Middle-East Journal of Scientific Research 24 (S1): 64-69, 2016
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2016.24.S1.16

Multitenant Cloud VM Security Using HyperCoffer and HSVM

[1]N.D. Sowmiya and [2]S. Shanthi

[1]PG student, Department of CSE, Valliammai engineering college, Chennai, India
[2]Assistant Professor, Department of CSE, Valliammai engineering college, Chennai, India

Abstract: Security of tenant's data mainly based upon infrastructure of multitenant cloud environment. However both hardware and software is combined being processed and controlled, there is no any surprise, leakage of data or even malicious cloud operator occurring. Challenge is to secure the virtual machine (VM).Unfortunately, none can make a solution to cloud platform but we can give protection to our VM by against controlling the physical machine. In this paper we analyzed the challenge of virtual machine by using both hardware and software stack processor. We introduce the HyperCoffer, hardware-software framework that used to protect the user data. HyperCoffer doesn't make any security assumption and only trust the processor chip. HyperCoffer use a VMShim that run in between guest VM and hypervisor. Another hardware virtualization security is used and it's called as hardware-assisted secure virtual machine (H-SVM) that uses PIC16f877a microcontroller which is inserted in between hypervisor and hardware. H-SVM that is used to protect the hardware data is a server side processor. In this paper we have implemented a prototype of HyperCoffer by QEMU-based full-system emulator and VMShim mechanism then HSVM by USART and pic microcontroller. By this process we also improve the resource management, performance ratio (Performance measurement using trace-based simulation), reliability.

Key words: Multitenant · Cloud computing · Virtualization security · HyperCoffer · H-SVM

INTRODUCTION

Privacy and integrity of user data highly based upon the infrastructure of cloud being secure [1]. Multi-tenant cloud data should be securely stored and processed. By default, we cannot make secure to cloud data and limited security assurance only maintain [2]. So there is no surprise about the leakage of tenant information and data also in recent survey over 1000 chief occur and it is declared by the IT managers and executives auditors [3]. None can give a security to cloud platform but we can give the security protection to our virtual machine by against controlling physical machine. A hypervisor, also known as virtualmanager, it is a program that allowsthe multiple operating systems to share a single hardwarehost. Each operating system seems to have thehost's memory, processor and other resources. However, the hypervisor is actually managing the host processor and a resource, allocating what is needed to each operating system in process and making sure that the virtual machines (called guest operating systems) cannot disrupt each other. A hypervisor software layer is created in between virtual machine and hardware even a security is maintained in software based virtualization but code size is high. More coding is need for improving security level. Hardware layer is inserted in two forms, First form in between VM and hypervisor. Second form, in between virtual machine monitor and hardware [4]. In a virtualized system, virtualization is the fundamental technological platform for cloud computing. The word virtualization refers to abstraction of computer resource from application and end user processing the service. Virtualization technology define multi-tenancy cloud business model by giving scalable, shared resource platform for all tenant.

Multi-tenant shares computer capacity, storage, network is may lead to work overload on same virtualization, security vulnerabilities and failure of mechanisms. The multi-tenant virtualization platform important security element used is multilateral security. Multilateral security uses a VPMS architecture which defines the multilateral security to consumers [5]. Virtual machine monitoring is processor which is based upon two approach implementation is made by software

Corresponding Author: N.D. Sowmiya, PG student, Department of CSE, Valliammai Engineering College, Chennai, India.

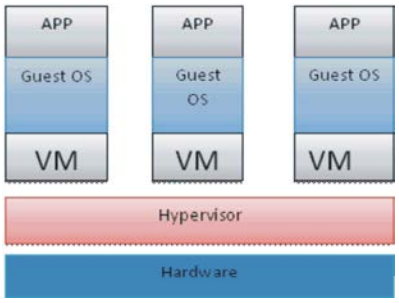

Fig. 1: VM block diagram

and hardware based approach. Software based approach is used to improve the security mechanism and execution is made by QEMU and hardware based approach is used to improve the viewpoint of performance, memory consistency and implementation is by using KVM,USART etc., that utilize hardware assists for virtualization of CPU [6-9]. Combined process of the software and hardware based approach is implemented by using Hybrid VMM [7].

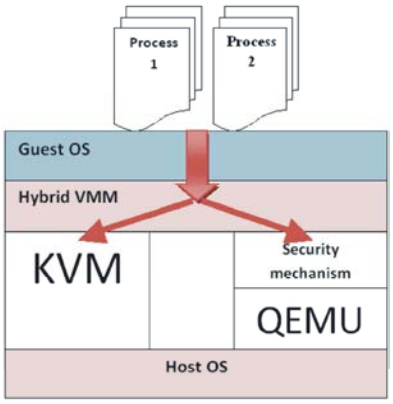

Fig. 2: Hybrid VMM overview

In today's memory virtualization techniques with highest privilege level, hypervisor can control both aspects of the memory virtualization, memory isolation and memory allocation through address translation. The role of hypervisor is to limit the memory allocation to use the physical memory more efficiently. Here we focus on the guest VMs protection even hardware is securely protected in data center even though leakage of data is occupied. With restricted thread model, we design hardware based VM isolation, called hardware-assisted secure virtual machine (H-SVM) architecture which extra translation layer in between hypervisor and hardware.

H-SVM that is used to protect the hardware data is a server side processor. For security critical applications, trusted computing base (TCB) or HW-based approach is highly desirable to protect guest VM and not only minimizes the surface attack also secure whole management OS that contains device driver and virtual machine management functionality. It is also important functionality to secure execution environment on virtualized computing platform under untrusted management OS or untrusted hypervisor that provide secure runtime environment network interface and secondary storage for virtual machine [9].

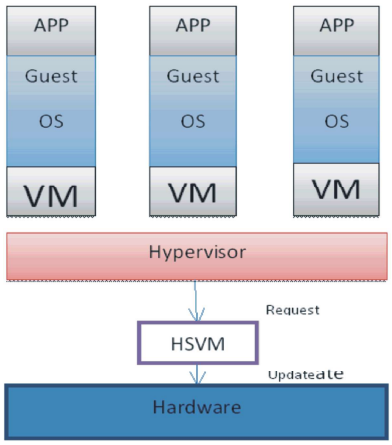

Fig. 3. HSVM VM security

The challenge is to secure the transparent VM protection by controlling software and hardware stack processor. to provide the strong and transparent VM-Level protection of multi-tenant cloud environment in Hyper Coffer that secure the privacy and integrity of tenant's VM. Hyper Coffer is Software-Hardware framework that only trusts the processor chip not makes any security assumption to external memory and device. Hyper Coffer extends the memory encryption concept for integrity checking of secure data communication in off-chip memory. HyperCoffer use a small piece of software that called VMShim which run in between OS and Hypervisor. We cannot protect the user data but give a security against the virtual machine by using HyperCoffer we can protect the tenant's data [2]. Other security and privacy issues in virtualizations are data leakage, data remanence issue in virtualization [10].

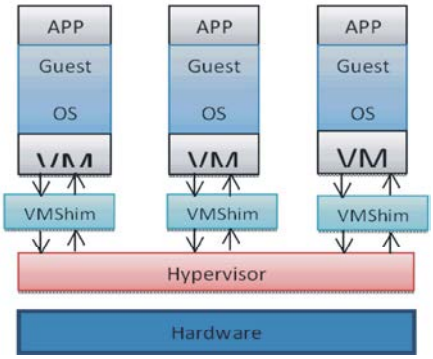

Fig. 3: Overview of VMShim

Security Benefits Due to Virtualization: The following are some of the benefits to security of the virtualization once it is introduced into the environment:

- The following are some of the benefits to security of the virtualization once it is introduced into the environment:
- Centralized storage of data is in virtualized environment prevent the loss of important data or information when the device is lost, compromised or stolen.
- When one application on the operating system is affected by attack, if the virtual machine and application are correctly and properly isolated and separated.
- Virtual environment provide Allows the sharing of system without necessarily having to share the critical information or data across the system when the system is configured properly and flexibility is maintained.
- Rollback is performed when the VM is infected get into the prior "secure" state that occurred or existed before the attack.
- Server can revert back the previous state in order to test what is occurred before and during an attack by the server virtualization.
- Desktop virtualization can be deployed to give more security to the user environment. An administrator can create and control "image" that is send to the user environment. This provides the better control of the operating system to secure that organizational requirement as well as security policies.
- Hypervisor is also known as virtual machine monitor that allow single hardware host to share the multiple operating system.

The following sections will explore the major areas of concern for virtualization security professionals works. They are,

- Hypervisor security.
- Host/Platform Security.
- Securing Communications.
- Security between guests.
- Security between host/guests.

The hypervisor cannot work properly if all data interfaces with VM are prohibited. However, separating protected and undefended data is a non-trivial work due to the multipart interface between hypervisor and VMs. Challenges of virtual machine are,

- Secure VM/Hypervisor Interaction.
- Interaction with the Outside World.

Background: Virtualization introduce host mode to run hypervisor and guest mode to run virtual machines (VMs). When VM executes a honored operation, its move from guest mode to host mode and which is called as VMExit. According the hypervisor VMExit defines different exit reason.to define about the VM,s CPU context, there is in-memory control structure (VMCS) for every virtual CPU, which summarizes the CPU context from both virtual machines (VM context) and hypervisor (hypervisor context).during VMExit processor is saves the VMCS and it is used by the hypervisor and handles the VMExit and resumes the VM's execution. H-SVM use microcode programs in hardware for the security of the memory protection of the processor. In HyperWall memory isolation is done by using Confidentiality and Integrity Protection (CIP). H-SVM handles the complex VM interaction, data sharing and protection by combing the guest Oss and hypervisor. Hyper Wall avoid the complex data interaction it only protect against the memory protection only.H-SVM and HyperVisor does not protect or secure against the external device but give secure to I/O data VMs.

Table 1: comparison between related system

	OS Transparent	TCB size	Phys. Attack	Cloud Function
HyperWall	No	CPU + Mem +IOMMU	No	Part
H-Svm	No	CPU + Mem + IOMMU		
	No	Full		
CloudVisor	Yes	All HW + CloudVisor	No	Full
HyperCoffer	Yes	CPU + VM-Shim	Yes	Full

Proposed System: Cloud user client's data are stored in the cloud data center that the cloud users are spread all over the globe with in cloud provider thousands of servers are communicated through the internet more risk are intended is well known. Cloud services are processed by using internet as communication infrastructure, cloud service is mainly based upon the IAAS provider. IAAS is upon the virtualization security. Virtualization security, or virtual-aware security, is important and must to computing systems and securing server computing systems that are virtualized or combined. A virtual machine (VM) is one pretending desktop or server network resources to create a virtual kind in which the framework is divided into single or multiple execution environments. HyperCoffer does not make any security assumption it only trust processer chip. HyperCoffer is used for protecting the user tenant's data from malicious actives against the both software and physical attacks at VM-level. However the security policies are connected secure processor difficult process is to reduce the sematic gap between virtual machine (VM) and secure processor. HyperCoffer uses a novel approach that which secure the secure processor provide the security enhancing mechanisms VMShim a small piece of software most virtualization security semantics. HyperCoffer provide both Hardware tents for running the VMShim and interactive of data for communication between hypervisor and VMShim.

HyperCoffer prototype is implemented by using QEMU-based full system emulation environment with VMM.Hardware-framework that transparently protects the guest virtual machine under untrusted hypervisor and even a physical attack. by using the QEMU we can reduce a low performance over head. HyperCoffer use AISE-based Data Encryption for Memory Privacy Protection is counter-mode address independent seed encryption for memory encryption. BMT for integrity Bonsai Merkle Tree Instead of blocking directly the encrypted /decrypted data generate the pseudo-random pad code. Pseudo-random pad is which XORed is then with plain text to generate the cipertext. Seed is composed into three forms they are LPID, counter and offset. LPID (Logical per-Page ID) is a unique ID to every physical memory page. Independent page address and initial time is assigned for every cache block counter and page off set is maintained. When over flow is occurred corresponding page assigned with new LPID and re-encrypted.by this malicious user cannot reuse the seed. Every LPID and counter value are saved in main memory and saved in the given physical address. HyperCoffer architecture consist of memory data protection, cache data protection, CPU context protection, extended page table protection, VM-table for multiplexing, VM-shim mode.

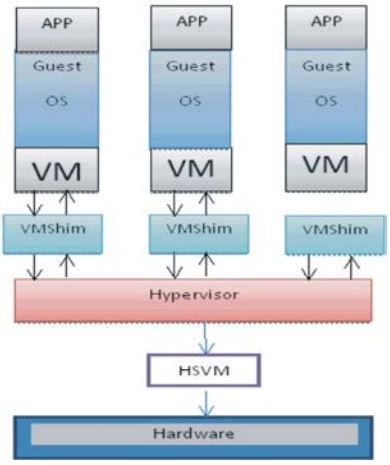

Fig. 4: Combined HyperCoffer and HSVM architecture

Designed used for VM isolation is done by using hardware-assisted secure virtual machine (H-SVM). Memory isolation is improved when we use the HSVM because direct blocking of hardware is protected and it is in the save region. When any changes are made modification is done to nested page table by HyperVisor. Nested page table from virtual machines (VMs) are stored in secured state of memory region, which can only processed by H-SVM hardware. If any changes in the VM memory allocation region HyperVisor at secure level will request to the H-SVM to update the nested page table for the VM. H-SVM checks whether request made may disturb memory isolation among the virtual machines (VMs) before updating nested page table. VM deallocate the physical memory page, if by setting all bytes zero when H-SVM cleans up the deallocated page. H-SVM is page ownership table, VM context information, nested page table. H-SVM is added in between HyperVisor and hardware.H-SVM cannot protect the CPU register status but by using HyperCoffer. Two prototypes are used in our paper, one is HyperCoffer and another is H-SVM. HyperCoffer is used to protect the client side data and H-SVM is used to protect and privileged level is to maintain in cloud data center.by this we improve the overall performance, reliability, security to data and VM.

Implementation and Interface

Hypercoffer Implementation and Interface: In virtualization environment security Hyper Coffer extends the traditional secure processor and rights. HyperCoffer use encryption technique as AISE,

integrity checker as BMT and and introduces VM-Table for multiprocessing.by this process all the data is protected including the data in CPU context, memory, input/output device, on- chip cache etc.Current running protected information is stored in the VM-table which was introduced in the HyperCoffer and it is saved in the save protected memory region. These instructions are used to operate in VM-Table {VMID, $K_{vm,}$ $HRoot_{vm,}$ $Addr_{cnt,}$ $Addr_{shim,}$ $Addr_{BMT}$} for each entry three more new instruction are used vm_install, vm_snapshot and vm_uninstall. Two non-volatile register are used they are, one for creating unique LPID for each page andanother is get updated in logging of VM is booted/ snapshot is made.

Table 3: key used by HyperCoffer

Key	Context	Protection
K_{vm}	Per VM	Encrypt VM memory and disk image
K_{mem}	Per Chip	Encrypt CPU reserved memory for VM-Table
SKcpu	Per Chip	Private key of the CPU

H-SVM Implementation and Interface: H-SVM is instead in between HyperVisor and Hardware. HyperVisor or VMs execute to use a special instruction to make request to H-SVM.Important instructions used in the H-SVM are Create VM, Delete VM, Page map, Page unmap, Context saveand Context restore.to initialize the VM context information, to update nested page table and VM to be scheduled there are four basic interface is used for processing. Create VM, HyperVisor request the H-SVM to create a new nested page table for the VM, then H-SVM initializes the VM context information. H-SVM also creates a unique per-VM encrypted key that which is used for hypervisor to request the page swap. Delete VM, to destroy the VM contact information from nested page table request send to H-SVM. before destroying VM clear the nested page table. To secure and guarantees the confidentiality of the VM. Finally VM clear the VM- encrypted and VM unique identifier key from nested page table.Page map, To VM it is used to assign the physical memory page is done by using page map, page table entry page map the frame (machine memory page) to guest physical page.

Page map, To VM it is used to assign the physical memory page is done by using page map, for updating nested page table entry page map operation maps the frame (machine memory page) to guest physical page. Page map is use to check the page ownership of the physical page is a critical component of memory isolation for every each page map by H-SVM. Before update H-SVM to nested page table have to check the page

Table 2: Instruction used in the HyperCoffer

New instruction	Environment	Instruction Semantic
vm_install, addr1, addr2	Hypervisor	Install vm_key (addr1) and vm_vector (addr2). Return VMID.
vm_uninstall, VMID	Hypervisor	Remove the vm_vector indexed by VMID from the VM-Table.
vm_snapshot,VMID, addr	Hypervisor	Encrypt the vm_vector indexed by VMID and save it to memory.
ept_st, addr, val	Hypervisor	Update data in EPT memory. Invalid cache only if an GPA_2_HPA mapping is modified or deleted.
VMEnter	Hypervisor	Resume VM-Shim instead of the VM.
VMExit (modified)	Guest VM	Transfer control to VM-Shim instead of the hypervisor.
shim_to_host	Shim	Trigger VMEXIT and switch to host mode.
shim_to_guest	Shim	Switch to guest mode and resume VM.
raw_st, addr, val	Shim/Guest	Store data into memory without encryption.
raw_ld, enc_on, addr	Shim/Guest	Load data without integrity check. Use enc_on to control encryption engine on or off.

ownership table whether physical page already owned by another VM.If already physical page is owned by another VM mean request map operation will be aborted. if the nested page table update entry for the VM request then VM become the owner of the physical page. by this illegal mapping of page table can be reduce. Page upmap is used for deallocation of physical memory from virtual machine. Page unmap request is made by HyperVisor to H-SVM. Then H-SVM modifies the page table entry and before completing clear the information or content of the memory page operation and also reset the page ownership table. Context save. This instruction is used to schedule a virtual machine to core by the HyperVisor. Register states the saved and restore the information of VMContext. Contact information contain register state and page table pointer to nested page table. H-SVM with the VM identifier HyperVisor request to context save. Save register to core running in the protect secure memory region by using H-SVM. This operation is similar to VMExit in AMD-V.Context restore. It is used to restore schedule process by the HyperVisor to H-SVM.H-SVM store the VM information in the core to process the register states. Only H-SVM can change and update the nested page table and register states, but HyperVisor cannot force to run the VM to the use cooperate the nested page table.

CONCLUSION

This paper analyses about the security to guest virtual machine of multitenant cloud environment. Security of tenant's data is mainly based upon the infrastructure of multitenant cloud environment. There no any surprise or leakage of data in cloud operator occurring but we cannot give security to our user data by protecting against virtual machines. Here we use two security mechanisms HyperCoffer and H-SVM to secure guest VM and cloud data. HyperCoffer is use to secure tenant's data and it doesn't make any security assumption. HyperCoffer use VMShim that in between guest OS and hypervisor. Another mechanism H-SVM, it is hardware based VM security which is instead in between HyperVisor and hardware which use PIC16f877a running H-SVM. H-SVM is server side processing which is used secures the cloud data and information. By this mechanism we can improve the overall performance, reliability, user information and VM security.

REFERENCES

1. Tech target HyperVisor processor definition http://searchservervirtualization.techtarget.com/
2. Yuban Xia, Yutan Liu and Habit Chen, 2013. "Architecture Support for Guest-Transparent VM Protection from Untrusted HyperVisor and Physical Attacks*"978-1-4673-5587-2/13/$31.00 ©2013 IEEE.
3. Circled Reporter, 2009. "Survey: Cloud computing 'no hype', but fear of security and control slowing adoption." http://www.circleid.com/posts/20090226_ cloud_computing_hype_security, 2009.
4. Dinesh Chandrasekaran, Ramesh Karri, Pratik Mathur, Vikram Padman and Nasir Memon, 0000. "Hardware-assisted Secure Virtualization".
5. Pengfei Sun, Qingni Shen, Liang Gu, Yangwei Li, Sihan Qing and Zhong Chen, 0000. "Multilateral Security Architecture for Virtualization Platform in Multi-tenancy Cloud Environment" IEEE conference,© corresponding author.
6. Seongwook Jin, Jeongseob Ahn, Jinho Seol, Sanghoon Cha, JaehyukHuh and Seungryoul Maeng, 2015. "H-SVM: Hardware-Assisted Secure Virtual Machines under a Vulnerable Hyper Visor" Ieee Transactions on Computers, 64(10), October 2015
7. Junya Sawazaki, Toshiyuki Maeda and Akinori Yonezawa, 2010. "Implementing a Hybrid Virtual Machine Monitorfor Flexible and Efficient SecurityMechanisms"978-0-7695-4289-8/10$26.00© 2010 IEEEDOI 10.1109/PRDC.2010.32.
8. Bellard, F., 2005. "QEMU, a fast and portable dynamic translator," in Proc. of the 2005 USENIX Annual Technical Conference (ATC '05), pp: 41-41.
9. Neiger, G., A. Santoni, F. Leung, D. Rodgers and R. Uhlig, 2006. "Intel Virtualization Technology: Hardware support for efficient process or virtualization," Intel Technology Journal, 10(3): 167-177, Aug. 2006.
10. Chunxiao, Li, Anand Raghunathan and Niraj K. Jha, 2012. "A Trusted Virtual Machine in and Untrusted Management Environment", Ieee Transactions on Services Computing, 5(4), October-december 2012.