Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 11 (2016) pp. 283-292


# Security Assessment Framework for Multi-tenant Cloud with Nested Virtualization

**Oussama MJIHIL[1], Dong Seong Kim[2] and Abdelkrim HAQIQ[3]**

[1]Computer, Networks, Mobility and Modeling laboratory, FST,
Hassan $1^{st}$ University, Settat, Morocco
*o.mjihil@uhp.ac.ma*

[2]Department of Computer Science and Software Engineering
University of Canterbury, New Zealand
*dongseong.kim@canterbury.ac.nz*

[3]Computer, Networks, Mobility and Modeling laboratory FST,
Hassan $1^{st}$ University, Settat, Morocco
e-NGN Research Group, Africa and Middle East
*ahaqiq@gmail.com*

*Abstract*: **Security assessment and mitigation have gained considerable attention over the recent years according to the information technology evolution and its broad adoption. Organizations are more aware of their data security, and they also have become more exigent in terms of extensibility and flexibility of their Information Technology infrastructures. Cloud computing is introduced as an evolution of information technology that offers major solutions and techniques to meet the evolving requirements of both tenants and clients. Therefore, extensibility and dynamic adjustment, which are among the most essential Cloud advantages, can make the security analysis a very hard task. There have been many approaches to analyze automatically the cyber security of traditional IT infrastructures without taking into account the dynamic nature of Cloud computing and its new features, such as the nested virtualization. Until now, there is a few work to assess the security of Cloud computing. In this paper, we propose a novel approach to design and develop Model-based Automated Security Assessment Tool for Cloud Computing named MASAT.**

***Keywords***: Attack Graphs, Attack Representation Models, Cloud Computing, Security Analysis, Distributed Systems, Virtualization.

## I. Introduction

Cloud computing provides a leap on the Information and Communications Technology (ICT) and allows the use of a massive amount of resources, in a scalable manner and with great automation.

The National Institute of Standards and Technology (NIST) defines the Cloud model as follows: "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." [1]

Cloud computing is becoming increasingly adopted by large and small businesses, thanks to its advantages and cost effectiveness. Although some unusual Cloud features slows down its growth. As never before, customers are invited to trust and put all their data in the hands of another organization, which raises worries and some questions about security and data protection. For these reasons, security is one of the determining factors of this new ICT model. Cloud computing should meet all security requirements of traditional ICT systems.

Virtualization, which is one of the most important functions, enables resource elasticity, pooling resources at massive scale and allows a greater automation and programmability, these features can be improved using software to facilitate the representation and the management of Cloud resources. Moreover, the Cloud Service Provider (CSP) can use virtualization in a nested manner or change dynamically the virtual machines' locations for different reasons.

In order to provide an adaptive Cloud security platform that takes into account the nested virtualization [2], we propose a Model based Automated Security Assessment Tool (MASAT) for the Cloud. MASAT will enable one to assess the security of virtual machines installed at different virtualization levels using distributed security agents. These agents are composed by several interoperable subsystems. Each agent uses a vulnerability scanner to collect information about the system's vulnerabilities, a representation tool that exploits information about the system and its vulnerabilities to generates an attack representation model (here, we use an attack graph) and a communication mechanism to allow the agents to exchange analysis results and other types of messages. Accordingly, we will also address some security issues related to Cloud computing particularities, such as the dynamic adjustment, live migration, scalability and in-

stallation balance of virtual machines, which have a direct influence on Cloud security assessment and mitigation.
MASAT is designed to be adapted with the evolving Cloud features, particularly the nested virtualization, thus the contributions of this paper are summarized as follows:

- MASAT allows one to assess the security of the Cloud infrastructure as well as the virtual machines installed at different virtualization levels.
- MASAT shares the security assessment task between several distributed agents, which makes the Attack Representation Model (ARM) (here, we use an attack graph), generated by each agent, less complex.
- Security assessment of separated subsystems can be performed in parallel by different agents, which can reduce the assessment time.
- This paper presents and discuss also some functional solutions for some current Cloud security challenges.

An earlier version of this paper appeared in [3]. In this extended version, we addressed some additional Cloud features and their influence on the security assessment (i) MASAT will be able to track and handle the dynamic adjustment occurred on the Cloud and (ii) MASAT agents scalability will be addressed and some techniques will be suggested to improve it.
This paper is organized as follows. Section **??** summarizes related works. A Cloud architecture for Nested virtualization is presented in Section **??**. We present the MASAT in Section **??**. We discuss the system complexity in Section V. We present evaluation in Section VI. We briefly discuss some challenges in Section VII. And finally, we conclude our paper in Section VIII.

## II. Related Work

In this section, we present previous researches related to the security analysis and Cloud computing.
**Security models.** For traditional ICT systems, many attack representation models (a.k.a., attack and defense models) have been proposed to analyze and enhance the cyber security, but these models suffer from scalability and dynamic adjustment problems, when the size of the network becomes very large. Accordingly, several researches have been proposed to provide high scalability and performance. Some of those approaches are Two Layer Attack Graph (AG) [4], ACT [5],HARMs [6].
**Cloud Security Analysis and Frameworks.** Cloud security analysis has been a dynamic research area. Many works have addressed security, risk assessment and other Cloud related issues. Takabi *et al.* [7] discussed some security and privacy challenges in Cloud computing considering the security in Infrastructure-as-a Service (IaaS) as a shared responsibility between the tenants and the Cloud service provider. Che *et al.* [8] presented the main existing security models and strategies. Rahman *et al.* [9] proposed incident handling strategies and frameworks to mitigate risks, integrity and availability in Cloud computing environments and their challenges.
Many frameworks have been developed to evaluate and assess the Cloud security. Khorshed *et al.* [10] proposed a proactive attack detection model using modern machine learning techniques. Shaikh *et al.* [11] proposed a trust model to evaluate the Cloud service security strength. Rebollo *et al.* [12], [13] introduced a security governance framework for the Cloud. Zissis *et al.* [14] introduced a Trusted Third Party to analyze security of new Cloud platforms.
Virtualization has been considered in several works. Chung *et al.* [15] proposed a vulnerability detection and countermeasure selection framework for the Cloud using Attack Graph model. Christodorescu *et al.* [16] presented a scalable centralized solution that protects virtual machines using a dedicated security VM. Varadharajan *et al.* [17] provided a security model that enables the Cloud service provider to certify security properties of the tenants' virtual machines and performs a dynamic virtual machine isolation as a countermeasure technique.
Cloud computing security is not only about vulnerabilities and attacks detection since we can improve the Cloud security and plan for some protection techniques that prevent vulnerabilities exploitation. Accordingly, Chonka *et al.* [18] proposed a framework that protects the Cloud against Denial of Service attacks using a back propagation neutral network. In addition to the network and application security, Cloud data security is also a determinant factor. Therefore, a framework combining the three security parameters: MAC (Message Authentication Code), Encryption and Classification of data and Index has been proposed by Sood *et al.* [19] to protect the Cloud data efficiently. Furthermore, other surveys [20, 21, 22] summarized different researches in Cloud security area.

## III. Proposed Cloud Architecture

### *A. Tools and Challenges in New Cloud Platforms*

To make the virtualized infrastructures more flexible, IBMs Turtles project [2] proposed an implementation of a nested virtualization in Kernel Virtual Machine (KVM) [23] hypervisor. With this addition, KVM becomes the first hypervisor for commodity hardware that implements this feature. Today, the nested virtualization is supported by many other hypervisors. This feature makes the Cloud highly flexible and easy to use, so virtual machines in different virtualization levels will depend only on the guest hypervisors where they are installed, which allows the users of private Cloud infrastructures to easily migrate their virtual machines to other private or public Cloud infrastructures.
Virtual machine migration is a basic technique supported by all well known Cloud Platforms. This functionality is used either by the tenants who want to migration their virtual machines between different Cloud Platforms or by the Cloud Service Providers for balancing, security or other strategic reasons. Generally, there are two types of migration, regular migration is when we should shut down the virtual machine and then reboot it after its migration to another host, this kind of migration interrupts the service delivery during the migration time. However, live migration enables the service delivery continuity while changing virtual machines' locations.
Another Cloud advantage is the greater automation. Actually, Cloud platforms allow a direct and easy network control using some new tools and approaches such as Software-

Defined Networking (SDN) [24] that makes the network control highly programmable and abstracts the underlying infrastructure for applications and network services. In our case, SDN can be used to change automatically the network configuration, isolate the compromised virtual machines and so on.
Although, the dynamic adjustment of the network configuration, virtual machine automatic migration and the significant use of the nested virtualization will impact negatively the efficiency and the accuracy of suggested security mechanisms or tools and make security analysis a very difficult task. In such environments, security assessment tools should be able to track any change in the network architecture or virtual machines' locations.
For our experimental example we selected OpenStack [25] among a very large variety of Cloud platforms for the following reasons:

- OpenStack is an open source software which brings together several enterprise and open source solutions to form a powerful and mature IaaS stack.
- OpenStack Compute (Nova) supports KVM [23], Xen and Other hypervisors to perform the management of virtual machine instances.

In this work we built an experimental Cloud architecture using OpenStack as a Cloud platform and KVM as a nested virtualization hypervisor. This experimental Cloud will be used to illustrate the operational mode of our proposed security assessment tool and also to carry out a real security analysis experiment.
Security assessment automation has been addressed in traditional information technology systems to enable security professionals to understand the current security status of their networks, which is not easy to track and understand when attackers use multi-host, multi-stage attacks in a large and complicated network topology. In other words, when the attacker can pass through several hosts to reach a victim host, security assessment becomes very complicated and it should be carried out using software tools. These tools generate a graphical representation of all possible attack scenarios, called Attack Representation Models (ARMs), to help security experts evaluate the network security and then choose the appropriate countermeasures.
Cloud computing is introduced as an evolution of IT systems, then new security platforms can use ARMs for the same purposes as in traditional networked systems but with some adaptations to the Cloud context and its new features. Consequently, when Cloud platforms allow the nested virtualization massive use, security frameworks should be able to analyze not only the main network but also virtual machines and networks installed in a nested manner inside it and take advantages of this feature to facilitate security analysis and countermeasure selection. Moreover, security frameworks should be able to handle the network reconfiguration and virtual machine migration.
In this work, we propose a security assessment system that considers the previously mentioned Cloud features. This system will be able to assess the security of the entire Cloud infrastructure including virtual machines and networks installed in nested levels, using distributed agents. Consequently, each Host or VM containing an underlying virtualization level should contain also a security agent, which can assess the security of its underlying infrastructure using an ARM. Furthermore, we are considering and addressing scalability, automation and dynamic adjustment problems in Cloud Computing.

### *B. Proposed Cloud Architecture Overview*

Cloud infrastructures can contain several virtual machines installed within one or more physical data centers using virtualization. Cloud service providers or owners can build many virtual networks using, in addition to virtual machines, other virtual components and software facilities enabling programmability and automation. Moreover, the nested virtualization allows both the Cloud service provider and the tenant to create other virtual levels, which can also include other virtual machines or networked systems.
Even though scalability and dynamic adjustment are highly recommended for Cloud computing, they are considered as major problems for security evaluation frameworks, because attack representation models suffer already from these problems. Therefore, we propose the Cloud architecture shown in Figure 1 as an experimental Cloud infrastructure.
The Cloud architecture we built to implement our real case scenario enables the installation of virtual machines and virtual networks in nested virtual levels. This architecture is composed of two main physical Cloud servers:

- Cloud server B runs services (e.g., Web server, Mail server, FTP server, DNS server and so on) that are directly accessed by the external users. This area is called Demilitarized Zone, and it's used to add an additional security layer to the Cloud infrastructure.
- Cloud server A hosts virtual machines VM1 and VM2, which contain other virtual levels using KVM hypervisor. In addition to virtual machines, this server contains a database server, security server and virtual networking components.

These Cloud servers are connected to the internet through a physical firewall, which monitor the network traffic exchanged between the internal network and the internet, which is supposed to be not secure. Additionally, to make this infrastructure more secure we used a virtual Gateway to enable the use of the Network Address Translation (NAT), which isolates and hides internal VMs from the internet users by using private and non-routable IP addresses.
To enable remote login to virtual machines we will use Secure Shell (SSH), which requires an encryption key exchange at the beginning of the connection. Thereafter, all TCP segments will be authenticated and encrypted. With this encryption, it becomes impossible to use a sniffer to intercept traffic exchanged between the network administrators and our secure cloud. SSH can be used also to enable other network services to operate securely.
Cloud computing is designed for multi-tenancy and greater automation support. Thus, new approaches to computer networking are created in order to minimize human intervention giving priority to programmability and automation. Accordingly, Software-defined networking (SDN) is increasingly used by Cloud administrators to manage network services

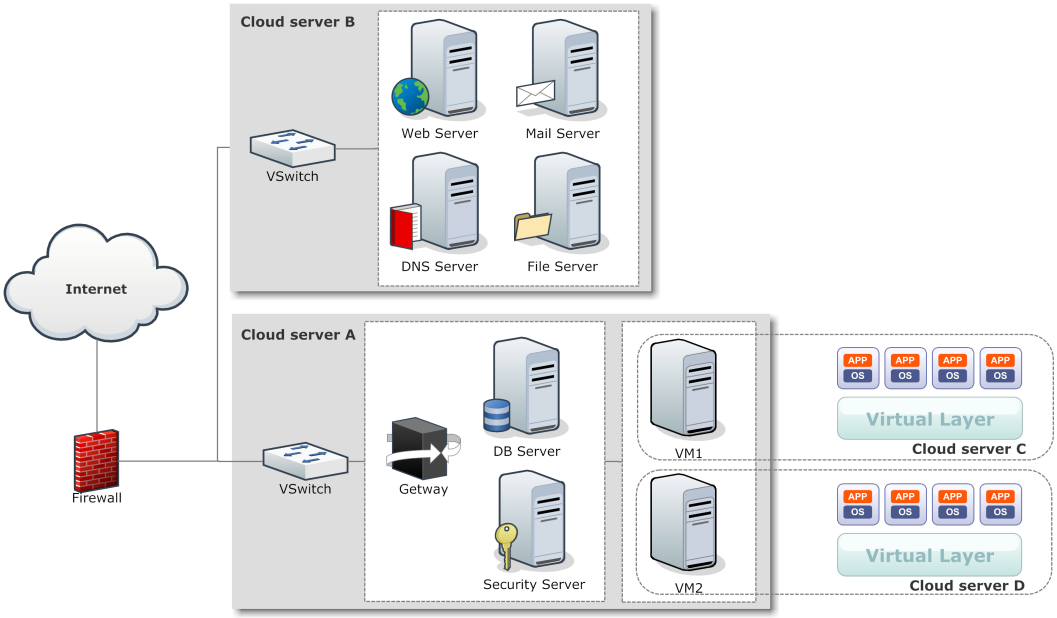


**Figure. 1**: The network topology of the proposed Cloud architecture.

through abstraction of higher-level functionality. This abstraction brings many advantages by enabling the network control to become directly programmable and enhancing network-related security applications.
virtual-machine instance can be moved from one location to another for maintenance, load redistribution or for other reasons. As addressed before, generally we can identify two kinds of migration, non-live and live migration. Since we used OpenStack as a cloud platform, we should notice that Compute service does not use live migration functionality by default. Consequently, one can use KVM-Libvirt to enable live migration and avoid several minutes of downtime while migrating a virtual machine.
From Figure 1, we can observe that VM1 and VM2 contain another virtual level including other VMs and virtual network components that can be automatically managed using software facilities. Consequently, Cloud computing security analysis frameworks should be designed to provide solutions to security analysis problems related to the Cloud features previously mentioned.

## IV. Cloud Computing Security Assessment Framework

Design and implementation of new security assessment frameworks should fit with the evolving Cloud features and paradigms. Cloud computing is increasingly gaining customers' trust, which leads Cloud service providers to do more efforts in terms of scalability and security of their infrastructures. A natural result of this growth is the introduction of new features and techniques helping Cloud administrators to manage their architectures. Obviously, the more we have evolution in Cloud computing, the more we should adapt our security frameworks to support its new features.
Nested virtualization, as explained before, will produce hierarchical virtualization levels. Each level contains virtual machines and/or virtual networked systems that should be also considered in security analysis process. So, MASAT attempts to solve this problem using a distributed and decentralized security assessment system made up of distributed security agents.
Figure 2 depicts the security analysis framework architecture. This framework is composed of a set of distributed security analysis agents, which are structured hierarchically to cover all virtual levels. Each agent is responsible for the security assessment of the underlying virtual infrastructure. An attack graph agent (AG agent) is installed in a virtual level only when it includes VMs and/or virtual networks. These agents communicate and comply to accomplish the overall security analysis of all virtual levels of a given Cloud.

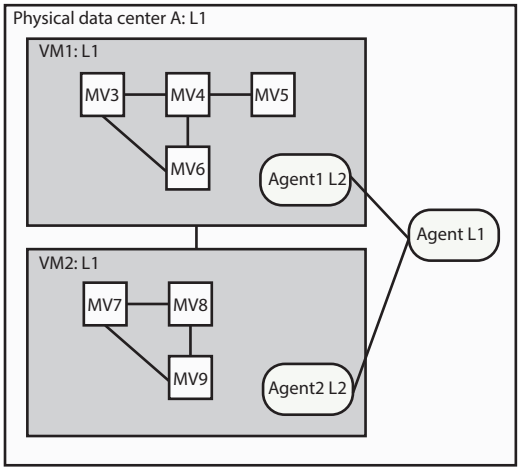


**Figure. 2**: The general architecture of the security analysis tool.

Agents are designed to operate according to the received request type, such as for local or recursive security assessment. When security assessment of a level is started, the corresponding agent scans the underlying network to find all its vulnerabilities and generate an AG according to the steps il-

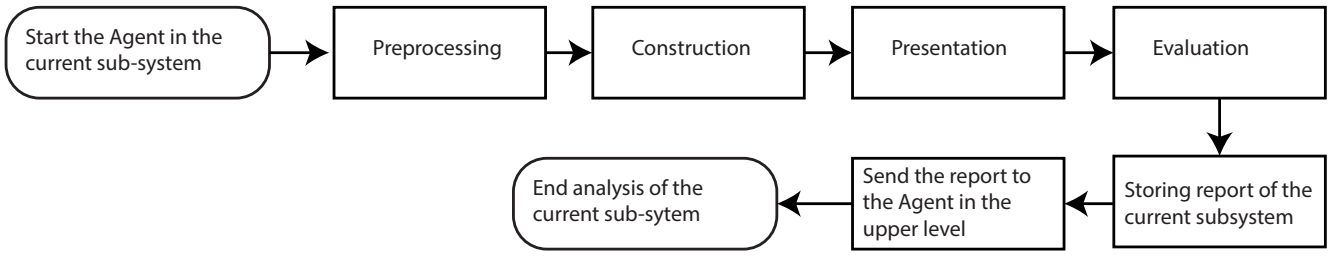


**Figure. 3**: Security analysis phases

lustrated in figure 3, and sends the result to the parent agent in the parent level. To perform a global security assessment of the entire Cloud infrastructure, AG agent in the parent level launches recursively the assessment of the underlying sub-levels, then it starts the analysis of its current level when all the sub-levels' reports are received.
In this distributed security system, an agent is designed to:

- Raise agents installed in the underlying levels (e.g., Agent1 L2 and Agent2 L2 for Agent L1).
- Receive analysis reports from agents of the underlying levels (e.g., Agent1 L2 and Agent2 L2 for Agent L1).
- Analyze the security of the virtual infrastructure installed on the top of the current hypervisor (e.g., VM1, VM2 and so on).

Decentralized execution of agents can provide many benefits, especially when they run on separated physical environments (parallel execution) or on multi-core architectures, it's also possible to start the assessment of each level separately, to analyze the system by stages or to scan it entirely. Moreover, AG agents provide information about vulnerabilities at each level, which facilitates local countermeasure selection and application.
Because the new Cloud platforms enable the nested virtualization, we can use a distributed system to analyze separately the security of each vitualization level. However, the architectural advantages of the nested virtualization can be exploited to enhance the scalability and the pertinence of the security analysis system. Cloud computing is also a model for enabling scalability and automation. As we discussed before, scalability problems can be significantly reduced using distributed systems. Moreover, automation, virtual machine migration and network reconfiguration are anticipated as a part of the operational mode of our suggested framework.
Unlike the previously proposed systems, which perform a centralized security analysis of the entire network, our proposed system consists of a set of distributed agents that comply to perform this task in a nested and decentralized way.
Communication between agents is one of the most essential features of this system. Each agent performs a local security analysis of a virtualization level, then it needs to communicate to other agents to share security analysis results. The main architecture of this system consists of a Remote Procedure Call RPC [26] system allowing agents to communicate using TCP protocol, when they are located in different physical servers, or a shared memory, when both the communicating agents are in the same physical server.
Figure 4 describes the communication between agents previously presented in Figure 2.

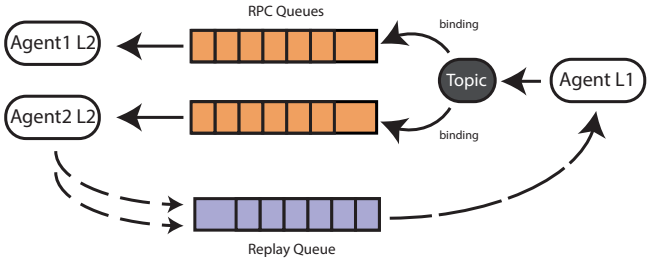


**Figure. 4**: Communication design overview.

To perform the integral security analysis of the Cloud infrastructure, Agents in different virtual levels should have the necessary mechanisms to exchange messages between each other. According to the RPC publish/subscribe pattern, agent that sends the message is called "Producer" and the receiver is called "Consumer". Moreover, producers do not send messages directly to consumers, and they do not have to necessarily know them. The producer can only send messages to a "Topic" that pushes them to the subscribed consumers through their message buffers, called "Queues". These messages can be a command for getting the security status, the network topology or other relative information.
This pattern allows producers to trigger the execution of the work (the security analysis, in our case) of several consumers simultaneously and wait for asynchronous replies. Once the message is received, consumer agents start the procedure described in Algorithm 1 and send the security report as a reply to the producer agent, which will continue the analysis of the virtual level on which it's installed.
Algorithm 1 shows a high level description of the distributed security assessment steps performed by MASAT in case of global Cloud security assessment.

## V. System Complexity

### A. *Message Complexity*

In distributed systems, it is always important to study the influence of the number of nodes or the system's components (in our case, security assessment agents) on the complexity of a distributed processing of a task. Accordingly, message complexity is the total number of messages transmitted during the execution of a distributed task (in our case, Cloud security assessment).
We have $M$ agents structured hierarchically, each agent can communicate with the superior (in the parent virtual level) and the subordinate (in the underlying virtual level) agents, so the number of exchanged messages is $M - 1$ in both the

**Algorithm 1** Algorithm for distributed security analysis system

**procedure** BROADCAST ALGORITHM
  $N \leftarrow Numberofchildnodes$
  **for** $i \leftarrow 0; i < N; i++$ **do**
    **Send** $AnalysisrequesttoChildi()$
  **end for**
  $Done \leftarrow 0$
  **while** $Done <= 0$ **do**▷ We wait for Child's answers
    **Receive** $ReplyFromChild() \leftarrow b$
    $Done+ = 1$
  **end while**
  $Analyzingthecurrentlevel$
  $StoringAnalysisresult$
  **Send** $ReplayToSuperiorAgent()$ ▷ Send the Analysis result to the agent in the upper level
  **return** $b$
**end procedure** ▷ The current level is analyzed

broadcast and reply directions, and the total communication performed by all agents to accomplish the security assessment for the overall cloud platform is $2(M-1)$. Thus message complexity is :

$O(M)$ where M is the number of AG agents.

### B. Asynchronous Time Complexity

In asynchronous algorithms, time complexity is the number of time units necessary to accomplish a distributed task, such as the security analysis of the overall system, in the worst case. Since asynchronous computations are non-deterministic, the computation time will depend on the size and the complexity of the analyzed subsystems. Moreover, the installation balance of virtual machines and networking have an inevitable influence on the security assessment time, since we are interested in computing the worst case scenario. In other words, some agents can have to assess the security of a number of components more than others, then this time will be given by the path containing the most big and complex underlying infrastructures. Assuming that each agent takes 1 time unit to perform the security analysis, the lower bound time complexity will be as follows:

$$O(depth(T))$$

depth(T) is the distance between the first agent and the farthest leaf agent.

### C. Security Analysis Complexity

In small infrastructures, security analysis and mitigation can be done manually by security experts, but in large and complex infrastructures these tasks are beyond the human being's ability. For this reason, security should be checked automatically using a model-based software tool.
The computational complexity is used to quantify the amount of resources required to perform the security analysis task, such as the execution time and memory usage.
Table **??** shows a comparison between the computational complexity of Attack Graph (AG) and Attack Tree (AT) in construction, evaluation and modification phases.

| ARM | Construction | Evaluation | Modification |
|---|---|---|---|
| AG [4] | $O(m^2n^2)$ | $O(m!^n n!)$ | O(mn) |
| AT [27] | O(m!n!) | O(m!n!) | O(m!n!) |

*Table 1*: Comparison between AG and AT

In addition to the distribution balance of virtual machines, The overall security analysis time varies also according to the used attack representation model and the complexity of the underling structure in each virtual layer, see the discussion section.

## VI. Evaluation

### A. Vulnerability Scanning

We presented in the previous sections the operational mode of MASAT. With this framework, vulnerability analysis in Cloud platform containing several virtual layers is performed by more than one agent. These independent agents are designed to have the same behaviour and do exactly the same tasks, because the nested virtual layers have the same characteristics, and also an agent can be a parent, child of another agent or it can play both roles in the same time.
We take the architecture given in Figure 1 as a case study. In view of the fact that agents are identical, we will use "Agent L1" as an example to demonstrate functionalities, life cycle and real application of MASAT.
In vulnerability scanning phase, the agent searches the system's vulnerabilities. During the scanning period, virtual machines may have many vulnerabilities, but other vulnerabilities may be discovered after a new software installation, a network reconfiguration or after a new vulnerability discovery (the zero-day attacks). Consequently, the more the scan is recent the more we can have relevant information about the system's security status, which helps agents and security experts to anticipate the appropriate countermeasures and protection techniques.
Even though security assessment is carried out, and all its results are stored, by the corresponding agent, we have no warranty that the current security status will persist. Some events, such as the network reconfiguration or virtual machine migration can make the previously stored security reports no longer valid. These events can massively occur in Cloud computing.
Vulnerability analysis can be carried out using tools. For this task we used Nessus Vulnerability Scanner [28] to collect vulnerabilities of the networked system monitored by the agent called "Agent L1". This network contains a web server, a file server, a database server and other virtual machines connected to the internet via a router.
In this phase, MASAT agents use Nessus vulnerability scanner to examine the underlying virtual levels and prepare information for next phases. In our use case, "Agent L1" detected three vulnerabilities in its supervision zone. Table **??** shows some information about these vulnerabilities.
As addressed before, "Agent L1" will consider VM1 and VM2 as final nodes. In other words, it will not assess the security of virtual infrastructure installed on their subsequent virtual levels, but it will delegate the analysis of these levels to the underlying agents: "Agent1 L2" and "Agent2 L2".

| Location | CVE | Authentication | CVSS [29], [30] |
|---|---|---|---|
| WebServer | CVE-2009-1956 | Not required | 6.4 |
| WebServer | CVE-2014-0226 | Not required | 6.8 |
| FileServer | CVE-2007-2318 | Not required | 9.3 |

*Table 2*: Vulnerabilities detected by Nessus.

### *B. Attack Graph Representation*

As a result of the previous phase, Nessus vulnerability scanner creates an output file containing the summary of the network's vulnerabilities and their information. This file will be used as an input in the representation phase.
An attacker can use "multi-host, multi-stage" attacks. Thus, he can use different combinations of vulnerabilities and passes through several hosts or equipments to reach his goal, which is not easy to observe from Nessus output file. Accordingly, agents give to these attacks a representation form based on a well known Attack Representation Model. In our current work, we used MulVAL [31] to generate an Attack Graph from the vulnerability information given by Nessus.
Figure 5 depicts the attack graph generated by MASAT using MulVAL and Nessus vulnerability scanner output files. The significant impact of these three vulnerabilities can allow an attacker to use different attack scenarios using multiple combinations of hosts and vulnerabilities to reach the target, especially with the Multiple Unspecified Format String Vulnerabilities discovered in our file server, which allows the attacker to execute arbitrary code within the context of the affected application.
Rather than representing the sequences of exploited vulnerabilities, an Attack representation model may give also information about critical hosts, the rank of the most important vulnerabilities and countermeasures and other important information. These features will be addressed in our future work.

### *C. Communication Between Agents*

Since we have several virtual levels, each agent is considered responsible for the security analysis of the virtual level in which it's installed. An agent complies constantly with other agents. For example, "Agent L1" analyses the initial level without taking into account the internal content of VM1 and VM2, which contain other networked systems. To complete the security analysis of the entire Cloud platform including all its nested virtualization levels, "Agent L1" raises the agents called "Agent1 L2" and "Agent2 L2" in the underlying virtual level. "Agent1 L2" and "Agent2 L2" do the same tasks as "Agent L1", they scan the network, they give the corresponding representations and they check if there are other underlying virtual levels to raise their agents. In addition, they send a report to the main agent, called "Agent L1", which will have a complete image about the entire Cloud security.
In case of dynamic adjustments, such us the network reconfiguration or virtual machine migration, agents exchange information to update their security status. For example, when some modifications occur in "Agent1 L2", its corresponding report already sent to "Agent L1" should be also updated. Then "Agent L1" receive a notification from "Agent1 L2" to update the status of its report.

## VII. Discussion

### *A. Communication Between Hosts*

We discussed before the ordinary use case of the nested virtualization, which involves the installation of one or more virtual machines and virtual networks on the top of a hypervisor. In practice, VMs may have the ability to communicate independently of their locations. Consequently, a network topology can include virtual machines located in different virtual levels.
We consider a communication between VM6 and VM8, as depicted in figure 6 . In this specific case, the agents should not break the link between these two nodes. Each agent will consider this link as an external communication that should be analyzed as well.

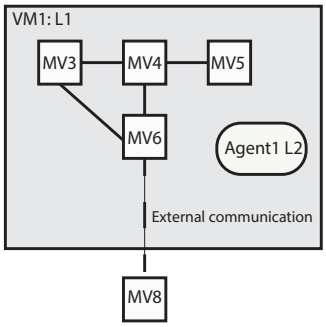


**Figure. 6**: Communication between virtual machines supervised by different agents

### *B. VMs Installation Balancing*

Time is extremely important in Cloud Computing security analysis. A distributed system is not always efficient, because the total required security analysis time is given in the worst case. In other words, the total execution time will be given by the path containing the most time consuming tasks, if the architecture is not well balanced, the anticipated benefits of the distributed system may not be very significant.

### *C. Dynamic adjustment*

Using software facilities, such us software-defined networking (SDN), the network can be reconfigured dynamically. When this reconfiguration occurs in a virtual level, security status will be changed immediately.
In case of virtual machine migration, either for live or regular migration, virtual machines can move to another location (e.g., another virtual level). Thus, the receiver agent will update its security status and notify agents concerned by this change.

### *D. Scalability*

MASAT is designed fundamentally to solve security assessment problems for Cloud platforms enabling nested virtualization and dynamic adjustment. In addition, Scalability can be improved using distributed agents, which does not mean that MASAT solves definitely the scalability problem.
In real case scenario of Cloud computing infrastructures, one virtual level can contain an important number of virtual ma-


**Figure. 5**: Attack graph generated by MulVAL for the test network

chines connecting to each other and having an important number of vulnerabilities. In such case, the controller agents may find a serious scalability problem, which is caused by attack graph itself. Even though many attack graphs have been proposed to deal with scalability problems while assessing network security, this problem remains a challenge, especially after the Cloud success and its broad adoption.
Some research works have presented solutions to improve AG scalability, using decomposition [32], reduction techniques [33] or using clustering and classification [34], [35], [36]. MASAT can be improved using these options in our future work.

### *E. Partial Security Analysis*

An agent can start security analysis in its area independently of other agents. This option will allow the Cloud tenants to control their own infrastructures using the same tool.
A tenant can administrate one or more virtual infrastructures located within the same Cloud, each infrastructure can contain several virtual levels. In this case, it makes sense to give the tenants the ability to monitor the security status of their infrastructures without returning to the Cloud administrators. Moreover, Partial analysis allows Cloud administrators to analyze the entire Cloud infrastructure, which is often very expensive, by stages.

## VIII. Conclusion

In this paper, we have presented MASAT, a Model-based Automated Security Assessment Tool that checks the Cloud security taking into account its new features, such as the nested virtualization. In other words, MASAT allows one to assess the security of the main Cloud infrastructure as well as the virtual machines located in different virtualization levels. MASAT is a distributed system that utilizes distributed agents to collect vulnerability information and generate the corresponding Attack Graph, which facilitates the security evaluation. Moreover, MASAT attempts to reduce the security analysis time by dispatching the work through several workers, performing analysis by stages and so on.
MASAT is composed by several distributed security agents, so it will be important to evaluate the performance, the reliability and other characteristics of this distributed system. Moreover, MASAT can use other Attack Representation Models, generate the corresponding security metrics and use them to perform an automatic countermeasure selection. These features should be investigated in future work.

## Acknowledgments

This research was supported by the NATO Science for Peace and Security Multi-Year Project (MD.SFPP 984425).

## References

[1] Peter Mell and Tim Grance. “The NIST definition of cloud computing”. In: (2011).

[2] Muli Ben-Yehuda et al. “The Turtles Project: Design and Implementation of Nested Virtualization.” In: *OSDI*. Vol. 10. 2010, pp. 423–436.

[3] Oussama Mjihil, Dong Seong Kim, and Abdelkrim Haqiq. "MASAT: Model-based Automated Security Assessment Tool for Cloud Computing". In: *IEEE 11th International Conference on Information Assurance and Security (IAS)*. 2015.

[4] Anming Xie et al. "Evaluating network security with two-layer attack graphs". In: *Computer Security Applications Conference, 2009. ACSAC'09. Annual*. IEEE. 2009, pp. 127–136.

[5] Arpan Roy, Dong Seong Kim, and Kishor S Trivedi. "Attack countermeasure trees (ACT): towards unifying the constructs of attack and defense trees". In: *Security and Communication Networks* 5.8 (2012), pp. 929–943.

[6] Jin Hong and Dong-Seong Kim. "HARMs: Hierarchical Attack Representation Models for Network Security Analysis". In: (2012).

[7] Hassan Takabi, James BD Joshi, and Gail-Joon Ahn. "Security and privacy challenges in cloud computing environments". In: *IEEE Security & Privacy* 6 (2010), pp. 24–31.

[8] Jianhua Che et al. "Study on the security models and strategies of cloud computing". In: *Procedia Engineering* 23 (2011), pp. 586–593.

[9] Nurul Hidayah Ab Rahman and Kim-Kwang Raymond Choo. "A survey of information security incident handling in the cloud". In: *Computers & Security* 49 (2015), pp. 45–69.

[10] Md Tanzim Khorshed, ABM Shawkat Ali, and Saleh A Wasimi. "A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing". In: *Future Generation computer systems* 28.6 (2012), pp. 833–851.

[11] Rizwana Shaikh and M Sasikumar. "Trust Model for Measuring Security Strength of Cloud Computing Service". In: *Procedia Computer Science* 45 (2015), pp. 380–389.

[12] O Rebollo, D Mellado, and E Fernández-Medina. "Introducing a security governance framework for cloud computing". In: *Proceedings of the 10th International Workshop on Security in Information Systems (WOSIS), Angers, France*. 2013, pp. 24–33.

[13] Oscar Rebollo et al. "Empirical evaluation of a cloud computing information security governance framework". In: *Information and Software Technology* 58 (2015), pp. 44–57.

[14] Dimitrios Zissis and Dimitrios Lekkas. "Addressing cloud computing security issues". In: *Future Generation computer systems* 28.3 (2012), pp. 583–592.

[15] Chun-Jen Chung et al. "NICE: Network intrusion detection and countermeasure selection in virtual network systems". In: *Dependable and Secure Computing, IEEE Transactions on* 10.4 (2013), pp. 198–211.

[16] Mihai Christodorescu et al. "Cloud security is not (just) virtualization security: a short paper". In: *Proceedings of the 2009 ACM workshop on Cloud computing security*. ACM. 2009, pp. 97–102.

[17] Vijay Varadharajan and Udaya Tupakula. "Countering security attacks in virtual machines in the cloud using property based attestation". In: *Journal of Network and Computer Applications* 40 (2014), pp. 31–45.

[18] Ashley Chonka et al. "Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks". In: *Journal of Network and Computer Applications* 34.4 (2011), pp. 1097–1107.

[19] Sandeep K Sood. "A combined approach to ensure data security in cloud computing". In: *Journal of Network and Computer Applications* 35.6 (2012), pp. 1831–1838.

[20] Mark D Ryan. "Cloud computing security: The scientific challenge, and a survey of solutions". In: *Journal of Systems and Software* 86.9 (2013), pp. 2263–2268.

[21] Mazhar Ali, Samee U Khan, and Athanasios V Vasilakos. "Security in cloud computing: Opportunities and challenges". In: *Information Sciences* 305 (2015), pp. 357–383.

[22] Farrukh Shahzad. "State-of-the-art survey on cloud computing security Challenges, approaches and solutions". In: *Procedia Computer Science* 37 (2014), pp. 357–362.

[23] Irfan Habib. "Virtualization with KVM". In: *Linux Journal* 2008.166 (2008), p. 8.

[24] Sakir Sezer et al. "Are we ready for SDN? Implementation challenges for software-defined networks". In: *Communications Magazine, IEEE* 51.7 (2013), pp. 36–43.

[25] *OpenStack Cloud*. `http://www.openstack.org/`. Accessed: 2015-10-01.

[26] Raj Srinivasan. "RPC: Remote procedure call protocol specification version 2". In: (1995).

[27] Bruce Schneier. *Secrets and lies: digital security in a networked world*. John Wiley & Sons, 2011.

[28] Jay Beale et al. *Nessus network auditing*. Syngress Publishing, 2004.

[29] Peter Mell, Karen Scarfone, and Sasha Romanosky. "Common vulnerability scoring system". In: *Security & Privacy, IEEE* 4.6 (2006), pp. 85–89.

[30] Peter Mell, Karen Scarfone, and Sasha Romanosky. "A complete guide to the common vulnerability scoring system version 2.0". In: *Published by FIRST-Forum of Incident Response and Security Teams*. 2007, pp. 1–23.

[31] Xinming Ou, Sudhakar Govindavajhala, and Andrew W Appel. "MulVAL: A Logic-based Network Security Analyzer". In: *USENIX security*. 2005.

[32] Jehyun Lee, Heejo Lee, and Hoh Peter In. "Scalable attack graph for risk assessment". In: *Information Networking, 2009. ICOIN 2009. International Conference on*. IEEE. 2009, pp. 1–5.

[33] John Homer et al. “Improving attack graph visualization through data reduction and attack grouping”. In: *Visualization for computer security*. Springer, 2008, pp. 68–79.

[34] Gary William Flake, Robert E Tarjan, and Kostas Tsioutsiouliklis. “Graph clustering and minimum cut trees”. In: *Internet Mathematics* 1.4 (2004), pp. 385–408.

[35] Steven Noel and Sushil Jajodia. “Understanding complex network attack graphs through clustered adjacency matrices”. In: *Computer Security Applications Conference, 21st Annual*. IEEE. 2005, 10–pp.

[36] Steven Noel et al. “Multiple coordinated views for network attack graphs”. In: *Visualization for Computer Security, 2005.(VizSEC 05). IEEE Workshop on*. IEEE. 2005, pp. 99–106.

## Author Biographies

**Oussama MJIHIL** received the BS degree in computer science from Hassan 1st University, Faculty of Science and Technology, Settat, Morocco, in 2010, and the MS degree in software engineering in 2013 from the same university. He is currently working toward the PhD degree in the Computer, Networks, Mobility and Modeling laboratory at Hassan 1st University. His research interests involve: Cloud Computing, security, networking and distributed systems. Since 2014, he is a member of the Nato Project SPS-984425 entitled Cyber Security Analysis and Assurance using Cloud-Based Security Measurement system. He is a student member of the IEEE.

**Dong Seong Kim** Dong Seong Kim is a Senior Lecturer (softly equivalent to an associate professor in the US, but permanent position) in Cyber Security in the Department of Computer Science and Software Engineering at the University of Canterbury, Christchurch, New Zealand. He received Ph.D. degree in Computer Engineering from Korea Aerospace University, South Korea in February 2008. He was a visiting scholar at the University of Maryland, College Park, Maryland, U.S.A. during the year of 2007. From June 2008 to July 2011, he was a postdoc at Duke University, Durham, NC, USA. His research interests are in security and dependability for systems and networks; in particular, Intrusion Detection using Data Mining Techniques, Security and Survivability for Wireless Ad Hoc and Sensor Networks and Internet of Things, Availability and Security modeling and analysis of Cloud computing, and Reliability and Resilience modeling and analysis of Smart Grid. More information is at http://cosc.canterbury.ac.nz/dongseong.kim

**Abdelkrim HAQIQ** has a High Study Degree (DES) and a PhD, both in the field of modeling and performance evaluation of computer communication networks, from the University of Mohamed V, Agdal, Faculty of Sciences, Rabat, Morocco. Since September 1995 he has been working as a Professor at the department of Mathematics and Computer at the Faculty of Sciences and Techniques, Settat, Morocco. He is the Director of Computer, Networks, Mobility and Modeling laboratory. His research interests lie in the areas of applied stochastic processes, stochastic control, queuing theory, game theory and their applications for modeling/simulation and performance analysis of computer communication networks. He is the General Secretary of the electronic Next Generation Networks (e-NGN) Research Group, Moroccan section, an International Steering Committee Chair of the international conference on Engineering Education and Research 2013, iCEER2013, and a TPC member and a reviewer for many international conferences. He is actually Co-Director of the NATO multi-year project SPS-984425 entitled Cyber Security Analysis and Assurance using Cloud-Based Security Measurement system.