entropy was used in [11] and, [12]. If we consider the results obtained by Olusola et. al. in [11], the most relevant attributes to consider for the attack types satan, ip sweep, port sweep and nmap are as presented in Table I. These attributes would therefore be relevant when investigating an incident with a DoS attack.

Table I: Attributes most relevant to attack types

Attack type	Attribute
satan	diff server rate
ipsweep	dest host name src port rate
portsweep	srv error rate
nmap	source bytes

The relevant attributes in Table I for a DoS incident have continuous values from set V in Equation (5). Before the next step in our algorithm where connections related to the attributes are inferred. For attributes that have continuous values, thresholds need to be set by an investigator. Based on these thresholds, connections that have any of these attributes above the threshold are included in set C in Equation (6). This is the role of function g in Equation (13). From the output of g, function h builds a subset of hosts.

Finally, the set of hosts H is reduced further based on the assigned weighs in Equation (14). H will finally comprise of reduced list of remote hosts that will be investigated.

V. Conclusion and Future Work

Digital forensics remains a challenge in Cloud environments despite developments in the security aspects of the Cloud. This is due to the lack of standards and tools that can be used in Cloud environments. This contributes to the escalation of costs when an investigation has to be conducted in a Cloud environment. An attempt is made in standardizing a digital forensic process through the draft standard in [9]. In this paper we contribute to the evidence detection phase of the standard. After an incident has been detected and reported in a distributed environment such as the Cloud, it is difficult to identify locations where evidence can be gathered. Crucial evidence may lie in a remote host that is connected to the incident scene. Our model identifies and prioritizes hosts that may contain evidence. The prioritization of the hosts to be investigated is based on the effort required to investigate the remote hosts given their proximity. We have demonstrated how this model can be applied in practice using the KDD99 training data set.

The model is aimed at minimizing costs involved in conducting a digital forensic investigation in a Cloud environment. As future work, we will model costs involved in conducting a digital forensic investigation in the Cloud. We will also demonstrate how the model presented in this paper minimizes the cost in terms of monitory values.

REFERENCES

[1] R. Buyya, C. Shin, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms : Vision , hype , and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[2] R. Lovell, "White Paper: Introduction to cloud computing." 2011.

[3] A. Mouhtaropoulos, M. Grobler, and C.-T. Li, "Digital Forensic Readiness: An Insight into Governmental and Academic Initiatives," *2011 European Intelligence and Security Informatics Conference*, pp. 191–196, Sep. 2011.

[4] R. Rowlingson, "A Ten Step Process for Forensic Readiness," vol. 2, no. 3, pp. 1–28, 2004.

[5] Y.-D. Shin, "New Digital Forensics Investigation Procedure Model," *2008 Fourth International Conference on Networked Computing and Advanced Information Management*, pp. 528–531, Sep. 2008.

[6] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Cost-based Modeling for Fraud and Intrusion Detection : Results from the JAM Project."

[7] A. Houmansadr, S. A. Zonouz, and R. Berthier, "A cloud-based intrusion detection and response system for mobile phones," *Security*, 2011.

[8] A. Mitrokotsa and C. Dimitrakakis, "Intrusion detection in MANET using classification algorithms: The effects of cost and model selection," *Ad Hoc Networks*, vol. 11, no. 1, pp. 226–237, Jan. 2013.

[9] "Information Technology- Security techniques- Investigation principles and processes," U.S. Patent ISO/IEC WD 270432012.

[10] M. Taber, "Maximum Security : A Hacker ' s Guide to Protecting Your Internet Site and Network."

[11] A. A. Olusola, A. S. Oladele, and D. O. Abosede, "Analysis of KDD ' 99 Intrusion Detection Dataset for Selection of Relevance Features," vol. I, 2010.

[12] H. G. Kayacık, A. N. Zincir-heywood, and M. I. Heywood, "Selecting Features for Intrusion Detection : A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets," pp. 3–8.

[13] "KDD Cup 1999 Data." [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[14] M. Tavallaee, E. Bagheri, W. Lu, and A. a. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, no. Cisda, pp. 1–6, Jul. 2009.

[15] I. Levin, "KDD-99 Clssifier Learning Contest LLSoft's Results Overview," vol. 1, no. 2, p. 67, 2000.

[16] M. S. Hoque, "A N I MPLEMENTATION OF I NTRUSION D ETECTION," vol. 4, no. 2, pp. 109–120, 2012.