Analytical Survey of Security in Virtualized Environment

Nilambari Joshi
Dept. of Computer Engineering and IT
Veermata Jijabai Technological Institute
Mumbai, India

Varshapriya J N
Asst. Professor, Dept. of Computer Engineering and IT
Veermata Jijabai Technological Institute
Mumbai, India

***Abstract*— Virtualization is a key component of current enterprise IT infrastructure. That is mainly because of the cost effective solutions and highly scalable and efficient ways of data storage and data processing it provides. On one hand this ensures business continuity for critical business applications viz. banking, e-commerce etc. but at the same time it confronts them to more security risks. The intrinsic characteristics of virtualization i.e. resource sharing and isolation, which are its strengths, if used without taking proper precaution can be exploited and can put whole virtualized environment and eventually the business at stake. This paper studies different reasons that put virtualization at security risk and also analyses different mechanisms to mitigate risks and to strengthen the environment making it less vulnerable to attacks over the network as well as within the host system on which it is deployed.**

Keywords- Virtualization, Hypervisor, Virtual Machine Introspection, Security

I. INTRODUCTION

A. Virtualization

Virtualization with respect to computing environment is the concept of logical system which is similar to and provides same functionalities as that of a physical system. In reality it is just part of actual physical system [7]. Virtualization Framework divides actual resources of a physical machine like CPU, memory, storage disk, etc. into different virtual machines such that each virtual machine can run as standalone entity with its own operating system and set of applications. This is mainly achieved by one of the techniques time-sharing, emulation, and partitioning [8]. For the user of an application that is running in a virtual machine (VM), the VM is a standalone self sufficient system but in fact it is working along with other VMs on a single physical system.

Virtualization brings many benefits along with it,

1. Infrastructure cost reduction – by reusing physical resources.
2. Reduction in power usage
3. High scalability – by easy addition of physical resources and integration
4. Less downtime / business continuity – by means of live migration of VM
5. Efficiency – by distributed computing
6. Availability – through replication

There are different types of virtualization like Desktop virtualization, Server Virtualization, Memory, storage, network virtualization. This paper mainly deals with Server virtualization in which logical computer called virtual machine or guest machine having its own operating system (guest OS) is created on a physical system by dynamically sharing physical resources of the system or host machine among all virtual machines.

There are three types of Server Virtualization

- Full Virtualization –In full virtualization mode, the guest machine operating system (OS) is unmodified. Privileged instructions are handled by hypervisor by means of binary translation and communicated to the hardware.

- Para Virtualization – In para virtualization mode, guest OS is modified to call hypercalls of the hypervisor instead of privileged instructions.

- Hardware Assisted Virtualization - In Hardware Assisted Virtualization, modification are made in the processor with additional privilege level and new set of instructions which identifies guest requested instructions and takes hypervisor's help to deal with guest requested privileged instructions This ensures guest OS to be unmodified and good performance.

B. Hypervisor

Hypervisor also known as Virtual Machine Monitor (VMM) is the component in computing system that drives virtualization. It emulates underlying host system hardware to the guest machines created on the host. It facilitates creation and management of virtual machine on host system. It is

basically an abstraction layer between physical host system hardware and virtual machines. The mechanism required to assist guest OS interaction with actual hardware system is implemented in hypervisor.

There are two types of hypervisors

- Type I – Type I hypervisor also termed as bare metal or native hypervisor runs directly on the host's hardware to control the hardware and to manage guest operating systems. A guest operating system runs on another level above the hypervisor. Hypervisor provides basic OS functionalities like resource scheduling, memory allocation etc in addition to managing virtualized environment.
  Oracle VM Server for SPARC, the Citrix XenServer, KVM, VMware ESX/ESXi, and Microsoft Hyper-V hypervisor are some of the type I hypervisors.
- *Type II* – Type II hypervisor also termed as host based hypervisor runs within a conventional operating system environment. With the hypervisor layer as a distinct second software level, guest operating systems run at the third level above the hardware. VMware Workstation and VirtualBox are examples of Type II hypervisors.

Hypervisor makes virtual machine feel that it has dedicated resources to run its application where as in fact it's sharing resources of underlying host with other virtual machines running on the same host. Hypervisor is mainly responsible for following tasks

- Isolation – It ensures that one virtual machine cannot interfere with the processes and data related to another virtual machine running on the same physical host.
- Resource Sharing – It should have proper policies and algorithms in place to manage resource allocation and sharing among virtual machine in such a way that at no point VM is short of resources as configured when it's created and have acceptable performance.

Other functionalities that hypervisor performs are

- Inspection -The VMM has access to all the state of a virtual machine viz. I/O controller, registers, memory, storage etc. So it is primary source of getting information about guest VM from outside the VM. [1]
- Interposition – VMM interpose on certain virtual machine operations like executing privileged instructions.

II. Virtualization Security Concerns

Virtualization has introduced new components like hypervisor, virtual machine at different levels of abstraction. This in turn adds different interfacing points and thus creates more attack prone surface that needs to be protected to secure virtualized system. Basic architecture of virtualized environment as shown in fig.1 gives the idea of attacking surfaces introduced by virtualization. This mainly includes interface between Hypervisor and guest OS and interface between hypervisor and base OS. Additionally any interface created for inter VM communication can be considered as a point of attack.

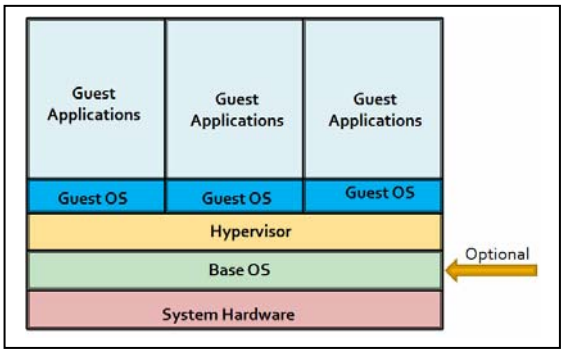

Fig. 1 Virtualization Architecture

A. *Causal Analysis*

Following intrinsic characteristics of virtualization puts the system at risk.

1. Resource sharing - All virtual machines on a host share host hardware resources, hypervisor is responsible to ensure isolation between VMs. But if this isolation is breached, or if it is not properly configured, it is quite easy to peep into another VMs data and hardware state and to change it. Some mechanisms like clipboard sharing, network configuration can give way to security breach.
2. Additional interfacing points – In server virtualization many virtual machines run on single physical host. To cater requests from guest OS to underlying hardware, hypervisor comes into picture which binds these two components seamlessly. This adds to different interfacing points like Guest OS –To- Hypervisor, Hypervisor – To- Hardware, External network –To- guest OS. Thus increasing attacking surface. Attackers find new ways of exploiting vulnerabilities at these different levels.
3. VM management by hypervisor – Hypervisor is main component commanding virtual machine. It starts stops and configures virtual machine. Thus if attackers manage to get control of hypervisor then he can have full control of the virtual machines running on the host.
4. Centralization – As a part of server consolidation, different applications run on different virtual machines but on the same physical machine. Thus any problem or attack on host causing unavailability of resources can put VMs at risk of Denial of Service. Thus guest machines are highly prone to get affected by an attack on the underlying host.
5. Negligence in handling isolation – Ideally one VM should not have access to application data and information on another VM deployed on same host machine. Hypervisor is mainly responsible to ensure this isolation. But sometime to support business needs some mechanisms are worked out viz. clipboard sharing, keystroke logging by hypervisor, disabling

isolation to run same application on multiple OS. This creates room for attacker to compromise the system.

6. Virtual Machines are logical (software) components. Some artefacts are created (image file) on underlying host system when VM is created and the file is accessible on host even when guest machine is turned off. It can be targeted and modified by an attacker.

B. *Types of Attacks*

There are different types of attacks which can be devised against virtual environment

1. VM escape – It happens when a program can bypass the virtual machine layer from inside a Virtual Machine and get access to the host machine. Since host machine has control over all VMs (by means of hypervisor), getting root privileges of host can manipulate with different virtual machines and communication among them. This kind of attack is normally possible by exploiting bugs on the VMM along with improper configuration of the host/guest interaction. Guest to Host Privilege Escalation attack is one of the ways to carry out VM escape. Virtunoid [13] by Nelson Elhage is a guest to host breakout for qemu-KVM. This is an exploit for CVE-2011-1751, a missing check in the qemu-kvm user space driver for the KVM Linux Kernel-mode Virtual Machine.

2. Denial of Service attack – Denial of Service attack is mainly focused on depriving a legitimate user of the services / resources they are entitled to use. In case of virtualized environment, VMs can make use of host resources like CPU, memory, networking interface etc. One virtual machine if requests all physical resources of host, no other VM can use the resource and thus are denied of the resources. One mechanism to avoid this is to restrict maximum amount of resources that a VM can avail of. But still there are certain situations for ex. deploying a bulky image in a VM can make use of most of the host resources.

3. Virtual Machine based rootkits – Originally rootkits were developed to replace standard Unix tools with versions that gave a user root or super-user privileges. They are designed in such a way that these activities are invisible to the user. Rootkits can hide themselves and make it difficult to get discovered by user or other processes. With the advent of virtualization, attackers are utilizing virtualization technology to do rootkit based attacks even more stealthily

Some examples of virtual machine based rootkits

- Subvirt - SubVirt operates at a level below the host kernel and remains inaccessible to the host operating system. The original host operating system is placed inside a virtual machine. The boot sequence is modified by the kernel module to load original operating system inside the Virtual PC (or VMware in case of a Linux).[11]
- Blue Pill – This is VM based rootkit designed by Joanna Rutkowska and Alexander Tereshkin. The main idea was that it should install itself without necessary any intervention of the machine, and would move the operating system into the virtual machine. This rootkit was itself a hypervisor that would allow to control the guest OS and cannot be detected using any integrity scanner. [12]

4. Remote management attacks – Most of the hypervisors provide VM management console with administrative privileges. The console can be accessed over the network as a web based application and thus is susceptible to many web based attacks like Cross Site Scripting, SQL Injection attack etc. Xen API HTTP Interface that had a Cross-site scripting (XSS) vulnerability, which allowed running a script code in a user's browser session in context of an affected site.

5. Networking attacks – Virtual Machines share same physical network interface as that of the host. Internally they are segregated through virtual switch /virtual hub. It is quite possible to exploit this virtual link to sniff traffic directed to one VM from another V deployed on the same physical host. ARP spoofing is one of such kind of attacks.

III. SECURITY MECHANISMS

As discussed in the earlier sections, virtualized system is a layered system. The components placed at each level of abstraction are prone to attacks which can affect the whole system with different intensity. There are different ways of securing standalone systems which are also relevant and useful in virtualized environment to deal with traditional attacks. Mentioned below are the traditional security techniques

- Firewalls – Firewall is very effective technique to introspect and regulate traffic moving in and out of a system. There can be hardware firewalls as well as software firewalls .Firewalls can be deployed and configured in host to implement any common configuration applicable to all VMs running on the host.
- Operating system patches against known attacks – All operating system vendors release patches with modifications in the OS to remove some existing vulnerabilities or to prevent known attacks. It is always advisable to keep guest as well as host OS updated with latest patches to mitigate risk of traditional system based attacks.
- Network based Intrusion Detection System (NIDS) – NIDS placed at the network interface provides high security against network based attacks. But they have very less knowledge about the state of actual system to be protected

- Host based Intrusion Detection System (HIDS) – HIDS has more visibility of the host's state as compared to NIDS. So it can be more effective to deal the attacks which bypass NIDS or which are initiated from within the network. However HIDS are susceptible to attacks when host is compromised.

The key challenge for traditional security systems is that real information about hardware and software states of a virtual machine cannot be obtained from within the system (as required by HIDS). Since VM works in virtual space, its actual state is known to and can be accessed by means of hypervisor. That is why to secure virtualized environment, it's necessary to devise more virtualization aware and virtualization friendly techniques to impose security on the system. There are different mechanisms applied at virtualization interface.

A. *sHype*

sHype, is a product of IBM Research aimed at securing x86-based virtualized environments [14]. In a virtualized environment multiple operating systems and applications are co-located on same hardware system. This being very beneficial from infrastructure point of view, it also causes undesirable interactions between those entities. Hypervisor ensures isolation among virtual machines so that the data of one VM is secure and not accessible from other VM. But in reality such stringent isolation is not desirable since organisations want communication between VMs which run logically related applications. So there is a need for secure resource sharing by enforcing access control between related groups of virtual machines. It places a secure access layer around the hypervisor running on a physical machine, mainly focussing on controlled resource sharing and information flows between VMs.

sHype considers two types of policies

1. Chinese Wall Policy – This enables administrators to ensure that certain VMs (and their supported workload types) cannot run on the same hypervisor system at the same time. This is useful to mitigate covert channels or to meet other requirements regarding certain workload types (e.g., workload types of competitors) that shall not run on the same physical system at the same time.

2. Type Enforcement Policy - It specifies which running VMs can share resources and which cannot. It supports the coalitions introduced by mapping coalition membership onto TE types. The TE policy defines the set of TE-types (coalitions) and assigns TE types to VMs (coalition membership). The TE policy rules enforce that VMs only share virtual resources if they have a TE type in common.

B. *KVMSec*

KVMSec is architecture (KvmSec) that is an extension to the Linux Kernel Virtual Machine aimed at increasing the security of guest virtual machines [5]. The KvmSec implements modules in both host and guest systems with their communication through secured channels. The core modules of the detection system are located on the host machine so that attacker in guest system is less likely attack it. Data collection is done through VM processes to get more comprehensive information. Each Virtual Machine uses its own private memory area for communicating with the host, so it is totally independent from other VMs.

Main characteristics of KVMSec are as below –

1. It is transparent to guest machines
2. It is hard to access even from a compromised virtual machine
3. It can collect data, analyze them, and act consequently on guest machines
4. It can provide secure communication between each of the guests and the host;

KVMSec provides communication link and signalling channel between host and guest using shared memory. In KVMSec shared memory is not directly managed by the hypervisor but by the main emulation process, that is Qemu-KVM. The choice of a communication channel using shared memory

C. *Virtual Machine Monitoring*

On one hand security of virtualized environment is important but at the same time the concept of virtualization has been leveraged to increase security of standalone host. The host system is moved into virtual machine and is monitored from outside the system i.e. VM. In such techniques the shortcomings of host base security systems like disguising HIDS with false information, attacking IDS are overcome.
There are two alternatives being developed and checked for validity in current research areas -

1. Active Monitoring - Active monitoring is done when the security tool places a hook inside the system being monitored. When execution reaches the hook, it will interrupt execution and pass control to the security tool. Active monitoring can also be done outside of the system being monitored (e.g., network and disk monitoring), however these monitors are restricted to the semantic level provided by the disk and network device abstractions. In active monitoring security critical code is placed in un trusted domain. It is required to secure this code which can be achieved by write protecting the memory where the code is placed.

2. Passive Monitoring - Passive monitoring is when the security tool monitors by external scanning or polling. As a result, it is unable to guarantee interposition on events before they happen. In passive monitoring security tool is placed in privileged VM (Dom 0 in case of xen). The security tool gathers guest related information from hypervisor which is in raw form. The tool needs to apply knowledge of guest OS (OS data structure semantics and positioning) to extract required information. This difference between the raw data provided by hypervisor and its meaningful analysis (with respect to guest OS) is called as semantic gap. In

case of Passive monitoring extra effort is required to handle semantic gap carefully.

Virtual Machine monitoring is being used to get run time state of guest machine and to use it for intrusion detection in the virtual machine. Lares [10] provides a novel approach of secure active monitoring in virtualized environment. Whereas active monitoring is used in Hypervisor Based Integrity Measurement Agent (HIMA) [9] to check integrity of the virtual machine.

D. Virtual Machine Introspection (VMI)

Virtual Machine Introspection is a way of doing passive monitoring introduced by Garfinkel [1]. The approach is used to carry out intrusion detection in virtual machine, VMI IDS, by directly observing hardware state and events and using this information to figure out the software state of the host. This offers visibility comparable to that offered by an HIDS. Directly observing hardware state offers a more robust view of the system than that obtained by an HIDS, which traditionally relies on the integrity of the operating system. Since introspection is done from hypervisor layer i.e. below guest OS, the information obtained is genuine even in case of guest OS compromise. VMI IDS is been used as a basis for many security applications mainly with an aim to provide more robust security service on existing system.

One of the application of VMI based Intrusion Detection is used in HyperSpector [2] which is virtualization based intrusion detection system for distributed network. HyperSpector is a distributed IDS where server is placed in one VM and intrusion detection component is placed in a separate VM on the same host as that of the server. The Server VMs across different hosts are connected through normal Ethernet LAN while IDS VMs are connected through virtual switch. Laureno [4] has used the concept of VMI to detect intrusion in virtual machines by observing sequence of system calls used in guest application processes.

IV. COMPARITIVE ANALYSIS

Three important security techniques studied so far viz. KVMSec, sHype and VMI can be compared as below –
KVMSec provides protection against VM integrity compromise, **sHype** ensures policy based access control to shared resources among VMs running under same hypervisor control. Whereas **VMI** is based on the idea of dealing with security from lower level of abstraction than that of attacking surface, making monitoring and security breach detection system out off visibility of attacker. It can be summed up as shown in table 1.

TABLE I. COMPARITIVE ANALYSIS OF DIFFERENT SECURITY MECHANISMS

Security Aspect / Security Mechanism	**KVMSec**	**sHype**	**VMI**
Description	Security measures built into KVM hypervisor	Policy based Mandatory Access Control for xen hypervisor	An architecture to get guest runtime information
Protection against VM integrity compromise	Yes	Yes in terms of content integrity	By comparing information available by Introspection and from guest OS
Protection against viruses, Trojans. DoS, rootkits	By means of secured communication between guest and host	No	By keeping monitoring and detection system out of VM being monitored
Cross-platform applicability	Integrated with KVM	Integrated with Xen but can be customized to work with other hypervisors	Hypervisor as well as guest OS independent

V. CONCLUSION AND FUTURE SCOPE

Different security measures are being discovered and applied to cope up with the new type of attacks targeting virtualized systems. Applying security measures at hypervisor level is one of the options to keep the base secure. But that is not the only point of concern. The approaches of active monitoring and virtual machine introspection are being used to leverage virtualization technology to provide security to the server. Considering security risks virtualization is exposed to, it is necessary to devise a solution that will protect virtual machines from network based as well as host based attacks. Virtual Machine Introspection applies security measures at hypervisor level. Hypervisor has realistic information of the hardware state of the virtual machine but that needs to be deciphered with respect to the guest OS, to reduce semantic gap. Whereas with active monitoring information can be obtained beforehand from within the guest with no concern of handling semantic gap. But it has risk of active/passive attacks on the security system. Considering benefits of the two approaches, a hybrid solution can be designed which extracts information from both the sources and amalgamate it to get comprehensive view of the activities going on in virtual machine. An Intrusion Detection and prevention framework based on this hybrid approach can be constructed to deal with attacks (network based as well as host based) targeting virtualized systems.

REFERENCES

[1] "A Virtual Machine Introspection Based Architecture for Intrusion Detection", Tal Garfinkel (VMware), Mendel Rosenblum (Stanford) - Network and Distributed System Security Symposium, February 2003.

[2] "HyperSpector: Virtual Distributed Monitoring Environments for Secure Intrusion Detection", Kenichi Kourai, Shigeru Chiba, 2005 ACM 1-59593-047-7/05/0006

[3] "kvm: the linux virtual machine monitor "Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori -in Proceedings of the Linux Symposium, vol. 1, (Ottawa, Ontario,
Canada), pp. 225_230, June 2007.

[4] "Intrusion Detection in Virtual Machine Environments", Marcos Laureano, Carlos Maziero, Edgard Jamhour - Proceeding EUROMICRO '04 Proceedings of the 30th EUROMICRO Conference.

[5] "KvmSec: A Security Extension for Linux Kernel Virtual Machines", Flavio Lombardi, Roberto Di Pietro - 2009 ACM 9781605581668/09/03.

[6] "Understanding Full Virtualization, Paravirtualization, and Hardware Assist", VMWare white paper

[7] Virtualization -Wiki - http://en.wikipedia.org/wiki/Virtualization

[8] "SECURITY CHALLENGES WITH VIRTUALIZATION", J Ramos, Masters thesis, University of Lisboa,2009

[9] "HIMA: A Hypervisor-Based Integrity Measurement Agent ", Azab, A.M., Peng Ning ; Sezer, E.C. ; Xiaolan Zhang , Computer Security Applications Conference, 2009. ACSAC '09.

[10] "Lares: An Architecture for Secure Active Monitoring Using Virtualization", Payne, B.D., Carbone, M. ; Sharif, M. ; Wenke Lee , Security and Privacy, 2008. SP 2008. IEEE Symposium on

[11] "SubVirt: Implementing malware with virtual machines", Samuel T. King Peter M. Chen, Security and Privacy, IEEE symposium 2006

[12] "Introducing Blue pill", Joanna Rutkowska, SyScan'06 Conference, Singapore

[13] "Virtunoid: Breaking out of KVM", Nelson Elhage, Black Hat USA 2011 / DEFCON 19

[14] "Building a MAC-based security architecture for the Xen opensource hypervisor", R. Sailer and et al., In Proceedings of the 21st Annual Computer Security Applications Conference (ACSAC 2005), Miami, FL, USA, Dec. 2005.

AUTHORS PROFILE

Nilambari Joshi is a post graduate student in computer engineering and IT department, VJTI, Matunga, Mumbai. She has 8 years of industry experience with Tata Consultancy Services Ltd.. Her area of interest includes Web Application Development, Computer Networking and Security.

Varshapriya.J.N is working as Assistant Professor in Computer Engineering & I.T Dept., VJTI, Matunga, Mumbai. She is M.E. in Electronicios & Telecommunication Engineering and has 4 papers to her credit at National and Interational level. She has taught various subjects such as Microprocessor & Microcontroller,Information Storage & management systems,Cloud Computing & Security, Data mining, etc at Graduate & Post Graduate Level. She has guided several projects at graduate and post graduate level and has combined experience of 9 years in Industry,Research & Teaching .She is also a student branch coordinator of CSI.