VNSS: A NETWORK SECURITY SANDBOX FOR VIRTUAL COMPUTING ENVIRONMENT

Gao Xiaopeng, Wang Sumei,Chen Xianqin

State Key Laboratory of Software Development Environment BeiHang University
Beijing 100191, China
gxp@buaa.edu.cn ,(wangsumei.buaa, chenxianqin)@gmail.com

ABSTRACT

With the number of applications running upon the virtualized system increased, the virtual network circumstance becomes more and more complicated; the consequent security problems thereby have been a concern for industrial and academic fields. However, the current solutions are mostly confined to the enforcement of several patchy-works on system which still requires proficient hacking skills for administrators and cannot ensure continuous protection for VM, resulting in potential security risks. In this paper we present a framework (VNSS) which provides both guarantee of distinct security level requirement and full-lifecycle protection for VM. We have implemented a prototype system based on Xen hypervisor to evaluate our framework. The experiment results demonstrate that our framework can provide continuous protection for virtual network environment.

***Index Terms*—**System Virtualization, Network Security, Virtual Machine, Stateful Firewall, Lifecycle

1. INTRODUCTION

System virtualization technology allows multiple operating systems to run on one computer. A single server can sustain a large number of applications, which raises the amount of VMs in virtual machine monitor (VMM) and adds up complexity of virtual network within virtualized system.VMM, also named hypervisor, is a software layer between the operating system and the computer hardware [1]. VMM manages all the physical resources, provides a set of virtual platform interfaces for each VM, such as virtual network interface cards (VNICs), virtual CPU, etc, supervises VMs access to the hardware resources and handles each virtual machine's communication with the CPU, the storage medium and the network.

The virtual network of VMM is implemented by virtual switches or bridges which connect the VNICs to physical network interface card of the host. Therefore, all the traffics in virtual network are visible for each VM that shares the same physical data links, which may potentially conduct security risk. In addition, most VMs employ virtual network connections to communicate with each other which cannot be tracked by existing network-based security systems [3], moreover, when a VM is moved from one host to the other, the target host might not have the security necessary to protect the VM.

Accordingly, the security of virtual network has been widely concerned by industrial and academic fields. Previous works on secure virtual network usually perform several patchy-works on host system, which are based on firewall scripts and existing network virtualization technologies, such as VLAN and VPN, etc. Nevertheless, these approaches have vulnerabilities to some extent, such as, lack of visibility into virtual machine traffic, no guarantee for the needs of distinct security level of each VM, no assurance of continuous protection, etc.

In this paper, we focus on the network security of virtualized system, and aim at providing a network security sandbox for the virtual network. We present a framework which not only guarantees VM with distinct security level by customizing security policies (SPs) for VM but also provides continuous uninterrupted protection through live migration. The implementation of our framework is based on stateful firewall technology and some userspace tools, such as iptables, conntrack-tools, etc. The main contributions of our framework are as follows:

1) Consider that each VM might have different security level needs.
2) Provide full-lifecycle protection for VM, that is, ensure network security policies are in place throughout the virtual machine lifecycle.

The rest of the paper is divided into the following sections: Section 2 summarizes the related work; Section 3 describes the prototype of the framework; Section 4 details the implementation and evaluation of the framework; Section 5 comes to a conclusion of our work.

2. RELATED WORK

VMware introduces VMware vShield Zones that in the virtual logical zones creates logical zones, each of which represents a distinct level of security. VMware vShield Zones leverages existing virtual network technologies, such as virtual switches and VLANs, to provide runtime visibility and enforcement of virtual network traffic [6]. VMware

978-1-4244-8886-5/10/$26.00 ©2010 IEEE

vShield Zones distinct the level of security based on zones. However, virtual machines in a same zone might need a different security level. In addition, it is a commercial product which lacks scalability for open source system.

Trust Virtual Domains (TVDs) [2, 5] are the framework for implementation of virtual networks in virtualized system, such as virtual data centers where all the virtual machines share the physical resources. Virtual machines which are assigned in the same TVD can access the resources belonging to the TVD without executing additional security protocols, while the resources belonging to different TVDs are strictly separated. [2] presents a secure network virtualization framework for realizing the abstraction of Trusted Virtual Domains (TVDs). The framework connects groups of related virtual machines which are running on separate physical, and provides security guarantee for the VMs. It tracks the inner connections by delivering traffic from internal physical appliances to external ones.

The existing network security technologies include the stateful firewall [4], VLAN tagging and VPNs, etc. Stateful firewall keeps the track of the network connections (such as TCP streams and UDP communication) travelling through it. It distinguishes legitimate packets for different types of connections and only accepts packets that match a known connection state. Most existing network security engines (SEs) are stateful, and all flows through them have security context (SC). Network security engines filter traffic based on both SC and packets' contents [4].

3. FRAMEWORK OF NETWORK SECURITY SENDBOX

In this section, we describe our framework, namely VNSS. It provides lifecycle protection for VM and ensures continuous uninterrupted protection for virtual network. Figure 1 illustrates that the framework consists of security sandbox controller (SSC), security policies create agent (SPCA), virtual machine create agent (VMCA), virtual machine migration agent (VMMA), security context migration agent (SCMA) and security policies migration agent (SPMA).

The SPCA is in charge of SPs creation for the virtual machine during its creation, while the VMCA is responsible for virtual machine instance. The VMMA, which takes charge of the migration of virtual machine, migrates virtual machine instance from the source host to the target one. Simultaneously, the SCMA synchronizes the security context related to the virtual machine and SPMA relocates the SPs which belong to the virtual machine from the source to the target. These five agents are all scheduled by SSC.

As is shown in figure 2, the framework workflow can be divided into three phases: VM creation phase, VM migration phase and VM destruction phase.

In VM creation phase, SSC loads and analyzes a modified virtual machine configuration file, and then invokes the VMCA to create the virtual machine instance and the SPMA to generate SPs for the virtual machine.

In VM migrate phase, SSC triggers the VMMA to migrate virtual machine instance, the SCMA to synchronize the related security context of the virtual machine, and as well the SPMA to resume the SPs on the destination.

In VM destruction phase, SSC destroys the virtual machine instance and removes the related security polices from security engines.

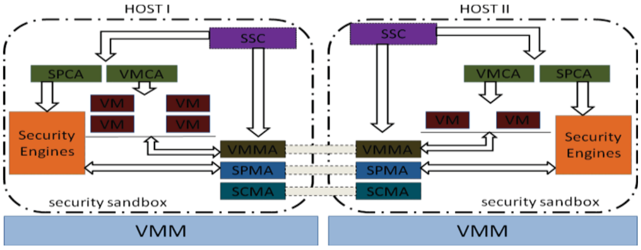


Figure 1. Architecture of the framework

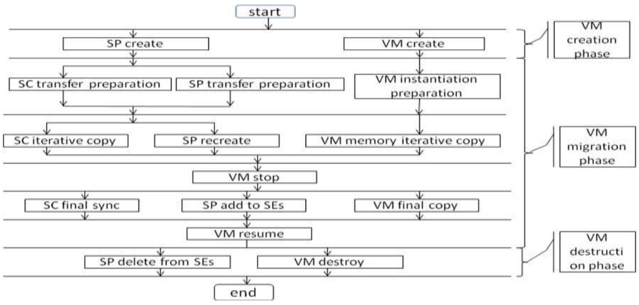


Figure 2. The workflow of the framework

4. EVALUATION

In this section, we describe the implementation of the prototype system and evaluate our framework under real application circumstances.

4.1 Prototype system

4.1.1 Overview

The prototype of our framework is implemented on xen hypervisor. Xen developed by University of Cambridge Computer Laboratory is an open source and splendid virtual machine software. The design of Xen hypervisor follows the isolation mechanisms and policies [1]. In charge of resources allocation, domain scheduling and physical device access control, the hypervisor (VMM) authorizes some tasks to domain 0 which is a privileged VM.

According to Figure 1, there are six components in our framework, which are VMCA, SPCA, VMMA, SCMA, SPMA and SSCA. We implement our prototype system by using current stateful firewall technology and some assorted userspace tools, such as iptables, xm commands program and conntrack-tools [8].

Iptables is the user space command line program used to configure Linux packet filtering rule set. Xm commands program is a Xen management user interface for managing Xen guest domains and the daemon. Conntrack-tools open source project is a set of userspace tools for Linux that allow system administrators to interact with connection tracking system [9] and allow user to synchronize the state among several replica firewalls. In our prototype, the SCMA synchronizes the VM related security context set in incremental synchronization mode between the source and the target during VM migration.

4.1.2 Implementation

The VMCA is implemented by re-implementing the xm commands program. The implementation of the SPCA is based on iptables and bash shell script. The VMMA is implemented by applying live migration tools and Xend daemon. The SCMA is achieved through applying the conntrack-tools open source project [8]. The SPMA is implemented by a bash shell script that is based on iptables. The SSC is implemented by a bash shell script based command too, and acts as the user interface for our prototype and parses command parameters and triggers the appropriate action.

SPCA firstly creates SPs to configure a stateful firewall which is embedded as the security engine in hypervisor.

VM creation phase: The VMCA creates VM instance, and SPCA parses the security needs of the VM and generates the related SPs for the VM so as to customize the security level for the VM as needed. The SSC manages the VMCA and SPCA working in parallel way to ensure that the VM related SPs becomes effective before the virtual network interface card of virtual machine is enabled.

The VM migration phase: The VMMA migrates VM instance from the source host to the target one, simultaneously, the SCMA synchronizes the VM related security context to the target and the SPMA relocates the VM customized SPs to the target. The SSC schedules the VMMA, SCMA and the SPMA working in parallel way to provide uninterrupted application and guarantee full-lifecycle protection for the migrating VM.

The VM destruction phase: The SSC analyses the configuration of the VM and deletes the VM related SPs from the security engine.

4.2 Experiments

In the following experiments, we aim to evaluate that whether our implementation can satisfy distinct security needs for each VM and provide full-lifecycle protection for VM. Experiments are based on the test-bed demonstrated in Figure 3.

4.2.1 Guarantee distinct security level

The following Test case (T1) aims at validating that the framework can guarantee distinct security level for each VM as needed.

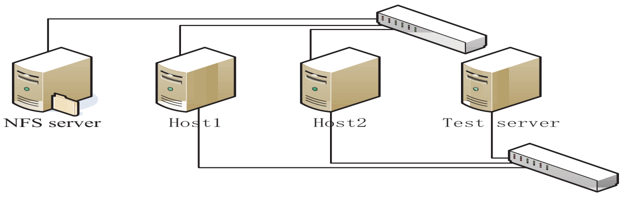
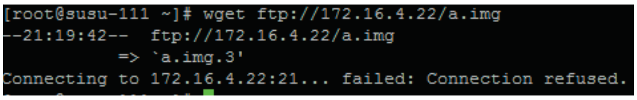


Figure 3. Test-bed

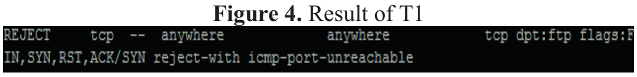
Test case1 (T1): Specify a demand of protecting FTP application in profile. Enable both VMCA and SPCA to create a VM with a FTP server installed on one of the two Hosts. The FTP server provides a 10GB sparse image file for downloading. After the VM creation, the FTP client on the test host tries to download the file from the FTP server.

Test results: Figure 4 demonstrates that the FTP application on the VM is protected after the VM is created. Figure 5 illustrates that the iptables rules for protecting the FTP application are right in the security engines when the VM is created. The results manifest that our implementation can guarantee distinct security level for each VM by customizing the needs in the profile, meanwhile, the security protection for the VM becomes effective immediately after the creation of the VM.



```
[root@susu-111 ~]# wget ftp://172.16.4.22/a.img
--21:19:42--  ftp://172.16.4.22/a.img
           => `a.img.3'
Connecting to 172.16.4.22:21... failed: Connection refused.
```

Figure 4. Result of T1



```
REJECT     tcp  --  anywhere             anywhere            tcp dpt:ftp flags:F
IN,SYN,RST,ACK/SYN reject-with icmp-port-unreachable
```

Figure 5. Rule for protecting FTP application

4.2.2 Guarantee full-lifecycle protection

The test case 2 (T2) and the test case 3 (T3) are enforced to prove that our implementation can provide uninterrupted application, while the test case 4 (T4) and the test case 5 (T5) are actualized to validate that our implementation can ensure protection for VMs even during migration.

For T2, T3, T4 and T5 in Table 1, the SPCA configures the firewall for hypervisor as follows: if the status of the connection is established, the packet will be accepted, otherwise the packet's content will be inspected; if the packet is a handshake SYN packet for telnet service, it will be accepted; if the packet cannot match with any rule tagged with accept action, it will be dropped.

For each entry of Table 1, we take the following steps:

1) Starts a FTP server on the VM and the server provides a 10GB sparse image file for downloading.
2) Downloads the image file from the FTP server by a FTP client on the test server.
3) Runs a script which measures the FTP client's throughput.
4) Starts a migration process.

Test Results: In Figure 6, the left graph demonstrates that when the SCMA is disabled, the downloading will be stopped after migration, because the stateful firewall of the target cannot recognize the connection used by FTP client and drop all the connection of it; the right graph shows that when the SCMA is enabled, the downloading will be continued after a short downtime. In Figure 7, the graph on the left illustrates that when the SPMA is disabled, as the VM related SPs cannot be migrated to the target together with the VM, the VM related SPs lose its effectiveness after migration, and the right graph evidences that when the SPMA is enabled, the VM related SPs are in place all the time.

The experiment results reveal that the needs of security level for each VM can be specified in its profile, and can be guaranteed by generating related SPs. Meanwhile the related SPs are in place throughout the VM full-lifecycle, form the moment the VM is brought online until it is finally destroyed.

Table 1 Test cases

Test case	configuration
T2	Create a VM without related security policies; VMMA enabled, SCMA disabled, SPMA enabled.
T3	Create a VM without related security policies; Start a FTP server on the VM; VMMA enabled, SCMA enabled, SPMA enabled.
T4	Use the VM created in T1; VMMA enabled, SCMA enabled, SPMA disabled.
T5	Use the VM created in T1; VMMA enabled, SCMA enabled; SPMA enabled.

5. CONCLUSION

In this paper, we identify the vulnerabilities of current solution of virtual network security. Correspondingly, we propose a framework (VNSS) which aims at providing both guarantee of security level requirement and full-lifecycle protection for VM. The prototype of the framework is implemented on Xen hypervisor by using existing stateful firewall technology, and wrapping the existing open source tools. The experiment results attest that our framework provides a network security sandbox that can guarantee VM's requirement of security level by customizing related SPs for VM and ensure the VM related SPs are in place throughout the VM full-lifecycle and even provide continuous uninterrupted protection through live migration.

6. ACKNOWLEDGEMENTS

We would like to thank our colleagues for their comments and suggestions. This work was partially supported by the fund of the State Key Laboratory of Software Development Environment (No.SKLSDE-2009ZX-02), the grant from China Aviation Science Fund (No.20081951) and the grant from the National High Technology Research and Development Program of China (863 Program)(NO.2007AA01Z183).

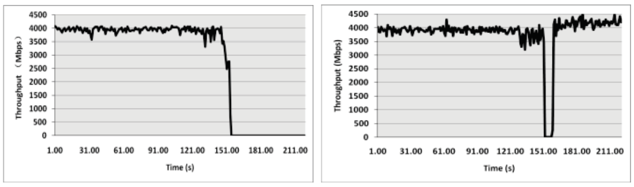


Figure 6. FTP throughput of T2&T3

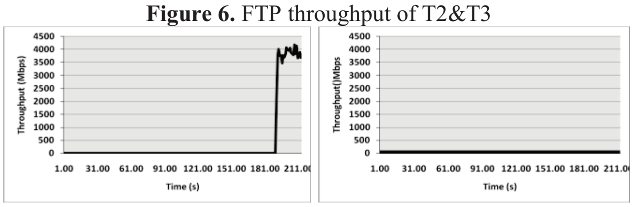


Figure 7. FTP throughput of T4&T5

7. REFERENCES

[1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the Art of Virtualization," *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP)*, Bolton Landing, Lake George, New York, October 2003.

[2] S.Cabuk, C.Dalton, H.V.Ramasamy, and M. Schunter, "Twards Automated Provisioning of Secure Virtualized Networks," I*n Proc. 14th ACM Conference on Computer and Communications Security (CCS-2007),*235-245, October. 2007.

[3] S.J.Vaughan-Nichol, "Virtualization Sparks Security Concerns," *Computer*, vol.41, no.8,13-15, Augeust 2008.

[4] M.G.Gouda, A.X.Liu, "A model of stateful firewalls and its properties," *Dependable Systems and Network*s, *2005*, DSN 2005, Proceedings, pp.128-137, 28 July 2005.

[5] A. Bussani, J.L.Griffin, B.Jansen, K.Julisch, G.Karjoth, H.Maruyama, M.Nakamura, R.Perez, M.Schunter, A.Tanner,L.van Doorn, E.V.Herreweghen, M.Waidner, and S.Yoshilhama, "Trusted Virtual Domains: Secure Foundation for Business and IT Services," *Research Report RC 23792*, IBM Research, November 2005.

[6]VMware, VMwarev Shield Zones. http://www.vmware.com/files/pdf/VMware-vShield-Zones-DS-EN.pdf

[7] C.Clark, K.Fraser, S.Hand, J.G.Hansen, E.Jul, C.Limpach, I.Pratt, and A.Warfield, "Live Migration of Virtual Machines," *In Proceedings of the 2nd ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Boston, MA, May 2005.

[8] P.N.Ayuso, The connrack-tools user manual, http://conntrack-tools.netfilter.org/manual.html, 2008

[9] Pablo Neira Ayuso, "Netfilter's connection tracking system," *The Magazin of USENIX*, vol.31,no.3 (Berkeley, CA:USENIX Association,2006, pp40-45).