Security in Cloud over the Virtual Environment

J. Fahamitha ME (CSE)
Dr.Nagarathinam's College of Engineering,
Mallur, Namakkal

***Abstract*—Cloud computing is becoming increasingly important for provision of services and storage of data in the Internet. However there are several significant challenges in securing cloud infrastructures from different types of attacks. The focus of this paper is on the security services that a cloud provider can offer as part of its infrastructure to its customers (tenants) to counteract these attacks. Our main contribution is a security architecture that provides a flexible security as a service model that a cloud provider can offer to its tenants and customers of its tenants. Our security as a service model while offering a baseline security to the provider to protect its own cloud infrastructure also provides flexibility to tenants to have additional security functionalities that suit their security requirements. The paper describes the design of the security architecture and discusses how different types of attacks are counteracted by the proposed architecture. We have implemented the security architecture and the paper discusses analysis and performance evaluation results.**

Index Terms—Cloud security, security architecture, security and privacy.

I. INTRODUCTION

cloud computing [1]–[3] has become an important technology where cloud services providers provide com-puting resources to their customers (tenants) to host their data or perform their computing tasks. Cloud computing can be categorized into different service deliver models such as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Virtualization [4] is one of the key technologies used in the IaaS cloud infrastructures. For instance, virtualisation is used by some of the major cloud service providers such as Amazon [2] and Microsoft [3] in the provision of cloud services. We will use the term tenant to refer to cloud customers who wish to access services from cloud providers. Tenants can themselves be using their virtual machines to provide services to their own customers; we will refer to customers (or users) as those who use the services of the tenants. Hence customers in our architecture are the customers of the tenants.

In general, the tenants in the cloud can run different operating systems and applications in their virtual machines. As the operating systems and applications of the tenants can be potentially large and complex, they may contain security vulnerabilities. Furthermore, there can be several tenants on the same physical platform sharing resources in a cloud infrastructure. The vulnerabilities in operating systems and applications can be potentially exploited by an attacker to generate different types of attacks. These attacks can be targeted against the cloud infrastructure as well as against other virtual machines belonging to other tenants. So there is a need to design security architecture and develop techniques that can be used by the cloud service provider for securing its infrastructure and tenant virtual machines.

However there are several issues that arise when developing Security as a service for cloud infrastructures. In the current environment, the cloud service providers do not generally offer security as a service to their tenants. For example, in [5] Amazon mentions that security of tenant virtual machines is the responsibility of the tenants since they are free to run any of the operating systems or applications[1] (though it claims to secure the underlying infrastructure). Hence tenants need to make their own arrangements for securing their virtual machines that are hosted in the cloud. Although tenants can use different security tools such as anti-virus and host based intrusion detection systems to secure their virtual machines, the limitations arise [6] due to these tools residing in the same system as the one being monitored and hence are vulnerable to attacks. Also some tenants may not be capable of securing their tenant virtual machines. Hence there is a need for the cloud service provider to offer security as a service to such tenants.

Furthermore, security requirements for tenants may vary and some tenants may opt for more security services from the cloud provider while others may opt for the baseline default security. For example, a tenant who is running financial services on its virtual machines is likely to need more security measures compared to a tenant who is providing basic web hosting. However, greater the level of security measures taken up by the tenant from the provider, greater is the possibility for the cloud provider to get to know more about the tenant's system. That is, the security mechanisms and tools offered by the cloud provider (as part of its security as a service) can gather more information about the operating system and applications running in the tenant's virtual machines. This in turn may lead to greater privacy concerns for the tenant. Here privacy concerns refer to the ability of the provider to find details about the services and applications' data in a tenant's machine.[2]

Our main contribution in this paper is a security architecture that provides a flexible security as a service model that a cloud provider can offer to its tenants and customers of its tenants. Our security as a service model while offering a baseline security to the provider to protect its own cloud infrastructure also provides flexibility to tenants to determine how much control they wish to have over their own virtual machines. The baseline security is needed by the provider to ensure that malicious tenants are not attacking the cloud infrastructure or even hosting malicious software. Every tenant has to have the security functionalities that forms part of the security baseline, which offers basic security guarantees in its default mode of operation. However there will be other tenants who would require additional security services (on top of the baseline) from the cloud provider to meet their requirements as well as to protect them from other malicious tenants

Hence our security as a service model provides options to have additional security functionalities that suit tenants' security requirements. These additional security functionalities can require the tenant to reveal more information about its services and applications, which may create privacy concerns for the tenant.[3]

Our approach offers a choice to the tenant to managing this tension between the privacy concerns and the security controls offered by the cloud provider. An important feature of our model is that it makes this trade-off between security and privacy explicit. Furthermore, the choice by a tenant to opt in for additional security services can provide the cloud provider to develop a framework for charging the tenants for these additional security services.

The paper is organized as follows. We describe the threat model in Section II and consider the different types of attackers and attacks that can occur in the infrastructure as a service cloud environment. Then we summarize the capabilities of the security architecture that is proposed in this paper. Section III describes the security as a service model for cloud infrastructure. It describes the design of the security architecture and discusses how different types of attacks are counteracted by the proposed architecture.

Section IV describes the implementation and analysis of the security architecture, and discusses the performance evaluation results. Section V describes relevant related work and provides a comparison with the capabilities of our security architecture. Finally, Section VI concludes the paper.

II.THREAT MODEL

Our system model involves cloud service provider which includes cloud system administrators, tenant administrators (or operators) who manage the tenant virtual machines, and tenant users (or tenant's customers) who use the applications and services running in the tenant virtual machines. Cloud providers are entities such as Amazon EC2 and Microsoft Azure who a vested interest in protecting their reputations.

The example, the malicious tenant can perform denial of service system administrators are individuals from these co-attacks by crashing the server or starving the resources to other portions entrusted with system tasks and maintaining cloud infrastructures, who will have access to privileged domains. We assume that as cloud providers have a vested interest in protecting their reputations and resources, the adversaries from the cloud provider perspective are malicious cloud system.

There can be attacks from tenant administrators on the tenant virtual machines. That is, the tenant administrators can exploit the vulnerabilities in the tenant virtual machine for malicious purposes. Such attacks can target both the cloud in-frastructure as well as co-located tenants. For example, attacks such as VM escape enable

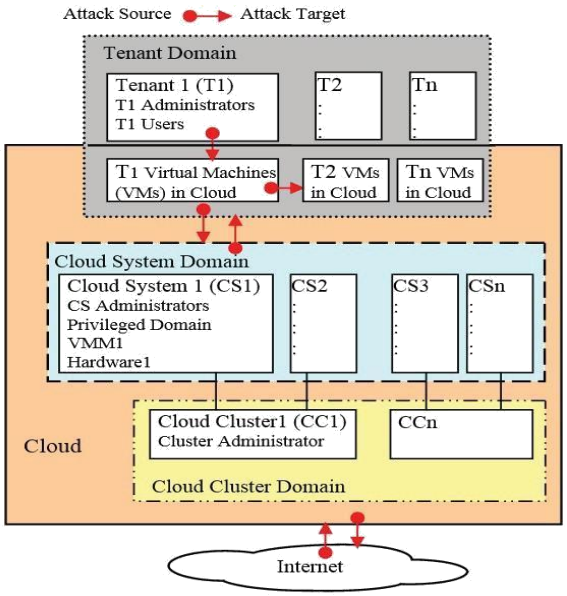

Fig. 1. Cloud Cluster

In Fig. 2, although VMM11, VMM13 and VMM14 are hosting virtual machines that belong to single tenant, VMM12 is hosting virtual machines that belong to different tenants. In such cases, tenants may want to ensure that data can be shared among their own virtual machines but at the same time may wish to protect their virtual machines from other tenant's virtual machines. For instance, in Fig. 2, if Tenant 1 has requested free communication between its virtual machines, and hence the cloud service provider decides not to monitor any of the communication between Tenant 1's virtual machines; then a malicious tenant user will be able to exploit vulnerabilities in TVM13 to generate attacks on the other virtual machines (TVM11, TVM12).

In this case, the tenant administrator can request the cloud service provider for additional security mechanisms to deal with such malicious insider attacks within the tenant organization.In a general situation, a malicious tenant user or more importantly a malicious tenant administrator can generate attacks against virtual machine belonging to another tenant (e.g. TVM21). The cloud service provider needs to provide secure isolation between the tenant virtual machines.

However the cloud service provider may not be aware of the operating systems and applications running in a tenant virtual machine. Hence it is not an easy task for the cloud service provider to enforce security policies on the tenant virtual machines. Fur- thermore since the elastic nature of cloud allows the ability to dynamically increase the resources allocated to tenant virtual machines, the attacker can use this capability in compromised tenant virtual machines to generate sophisticated attacks.

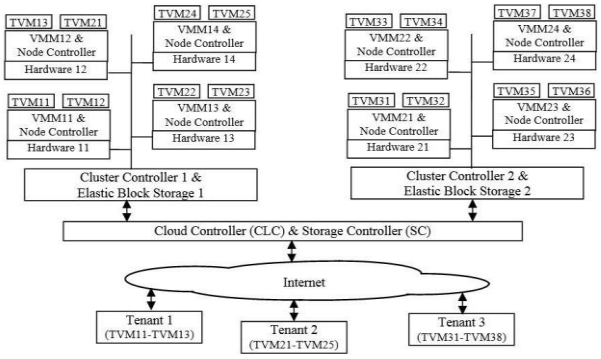

Fig. 2. Cloud scenario.

Hence there can be: (i) insider attacks from the tenant do-main such as a malicious tenant user described above, and (ii) insider attacks from the cloud service provider administrators

Our Approach: Our security architecture assumes that the cloud service provider provides a trusted VMM platform (for example, equipped with Trusted Platform Module (TPM) [15]). We also assume that the security components of our architecture embedded within the VMM are trusted. The cloud provider also provides controls and auditing procedures which ensure the physical security of the cloud infrastructure to overcome hardware based attacks such as cold-boot attacks. Our security architecture allows a cloud provider to provide a default baseline set of security measures even if the tenants do not require any security feature for their virtual machines.

The rationale behind this design choice is as follows. If security measures are not provided by the cloud provider, then the provider will not be able to detect any attack on its own infrastructure itself as well as on other tenant virtual machines and hosts on the Internet. For example, without any monitoring done by the cloud provider, malicious tenants can use their virtual machines to flood other tenant virtual machines or target different services (such as DNS and web server) within the provider infrastructure. On the other hand, greater the level of security measures, greater the possibility for the provider to get to know more details about the tenant's operating system and applications that are running in its virtual machines. It is up to the tenant to decide whether it is acceptable depending on its privacy concerns.

Our security architecture protects the cloud infrastructure from attacks generated within a tenant virtual machine by the tenant administrator and tenant users. It also protects co-located tenants from the attacks generated by such tenant entities. In our architecture, the baseline security mechanisms (in the SPAD component, Section III.D) allow the cloud service provider to protect their infrastructure, legitimate ten-ants as well as external hosts from the attacks by malicious tenants. The security policies and mechanisms (in the security component TSAD, Section III.D) in our architecture deal with the insider attacks from malicious tenant users.

Our security architecture also protects tenants from threats posed by cloud system administrators who misuse their privi-leges and exploits against privileged domain. It enforces role based access control to partition the system administrators to different roles which enables to restrict their privileges to specified set of actions. This in turn helps to contain the harm of the insider attacks; in particular, it removes the possibility of a single administrator (or a single role) having all the privileges.

Our security architecture also provides mechanisms to deal with some attacks on the VMM. This is done using a Security Gateway component which specifies cluster wide policies and mechanisms to detect attacks on the VMM platforms.

Finally our security architecture provides the ability to charge a tenant depending on the security services that are required by the tenant. For example, a tenant virtual machine that is running financial services may need more security measures than a tenant that is running basic web hosting.

III. SECURITY AS A SERVICE MODEL

In this section, we describe our security as a service model for cloud infrastructure. First we give an outline of cloud architecture in Section A. We describe the assumptions made in the design of the security architecture in Section B. Section C gives an overview of the basic security architecture. Section D describes the details of the components in the security architecture. Section E considers the different types of attacks and how our architecture is able to deal with such attacks. Finally Section F extends the security architecture to counteract further attacks on the VMM such as VMM compromise.

A.Cloud Architecture overview

Let us consider a generic cloud service provider architecture as shown in Fig. 2. Tenants (T1, T2, T3) are hosting one or more virtual machines on the cloud service provider infrastruc- ture and remotely managing their virtual machines. The Cloud Controller (CLC) is the main interface for the cloud tenants and it is the top level management for the IaaS cloud. It can query other controllers such as the Cluster Controllers (CC) and Node Controllers (NC), Storage Controller (SC) to make high level decision on the implementation of the tenant virtual are temporarily using the cloud may use the default virtual machine images provided by the cloud service provider. In this case, the cloud tenants may not have complete knowledge of the applications running in the default images. However the cloud provider will be providing the security baseline. We do not consider the situation of a malicious cloud provider providing malicious applications to its tenants in this paper. As we mentioned earlier in the threat model section, the cloud service provider has a vested interest in protecting his/her reputation, and hence does not deliberately provide malicious applications.

B.Assumptions

Let us now consider the assumptions made in our architec- ture. We assume that tenant virtual machines accept a security baseline (mentioned earlier) functionalities specified by the cloud service provider. If there are any special requirements for the tenant which do not comply with the baseline security requirements of the cloud service provider, then these need to be resolved at the time of the registration. The security base- line is enforced by our architecture in the node controller. With respect to the applications running in the tenants, we assume that the tenants are aware of the applications that are running in their own machines. We also assume that tenants may have their own host based security tools (HBST) running in their virtual machines. Furthermore, default security baseline provides the best option for those tenants who are concerned about the privacy of the applications and services running on their virtual machines.

C.Security Architecture Overview

Consider the basic security architecture diagram shown in Fig. 3. As mentioned above, the tenants may wish to have their own host based security tools (HBST) to run on the virtual machines that they are obtaining from the cloud provider. Since host based security tools have good visibility into the system being monitored, this acts as a primary layer of defense in our security architecture. The other important components in our security architecture shown in Fig. 3 are the Service Provider Attack Detection (SPAD) and the Tenant Specific Attack Detection (TSAD) components. First let us look the

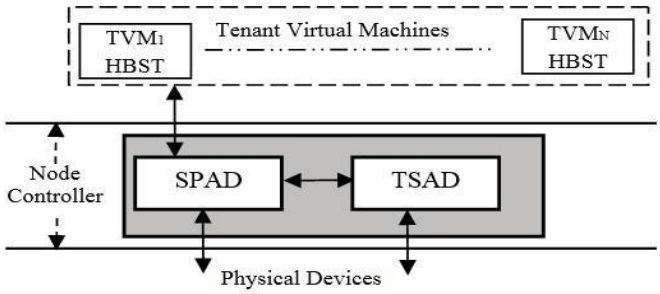

Fig. 3. Basic security architecture.

enforced on all the tenant virtual machines, they are designed to be lightweight and provide basic security base- line with minimal privacy violations. As the SPAD security mechanisms are needed for secure operation of the cloud provider, we envisage that these services will be offered to the tenant by the provider without any additional charges. In Section III.D we discuss the SPAD security mechanisms in detail and describe the types of attacks that these mechanisms are able to counteract. The TSAD enforces tenant specific security policies on the tenant traffic

The TSAD component contains policies based on signatures and anomalies to detect attacks. These include dropping traffic that match with attack patterns, rate limiting traffic above specified thresholds (for instance, for ICMP, UDP and SYN packets), dynamic allocation of resources to tenants as well as identifying malicious communications using access policies on protocols and ports.

TSAD policies also address runtime state based mechanisms not only to detect suspicious processes running in the tenants but also for ensuring the specified security related processes are running in the tenants, thereby validating security of tenants. We describe the mechanisms associated with the TSAD policies in Section III.D. Specific examples of policy specifications are given in the appendix.

D.Component Description

1.Service Provider Attack Detection (SPAD): SPAD is designed to enforce security policies in the baseline thatoffered by the cloud provider. They are intended to minimize the attacks on the cloud provider infrastructure as well as preventing attacks between the tenant virtual machines. Note SPAD policies are enforced on all the tenant virtual machines. In our architecture, the basic SPAD security policies prevent attacks with spoofed source address from the compromised tenant virtual machine and maintain traffic logs originating from the tenant virtual machines for detecting anomalies

2.Tenant Specific Attack Detection (TSAD): TSAD compo- nent enforces tenant specific attack detection policies. In our architecture, the tenant can request the cloud service provider to enforce signature based detection and/or anomaly based detection policies in the TSAD. First, the tenant is able to specify application specific attack signatures depending on the applications that are running in its virtual machines. This will vary with the tenant as different tenants can have different applications and services at different times. Furthermore, greater the information that a tenant is willing to reveal about the applications and services that are running in its virtual machines, more specific security mechanisms can be implemented by the cloud service provider to detect service or application specific attacks. Though this provides a higher level of security, the tenant is revealing more information to the provider and hence potentially its privacy can be reduced. Let us now consider how such fine granular security can be enforced by the cloud service provider on the tenant virtual machines with the process validation (Pro_Val) module in the TSAD.

Pro_Val first checks if the host based security tool related processes (see Fig. 4) are running in the tenant virtual ma- chine. If the tenant virtual machine is compromised, then the processes related to the security tool in the tenant virtual ma- chine will not be detected in the Pro_Val report. In such cases, the tenant virtual machine is considered to be compromised with malware. Hence attacks such as *conficker*

and *torpig* which disable security tools in the tenant virtual machine are detected by Pro_Val.

E. Detection of Attacks

Insider Attack from Tenant Domain

In this section, we discuss how our architecture can be used by the tenants to deal with insider attacks from their users (tenant users in their domain).

In our architecture, the tenant administrators can make use of the TSAD component to detect insider attacks. TSAD can be used to log the activity of the users on their systems. The logs can help to extract the user behaviour, identify security policies that need to be enforced on the user and also to analyse the attacks if the malicious insider is successful in exploiting the vulnerabilities in the tenant virtual machines. However monitoring user activity may not be effective against malicious insiders who have joined recently and who do not have much history. Hence TSAD detects the attacks byTSAD maintains logs of all activities of users and processes within the system. For example, the execve system call corre-sponds to the commands typed by the user. The user activity is extracted by filtering the execve system calls and relating the activity based on the timestamps and events such as login and chdir

Insider Attacks from Cloud Service Provider

In this paper, as mentioned earlier, we consider the Cloud service provider to be a trusted entity. However there can be cloud system administrators involved in the management of the cloud infrastructure performing a range of activities such as software development, support, testing, and system and network administration. These system administrators will need different levels of access to the resources in the cloud to perform their tasks. However the privileged domains in the current VMMs do not support fine granular access control for the cloud administrators. An important principle in the design of secure systems is the notion of least privilege; that is, system administrators will need to have only those privileges that are needed for the tasks at hand.

Denial of Service Attacks

Not only a tenant machine can be subjected to a denial of service attack, but also the compromised tenant virtual machine can be used by the attacker to generate further attack traffic. For instance, a tenant virtual machine can be used to flood the victims with attack traffic such as ICMP, UDP and TCP SYN floods.

Also the attackers can make use of the elastic nature in the cloud to allocate more resources to the compromised tenant virtual machines and use them in the generation of the attack traffic. This could lead to serious attacks in the Internet. Hence there is a need for techniques to minimise denial of service attacks originating from tenant virtual machine

IV. ANALYSIS

In this section, we discuss the implementation and analysis of our security architecture. Section A presents the imple-mentation setup. Section B describes the implementation of our security architecture. Section C discusses performance evaluation results.

Implementation Setup

We have used the open source based system Xen hypervisor to implement our architecture. However it is to be noted that our security architecture can be implemented using other VMM based systems such as VMWare or HyperV. Fig. 6 shows the basic implementation of our security archi-tecture at a single VMM platform level using Xen hypervisor. A tenant hosts its services on virtual machines that are running on Xen hypervisor, which belongs to the cloud provider. We have used different subnets for the cloud provider network, tenant domain, the tenant users (who are the customers of the tenant), and the attack domain.The tenant admin manages the tenant virtual machines that are hosted in the cloud. The tenant users are the customers of the tenant. RIP protocol is used between the routers R1 and R2. The SPAD and TSAD functionalities are implemented in the privileged domain Dom0. The attacking sources in the attack domain generate the different types of attack traffic on the tenant virtual machine that we have experimented with.

First we present an overview of Xen that is relevant for our architecture and then discuss the implementation of the components of our security architecture. We then discuss how our architecture deals with different attack scenarios described above in Section III.

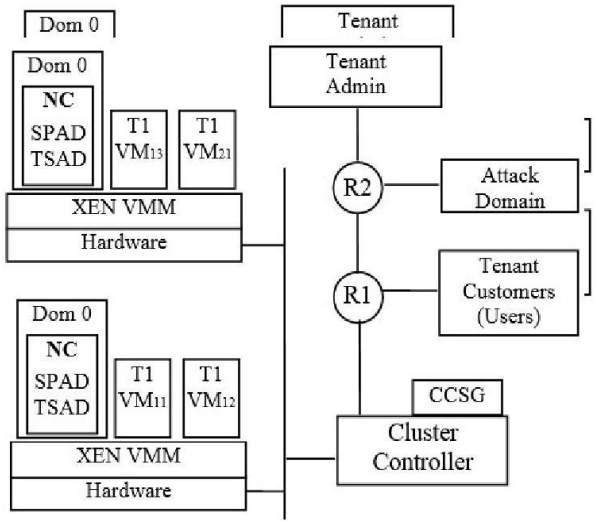

Fig. 6. Implementation setup.

V. RELATED WORK

During the course of the description of the security architecture, we have already referenced many related works. In this section we consider additional relevant related works and compare them with our architecture.

CloudVisor [24] uses nested virtualisation to deal with the compromise of the hypervisor. In this technique a secure hypervisor is introduced below the traditional hypervisor and the interactions between the traditional VMM and virtual machines are monitored by the secure hypervisor. However since the resource management is still performed by the traditional VMM, the compromise of VMM can impact the operation of the virtual machines. Compared to CloudVisor the main focus of our work is securing the network interactions of tenant virtual machines. The technique proposed in [25] allocates a separate privileged domain for each tenant. The tenants can use this for the enforcement of VMM based security on their virtual machines. However the model can become more complex as different tenant virtual machines can be hosted on the same physical server. Furthermore, such models cannot deal with the case of malicious tenants that misuse the cloud resources to generate attacks on other hosts. Our architecture considers the case of malicious cloud administrators and malicious tenants.

There have also been some prior works addressing privacy related issues in the cloud. Butt et al [26] proposed self service cloud which splits the privileged domain into system wide domain (Sdom0) and privileged client domains. Each tenant has their own privileged domain for enforcement of security policies on their virtual machines. However, since several tenant virtual machines can be implemented on the same physical server, a separate client administrative domain has to be created for each tenant. This makes the model considerably complex. Furthermore, an attacker who has control of Sdom0 can cause resource starvation to tenant virtual machines. In our architecture, we enforce different level of access to the cloud administrators using role based access control.

The techniques proposed in [27] consider making the cloud services scalable to the dynamic changes in the runtime environment. In the proposed architecture, the cloud service provider monitors the load (active connections) on the tenant web server and dynamically varies the number of virtual machines allocated to the tenant. Attacks can lead to increase of load on the tenant virtual machines. Our architecture is able to identify the increase in load caused by the attack traffic. There have also been some prior works which make use of cloud for securing the traditional systems and networks. For instance, Beaty et al [28] proposed network based access control for different cloud deployments. In this technique, the administrators from different cloud deployments report to a Cloud Access Manager on their requirements. However, our model also detects system level attacks such as rootkits.

VI. CONCLUSION

In this paper we have proposed a security architecture that provides a security as a service model that a cloud provider can offer to its multiple tenants and customers of its tenants. Our security as a service model while offering a baseline security to the provider to protect its own cloud infrastructure also provides flexibility to tenants to have additional security functionalities that suit their security requirements. The paper described the design of the security architecture and discussed how different types of attacks are counteracted by the proposed architecture. We have described the implementation of the security architecture and gave a detailed analysis of the security mechanisms and performance evaluation results.

APPENDIX

Policy Specification Examples: Several policy languages have been proposed over the years. We have chosen XACML [36] as it has the necessary constructs to specify the types of policies even though it is generally regarded as an access control policy language. Also it is an international standard, and XML based schemas allow verification of the structure of the policy file; we have also developed an evaluation engine. Furthermore, Intrusion Detection Message Exchange Format (IDMEF) [37] which defines data formats and exchange procedures for sharing information of interest to intrusion detection and response systems also makes use of XML. Hence we make use of the XACML since the tenants and cloud service providers can be using different security tools for the enforcement of the security policies. Below we give examples showing the use of such a language; Fig. 13 shows an example of security policy specification, whereas Fig. 14 shows how information such as alerts can be exchanged between different security agents implemented in different physical servers. the case for signature based detection of Slammer attack and anomaly based threshold for ICMP traffic to 3 messages per second. The alert message ID refers to the attack. The analyser identifies the cluster ID and VMM within the cluster. There are 12 defined categories (such as DNS for domain name system, NT for windows domain) defined by the IDMEF[37]. In this case, the cloud provider does not have to be aware of the services in the tenant virtual machine. Hence we use the category 0 which refers to “Domain unknown or not relevant.” Name identifies the specific sensor “SPAD” that detected the attack.

REFERENCES

[1] L. Youseff, M. Butrico, and D. Da Silva, “Towards a unified ontology of cloud computing,” in *Proc. 2008 Grid Computing Environments Workshop.*

[2] Amazon Inc., “Amazon elastic compute cloud (Amazon EC2),” 2011. Available: http://aws.amazon.com/ec2/

[3] “Windows Azure.” Available: http://www.windowsazure.com/en-us/

[4] J. E. Smith and R. Nair, “The architecture of virtual machines,” *IEEE Internet Comput.,* May 2005.

[5] “AWS security center.” Available: http://aws.amazon.com/security/ *Symp.*

[6] T. Garfinkel and M. Rosenblum, “A virtual machine introspection based architecture for intrusion detection,” in *Proc. 2003 Netw. Distrib. Syst. Security Symp*