TELKOMNIKA, Vol. 11, No. 11, November 2013, pp. 6935 ∼ 6942
ISSN: 2302-4046

A Virtual Machine Cloning Approach Based on Trusted Computing

Wei Ma*, Xiaoyong Li , Yong Shi , and Yu Guo
School of Computer and Information Technology, Beijing Jiaotong University
Beijing 100044, China
*corresponding author, e-mail: wei.ma1222@gmail.com

Abstract

Cloud computing has become more and more popular and the technology of rapid virtual machine (VM) cloning is widely used in cloud computing environment to implement the fast deployment of VMs to meet the need of bursting computational request. However, the security issue of VM cloning technology is not considered thoroughly in current approaches. This paper proposes a virtual machine cloning approach based on trusted computing which deals with memory and disk of a VM separately. This approach resolves three problems: the identities verification of involved servers; the attestation of source VM and destination VM; the protection of integrity of transmitted data.

Keywords: *cloud computing, virtual machine clone, trusted computing, vTPM*

1. Introduction

Cloud computing, as an emerging industry, has achieved great development. As the definition given by NIST (National Institute of Standards and Technology), there are three service models of cloud computing[1]:

- Software as a Service (SaaS). The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure.
- Platform as a Service (PaaS). The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider.
- Infrastructure as a Service (IaaS). The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources.

Virtualization technology is the foundation of cloud computing. Iaas, such as Amazon EC2[1], is able to provide a number of virtual machine instances for users with a smaller amount of physical machines. With this characteristic, users are able to distribute their work to different VMs. Amazon calls this feature "Elastic Cloud Computing (EC2)". However, this VM assign model is lack of agility which is very important when customers are eager to have one new VM or more new VMs to expand their computation scale. In EC2, instantiating new VMs is a slow operation and it always takes minutes[1]. Meanwhile, the newly instantiated VMs are typically created using replicas or templates provided by the CSP, which are initialed freshly and not aware of the user's computing environment. This lack of agility ,which forces users to apply more VMs to meet their computing needs or less VMs but take more time loss, restricts the availability and usability of cloud computing.

VM cloning technology was introduced to resolve the issue described above. With fast instantiate mechanism, VM cloning technology is able to leverage the resource assignment of cloud computing platform. Technologies, such as Copy-On-Write (COW), memory snapshot, are used to rapidly clone a runtime VM to one local or remote physical machine in the cloud computing

Received June 1, 2013; Revised July 30, 2013; Accepted August 10, 2013

cluster. Different from newly initialed VMs, there is a feature that the cloned VMs can totally inherit the memory states of the "parent" VM, which enables the cloned VM to be at user's service immediately without some additional configuration. For example, when a customer has a typical LMAP (Linux/MySQL/Apache/PHP) application stack deployed in a VM, and now he needs more VMs to meet his increasingly service request. If the CSP provides him with a series of traditional newly initialed VMs, he would have to install and configure software all he need to use, which will take more time to. But if VM cloning mechanism is introduced, the new VMs will be ready to work when they are delivered, which can reduce time cost of customer.

Three demands should be meet with VM cloning technology: rapid instantiation of a new VM; to dynamically adjust the assignment of resources in cloud computing environment; to answer gusty request computing request.

There are several current approaches of VM cloning which work well. However, the security issue is not considered thoroughly. During the process of cloning, although which takes very short time, there are risks such as leaking of sensitive information, illegal cloning, information stolen, etc. Hence, besides of the three demands mentioned above, the security issue should be considered as a basic principle. According to this issue, this paper raises three goals to meet: Remote attestation between two physical machines before cloning process. Identity verification between to be cloned VM and generated VM. Integrity measurement during cloning process.

1. Remote attestation between two physical machines before cloning process.
2. Identity verification between to be cloned VM and generated VM.
3. Integrity measurement during cloning process.

In our previous work[15], we discussed a virtual machine clone model. In this paper we expand our previous work and present a virtual machine cloning approach based on trusted computing, which deals with memory and hard disk separately, to enhance the security of VM cloning technology. This approach aims to meet the above goals to guarantee that there is no information leakage, stolen, or illegal clone. Based on Xen architecture, this approach relies on the concept developed in TCG (Trusted Computing Group) and take use of hardware elements such as TPM (Trusted Platform Module), and the emulated TPM for virtual machines vTPM is involved. And some experiments are made to observe the performance of our approach.

This paper is consisted of five sections. Section 2 depicts related work while section 3 briefly introduced architecture of vTPM, Xen and Trusted Computing. The virtual machine cloning approach is described in section 4. Section 5 illustrates the performance of our approach briefly. Finally, section 6 concludes this paper and proposes some perspectives.

2. Related Work

Generally speaking, there are three stages to describe the development of VM cloning technology. Firstly, the primitive method of cloning a VM was simply suspend-copy-restore a VM. User can suspend a targeted VM, and then restore states of this VM's memory and disk, and create a new VM using the restored. When the targeted VM and the cloned VM are on the same host, this method is simple and valid. But in network environment, it's hard to copy needed information from one host to another remote one, especially memory pages. Meanwhile, the efficiency of this approach is too low for cloud computing and with too high time cost.

VM cloning technology entered into the second stage when VM live migration went mature. Normally, VM live migration is used to ensure the high availability of VM applications. There are three steps when a VM is migrated: first, some unused memory pages are transmitted from source host to destiny host, in which some iteration algorithms are used to ensure the efficiency of the transmitting process; second, stop the source VM, and then transmit those memory pages which are in use at the first step; in the end, boot the destiny VM, and if there are mistakes in some memory pages, re-transmit them and after that, destroy the source VM. With this approach,

services and applications provided by the VM would not be stopped too long when the VM is migrating. Similarly, Remus uses an asynchronous VM replication on a remote host and heartbeat packet between two hosts to ensure the high availability of service. However, strictly speaking, VM live migration and Remus are not method of VM cloning, but some features involved in those two methods inspired the development of researches on VM cloning technology.

The third stage is almost VM cloning itself. Zayas[2] introduces the technology called copy-on-reference to transmit address space, which has been used in Snowbird[3] and Kozuch[4] to implement their VM suspend/restore operations. Potemkin[5] uses COW (Copy-On-Write) technology to clone VMs which are light but with very short life circle to deploy honey pot on a big scale IP address. Shadow memory page table plays a very significant role in Potemkin. Using shadow memory page table, it's convenient to allocate new memory page for a new VM and replace old out-of-time page. Also, shadow memory page table is used in SnowFlock[6], which aims to clone a batch of VMs in cluster or cloud computing environment with low latency. Compared with Potemkin, SnowFlock is able to clone a VM on a remote server while Potemkin cannot. Furthermore, paralleled cloning is introduced in SnowFlock, which reduces time cost significantly.

Moreover, VM cloning technology is also widely used in cluster. Emeneker[7] proposes dynamic clustering to provide and manage resources. Emulab[8] initializes dozens of nodes with virtualization technology involved, and muti-cast protocols defined by Frisbee[9] are used to distribute disk information to every single node.

3. Context and Assumptions

3.1. Trusted Computing

Trusted computing[10], which aims to resolve security issues in the foundation of architecture level, is defined and developed by TCG (Trusted Computing Group), which is sponsored by many vendors and organizations such as IBM, Intel, Compaq and Microsoft. It is defined as: one entity is trusted if this entity acts toward prospective goal with prospective actions. A secure cryptography co-processor called TPM (Trusted Platform Module), which is the core of Trusted Computing, enables that sensitive data such as cryptographic keys can be secured in shielded location. Generally, there are several manners used to enhance security of the protected system in Trusted Computing as followed.

Transitive Trust: Transitive Trust is used to assure the integrity of a platform. When a platform booted, the Root of Trust would determine the trust level of next functional modules acceptable of not, and then expand the trust boundary from the former Root of trust to include the next functional modules if the trust level is acceptable, and the trust will be transmitted to next level in this iterated process which leads to trust the entire system and build a trust chain from the Root of Trust to the top-level user space.

Attestation: Attestation is the process of proving the integrity of a platform. The measurements of a platform's integrity is restore in the PCRs (Platform Configuration Registers) in TPM, which can be digitally signed by the AIK (attestation identity key) of TPM to be attested the integrity of the platform.

Sealing Storage: Sealing Storage is used to protect the sensitive data with cryptography. A non-migratable asymmetric key and a set of PCR values are used to encrypt the sealing data which can only be decrypted on designated platform with matched PCR values.

3.2. Xen

Xen[11], which is developed and first released by the University of Cambridge Computer Laboratory in 2003, is a Hypervisor providing services that allow multiple computer operating systems to execute on the same computer hardware concurrently.

Xen is a native hypervisor which runs in a more privileged CPU state than any other software on the machine. Xen boots from a bootloader such as GNU GRUB, and then usually loads a para-virtualized host operating system into the host domain called domain-0 which is the only virtual machine which by default has direct access to hardware. From the dom0 the hypervisor can be managed and unprivileged domains (domain-U) can be launched[12].

The key features of Xen involved in this paper are the isolation and memory management. The isolation ensures that the different VMs are separated without stealthy shared information and the memory management, which using shadow page technology, ensures that the memory of VM is able to be replaced or refreshed quickly and effectively.

3.3. vTPM

vTPM[13], an exclusive software TPM equipped in each VM by virtualizing physical TPM chip, is designed to overcome the limitation that there is only one TPM chip on a physical server while there are several VMs running on it. Series of software components are used to ensure vTPM works like actual TPM chip and share one TPM. vTPM instance is a process named vtpmd which enables each VM run under TCG v1.2 standards. vTPM management tool is responsible for operations of vTPM instance such as creating, managing and canceling. And communications between a VM and its vtpmd are implemented by an emulated TPM module. Both vtpmd and vTPM management tool are running in domain-0.

Different with TPM, the vTPM owns one extended command set extended from TPM v1.2. Some vTPM management commands such as $TPM\_CreateInstance$, $TPM\_DeleteInstance$ and $TPM\_SetupInstance$, some vTPM utility commands such as $TMP\_TransportInstance$ are defined in this command set. Take advantage of this command set, we are able to generate and manage a new vTPM instance dynamically when cloning a VM.

3.4. Attacker model

In this paper, we have several assumptions. First, we assume that the VM_S which is going to be cloned is a VM running on a local server S_L; the VM is going to be cloned as VM_D which will run on a remote server S_R. And then we assume that there is an attacker $\mathcal{A}$, who is an owner of some VM but is malicious to the other VMs. $\mathcal{A}$ is assumed to be able to eavesdrop, modify, delete information over the network. In the process of cloning a new VM, we assume that $\mathcal{A}$ is interested in acquiring information or data that belong to other users to benefit itself. Furthermore, we assume that $\mathcal{A}$ is able to exploit vulnerabilities of software even hypervisor to perform malicious activities.

However, the capability of accessing physical machines in the cloud is not equipped by . So $\mathcal{A}$ is not able to modify or compromise the software stack or hypervisor on each physical machine. And the physical TPM chip is not accessible to $\mathcal{A}$ too. This assumption ensures that the physical platform is safe and trustworthy so that the attacker is not capable of violating the approach presented in section 4 by exploiting hardware components or privileged software.

4. Trusted Virtual Machine Clone Approach

4.1. Memory Cloning

Three functions are accomplished in memory cloning: first, before the cloning procedure begins, S_L and S_R should be able to authenticate the identity of each other; second, VM_S should be able to authenticate the identity of newly generated VM VM_D; finally, during the procedure, S_R should be able to authenticate the memory data transmitted from S_L.

There are two phases in memory cloning: phase 1 is an attestation process between servers and VMs, and phase 2 is integrity measurement of memory pages during the cloning process. Figure 1 shows the architecture of memory cloning.

Pahse 1.

The attestations between servers and VMs are accomplished in phase 1.Before the cloning begins, the local server S_L would verify the identity of the remote server S_R to avoid that the target VM is cloned on an untrusted platform. First, the TPM of S_L, TPM_L, will generate a nonce n1, and then TPM_L signs n1 with its AIK and sends it to S_R. After receiving the signed n1, the TPM of S_R, TPM_R, will generate a new nonce n2 and call the function TPM_Quote which provides the cryptographic hash of the PCRs values. The n1, n2 and hashed values of PCRs will be signed by the AIK of TPM_R and forward to S_L. At the last step of this phase, S_L will verify the

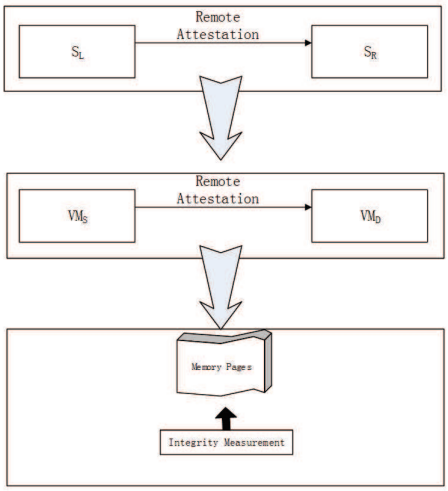

Figure 1. Virtual machine cloning architecture of memory

identity of S_R by checking the signature of the nonces and comparing the received PCRs values. Secondly, the function $TPM\_CreateInstance$ will be called to create a new vTPM instance for the new VM which will be generated to start the cloning process. After that, the new $vTPM_D$ which is owned by the new VM VM_D, will communicate to $vTPM_S$, the vTPM of VM_S. Normally, the values of PCR 1-7 of the newly generated vTPM are inherited from the TPM on the physical server, and the other PCRs will be extended in following process. Hence, in this process, only PCR 1-7 are filled with positive values and so only those PCRs will be involved. Like the attestation between servers, the VM_S will verify the identity of VM_D by checking the values of PCR 1-7 of $vTPM_D$ with $vTPM_S$. We don't give unnecessary details because the two procedures are too similar. After this attestation, the cloning process moves to phase 2.

The process of phase 1 is illustrated in Figure 2.

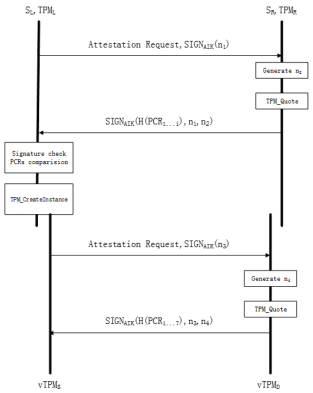

Figure 2. Phase 1 of memory cloning

Pahse 2.

In phase 1, by verifying the identities of involved entities, we can avoid that the VM is cloned onto unauthorized platform. In followed phase 2, we should consider the integrity of data being transmitted in the cloning process. As the attacker assumed able to modify the data over the network, so we should ensure that the VM_D receive correct memory data from VM_S. First, when VM_S is requested to transfer memory data to VM_D, the selected memory page will be set as

read-only in the shadow page table of VM_D in case that the memory page is modified during the transmit process. Secondly, the chosen memory page will be hashed as $H(MemoryPageData)$ and the hash value will be signed by the AIK of VM_S $SignAIK(H(MemortPageData))$ A packet that includes the hash value of the memory page, the signature and the memory page itself will be combined as $SignAIK(H(MemortPageData))||H(MemortPageData)||MemoryPageData$ and this packet will be sent to VM_D. In the end, after receiving the packet from VM_S, VM_D will verify the integrity of the memory page by verifying the signed hash value received. If the result of the integrity measurement shows trusted, VM_D will copy the received data to its own memory page newly allocated by S_R. The shadow page table of VM_D will be refreshed to add the new page table entry which is cloned from VM_S, and VM_S will be about to get ready for the transition of next memory page after setting its shadow page table to read-write mode. This process will be executed repeatedly until all memory pages are cloned. Moreover, to avoid the reduplicate transition of memory pages, there is a bitmap shared by S_L and S_R which indicates whether the page is transmitted or not by simply setting the corresponding bit of that page 1 or 0.

4.2. Hard Disk Cloning

After the clone process of memory pages, the runtime VM VM_D is generated on the remote server S_R. To maintain the availability of VM_D, the virtual disk should be cloned too.

Due to that before the cloning process start the identities of involved entities have been attested, so in hard disk cloning process we only need to consider the integrity of to be cloned virtual disk. Unlike virtual memory, virtual disks are always with big size and hard to be transferred over network. To reduce the overhead, we design a mechanism to clone virtual disk in hard disk cloning process. When VM_D is generated, an empty virtual disk is assigned to it and a bitmap is associated to it. This bitmap indicates the status of every data block whose size is defined by Xen and with this bitmap the virtual disks of VM_S and VM_D are logically connected. VM_D determines how to read/write data from/to its virtual disk by inquiring this bitmap. If VM_D needs to read or write data, it will check the corresponding bit with the block to access 0 or 1, where 0 means that the data block is still empty so that VM_D should access data from the virtual disk of VM_S and 1 means it can operate the data locally. When VM_D reads data from local virtual disk and meet a 0 in the bitmap, the data will be transferred from the virtual disk owned by VM_S over the network; when a 1 is met in the bitmap, the data will be read from the local disk. Something is different when the operation is not read but write. During the cloning process, VM_S execute the write operation to its virtual disk should be reflected to VM_D's virtual disk. So when VM_S writes data to its own virtual disk, VM_D will write the same content to the same block of VM_D's virtual disk. So the security issue goes to verify the integrity of data transmitted from VM_S before VM_D use the data. Simply, this process is like memory integrity verification: before the data is transmitted, it will be hashed and $vTPM_S$ will use its AIK to sign the hash value which will be sent to VM_D with hashed value and the data itself. By checking the hash value of received data, VM_D determines to trust the data or not.

This mechanism ensures the efficiency of the virtual machine cloning approach. The Figure 3 illustrates the process of hard disk cloning.

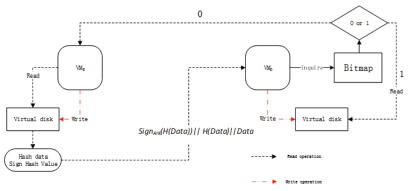

Figure 3. Hard disk cloning process

5. Performance

According to our approach, there are mainly two kinds of performance overhead. The first one is the performance overhead during the process of memory cloning and the second one is the hard disk cloning overhead. We made series of experiments based on snowflock[6], and work on the extra overhead of security mechanism.

During the process of memory cloning, the extra overhead is always brought by the computing of hashing memory pages, and in the process of hard disk cloning, the extra overhead is the computing of hashing the virtual hard disk.

We make experiment to calculate the time overhead when hashing memory pages and virtual hard disk. Due to the memory pages are copy-as-need, it's not necessary to copy every memory page to the cloned new virtual machine. We make several contrast for several different situations to verify the different time overhead. The result of experiment is illustrated in figure 4.

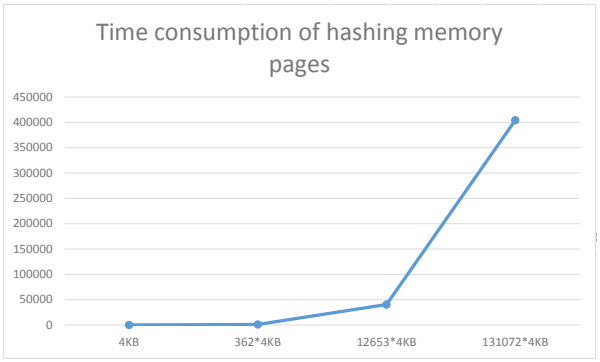

Figure 4. Time consumption of hashing memory pages

As shown in Figure 4, time overhead of hashing a single 4KB memory page is very tiny, but at worst when we need to hash every memory page as the amount of 131072, the time over head will be tremendous. So it is important to reduce the number of necessary memory pages.

The hard disk hashing overhead is simpler than the memory hashing. Because there are only one hard disk associated with the virtual machine.

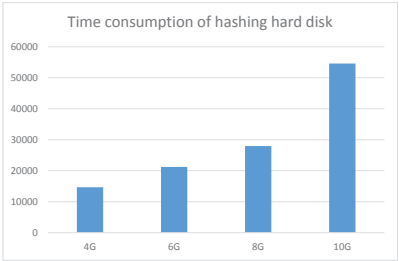

Figure 5. Time consumption of hashing hard disk

The virtual hard disks with different size hashing time overhead is illustrated in figure 5.

6. Conclusion

In this paper, we introduce the related background of VM cloning technology and the security issue about it is analyzed. Based on the current existed problems, we present a virtual machine cloning approach based on trusted computing which deals with memory and hard disk of VM separately. With this approach, three problems are resolved: first, the local server is able to verify the identity of the remote server before the cloning process begins; second, the newly

generated destiny VM can attest itself to the source VM to be cloned; third, the integrity of the transfer data is protected. This approach is capable of working with the Xen hypervisor and vTPM architecture which are widely used in current cloud computing environment. Furthermore, due to the VM cloning technology has a brilliant perspective of future other kind cloud computing such as Hadoop, we believe that the security issue in cloning technology will be more prominent and our approach will play an important role in the develop process of cloud computing.

Acknowledgement

This paper is supported by Research Fund for the Doctoral Program of Higher Education of China (RFDP20120009110007), Program for Innovative Research Team in University of Ministry of Education of China (IRT201206) and Program for Science and Technology Research and Development of Ministry of Railway of China (2012X010-B).

References

[1] Mell P, Grance T. "The NIST definition of cloud computing (draft)." NIST special publication, 2011, 800: 145.
[2] Zayas E. "Attacking the process migration bottleneck." ACM SIGOPS Operating Systems Review. ACM, 1987, 21(5): 13-24.
[3] Lagar-Cavilla H A, Tolia N, De Lara E, et al. "Interactive resource-intensive applications made easy." Middleware 2007. Springer Berlin Heidelberg, 2007: 143-163.
[4] Kozuch M, Satyanarayanan M. "Internet suspend/resume." Mobile Computing Systems and Applications, 2002. Proceedings Fourth IEEE Workshop on. IEEE, 2002: 40-46.
[5] Vrable M, Ma J, Chen J, et al. "Scalability, fidelity, and containment in the potemkin virtual honeyfarm." ACM SIGOPS Operating Systems Review. ACM, 2005, 39(5): 148-162.
[6] Lagar-Cavilla H A, Whitney J A, Scannell A M, et al. "SnowFlock: rapid virtual machine cloning for cloud computing." Proceedings of the 4th ACM European conference on Computer systems. ACM, 2009: 1-12.
[7] Emeneker W, "Stanzione D. Dynamic virtual clustering." Cluster Computing, 2007 IEEE International Conference on. IEEE, 2007: 84-90.
[8] Hibler M, Ricci R, Stoller L, et al. "Large-scale virtualization in the Emulab network testbed." USENIX 2008 Annual Technical Conference on Annual Technical Conference. 2008: 113-128.
[9] Hibler M, Stoller L, Lepreau J, et al. "Fast, scalable disk imaging with frisbee." Proc. of the 2003 USENIX Annual Technical Conf. 2003: 283-296.
[10] TCG. http://www.trustedcomputinggroup.org
[11] Barham P, Dragovic B, Fraser K, et al. "Xen and the art of virtualization." ACM SIGOPS Operating Systems Review, 2003, 37(5): 164-177.
[12] Xen Wiki. http://en.wikipedia.org/wiki/Xen
[13] Perez R, Sailer R, van Doorn L. "vTPM: virtualizing the trusted platform module." Proc. 15th Conf. on USENIX Security Symposium. 2006: 305-320.
[14] EC2:Amazon elastic compute cloud.http//aws.amazon.com/ec2/
[15] [15] Wei Ma, Xiaoyong Li, Yong Shi, Yu Guo, "A Virtual Machine Cloning Model in Cloud Computing Based on Trusted Computing." JCIT, Vol. 8, No. 4, pp. 490 497, 2013