ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 8, October 2012

Survey on virtual machine security

Bright Prabahar P
Post Graduate Scholar
Karunya university

Bijolin Edwin E
Assistant professor
Karunya university

***Abstract*— Virtualization takes a major role in cloud environment. It is the creation of a virtual (rather than actual) version of something, such as a hardware platform, operating system (OS), storage device, or network resources This paper deals with various virtual machine security mechanism in cloud environment where we can access virtual hard disk, virtual servers in pay as you go manner. It also explains about kernel attack, critical problem in virtualization and how the security architectures are used to solve that issue are explained in this paper. It is the survey of various cloud virtual machine architectures, advantages & disadvantages.**

***Keywords*— virtual machine security, kernel attacks**

1. INTRODUCTION

Cloud computing is a process of storing in the remote location hence it can be accessed by any device which is connected with the internet at anywhere at any time. This concept came in early 1961, there is a Professor named John McCarthy who suggested there is a computer time-sharing technology which lead to a future where computing power and even specific applications is sold through a utility based business model. The term utility based means pay per use. This idea became most popular in the late 1960s, but in 1970s the idea faded away when it became clear that the IT-related technologies of the day were unable to sustain such computing model

Some of the cloud service providers are Google, Amazon [1] etc.., cloud provides three services they are SAAS, PAAS, and IAAS. SAAS means software as a service where we can rent softwares as per our use some of the examples are Google drive, Microsoft's skydrive etc.., in that examples they provide office softwares which are used to create, edit documents, spread sheets, presentation etc.., these office softwares are give free of cost but some of the softwares like Photoshop, academic softwares are given on the basis of pay per use in cloud environment. In order to cloud you need a browser with basic functionalities. Through the internet we are accessing higher end machine which is capable of running higher softwares not only softwares we can run operating system also through Amazon's web services[1]. Google's Chromebook[3] is a best example of cloud computing where it consists of hardware which is capable of running a browser (chrome) initially it is suffered from kernel attacks which are discussed in coming sections.

PAAS means Platform as a Service which is used to get operating system as a service. Microsoft's windows azure is an example for this type.In IAAS they are providing infrastructure as a service some of the examples are Virtual machines[9], Virtual servers[9], Storage, Load balancing etc.., are comes under this.

Hypervisors are called virtual machine managers or virtual machine monitors which are used to run multiple operating systems in a same host.Hypervisors are of two types.First type of hypervisor runs directly on the hardware hence we can run multiple virtual machines[9] on it. Some of the examples of type 1 hypervisors are Citrix XenServer, VMware, and Microsoft's hyper-V hypervisor. In type 2 hypervisor runs above the operating system some of the examples of type 2 hypervisors are VMware workstation, virtual box etc..,

2. SECURITY ARCHITECTURES IN CLOUD ENVIRONMENT

There are lots of security mechanism are present apart from that there are six major security architecture are commonly used in virtualization which are HIMA[2], KVMSEC[3], LARE[4], XENaccess[5] and Vmscope[6] which are explained in comming sections.

A. HIMA

HIMA measures the integrity of Virtual Machines (VMs) running on top of the hypervisor, which provides both capabilities identified above. HIMA[2] performs two complementary tasks: (1) active monitoring of critical guest events and (2) guest memory protection. It guarantees that the integrity measures are refreshed whenever the guest VM memory layout changes (e.g., upon the creation of a processes), while the latter it ensures that the integrity measurement of user programs cannot be bypassed without HIMA's knowledge

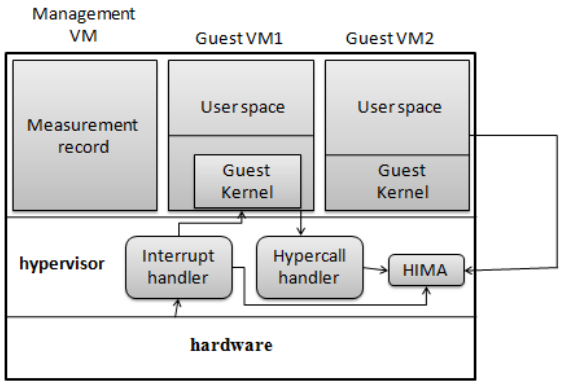

Fig.1 HIMA architecture

Looking inside the hypervisor, HIMA is properly isolated from the targets which are measured previously. It generates and maintains the certain hashes of the code segments of all kernel components and the user programs will run inside the guest VMs. below Figure shows the architecture of HIMA. HIMA places hooks inside the hypervisor's code which handles guest VMs' privileged events. Thus the HIMA intercepts all these events, which are hypervisor service requests (e.g., hypercalls, VMExit), system calls and hardware interrupts. The created measurement lists can be stored in the management VM for the validation. HIMA acts transparent to guest VMs. It only relies on the escalated privileges available to generic hypervisors, and it can be implemented (or ported) to any hypervisor platform. In this architecture they implemented a prototype of HIMA on Xen in X86/64 architecture. The current implementation is customized to measure [8]para-virtualized Linux guests. It can also be easily modified to support other guest systems (e.g., fullyvirtualized guests). HIMA utilizes the hypervisor's control on guest VMs by initiation & measuring their booted kernels and initial loadable modules. Beyond guest VMs booting, HIMA performs two major tasks which are used to measure the guest VMs' integrity and consistency of measured user programs: (1) guest events monitoring (2) memory protection in runtime.

Advantages:

- Dedicated management VM is present only to measure the events
- It can monitor the active events and also it can also protect guest memory

Disadvantages:

- It doesn't handle app writable and executable on memory pages (java)
- Monitoring of netapp is not possible
- In this architecture we use xen hypervisor hence driver problem will arises in case of updation

B. KvmSec: A Security Extension for Linux Kernel Virtual Machines:

The ultimate aim of this architecture (KvmSec)[4] is to increase the security of guest virtual machines. KvmSec can protect guest virtual machines against malicious attacks such as viruses and kernel rootkits. KvmSec enjoys the following features: it has transparent guest machines architecture; it is hard to access from a compromised virtual machine; it can collect data, analyze them, and also act consequently on guest machines. It can also provide secure communication between each of the guests and the host; and, it can be implemented in Linux hosts and at present supports Linux guest machines. These features are used to implement a real-time monitoring and security management system.

The most popular open source virtualization architectures: Xen and KVM[10], It uses Full-virt technology that exploits CPU virtualization support (AMD-V and Intel-VT). CPUs supporting such a new technology that feature ringlevel that allows the operating systems in guest VMs to run unmodified and unaware of the existence of an additional higher privileged ringlevel. The other approach to virtualize is [8]para-virt. Such approach does not exploit CPU

virtualization support and usually requires changes to guest operating systems

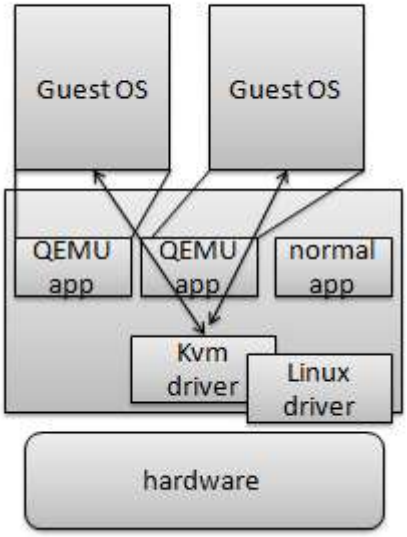

Fig.2 KVMSEC architecture

.

Xen is the most widely adopted virtualization solution in research, it is composed of The Xen hypervisor (or VMM), a privileged VM (Dom0) and a common VM (DomU) where the guest OS is executed. Xen features: 1) Shared Memory concept: a communication channel between VMs; 2) Ether Channel concept: a signaling channel between VMs; 3) Shared Memory Access Control concept: an access matrix describing which VM can access shared memory. Xen drivers are conned at the privileged Dom0 domain. All I/O requests by the various domains are performed by Dom0 and it is the hypervisor responsibility to switch from DomU to Dom0 when a I/O operation is called.

The new mainstream Linux virtualization solution is KVM, it is the part of the logical Linux kernel source since version 2.6.20. KVM[10] consists of a hypervisor (a Linux kernel module) and a modified version of Qemu emulation software. KVM is a standard kernel module, as a consequence it exploits the standard, it is reliable and frequently updated Linux device drivers. This is main reasons why KVM is less vulnerable to attacks than Xen, whose driver development is slower when compared to standard Linux. On the other hand, kernel codebase is overall larger than when compared to Xen and it can potentially contain more vulnerabilities and bugs. It appears that KVM[10], while not being fully mature yet, The main advantages over Xen is it has wider hardware support and increased exibility (e.g the redeployment of a newer KVM version does not require a host reboot).

Advantages:

- It can collect data and react on guest machines but its core resides in the protected host machine
- It can be deployed on most x86 and x86 64 machines

Disadvantage:

- The overall performance of the system is low

C. LARE

LARE[5] architecture takes a hybrid approach, giving security tools the ability to do all active monitoring while still benefiting from the increased security of an isolated virtual machine. This architecture discuss a prototype implementation that can process hooks from a virtual machine running Windows XP on Xen hypervisor, this security analysed and shows the performance of a single hook is 28 μsecs.

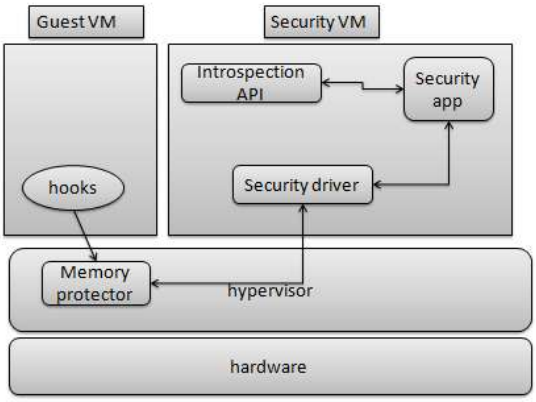

Fig..3 LARE

There is a fundamental difficulty in case of protection and flexibility in hook placement. Flexibility allows the hooks to be placed anywhere inside the untrusted system, which is in a traditional scenario would make them prone to tampering by intruders with system-wide privileges. In order to solve this conflict is Lares[5] came. At a high-level, it does this by splitting the security application into two Virtual Machines and using a special memory protection mechanism to provide integrity solution to hooks. As shown in Fig Lares has two VMs: the untrusted guest VM and a security VM that is part of our TCB (Trust Computing Base).

Since the guest Virtual Machine is untrusted, software placed inside it requires special protection. It is difficult to achieve if the components are too large or too integrated with the surrounding OS, so we use them in minimum required. These include the hooks for intercepting events, and a small specially-crafted trampoline which pass the code events signaled by the hooks to the hypervisor. These components has the capability of self-contained and simple enough that write-protect their memory footprint is sufficient to guarantee their correct behavior. It is a special mechanism of the hypervisor which provide these memory protections, along with an inter-VM communication functionality used for event passing.

The security VM contains the core which full of the active monitoring application, where the processing and decision making associated with its functions are done. Certain techniques like memory and disk introspection can be used as part of this decision making to gather the additional information about the events sent from the hooks in the guest VM. After a decision is made, the security application sends information back to the guest VM, where the decision is enforced.

As an example scenario, an anti-virus application is placed its signature matching and containment algorithms in the security VM, whereas the monitoring hooks would go into the guest VM. These hooks will trigger when certain monitored events were executed by the guest OS and it is transmitted to the security VM by the trampoline with the help of the hypervisor. The anti-virus' core engine will receive these events and use introspection to enrich them with right information, which will then processed by its signature matching algorithms and heuristics. After taking a decision, it will sent back to the guest VM's trampoline, where a response measure is carried out, such as preventing a process from loading a file or being written to disk.

Advantage:

- Real-time attack prevention by using host based intrusion detection systems called hook

Disadvantages:

- Security applications into an in isolated VM, these architectures do not support active monitoring.
- Security mechanism impact on system performance

D. XENACCESS

A monitoring technique which is known as introspection, but these prior works is focused only on the applications. This architecture is all about design of XenAccess[6], a monitoring library for operating systems running on Xen. XenAccess incorporates virtual memory introspection and virtual disk monitoring, it allows to monitor applications safety and efficiently access of the memory state and disk activity of a target operating system.

Xen uses a paravirtualized[8] approach to apply virtualization. This technique consists of altering the guest OSes by replacing susceptible instructions that cannot be virtualized with special hypercalls, that is, calls that are made directly to the VMM. This approach has an advantage only if good performance is good, since no trapping is done, and also allowing virtual machines to run on top of non-virtualizable architectures (such as x86). Yet the drawback of paravirtualization is that the guest OSes can be modified. Recent versions of Xen have the ability to rununmodified OSes by using the new Intel VT-x and AMD-V technologies. XenAccess uses paravirtualized domains without violate the property because the changes necessary for paravirtualization are not rigorously a part of the XenAccess architecture and do not make it any easier for the target OS to interfere with the monitoring code.

Xen divides the domain architecture, meaning that regular guest OSes are kept in unprivileged domains (domU), whereas a single administrative domain exists as Domain 0 (dom0). Dom0 can be seen as a domain-level of Xen where all the management functionalities are located. It has complete access to all virtual machines being run and also which can works as a device driver proxy for domU's virtual devices. The VMM itself is a simple and thin software layer whose main job is to assurance proper isolation between virtual machines, performing minimal resource management. This isolation is quite strong, since Xen relies directly on hardware-level protection mechanism and has a much narrow interface than a regular operation system (e.g., Linux)

Advantage:

- Monitoring of Virtual machine & memory

Disadvantages:

- As it is Para-virtualization it is suffered from kernel attacks
- Lac of drivers

E. VMSCOPE

This architecture is all about virtualization-based monitoring system called VMscope[7] which gives us the same deep inspection ability as existing internal monitoring tools (e.g. Sebek) while being as transparent and tamper-resistant as existing external monitoring tools (e.g. a network sniffer). By deploying itself totally outside the VMbased honeypot, VMscope is tamper-resistant and transparent to the monitored system. Further, without require any modification to the monitored system, VMscope runs at the virtual machine monitor (VMM) layer and is able to observe, recording, and understanding the parameters and semantics of various VM-internal system events such as including a system calls. which gives us the same monitoring ability as existing internal sensors even though we do not have any sensors inside. As an example, once a *sys read* system call of a VM is noted, VMscope will study from outside the VM the corresponding system call parameters and it understand which is being opened for this read operation and what will be the return value or content after the system call is completed also these semantic-level information will be aggregated and stored outside the vulnerable honeypot system, which gives us better tamper-resistance than other usual approaches.

More specifically, it enables .out-of-the-box. Monitoring, VMscope uses an extends one key software-based virtualization technique called *binary translation* serve, interpret, and record involved VM events at runtime. Note there exists another similar virtualization technique called *para-virtualization* (implemented in Xen and User Mode Linux),The reasons are: (1) Binary translation allows us to clearly support legacy OSes in VMs without any alteration on the guest OSes while para-virtualization requires modification and recompiling of the guest OSes. Such a alteration of the VM-based honeypot not only violates the transparency condition but also introduces the risk of being detected and subverted; (2) Para-virtualization requires the access and alteration of guest OS source code, which could significantly limit our choices of deploy commodity (commercial) OSes as honeypots. It point out that this design choice differentiate our approach from earlier Xen.

To show the feasibility of .out-of-the-box. monitoring, they have implemented a proof-of-concept prototype based on an open-source binary translation-capable VMM prototype called QEMU[10]. Our experimental results with real-world honeypot deployment as well as the association with the de-facto honeypot monitoring tool (i.e., Sebek) show that VMscope can achieve the same deep inspection ability as internal monitoring tools while, at the same time, being transparent and tamper-resistant against advanced attacks (e.g., NoSEBrEaK).

Advantages:

- Active monitoring of virtual machines

Disadvantages:

- Attackers can compromise VMscope with TCB(trust computing base)
- Attackers get access to change interrupt table leads to kernel attack

Table1.Comparison of various cloud security architectures

	objective	Virtualization type	Advvantages	Dis-advantages
HIMA	A good solution to integrity measurement has to provide both strong isolation between the measurement agent and the measurement target and Time of Check to Time of Use (TOCTTOU) consistency HIMA performs two complementary tasks: (1) active monitoring of critical guest events and (2) guest memory protection	Full virtualization	Management virtual machine Prevent kernel attacks because of full virtualization	Doesn't handle app writable and executable on memory pages (java) Netapp monitoring is not present
Xen access (Domain 0) (Domain 1)	XenAccess incorporates virtual memory introspection and virtual disk monitoring capabilities, allowing monitor applications to safely and efficiently access the memory state and disk activity of a target operating system	Para virtualization	Monitoring of VM & memory	As it is para-virtualization it is suffered from kernel attacks Lack of drivers
Lare	An architecture that takes a hybrid approach, giving security tools the ability to do active monitoring while still benefiting from the increased security of an isolated virtual machine. this architecture and a prototype implementation that can process hooks from a virtual machine running Windows XP on Xen.	Full virtualization	Real-time attack prevention host based intrusion detection systems called hook	Security applications into an in isolated VM, it does not support active monitoring. Security mechanism impact on system performance
VMscope	Virtualization-based honeypot monitoring system that is capable of inspecting and interpreting system internal events from outside the VMbased honeypot	Full virtualization	Active monitoring	Attackers compromise VMscope with TCB(trust computing base) Attackers get access to change interrupt table leads to kernel attack

KvmSec	It protect guest virtual machines against attacks such as viruses and kernel rootkit, it can collect data, analyze them, and act consequently on guest machines, it can provide secure communication between each of the guests and the host	Full virtualization	It can collect data and react on guest machines but its core resides in the protected host machine It can be deployed on most x86 and x86 64 machines	System performance is low

3. CONCLUSION:

In Cloud computing virtual machines, virtual networks take major role.We have studied various hypervisors and virtual machine architectures and its advantages and disadvantages.A comparitive study about the various hypervisors hence this survey ensures the security mechanism in cloud environment. We have discussed the various kernel attacks and how these architectures overcome such vulnerabilities and how the para-virtualization and full-virtualizations helps to prevent the kernel attacks are also discussed. The main contribution of this paper is to understand the various Virtual machine security mechanisms in cloud.

4. REFRENCES:

[1] Amazon web services http://aws.amazon.com/documentation/

[2] Ahmed M. Azab, Peng Ning, Emre C. Sezer, Xiaolan Zhang, HIMA: a hypervisorbased integrity measurement agent, in: Proceedings of the 25th Annual Computer Security Applications Conference, ACSAC'09, December 2009,Honolulu, Hawaii, USA

[3] chrome Operating System: chromebook http://www.google.com/intl/en/chrome/devices/

[4] L. Flavio, P. Roberto Di, KvmSec: a security extension for Linux kernel virtual machines, in: Proceedings of the 2009 ACM Symposium on Applied Computing, ACM, Honolulu, Hawaii, 2009

[5] B.D. Payne, M. Carbone, M.I.S.W. Lee, Lares: an architecture for secure active monitoring using virtualization, Presented at IEEE Symposium on Security and Privacy, S&P 2008, Oakland, California, USA, 2008

[6] B.D. Payne, W. Lee, Secure and flexible monitoring of virtual machines, Presented at 23rd Annual Computer Security Applications Conference, ACSAC, Miami Beach, Florida, USA, 2007.

[7] "out-of-the-box" monitoring of VM based High-Interaction Honeypots- Xuxian Jiang, Xinyuan Wang

[8] paravirtualization in cloud environment http://www.petri.co.il/hypervisor-vs-paravirtualization.htm

[9] basic concepts of virtualization http://en.wikipedia.org/wiki/Virtualization

[10] KVM & QEMU documentation http://www.linux-kvm.org/page/Documents