Protecting the Security and Privacy of the Virtual Machine through Privilege Separation

Cong Yu; Lixin Li; Kui Wang; Wentao Yu
Information Science and Technology Institute
Zhengzhou, China
tieshan0216@sina.com; picric@139.com; 381247177@qq.com; yuwentao030@126.com

***Abstract*—Virtual machine security issues have been the focus of attention. The permissions of traditional administrative domain Dom0 are too large, so that the user's privacy is threatened. Once the attacker compromises Dom0, it can threaten the entire virtualization platform. This paper introduces a privilege separation virtual machine security model (PSVM). Dom0's privileges are split into two parts: the operations about the user's privacy form a DomU management domain, responsible for managing the user's privacy; remaining forms Thin Dom0. Users and virtualization platform for server-side need mutual authentication. It can prevent unauthorized users and counterfeiting Virtualization platform invading system. The user's privacy is under its own management to prevent the Virtualization platform snooping. However, it affects only one user, even if the management domain is compromised. Combined with the model, the prototype system is implemented and security analysis and performance testing is done.**

Keywords-mutual trust; separation of privilege; virtual machine security; security model

I. Introduction

Virtual machine technology has brought new opportunities for the development of information technology, but security issues cannot be ignored. The vulnerability in Xen CVE published [1,2,3,4], has brought a serious security threat to the virtual machine, causing paralysis or user information leakage. To this end, IBM's [5] sHyper, for the Xen development of safety architecture, has deployed a CW (Chinese Wall) and STE (Simple Type Enforcement) policy effectively isolating the virtual machine. Chunxiao Li [6] from Princeton University proposed a way to prevent Dom0 and other virtual machines spying on its memory information by encrypted memory. The Cloud Visor project [7] proposed, uses nesting to detect the VMM (Virtual Machine Monitor) to prevent the attack from the inside. It can effectively protect the security and privacy of user's information. The sHyper isolated virtual machine, but not for the Dom0 protection measures. For its large amount of code and a great privilege, in theory, it can check the memory and register state of the user domain. The user's privacy is not protected, since the attack from Dom0 will endanger the security of the virtual machine host. Although Chunxiao Li's mechanism can enhance security, the frequent encryption will cause the system overhead increasing. However, while the Cloud Visor can protect the user's privacy, the nested performance loss is not inconsiderable which the user may not be able to tolerate. In this paper, we split the Dom0's privileges into two parts (one is held by per-user, the other is remained by the Thin Dom0). The system does not consider the hardware attack and Dos (Denial of service) attack. Hardware is generally placed by special that the attacker is difficult to directly contact. The Dos attack is very easy to implement, for example, shutting down the server or a large amount of data may work. Therefore, it's not to consider in this paper.

II. The PSVM Platform

A. Architecture

For Dom0's privilege is too large and it could easily lead to a variety of security threats, this paper proposes a separate privilege virtual machine security model based on mutual trust. We remove the device driver from traditional Dom0 to a driver domain (Driver Dom), easily producing security vulnerabilities. And we remove the operation affecting users' privacy to a per-user management domain (DomU Manage_VM). Dom0 become Thin Dom0 after separation, mainly responsible for the creation of the user domain, resource scheduling, and other basic operations. The virtual machine monitor (VMM, Virtual Machine Monitor) and hardware are at the bottom. The architecture is shown in Figure 1, and the dark part of the system is the TCB (trusted computing base):

Thin Dom0: responsible for the creation and management of the user domain (DomU, including DomU App_VM and DomU Manage_VM), and other operations.

DomU: users domain, including the DomU management domain (DomU Manage_VM), and DomU application domain (DomU App_VM).

DomU Manage_VM: responsible for the management of the user's privacy, for example, view memory, log management, vTPM management, remote attestation and other sensitive operations.

DomU App_VM: conversation with the users, equivalent to the DomU in the traditional Xen architecture.

Driver Dom: the back of the device driver, management of the I/O devices.

TPM Driver Dom: the TPM driver and management of the vTPM.

Published by Atlantis Press, Paris, France.

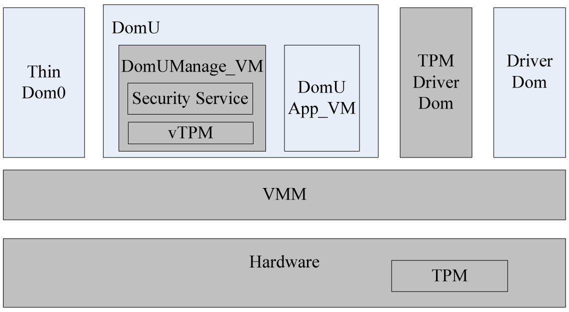

Figure 1. Architecture of the PSVM

B. Building of the System

a) Thin Dom0 building

After the computer powers on, the system will be startup as follows:

Step 1: The built-in TPM chip trusted boot process will measure the BIOS. Then it will load the BIOS and find master boot register MBR.

Step 2: Measure the MBR, then, run the Grub bootloader in the MBR. The Grub programs include the stage and grub.conf. The specified operating system kernel will be loaded by configuring grub.conf.

Step 3: Measure the xen.gz which stores the VMM, and then load xen.gz. Because the Thin Dom0 is not in the TCB, it needn’t to be measured. After then, loaded Thin Dom0 image file vmlinuz_thin dom0 and initrd.img_thin dom0.

b) DomU building

After Thin Dom0 created, it can load the DomU Manage_VM and DomU App_VM.

Step 4: User1 ask for the VMM authentication and integrity check, then request load DomU. Since the Thin Dom0 is untrusted, there may appear the following questions: (1) refuse to load, (2) tampering loaded, (3) does not release memory mapping, used to view its contents of DomU. This article does not consider the Dos attack, only to consider the problem (2) (3).

Step 5 and 6: Remote user1 provides hash values of dom1_manageVM.img and dom_appVM.img. The VMM is responsible to measure: if the values match, then start DomU; otherwise, an error is returned. It can defend against the attack (2). Finished loading DomU, the Thin Dom0 external mappings are released. We want to ensure that no DomU's memory mapping is in Thin Dom0, and then start DomU. It can defend against the attack (3).

C. Privilege Separation

During the system runtime, Dom0 manages the DomU. Once Dom0 is attacked, or misused, it might pose DomU security threats, and the secret information stored in DomU will be leaked to the malicious user or administrator. We split operation about the user's secret from the Dom0 privileges. The remaining forms Thin Dom0. The DomU Manage_VM is responsible for these sensitive operations of the corresponding DomU App_VM.

We analyze the operations and the types of information Dom0 participating in the DomU's life cycle to determine which information will be destroyed DomU information security.

When DomU is created, Thin Dom0 passes DomU through start info page, including memory size and other information, then load DomU system image through foreign mapping. At the creation time, DomU does not contain any secret information. It can be performed by Thin Dom0 without modification. Once the DomU starts, Dom0 releases foreign mapping of DomU. During the DomU App_VM hanged time and migration, Thin Dom0 needs to save the P2M table and page table to record the physical address to machine address mapping and virtual address to the machine address mapping, so that to position machine address in the recovery stage. This information does not contain the user's secret information, and can be handled by Thin Dom0. However, for the saving of the DomU App_VM image contains the user's privacy, it should be handed to the DomU Manage_VM to encrypt the image (including the memory and virtual register state) and save. In the recovery stage, the encrypted image can be loaded by Thin Dom0, for its encrypted view, the Thin Dom0 can’t spy into it.

D. Virtual TPM

TPM (Trusted Platform Module) is defined by the TCG specifications [8]. Here, especially the TPM chip, which is attached to the motherboard, performed by the hardware. Memory in general cannot access it directly, only through a special interface to access. It can generate a pseudo-random number, the key for encryption and decryption operations, to achieve the secure storage of keys, remote attestation, and the integrity verification.

In order to provide a secure and trusted computing environment to the user, the TPM chip is placed in the bottom of the hardware layer. Each DomU has a vTPM, providing services for each user. DomU Manage_VM contains each vTPM drive front. An isolate TPM Driver Dom is built for good security, as each vTPM device backend driver and vTPM Manager, used to manage and dispatch various vTPM instance. Its structure is shown in Figure 2:

vTPM provides functionality includes:

- Cryptographic operations, including random number and the key generation, encryption and decryption.
- Remote attestation. The identity of the platform can be verified by the remote user through digital signature of the data by AIK (Attestation Identity Key).
- Protection of secret data. Secret information such as keys or the hash values can be stored in security.

Published by Atlantis Press, Paris, France.

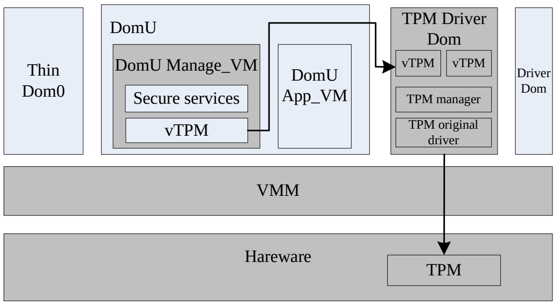

Figure 2. The frame of TPM and vTPM

E. *Security Services*

The user can customize security services to meet individual needs in the DomU Manage_VM, such as encryption, log management, auditing and so on. Due to these security services separately in each DomU Manage_VM, only for its own DomU App_VM, it can effectively prevent the leakage of users' information. The event of failure or being attacked, only limited to users in the DomU loss, does not spread to other DomU effectively isolate the fault.

F. *Virtual I/O*

In the conventional virtual machine system, the device drivers are placed in Dom0. The virtual machine can access to the peripherals through the front and back ends drivers of the device. Device drivers expose too much vulnerability, likely to pose a security bottleneck. However, we separate device drivers from Dom0 to establish an independent driver domain (Driver Dom). It contains backend drivers and the original drivers, accessing the hardware directly. Since it is not safe, if you want to access the network card or hard disk, it will result in the disclosure of information that can be solved by encryption. Information before entering the Driver Dom, sent to the DomU Manage_VM, encrypted and then transmitted to Driver Dom, so that even if the driver domain can observe that, it will not affect the confidentiality of the information. It is shown in Figure 3:

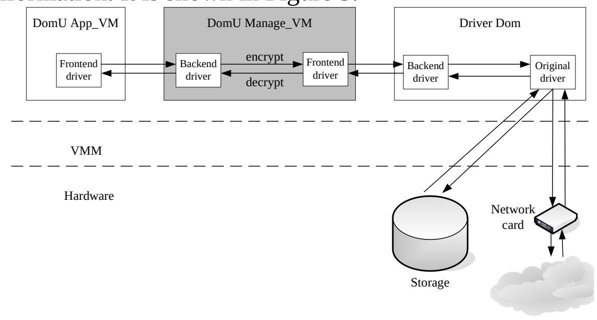

Figure 3. The I / O process in PSVM

III. THE IMPLEMENT OF PSVM

In Figure 1, we can see that there are five virtual domain need to build in PSVM. The recent research has indicated that the driver has been separated from Dom0 of Xen to build Driver Dom. Thin Dom0 is based on the traditional Dom0, removing some specific process, which will be described in the subsequent section. The DomU Manage_VM is based on the Xen streamlined Mini OS kernel, adding the operation of the user's privacy, customized security services according to the users' needs. It has support vTPM after Xen 3.0, and the front-end and back-end driver are contained[9]. Only need to do is to separate this part from Dom0 to ensure its reliability.

A. *System Startup*

Trusted boot of the system, starting from the BIOS hardware powering on to the startup of DomU, need to be measured to ensure the credibility of the system startup.

Trust Grub [10] is used to boot from the BIOS to Thin Dom0 instead of the traditional Grub. The grub.conf startup items are configured to achieve the security purpose. Its configuration is as follows:

```
title Xen-3.4.1,Fedora 13 x86_64
root (hd0,11)
measure -pcr=8 /boot/xen.gz
kernel /boot/xen.gz console=vga
module /boot/vmlinuz-thin dom0  root=/dev/sda12
module /boot/initrd.img-thin dom0
```

Whether to start Thin Dom0 can be determined by measuring the value of the PCR.

The startup of the DomU should to be performed together by remote users. The specific steps are as Figure 4:

Step 1: User sends to the server-side VMM a timestamp N, and using its own private key to encrypt, and then the server's public key to encrypt.

Step 2: The server VMM receives and decrypts the message with its own private key and the user's public key to get N. After that it produces a session key Ksession, together with N to obtain a hash value using the SHA algorithm. With the value as an argument and PCR, instruction TPM_quote is used to produce the testimony, together with the signature of the encrypted Ksession to the user.

Step 3: The user receives and decrypts Ksession, The value of PCR can be calculated. It would be used to verify the identity and integrity of the server-side VMM.

After the identification, the user sends to VMM the hash value of DomU Manage_VM and App_VM image, encrypted with Ksession, to verify its correctness. It should be noted that this value can be calculated and returned to the user at the last close. If it is the first time to create an initial system, a unified security system hash value can be provided by the VMM.

Step 4: The VMM validates the hash value of DomU Manage_VM image.

Step 5: If adopted, the DomU Manage_VM is created by the Thin Dom0.

Step 6: After created, the foreign mappings of Thin Dom0 are cleared by the VMM. Thin Dom0 is no longer mapped to its memory to spy on the user's information once the DomU started.

Published by Atlantis Press, Paris, France.

Step 7: The VMM validates the hash value of DomU App_VM image.

Step 8: The DomU App_VM is created by the Thin Dom0.

Step 9: The VMM purges all the foreign mappings of Thin Dom0.

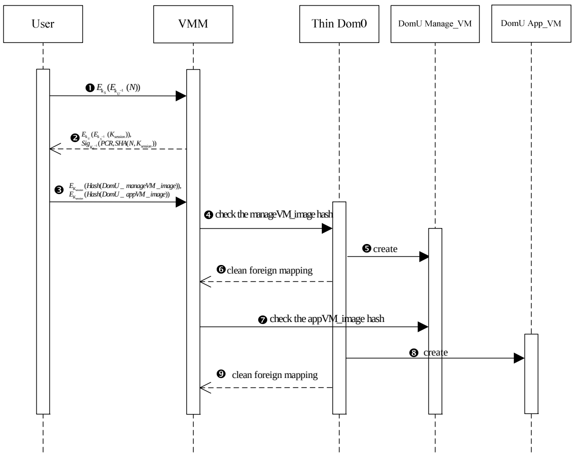
User
VMM
Thin Dom0
DomU Manage_VM
DomU App_VM
❶ $E_{k_S}(E_{k_U^{-1}}(N))$
❷ $E_{k_U}(E_{k_S^{-1}}(K_{session}))$, $Sig_{K_S^{-1}}(PCR, SHA(N, K_{session}))$
❸ $E_{K_{session}}(Hash(DomU\_manageVM\_image))$, $E_{K_{session}}(Hash(DomU\_appVM\_image))$
❹ check the manageVM_image hash
❺ create
❻ clean foreign mapping
❼ check the appVM_image hash
❽ create
❾ clean foreign mapping

Figure 4. The boot process

B. Privileges Allocation

User privacy-related operations are handed over the DomU Manage_VM. It can be completed the allocation of privileges by XSM (Xen Security Modules) framework [11]. The XSM is a Xen generic security framework, in which safety hooks are inserted into the VMM layer to enforce the security policy. The research was restricted by security policy configuration. Allocation of domain and privileges is shown in Table 1:

TABLE I. ALLOCATION OF PRIVILEGES IN PSVM

	Thin Dom0	Manage_VM	App_VM	TPM Driver	Driver Dom
Thin Dom0	I	M	M	M	M
Manage_VM		S, I, T	S, I, T	S, T	I
App_VM		I, T	I, T	T	I
TPM Driver	T	T	T	T	
DriverDom	I	I	I		I

VM management operations (M): VM creation, management and other operations.

Privacy operations (P): operations about the user privacy.

Security services operations (S): the user security services, such as encryption, auditing and other functions.

I / O operations (I): I/O access, device drivers.

TPM operations (T): the TPM and vTPM operations, such as data security storage.

C. Security Services

In the DomU Manage_VM, it can provide security services to the users, such as encryption, log management, audit and other functions. These services can be provided by the VMM unified, can also be provided by a trusted third party. Dom Manage_VM can selectively load these security services according to user needs. Currently, we can only provide encryption services. The data can be encrypted by a secure key produced by vTPM with 128-bit AES algorithm to ensure sensitive information that flows to non-trusted domains encrypted without leaks of information.

IV. SYSTEM ANALYSIS

The system TCB is composed of two parts, one from the virtualization platform, the other one from the user. The TCB of the virtualization platform includes the VMM and TPM driver domain. And user TCB includes the DomU Manage_VM. The bulky Dom0 is removed from the TCB to reduce the attacks. In addition, the system can improve security as follows:

- Users and virtualization platform need mutual authentications, It can prevent malicious users or fake virtualization platform. The random number is added in certification to prevent replay attacks.
- Privilege separation. The traditional virtual machines unify all the privilege on Dom0. Once Dom0 is compromised, it will cause damage to the entire system, affecting every user. In PSVM, privileges are separated to per-user domains. Each the privileged domain is only responsible for managing its own application domain.
- It can avert leakage of private data. Privacy protection is an important issue of virtualization. In PSVM, the user's privacy is administered by the user, rather than the system administrator, effectively preventing the leakage of user privacy data.
- Fault isolation. The compromised management domain can only make the corresponding user suffered, but not the others.

Furthermore, it supports customization services. The user can customize personalized service according to his own needs. High security user can customize more security services. And the performance but low security users can streamline the system, to achieve higher operating speed. The services can be provided by third parties, but not just rely on virtualization platform to reduce maintenance costs.

V. EVALUATION

In order to test the performance of the system, our experimental platform is based on Xen-3.4.1, Fedora 13 x86_64 , Intel Core 2 Duo / Quad processor, 8GB of DDR2 memory, 160G SATA hard disk, TPM 1.2. As a user, with a Pentium (R) 4 2.93GHz processor, 512M memory computer and connected to virtualization platform in 100M LAN.

The system start-up time is tested in the prototype system, including the time from powering on to DomU starting on traditional Xen, and time from powering on to the DomU App_VM starting in PSVM, the test is done 20 times , Table 2 lists the results of the average test time.

It can be seen that until the Dom0 startup, the difference between PSVM prototype system and Xen is not large. But the start-up phase of DomU, due to cooperation with remote users and network bandwidth, it results large delay. However, considering the security and system startup not a large proportion of the system running time, the delay is still acceptable.

TABLE II. STARTUP TIME COMPARISON

Phases	Traditional Xen	PSVM Prototype System	
Dom0 startup	56.093s	60.177s	
DomU startup	13.925s	Manage_VM	32.801s
		App_VM	16.943s

We also test the time-consuming of data written to the hard disk. Respectively, 16M, 32M and 128M data blocks of storage, the data stored encrypted, non-encrypted stored in PSVM prototype system and stored directly in traditional Xen are compared, as Figure 5.

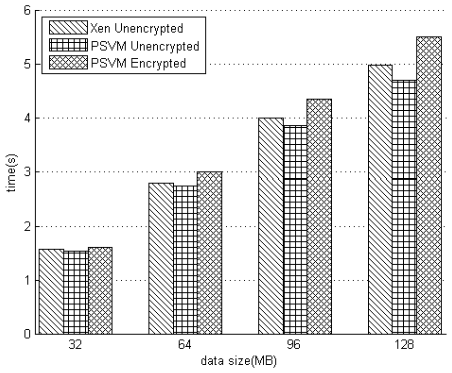

Figure 5. The storage performance contrast between PSVM and Xen

In the case of non-encrypted, the storage time required in PSVM prototype system is almost the same as in traditional Xen, even a slight improved. The encryption time consumes at an average of about 8%, within the acceptable range.

VI. RELATED WORK

There is a lot of domestic and foreign research for the security of the virtual machine.

The access control model Researchers have exploited various access control models, such as BLP, RABC, Biba. IBM's [5] sHyper is the most popular. It has deployed a CW and STE policy to control the information flow between the virtual machines.

The nested virtualization Both the XenBlanket project [12] and the Cloud Visor project [7] use the nested virtualization to detect the VMM and protect the users' security from the Dom0.

The encryption of memory Chunxiao Li [6] encrypts all the memory before it transmitted to the other domain. It can protect the user's privacy. Hidekazu Tadokoro [13] also protects the information of memory by encryption. Every domain has an encryption-map, it shows which memory is encrypted.

VII. CONCLUSION

PSVM presents a new architecture of the virtual machine. It separates the traditional Dom0's privileges to per-user. It can reduce the risk, improve the security and protect the users' privacy. We implement a prototype system based on Xen, and the results show that the model can protect the user's privacy and the performance consumed is within an acceptable range.

ACKNOWLEDGMENT

We thank the anonymous reviewers and our tutor, Li Li-xin, for their comments on the paper.

REFERENCES

[1] CVE-2012-3433。HVM destroy p2mhost Dos Xen HVM Guest p2mTeardown Denial of Service Vulnerability.

[2] CVE-2007-5497. Integer overflows in libext2fs in e2fsprogs.

[3] CVE-2008-1943. Buffer overflow in the backend of XenSource Xen paravirtualized frame buffer.

[4] CVE-2007-4993. Xen guest root escapes to dom0 via pygrub.

[5] IBM Corporation. Xen Users' Manual [EB/OL]. 63-64

[6] Chunxiao Li, Anand Raghunathan, Niraj K. Jha. Secure Virtual Machine Execution under an Untrusted Management OS [C]. 2010 IEEE 3rd International Conference on Cloud Computing.2010

[7] Zhang, F., Chen, J., Chen, H. and Zang, B. CloudVisor: Retrofitting Protection of Virtual Machines in Multi-tenant Cloud with Nested Virtualization. In ACM SOSP, 2011.

[8] Trusted Computing Group: TPM Main Specification Version 1.2.http://www.trustedcomputinggroup.org

[9] S. Berger, R. Caceres, K. Goldman vTPM: Virtualizing the Trusted Platform Module. In USENIX Security, 2006.

[10] Applied Data Security Group. What is Trusted GRUB [DB/OL]. http://www.prosec.de/trusted_grub.html

[11] George Coker. Xen Security Modules (XSM) [EB/OL]. National Information Assurance Research Lab. National Security Agency (NSA). 2007

[12] D. Williams, H. Jamjoom, and H. Weatherspoon. The Xen-Blanket: Virtualize Once, Run Everywhere. ACM EuroSys, 2012.

[13] Hidekazu Tadokoro, Kenichi Kourai, Shigeru Chiba. Preventing Information Leakage from Virtual Machines' Memory in IaaS Clouds. IPSJ Transactions on Advanced Computing Systems Vol.5 No.4 101–111.2012