Virtualization in Intrusion Detection Systems: A Study on Different Approaches for Cloud Computing Environments

Josenilson Dias Araújo and Zair Abdelouahab,

Federal University of Maranhão, Brazil

Summary
This article presents an overview of some of the major works that focus on the use of virtualization in intrusion detection systems to protect against threats in cloud computing environments. The elasticity and abundant availability of computational resources are attractive to attackers in order to exploit vulnerabilities of the cloud, and launch attacks against legitimate users to gain access to private and privileged data. To effectively protect the cloud users, an IDS should have the ability to expand, increase or rapidly decrease the quantity of sensors according to the quantity of resources, as well as the ability to isolate access to the system levels and infrastructures. For this purpose, characteristics of virtual machines as quick startup, fast recovery, stop, migration between different hosts and execution across multiple platforms can be exploited in VM-based IDS, making it a great alternative for monitoring intrusions in cloud computing environments.

Key words:
Cloud computing, IDS, Virtualization, VM.

1. Introduction

In recent years, security for cloud computing environments has received great interest. Protection mechanisms, such as Intrusion Detection Systems, IDS, need to be adapted to effectively protect these new environments. Due to the elasticity and distributed architecture, the computing resources in cloud environments are available, removed or redirected to users according to consumer demand. Therefore, to effectively monitor a number of resources that can increase on demand, the IDS must be dynamically expandable. One possible solution is the use of virtualization, where IDS-based virtual machine (VM) can increase or decrease the amount of sensors for monitoring the expansion of the cloud, making it a good alternative for detecting intrusion into these environments.

Virtualization provides some desirable features to IDS, such as insulation, quick recovery, ability to migrate between different hosts and execution across multiple platforms [1]. The application of these features in VM-based IDS can isolate access to infrastructure and system levels, reducing the impacts of possible attacks by covering each physical node or a virtualized instance of the IDS (IDS VM) that, upon the occurrence of a suspicious event, it can send an alert to another one or to a central element to take the necessary countermeasures, preventing unauthorized access to cloud resources.

In this paper, we discuss different virtualization solutions for IDSs in cloud computing, virtualization techniques employed in such solutions, analyzing possible strengths and limitations.

This paper is organized as follows: section 2, 3 and 4 present a review of the main concepts of cloud computing, virtualization and IDS, respectively. Section 5 shows the work done in the area of intrusion detection for cloud computing using IDS virtualization. Finally, in section 6, we present the conclusion with an account of the possible work improvements that can contribute to achieve better protection in cloud computing environments using IDS virtualization.

2. CLOUD COMPUTING

The paradigm of cloud computing presents a new way of using computing resources. The consumption of computing resources is charged per use where a user contracts services from a provider paying according to what it consumes, thus outsourcing software development and server administration. Virtualization becomes a key element to provide a set of resources such as storage, software, processing power and other computing resources as services to users. A user needs only a browser and an internet connection to consume these resources [2]. According to [3], cloud computing can be defined as a set of virtualized computing resources (virtual machines) available to users through the Internet. With the increase of needing to use more computational resources (such as software, hardware and services) in several areas, cloud computing presents itself as a good alternative for businesses by providing IT services with payment based on usage.

Since cloud computing is new, its model, its specifications and definitions are not standard. One of the most widely accepted definitions by several researchers is the one of NIST (National Institute of Standards and Technology). The NIST [4] defines cloud computing as a model that allows a convenient access on-demand, to a set of configurable computing resources (e.g., networks, servers,

Manuscript received November 5, 2013
Manuscript revised November 20, 2013

storage, applications and services) that can be rapidly acquired and released with minimal management effort or interaction with the service provider.

NIST defines a model of cloud computing which consists of five essential characteristics, three service models, and four deployment models.

The main characteristics of cloud computing are self-service on-demand or on-demand self-service, virtualization features, location independence with access to Internet resources, elasticity and payment model based on consumption [4], [5].

The self-service on demand provides the allocation of resources to a user of the cloud, without the need for intervention from the provider, as needed. The location independence is achieved with the full availability of services, making them accessible from anywhere that has access to the network infrastructure, where the clouds appear to be a single point of access for all the computing needs of the users. Elasticity is the ability to allocate and quickly remove large amounts of resources at runtime. Payment based on consumption is another key feature in cloud computing, where users need to pay only the provider for the services which are used.

The cloud resources are made available to users according to one of the three service models that are:

- IaaS (Infrastructure as a Service), provides computing resources such as processing, storage and networking through the use of virtualization to provide multiple virtual machines on a single hardware shared by the hypervisor.
- PaaS (Platform as a Service), a service that allows the user of a cloud infrastructure to build and deploy their own applications using languages, libraries and tools offered by the provider.
- □ SaaS (Software as a Service), provides applications in the cloud to be consumed on demand.

The deployment models, depending on how resources are organized and how they are available to users can be classified as:

- Public cloud, where access is available to the general public and may be owned and managed by a private, academic, government, or any combination.
- Private cloud infrastructure is provisioned for exclusive use by a single organization.
- Community Cloud, infrastructure is provisioned for exclusive use of a specific user community (organizations with common interests).
- Hybrid cloud, where the cloud infrastructure is made up of any combination of different infrastructures (private, public or community), becoming a single cloud.

The advantage of using cloud computing is to allow hiring new features as they are needed, reducing costs in infrastructure and maintenance.

3. VIRTUALIZATION

Over the years computers have evolved rapidly, with a large increase in processing power. However, that processing power has not been used efficiently, causing an under-utilization of computational resources. To solve this problem, aiming to reduce this idle processing, network administrators started using virtualization technology. Virtualization is a key technology of cloud computing, where the virtual machine is the basic unit of cloud computing platforms [6]. The use of virtualization is the creation of virtual machines that allow operating systems to run concurrently on a single physical machine.

Virtualization is not a new technology. It originated in the 60's with the introduction of a hypervisor technology in IBM mainframes. The hypervisor or virtual machine monitor (VMM) acts as an abstraction layer between the hardware and the various systems running in VMs, allowing partition a single computer system (called host) in several others (called guests). A virtual machine environment is provided by the hyperisor, also known as "operating system for operating systems" [8]. Each VM provides a complete environment very similar to a physical machine having its own operating system, applications and network services, and can interconnect (virtually) every one of these machines forming a virtual network (VN). A virtual machine is an efficient and isolated copy of a real machine [7]. The hypervisor provides an interface (hardware multiplexing) identical to the underlying hardware and controls the VMs. The monitor abstracts the complexities of hardware and system for VMs that can run applications or a "guest system" believing that it has exclusive control of a conventional environment with direct access to the hardware.

4. INTRUSION DETECTION SYSTEMS

With the popularization of computers in the 80s and 90s and the development and a broad access to computer networks and the Internet, more and more computational resources become part of companies and individuals, facilitating tasks and improving productivity. However, with the advance of computer networks and the Internet, numerous cyber threats appeared that target information, which is the main asset of many organizations. Therefore, it is necessary to deploy protection mechanisms against from automated tools or source code that exploits failures of systems that promote access to network servers, compromising their safety.

The intrusion detection system or IDS aims to enhance security in a computer network. The IDS includes monitoring processes, identifying and reporting instances of malicious or suspicious activity.

The IDS attempts to recognize behavior or intrusive action to alert an administrator or automatically trigger countermeasures [9]. It works with the operating system or a network of computers trying to identify malicious activities. It acts as a security tool, and works with others such as antivirus, firewalls and access control systems; it is designed to enhance the security of information systems and communication [11].

According to the classification schemes of IDS [10], the techniques used for intrusion detection can be: signature-based detection and anomaly-based detection. The signature-based detection identifies patterns of known attacks for possible intrusion attempts. These signatures are formed by a set of rules characterizing the attacker, having the advantage of an almost immediate detection and preventing the occurrence of false positives. The anomaly detection based on classifying activities outside the standard (or normal) behavior as anomalies in network traffic [11] [12], and abnormal system behavior, identifying suspicious activity by presenting deviant behavior, which could indicate the presence of malicious activities in the network. The advantage of this method is the detection of unknown threats, however, can produce a high rate of false positives due to the unpredictable behavior of users.

IDSs can be further classified according to the data collection:

- NIDS (Network Intrusion Detection-based) detection systems based on network data capture network traffic for analysis, looking for signatures of known attacks and / or abnormalities in the activities of monitored hosts on the network. The IDS should be located in a point with full visibility of network traffic monitor.
- HIDS (Host-based Intrusion Detection): detection systems, host-based monitor the activity of a local host. It can use log files of operating system events or databases for analysis. The goal is to identify attacks and attempts of unauthorized access to the machine itself. In this case, the IDS is located on the machine monitor.

5. VIRTUALIZATION OF IDS AND ITS APPLICATION IN CLOUD COMPUTING ENVIRONMENTS

An intrusion detection system should be available anywhere in the cloud, to monitor physical and virtualized machines. Since the cloud is a distributed environment, the ideal is that the IDS should also be distributed and use virtualization to monitor the elasticity of the cloud, and thus achieve an effective protection. Some works have been proposed based on a distributed architecture and using virtual machines as in [13]. This work presents an architecture that provides security through three types of services: VM security service, security service and virtual network service management of access control policy. Virtualization is used to address vulnerabilities in the Xen hypervisor [14]. It presents a system for monitoring integrity detection to manage virtual machines in distributed environments with a focus on ensuring the trustworthiness of VMs management that directly manipulate the hardware. In [15] an architecture is proposed to protect the virtual machine monitor (VMM), acting as a filtering layer to examine the system calls made by the hypervisor. In [22], it is proposed an intrusion detection system for grid computing environments and cloud. The system middleware is implemented at the level of the cloud and is protected from intrusion by insulating characteristics of virtualization. The IDS has a distributed architecture, so that each cloud node environment is monitored by a portion of the intrusion detection system; when an attack occurs, an alert is sent to other nodes in the environment. In the work proposed in [16] the log files of intrusion detection systems are analyzed using Hadoop MapReduce algorithms, providing robust reporting for administrators to quickly understand attacks.

The characteristics of virtualization provide greater coverage and resilience against threats within IDS solutions for cloud computing environments. Below, we present works based on this solution.

5.1 Intrusion Detection System in Cloud Computing Environment

In [17], an architecture for intrusion detection in cloud computing environments is proposed. The proposed solution creates separate instances of IDS for each user and uses a single controller to manage the instances. In this IDS architecture, signature as much learning can be used as a detection method.

Figure 1 shows the proposed architecture.

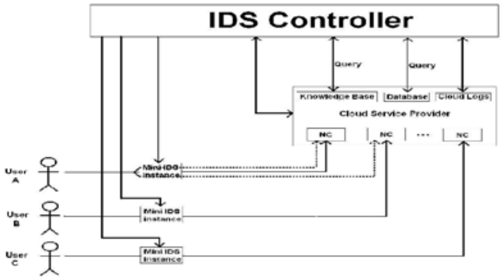

Figure 1. Intrusion Detection System in Cloud Computing Environment [17]

In the proposed architecture, the IDS is composed of instances classified as "mini IDS" created by a central IDS controller. The instances are deployed between each user and the cloud service provider. According to the authors, the main advantage is the reduction in workload because this one is split between multiple instances of the IDS to carry out their work in a better way than with a single IDS for the whole cloud. Thus, whenever a user wants to access cloud services an instance of the IDS is provided by IDS Controller, with the responsibility monitoring and achieving protection.

Therefore, all user activities are monitored by the mini IDS which sends a record of all activities performed by the user to the IDS controller at the end of each session. The controller in turn stores these records in the cloud, and next time the user starts a new session, the IDS controller retrieves the information of the user from the knowledge base. The knowledge base is stored in the cloud and contains information about the patterns of user activity based on information in the log of the cloud. This knowledge base can use neural networks or can be static. Every time, an instance of IDS is provided for a particular user, information about their previous activities are required by IDS controller from the knowledge base. This pattern of activity can be used to detect any intrusion from the user profile; it is possible to apply different rules to different users. In this architecture, there is a one to one relationship between user and instance of IDS and many to many between the IDS instance and node controller. The architecture includes some terms such as Agents, Directors and Notifiers. Information from data sources of log files, processes, and network are captured by Agents and send them to the Directors. The Agents are located within the IDS instances. The Directors, located in the IDS controller, make the analysis of information, which determines whether an attack is happening. In case of a possible attack, Notifier takes the necessary actions.

5.2 An Extensible and Virtualization-Compatible IDS Management Architecture

The work done in [18] presents an extensible IDS architecture that consists of multiple sensors embedded in virtual machines controlled by a central component that analyzes the collected results. Different IDS sensors can be used, and the exchange of messages between them is done with standard IDMEF (Intrusion Detection Message Exchange Format). The architecture is based on characteristics of virtual machines, such as isolation and fast recovery in case of compromise, and uses also the IDMEF standard for exchanging messages between sensors and central unit management. The use of IDMEF provides a standardization of alert information and also allows the use of different IDS sensors.

The VM IDS management system consists of several VM IDS sensor and a central management unit called IDS Management Unit. Each component VM IDS sensor consists of IDS sensors embedded in virtual machines. Users can control the management of the IDS by directly interacting and configuring core components. The structure of VM IDS management system consists of the following components:

- The virtual machines called VM IDS host the sensors and IDS Event Gatherer.

- The central management unit that controls the VM IDS which has four components: Gatherer Event, Event Database, Component Analysis and Remote Controller IDS.

In Figure 2, we can see the elements of this structure.

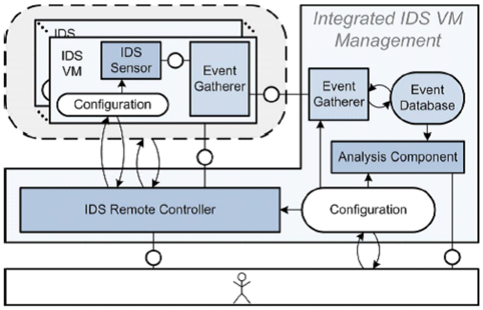

Figure 2. . IDS Management Architecture [18]

IDS sensors are independent processes running within VMs, which can be much like NIDS or HIDS. They identify malicious behavior producing alerts that are transmitted to the central unit via a management alert component connected to its output data called Event Gatherer. The Event Gatherer is present on both the sensor and in the management unit, having the function of collecting all the events coming from various types of sensors and standardize the output for IDMEF messages, and perform the logical communication between sensors and management unit. The gatherer consists of various plug-ins:

- Receivers: are used to read alerts and then converts them to the standard IDMEF.
- Senders: are used to write alerts to the destination, such as network, a database or a file.
- Handlers: are used to modify alerts in phase of processing.

Figure 3 shows the components of the gatherer.

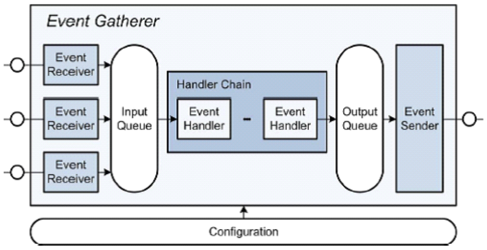

Figure 3. Event Gatherer [18]

The Component Event Database is a database that stores information about all received events and can be accessed via Component Analysis. Each event is stored permanently in the Event Database. A gatherer may be running an instance of a management component that accepts IDS connections and write events to a database.

The Component Analysis has the role of representing the retrieved events and analyzes them by correlating isolated events that may compose complex scenarios of attacks, and having access to the Event database.

The IDS Remote Controller remotely configures and controls all sensors connected to IDS, having access to each of the configurations of sensors and can be controlled and configured by the user. The remote controller can be a software for remote monitoring and control. To manage the IDS in virtual machines, the remote controller can communicate with the VMs IDS and their encapsulated sensors. Three important features of virtual machines management, such as control of VM, VM monitoring and VM configuration are integrated inside the remote controller which has operations of: start, stop, pause, resume, retrieve and update of VMs as well as can read and modify the settings of the IDS sensors. Through the system, one can view information of its sensors, and its virtualized environment as the status of virtual machines (whether a VM is running or not, what is the processing load on the VM, and other information). Several parameters of the IDS running VM can be configured directly through management of IDS, such as basic system files, usable disk space and memory. The recovery of VMs is used as a mechanism for preventing attacks of compromised VMs. Another possible attack prevention can be a temporary shutdown of a VM.

5.3 IDSaaS: Intrusion Detection System as a Service in Public Clouds

The proposal of [19] is to provide a scalable and adjustable service for cloud users, by providing the ability to monitor and react to attacks on several existing VMs in the virtual private network. For this purpose IDSaaS (Intrusion Detection System as a Service) is implemented using the Amazon EC2 web service (Elastic Compute Cloud) [20]. IDSaaS creates a virtual network environment for users through VPC (Virtual Private Cloud) Amazon service, where instances of EC2 VM type are created to store and run security components at the level of infrastructure (IaaS) by providing detection mechanisms completely controlled by users. IDSaaS is an IDS with signature-based detection model and uses the network as a source of data collection. It is scalable, portable, on-demand, user-controlled and available through pay-per-use model. It targets the level of cloud infrastructure, where its first task is to monitor and record suspicious activity in network traffic between virtual machines within a pre-defined virtual network in the public cloud. Figure 4 shows the layout of its components IDSaaS that is built and packaged in the format AMI (Amazon Machine Images). The VPC service is used to create two subnets, one public, where the VMs reside with IDSaaS, and a private that keeps business applications protected.

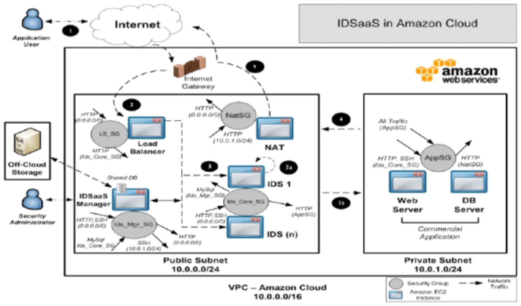

Figure 4. IDSaaS: Intrusion Detection System as a Service in Public Clouds [20]

The VMs containing IDSaaS can be of three types: IDSaaS Manager, Core and IDSaaS LoadBalancer. IDSaaS Manager is the access point of the security administrator, where several monitoring tasks can be made to set up other VMs in both the public and private subnets. The IDSaaS

Manager contains a database Event Database. IDSaaS Core is the gateway to the VMs containing business applications in the private subnet, inspecting all traffic using intrusion detection mechanism, besides other replicas can be created to distribute traffic avoiding single points of failure. LoadBalancer increases IDSaaS system availability in the cloud by doing the balancing of traffic among multiple VMs running IDS core.

5.4 CIDS: A framework for Intrusion Detection in Cloud Systems

A framework for cloud-based IDS is proposed in [21]. The goal is to deal with attacks like masquerade attacks (where threats pose as legitimate users), Host-based attacks (which may be a consequence of masquerade attacks) and Network-based attacks. CIDS also summarizes the intensive network IDS alerts by sending summary reports to the administrator of the cloud.

The solution acts at the level of cloud middleware using its mechanisms, for example, the messaging system and memory insight. CIDS monitors and protects the runtime environment for users (residing in virtual machines), It also maintains its own components protected from threats that may affect the VMs, because the components of the CIDS are located outside of the virtual machines. This protection feature is made possible by the isolation of VMs. CIDS is scalable without the central coordinator and uses a P2P solution. Its architecture distributes the processing load over several local clouds. User tasks are isolated by running them in monitored VMs. CIDS uses both a database of knowledge as well as a database of behavior for analysis, by collecting events and auditing information VMs to increase coverage of attacks.

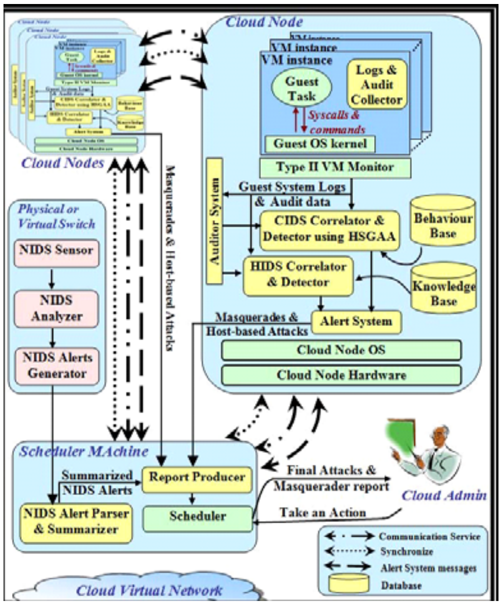

Figure 5. Cloud Intrusion Detection System [21]

Figure 5 shows the architecture of the framework where each node includes an auditing system that monitors log messages between nodes and the system middleware. Each node has a database of known attack signatures (knowledge) and one with abnormal patterns of behavior (behavior). Logs and events of VMs are collected. Through the exchange of information from the databases of knowledge and behavior and by auditing components present in each node, CIDS can detect non legitimate users (designated masqueraders) that try access from multiple nodes and also detect host and network attacks.

The CIDS framework components are:

- Cloud node: contains the resources that are accessed uniformly through the cloud middleware.
- Guest Task: sequence of actions and commands submitted by the user to an instance of VM.
- Logs & audit collector: It is located inside the VM and acts as a sensor for both the CIDS detector as well as the HIDS detector for collecting logs, audit data and action sequences and user commands.
- Instance VM: encapsulates the system to be monitored using VMM (Virtual Machine Monitor). The detection mechanisms are implemented outside the VM, staying away from attackers. A single instance of VM monitoring can observe multiple VMs.
- Type II VMM: CIDS uses VMM of type II, implemented as a process in the host operating system on the physical machine.
- The audit System: it implements three main functions which are: monitoring messages exchanged between nodes by extracting their behavior; monitoring the logging system of the middleware itself and collecting all audit data and events in middleware such as user login and submitted tasks.
- CIDS correlator and detector: correlates and analyzes user behavior according to its own heuristic.
- HIDS correlator and detector: correlates log users and subscriptions collected from various sources.
- Behavior-based database: database and historical profiles of cloud users.
- Knowledge-based database: the database of rules and signatures of known attacks.
- Alert System: uses the mechanisms of middleware communication to alert other nodes in the case of components of (HIDS or CIDS) find some pattern of attack.

- Parser and Summarizer: summarizes the alerts sent by the NIDS.
- Report producer: collect any cloud IDS alerts and sends a report on the attack to the cloud scheduler.

Each physical node has two IDS detectors, one CIDS and one HIDS, where each node participates in the intrusion detection by identifying local events that can pose security breach and exchanging their audited data with other nodes. By having their own analyzer and detector, the nodes perform locally the tasks of analysis and detection by reducing the exchange of information between them, reducing the complexity of analyzing data from multiple locations. This, however, increases the processing overhead within the nodes. This overhead is reduced through the use of HSGAA approach (Semi-Heuristic Approach Global Alignment) that detects attacks based on masquerades alignment algorithm of Smith Global Semi-Waterman [23].

6. ANALYZIS AND COMPARISONS

The solutions proposed by [17] and [18] use a centralized architecture, and may present potential bottlenecks caused by the processing of information collected from different nodes. To avoid the falling of cloud performance, the analysis of the information is made locally, reducing the transmission of information to the central element which undertakes to analyze a small set of data or just display alerts to the system administrator. The availability of an IDS to users of the cloud as a service is proposed in [19] which uses the structure of the Amazon EC2 and VPC services to create VMs and virtual networks used in architecture. The structure of a public cloud facilitates the construction of an IDS cloud, but the applications and user data protected by IDS are exposed to a greater number of threats and also the potential vulnerabilities not yet found. Another possibility is presented in [21] where a distributed architecture without central element is proposed, balancing the workload across the nodes of the cloud and thus avoiding a single point of failure for not having central element. However, the constant exchange of information between nodes to maintain the consistency of the databases, can reduce system performance.

Table I shows a comparison between IDSs based on VM for cloud environments presented above. This comparison takes into account the architecture of the proposed system, as well as the features of VM used in intrusion detection and system protection.

Table I
IDS-based on VM for Cloud Computing

Characteritics	Distributed Architecture	Centralised Architecture
Use of VM characteristics for intrusion detection	[13] [21] [22]	[17] [18] [19]
Use of VM characteristics for IDS protection	[21] [22]	
Use of IDS sensors in a VM	[13]	[17] [18] [19]
Use of IDS Sensors outside a VM	[21] [22]	
Use of virtualization to protect or fix problems in the hypervisor		[14] [15]

7. CONCLUSION

In this paper, we have presented some of the main works on existing intrusion detection for cloud computing environments based on virtual machines. The main advantage in using virtualization in IDS is the isolation of the monitored environment, providing an added layer of security and preventing threats having access to user information or to disable protection in the underlying system. Other features of virtual machines such as fast startup, shutdown and recovery, among other features provide greater coverage against attacks.

A possible extension to the works above is to consider the issue of elasticity which is a characteristic of cloud computing in an IDS. To achieve this purpose, a component which constantly monitors the cloud resources can be added to the IDS based on VM, enabling it to accompany expansion of the cloud. As the cloud environment provides more resources for users, the IDS can increase the number of sensors to monitor the growth of the cloud. With this elasticity, the IDS becomes more effective in detecting intrusion in cloud computing environments thanks to this expansion.

Acknowledgments

Financial support of CAPES and FAPEMA are gratefully acknowledged.

References

[1] X. Zhao, K. Borders, A. Prakash, Virtual machine security systems, book chapter, Advances in Computer Science and Engineering, pp 339–365, 2009.

[2] Sousa, F. R. C.; Moreira, L. O.; Machado, J. C. *Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios.* Ercemapi 2009: Edufpi, pp 1-26, 2009.

[3] Zhang, Chen, Huo. Cloud Computing Research and Development Trend, in Second International Conference on Future Networks. IEEE Computer Society, pp.93-97, 2010.

[4] NIST Definition of Cloud Computing v15 – Accessed on 26/08/2012.

[5] Vaquero, L. M., Rodero-Merino, L., Caceres, J., and Lindner, M. A break in the clouds: towards a cloud definition: towards a cloud definition., ACM SIGCOMM Computer Communication Review, Volume 39 Issue 1, pp 50-55, 2009.

[6] Tan, Xiang & Ai, Bo. The Issues of Cloud Computing Security in High-speed Railway. International Conference on Eletronic & Mechanical Engineering and Information Technology, pp 4358-4363, 2011.

[7] POPEK, G. e GOLDBERG, R. Formal Requirements for Virtualizable Third Generation Architectures. Communications of the ACM, 1974. Vol 17, Nº 7, P 412 – 421.

[8] KELEM, N. e FEIERTAG, R. A Separation Model for Virtual Machine Monitors. Research in Security and Privacy, 1991. Proceedings., 1991 IEEE Computer Society Symposium on 1991, Oakland, California – USA. P. 78 – 86.

[9] LAUREANO, M., MAZIERO, C. e JAMHOUR, E. Detecção de Intrusão em Máquinas Virtuais. 5o Simpósio de Segurança em Informática – SSI. São José dos Campos, pp 1 – 7, 2003.

[10] Nakamura, Emilio Tissato & Geus, Paulo Lício de. Segurança de Redes em ambientes cooperativos. 1. ed. São Paulo: Berkeley Brasil, 2002.

[11] García, T. P. and Díaz, V. J. and Maciá, F. G. and Vázquez, E. (2009) “Anomaly-based network intrusion detection: Techniques, systems and challenges” in: Elsevier Computers & Security,v 28, N1, p 18-28.

[12] Sabahi, F. and Movaghar, A. (2008) “Intrusion Detection: A Survey”, in: The Third International Conference on Systems and Networks Communications, IEEE, p 19-23, 2008.

[13] Jianxin Li, Bo Li, Tianyu Wo, Chunming Hu, Jinpeng Huai, Lu Liu, K.P. Lam. CyberGuarder: A virtualization security assurance architecture for green cloud computing. Elsevier Future Generation Computer System Journal, p 379-390, 2011.

[14] Haifeng Fang, Yiqiang Zhao, Hongyong Zang, H. Howie Huang, Ying Song, Yuzhong Sun, Zhiyong Liu, "VMGuard: An Integrity Monitoring System for Management Virtual Machines," icpads, pp.67-74, 2010 IEEE 16th International Conference on Parallel and Distributed Systems, 2010.

[15] Bharadwaja, S.; Weiqing Sun; Niamat, M.; Fangyang Shen; , "Collabra: A Xen Hypervisor Based Collaborative Intrusion Detection System," Information Technology: New Generations (ITNG), 2011 Eighth International Conference on , vol., no., pp.695-700, 2011.

[16] Shun-Fa Yang; Wei-Yu Chen; Yao-Tsung Wang; , "ICAS: An inter-VM IDS Log Cloud Analysis System," Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on , vol., no., pp.285-289, 15-17 Sept. 2011.

[17] Dhage, S.N., Meshram, B.B.,Rawat, R., Padawe, S., Paingaokar, M., Misra, A. Intrusion Detection System in Cloud Computing Environment, in International Conference and Workshop on Emerging Trends in Technology(ICWET 2011) – TCET, Mumbai, India. 2011.

[18] Roschke S, Feng C, Meinel C. An extensible and virtualization compatible IDS management architecture. In: Fifth international conference on information assurance and security, 2; 2009: pp. 130–134.

[19] Turki Alharkan, Patrick Martin. IDSaaS: Intrusion Detection System as a Service in Public Clouds. In 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp 685-687, 2012.

[20] Amazon Elastic Compute Cloud (Amazon EC2), Jul 2012, Website: http://aws.amazon.com/ec2.

[21] Hisham A. Kholidy, Fabrizio Baiardi. CIDS: A framework for Intrusion Detection in Cloud Systems, in Ninth International Conference on Information Technology- New Generations, pp 379-385, 2012.

[22] K. M. Vieira, A. Schulter, C. B. Westphall, C. M. Westphall. “Intrusion Detection for Grid and Cloud Computing,” in IT Professional Magazine, pp 38-43, 2010.

[23] Scott E. Coull, Joel W. Branch, Boleslaw K. Szymanski, Eric A. Breimer. 2008. “Sequence alignment for masquerade detection”. Journal of Computational Statistics & Data Analysis. 52, 8, pp 4116-4131, 2008.

Josenilson Dias Araujo received the B.S. and M.S. degrees in Computer Science from the Statual University of Piauí (UESPI) in 2001. He is now coursing the MSc degree in Computer Science, at the University of Maranhão. His research interest is in Network Security.

Zair Abdelouahab received his BSc, MSc and Ph.D degrees in Computer Science from University of Setif (Algeria in 1985), Glasgow University (UK in 1988) and Leeds University (UK in 1993) respectively. He is now a professor of Computer Science at the Federal University of Maranhão (UFMA) in Brazil. His research interests include distributed systems, Security Networks, and requirement and software Engineering.