An Architecture for Providing Security to Cloud Resources

Niranjana Padmanabhan
Lecturer
CS Department
S.N College of Engineering and Technology, India

Bijolin Edwin E
Lecturer
IT Department
Karunya University, India

ABSTRACT
One of the major challenges in Cloud computing is providing security to the cloud resources. In present paper, we make use of the concept of virtualization to protect the cloud components and the integrity of guest virtual machines. To guarantee increased security to cloud resources, an architecture called Cloud Protection System (CPS) is proposed. CPS remains fully transparent to the cloud components and the guest virtual machines since it is implemented on the base machine and monitors the integrity of guest virtual machines. Also, we propose an architecture called HypeSec, which can be integrated in the hypervisor Qemu, where it controls all inter-VM communication according to formal security policies. The architecture CPS is fully implemented using Eucalyptus cloud environment, and Qemu as the hypervisor. The effectiveness of the prototype is shown by testing it against the Sebek rootkit attack.

General Terms
Cloud computing, virtual machine, security.

Keywords
Eucalyptus, Hypervisor, Qemu, virtualization.

1. INTRODUCTION
Cloud computing is a consequence of economic, commercial, cultural and technological conditions that have combined to cause a disruptive shift in information technology towards a service-based economy. The underlying driver of this change is the commoditization of IT. Even though unresolved security and privacy issues are slowing down their adoption and success, cloud nodes are increasingly popular. Since the cloud nodes are exposed to third parties of services and interfaces, they are more vulnerable to cyber attacks. The cloud in fact is the internet, with all the positives and negatives of it. Hence providing security to the cloud is a challenging task. Thus it is crucial to identify the possible threats that could occur and to establish security processes to protect the cloud from attacks. Virtual Machines (VM's) on the Internet are exposed to many kinds of interactions that virtualization technology can help filtering while assuring a higher degree of security. To provide monitoring of VMs, allowing easier management of the security of complex cluster, server farms, and cloud computing infrastructures, virtualization can also be used as a security component. However, with respect to security, the virtualization technologies create new potential concerns. It will be difficult to maintain the consistency of security and ensure that records can be audited due to the dynamic and fluid nature of virtual machines. Also, the co-location of multiple virtual machines increases the attack surface and risk of virtual machine-to-virtual machine compromise. The main focus of this paper is on the controlled sharing of resources. Such sharing is not controlled by any formal policy in current hypervisor systems. This lack of formality makes it difficult to reason about the effectiveness of isolation between VMs. In the following sections we show how CPS can leverage full virtualization to provide increased protection to actually deployed cloud systems such as Eucalyptus. Also, HypeSec architecture is integrated in the hypervisor Qemu, where it controls all inter-VM communication according to formal security policies.

2. RELATED WORK
The survey on cloud computing presented in Armbrust et al. (2009) have been the starting point of our work. There are many research papers on integrity checking mechanisms and intrusion detection solutions. Those mechanisms can be successfully applied to cloud computing as well. For example, the Filesystem Integrity Tools and Intrusion Detection Systems such as Tripwire (Kim and Spafford, 1994) and AIDE (AIDE team, 2005) can be deployed in virtual machines. But they are subject to attacks possibly coming from a guest machine user who has turned the machine into a malicious one. In addition to this, when an attacker finds out that the target machine is in a virtual environment, it may attempt to break out of the virtual environment through vulnerabilities (Secunia, 2009) in the Virtual Machine Monitor (VMM). Most of the approaches present today, leverage VMM isolation properties to secure VMs by leveraging various levels of virtual introspection. Virtual introspection (Jiang et al., 2007) is a process in which the VMM monitors the state of a VM. SecVisor (Seshadri et al., 2007) Lares (Payne et al., 2008) and KVM-L4 (Peter et al., 2009), to name a few, leverage virtualization to monitor the integrity of the guest kernel code from a privileged virtual machine or from the Virtual Machine Monitor, also known as the hypervisor. Finally, the paper called Transparent security for cloud (Lombardi and Di Pietro, 2010) was studied to know what are the security measures taken for protecting the integrity of the virtual machines in the cloud. It can be seen that this paper and our work share some similarity in terms of positioning of the protection components. In Transparent security for cloud, the authors have considered the case in which security is provided in an environment where there is only a single virtual machine that has gone malicious. Furthermore in Secure virtualization for cloud computing (Lombardi and Di Pietro, June 2010), again by

the same authors considers the scenario where there are two virtual machines connected to a host and one of it has turned out to be malicious. But the chance of a compromised virtual machine that has become malicious affecting the next virtual machine (due to the dynamic and fluid nature of virtual machines) is vaguely explained. Our work shows that by providing an access control feature in the VMM, we can safe guard a virtual machine from being infected by a malicious one.

3. BACKGROUND

Cloud computing is Internet-based computing, whereby shared resources, software, and information are provided to computers and other devices on demand. In cloud computing, details are abstracted from the users, who no longer have need for expertise in, or control over, the technology infrastructure "in the cloud" that supports them. Cloud computing typically involves over-the-Internet provision of dynamically scalable and often virtualized resources. In general, cloud computing customers rent usage from a third-party provider thus avoiding capital expenditure. The cloud customers consume resources as a service and pay only for the resources that they use. We can say that cloud computing is a form of utility computing wherein the customers are charged according to the amount of resources that they use. The utilization rates are improved by sharing "perishable and intangible" computing power among multiple tenants, as servers are not unnecessarily left idle (which can reduce costs significantly while increasing the speed of application development). But a side-effect of this approach is that the overall computer usage rises dramatically, as customers do not have to engineer for peak load limits. In addition, it is possible to receive the same response times from centralized infrastructure at other sites due to "increased high-speed bandwidth". The concepts such as virtualization, distributed computing and utility computing are applied within the cloud paradigm. Cloud services are available at different layers like Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). The traditional way of software distribution, wherein the software is purchased for and installed on personal computers, is sometimes referred to as Software-as-a-Product. Software-as-a-Service is a term given to the software distribution model in which the applications are hosted by a vendor or service provider and made available to customers over a network, typically the Internet. As the underlying technologies that support web services and service-oriented architecture (SOA) mature and new developmental approaches become popular, SaaS is becoming an increasingly prevalent delivery model. SaaS is also often associated with a pay-as-you-go subscription licensing model. The next cloud service known as the Platform-as-a-Service allows us to include platforms for building and running custom web-based application. It is an outgrowth of the SaaS application delivery model. We can say that the PaaS model makes all of the facilities required to support the complete life cycle of building and delivering web applications and services entirely available from the Internet, without any software downloads or installation for developers, IT managers, or end users. In the IaaS model, the developers can/may create a specific operating system instance with home grown applications running. Unlike this model, PaaS developers are concerned only with web based development and generally do not care what operating system is used. Rather than complex infrastructure, PaaS services allow users to focus on innovation. IaaS can be defined as the delivery of computer infrastructure (typically a platform virtualization environment) as a service. IaaS leverages services, significant technology and data center investments to deliver IT as a service to customers. Unlike traditional outsourcing, which requires complex, lengthy contract vehicles, extensive due diligence, negotiations ad infinitum, IaaS is centered around a model of service delivery that provisions a predefined, standardized infrastructure specifically optimized for the customer's applications. IaaS customers essentially rent the resources as a fully outsourced service rather than purchasing data center space, servers, software, network equipment, etc. Usually, the service is billed on a monthly basis, just like a utility company bills customers. The customer is charged based on the amount of resources he has consumed. In our work, we have focused mainly on the 'lowest'' computational layer (i.e. IaaS) because we can more effectively provide a security foundation on top of which more secure services can be offered. Out of the cloud computing systems that exist today, most of them are proprietary (even though APIs are open and well known) and they do not allow integration with other systems or any kind of enhancements for research purpose. Because of this is reason, we have chosen Eucalyptus.

4. CLOUD PROTECTION SYSTEM

In the proposed Cloud Protection System (CPS), the guest virtual machine is monitored by the host to ensure that the integrity of the virtual machine is protected. We mainly monitor the kernel code or data that would be targeted (or) affected by attacks to provide protection to the virtual machines and the cloud infrastructure. Thus any modification to the kernel code and data is detected by monitoring the cloud components and the kernel (of virtual machine). This monitoring guarantees that the integrity of the virtual machine kernel and the cloud middleware have not been compromised. Now how we monitor the integrity of cloud components is by logging in and verifying the checksum of cloud libraries and executable files periodically. The high level description of CPS is shown in Figure 1. The monitoring data flows are depicted as continuous lines in green color where as the dangerous data flows are shown as dashed lines (red). All the CPS modules- the Interceptor, Warning Recorder, Warning Queue and the Evaluator are located on the base machine (host). The Interceptor component notices any suspicious guest activities like for example, system_call invocation and it is recorded by the Warning Recorder into the Warning Queue (WQ). Then the threat will be evaluated by the Evaluator component. Our protection system called CPS is implemented over Eucalyptus cloud environment. Eucalyptus (Nurmi etal.,2009) consists of: a Node Controller (NC) that controls the execution, inspection, and termination of VM instances on the host where it runs; a Cluster Controller (CC) that gathers information about VM and schedules VM execution on specific node controllers; further, it manages virtual instance networks; a Storage Controller (SC)—Walrus—that is, a storage service providing a mechanism for storing an accessing VM images and user data; a Cloud Controller (CLC), the web services entry point for users and administrators that make high level scheduling decisions. The NC runs on every node hosting VM instances. The NC activity and integrity is mainly monitored, as it is the key component for our cloud implementation.

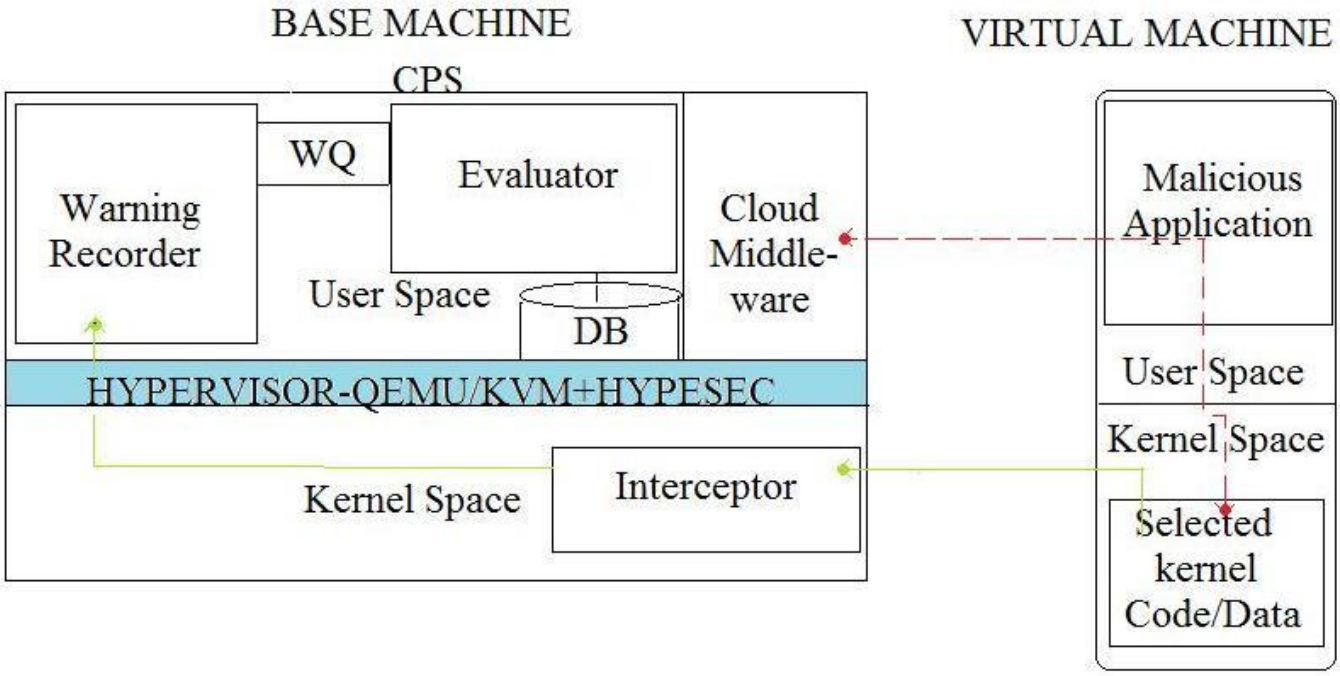

Fig 1: Cloud Protection System

Now, if any dangerous alteration in the guest VM is detected, CPS can take actions like shutting down the VM or restarting a clean image. An attack can be implemented by inserting a rootkit in the guest VM. For instance we can insert Sebek, which is a kernel module that hides its presence and intercepts file system and network activity. It alters the syscall table and changes the execution flow to execute any malicious code. CPS can detect both the alteration of the syscall table and the change in the checksum of kernel files on virtual storage. Now if there are many virtual machines installed in a single system and one virtual machine has gone malicious, then it will affect the remaining virtual machines in no time. That is, the co-location of multiple virtual machines increases the attack surface and risk of virtual machine-to-virtual machine compromise. Hence along with CPS, we can include a security feature in the hypervisor Qemu to provide better protection. We kept the name as HypeSec since we add the security feature in the hypervisor. The block diagram of HypeSec is depicted in Figure 2.

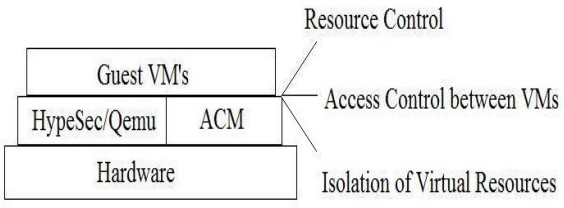

Fig 2: Block diagram of HypeSec

This architecture named HypeSec can have an Access Control Module (ACM) incorporated with the Qemu hypervisor which will exercise access control between VMs, isolation of virtual resources, resource control etc. The ACM authorizes access of VMs to resources based on certain policy rules attached to VMs. One policy can be like administrators must ensure that certain VMs (and their supported workload types) cannot run on the same hypervisor system at the same time. Based on such policies, the ACM can decide on whether to allow communication among virtual machines or not. This feature enhances the security provided by the CPS.

5. ATTACK IMPLEMENTATION

First we checked how our protection system reacted by making a single virtual machine malicious. We did so by inserting a module inside the kernel of the virtual machine which altered its syscall table and changed the execution flow so as to execute the malicious code. This alteration was identified by the CPS components at the base system by the change in the value of checksum generated after the syscall table was altered. After detecting the alteration, the virtual machine was made to be powered off. When this was found successful, we next made some additions to the Qemu code so that the Qemu hypervisor will exercise some access control method when there is more than one virtual machine on the base system and one among those is compromised. That is, once it is found out that a virtual machine is malicious, the Qemu hypervisor will change the access rights of the compromised virtual machine in such a way that it cannot communicate with the other VMs thus avoiding

the attack on the non malicious VMs. Then the compromised virtual machine was made to be powered off in order to avoid it from affecting the critical kernel code or data of the base system. This was done after the CPS components detected the change in the syscall table checksum of the malicious VM. There is of course a small amount of overhead introduced by this technique but compared with the detection capability of our system, it can be neglected.

6. CONCLUSION

In this paper, we have introduced an architecture named Cloud Protection System that can provide security to the cloud resources via virtualization. CPS monitors the guest and the middleware components and ensures that the integrity has not been compromised. To enhance the security provided, HypeSec architecture is proposed which is integrated along with the hypervisor Qemu. CPS combined with HypeSec can be deployed on any cloud implementation. Our protection system ensures that the integrity of the virtual machines is not compromised.

7. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards the development of this paper.

8. REFERENCES

[1] Armbrust M, Fox A, Griffith R. Above the clouds: A Berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, February 2009.

[2] Bellard F. Qemu, a fast and portable dynamic translator. In ATEC ’05: Proceedings of the annual conference on USENIX annual technical conference, Berkeley, CA, USA, 2005. USENIX Association, p. 41.

[3] Seshadri A, Luk M, Qu N, Perrig A. Secvisor: a tiny hypervisor to provide life time kernel code integrity for commodity oses. In SOSP’07: Proceedings of twenty first ACM SIGOPS symposium on operating systems principles, ACM, New York, NY, USA, 2007. p. 335–50.

[4] Payne BD, Carbone M, Sharif M, Lee W. Lares: An architecture for secure active monitoring using virtualization. In SP ’08: Proceedings of the 2008 IEEE symposium on security and privacy (sp2008), IEEE Computer Society, Washington, DC, USA, 2008. pp. 233-47.

[5] Lombardi F, Di Pietro R. Kvmsec: a security extension for linux kernel virtual machines. In SAC ’09: Proceedings of the 2009 ACM symposium on applied Computing, ACM, New York, NY, USA, 2009. pp. 2029–34.

[6] Qumranet. Linux kernel virtual machine. http://kvm.qumranet.com.

[7] Peter M, Schild H, Lackorzynski A, Warg A. Virtual machines jailed: virtualization in systems with small trusted computing bases. In VDTS ’09: Proceedings of the 1st EuroSys Workshop on virtualization technology for dependable systems, ACM, New York, NY, USA, 2009. p.18–23.

[8] Rhee J, Riley R, Xu D, Jiang X. Defeating dynamic data kernel rootkit attacks via vmm-based guest transparent monitoring. Availability, Reliability and Security, 2009. ARES ’09.

[9] Lombardi F, Di Pietro R. Transparent security for cloud. In SAC’10: Proceedings of the 2010 ACM symposium on applied computing.

[10] Lombardi F, Di Pietro R. Secure virtualization for cloud computing. In Elsevier, June 2010: Journal of Network and Computer Applications.