SELinux-based Secure Server Virtualization

Xunyi Ren,Yang Yu
School of Computer
Nanjing University of Posts and Telecommunications
Nanjing, China
renxy@njupt.edu.cn

Abstract—Server virtualization is to create and run several independent operation systems at the same time to maximize the utilization efficiency and flexibility of IT resource, the most serious security problem of which, however, is the unauthorized access of unsafe virtual machines. This paper presents a SELinux-based secure server virtualization method against the issue. With SELinux access control, virtual machine process is isolated from system processes by setting up different type labels and access control policies. What's more, different virtual machine processes can be independent with each other by introducing Multi Category Security (MCS) protection mechanism, thus to achieve secure access of server virtualization. Finally the realization of the method is presented.

Keywords-server virtualization; security; SELinux; access control

I. INTRODUCTION

Numbers of virtual machines (VMs) can be running on the same hardware platform with server virtualization technology [1], which brings serious security hidden troubles [2] as well as convenience. As the hardware platform is shared, a malicious VM may interfere with others on the same platform by stealing and modifying the private data and even making it crash.

Currently the security issues of server virtualization are as follows. VM Escape. Virtual machine monitor (VMM) provides a unified resource abstraction for VMs with new security hidden troubles introduced by resource sharing technology. VMM theoretical model is incomplete and attacks based on the potential vulnerability of VMM can easily result in VM overflow, also known as VM escape. Namely attackers can get the access to the host system and other VMs running on the host, which is considered the most serious threat to VM security. VM Hopping: a VM can monitor another and even has the access to the host. Suppose two VMs A and B are on the same host, an attacker on A can get the access to B by obtaining the IP address of B or access rights of the host. The attacker monitors the traffic of B and changes its status from running to offline to bring about communication interrupt by traffic attack or modifying the configuration file. The communication would need to be restarted when the connection being reestablished. Troubles brought about by the changes of network architecture. As the most changed part during server virtualization, the changes of network architecture would correspondingly bring special security issues. In traditional mode, there is a set of independent security protection products on each physical machine or server as well as products such as firewall, gateway safety protection equipments and WAF deployed at the periphery. However, new network model is used in virtualized data center, where dozens of operating systems or application programs are deployed on physical server in the form of VMs so that the hardware resources are shared and external networks can be imperceptible to the network traffic among the VMs. When a VM is of trouble, the security threats would spread to other VMs through network. Defects of DoS. Since resources (like CPU, memory, hard disk and network) are shared by VMs and the host in virtualized mode, DoS attack may be taken to a VM to obtain all the resources of the host and then all the requests from customers will be rejected for lack of accessible resource.

SELinux is security-enhanced Linux with flexible and fine-grained mandatory access control (MAC) implemented on Linux system. Process privileges can be limited to minimum and the integrity and confidentiality of process and data can be protected with the integrated strong MAC mechanism. This paper presents SELinux-based secure virtualization thus to improve the security performance of server virtualization.

II. INTRODUCTION OF SELINUX

Traditional Linux operating system generally use discretionary access control (DAC) to ensure system security, which leads to many security hidden troubles. First, there is a privileged user root. Anyone with root privileges can do whatever he wants for the entire system. Secondly, the division of file access rights is not detailed enough. There are only "owner", "owned group" and "others" in Linux system and there are no ways to make further division of the users in "others". Finally, privilege upgrades of SUID program. It's easily to be exploited by attackers if there is vulnerability in a program with SUID set.

To solve the hidden troubles, the U.S. National Security Agency (NSA) developed the security-enhanced Linux, namely SELinux [3]. MAC policy is carried out in SELinux kernel in accordance with the principle of least privilege to make that user programs and system servers only get the minimum number of privileges to meet the needs of performing tasks, with which the potential harm would be greatly weakened or eliminated when user programs and system daemons are being infringed. MAC is without the concept of super user root and irrelevant to the shortcomings of traditional Linux security policy which and MAC are

Published by Atlantis Press, Paris, France.

independent of each other. SELinux is complementary to the existing security accomplishment of Linux.

All OS access control [4] is based on a type of access control attribute of associated objects and subjects, where access control attribute is called security context in SELinux and all objects (like files and sockets) and subjects (processes) have associated security context. Each security context consists of three parts which are user, role and type identifier. As shown in figure 1, the marked part is the security context of file "install.log", where "root", "object_r" and "user_home_t" respectively the user, role and type identifier.

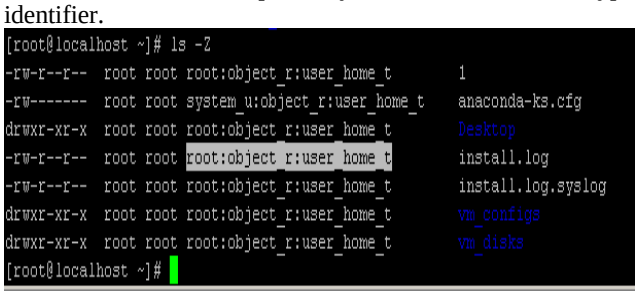

Figure 1. SELinux security context

In SELinux, access control attribute is always a security context and all the objects and subjects have an associated one. However the access mode in standard Linux is based on process user/group ID and file where user/group ID is accessed or denied. Since type enforcement is the main access control in SELinux, the type identifier of security context determines the access rights. The contrast of access control attribute between standard Linux and SElinux is summarized in table 1.

TABLE I. THE CONTRAST OF ACCESS CONTROL BETWEEN STANDARD LINUX AND SELINUX

	Standard Linux	**SELinux**
process security attributes	a real and effective user and group ID	security context
object security attributes	access mode, file user and group ID	security context
base of access control	file-based access mode of user and group ID	allowed access between process type and file type

III. SELINUX-BASED SECURE SERVER VIRTUALIZATION SCHEME

SELinux access control models include type enforcement (TE), role-based access control (RBAC) and optional multi-level security (MLS) model, which are integrated into a unified access control model where TE and RBAC are highly configurable. TE is the most basic model in SELinux on which other models are built based. A system is considered a set of subjects and objects in TE model and all of them are identified with types. In SELinux, all access must be expressly authorized and any access is not allowed by default regardless of Linux user or group ID, which means that there is not a super user. TE model affirms an operation permission of a subject on an object with "allow" sentence thus to meet the need of least privilege. SELinux also provides the TE-based RBAC model where the types the process can change into can be limited by the role identifier of the process security context. So the policy writer can create a role and allow it to be changed into a domain type (assume that the change is allowed in TE policy) so as to define the role limitations.

Every object in SELinux has been given a security context containing user, role and type, and TE is the most basic and important access control mode. That different processes have different operation rights on different objects is provided in SELinux access control policy which has been customized by system administrator and can't be changed by common users. Users even if the root user can't arbitrarily get the access to all kinds of objects if the corresponding "allow" policy is not defined in TE model, which indicates the MAC of subjects on objects. Therefore, according to SELinux security policy, independent processes can be isolated from each other and a process is not given the access to the files not needed, which fully reflects the security features of least privileges and separation of privileges and responsibilities [7].

With SELinux in virtualized server [5], VM processes can be separated from other application and system processes by setting up different type labels and access control policies. What's more, different VM processes don't influence each other with the introduction of multi-category security (MCS) protection mechanism. In other words, a VM process only has the access to the corresponding image file and shared files but not other system files and image files with SELinux, where VM image files are unauthorized to other processes, thus ensuring the security of virtualized server.

Before virtualization, different servers are physical separated from each other and the attacks suffered are mainly from network, as shown in figure 2. Thus servers can be effectively protected with intrusion detection tools and antivirus software. However, shown in figure 3, numbers of VMs run in the same server after virtualization so that a malicious VM can directly attack other VMs even the server.

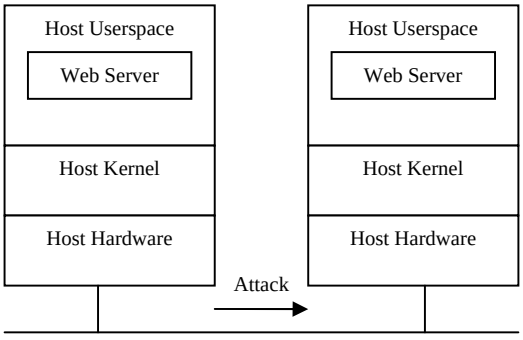

Figure 2. servers before virtualization

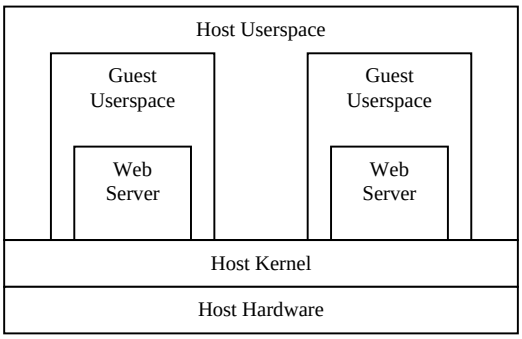

Figure 3. servers after virtualization

Secure server virtualization is to make that each VM is separate from other VMs and the server. A VM process only has the access to the corresponding image file but not other system files, VM image files unauthorized to other system processes. As shown in figure 4, VM 1 and its image file are respectively endowed with a type label like virtd_isolated_t:1 and virt_image_t:1. Similarly, VM n and the image file are endowed with virtd_isolated_t:n and virt_image_t:n. It's determined in SELinux access control policy that files with virt_image_t:n are only allowed for the process with virtd_isolated_t:n but not other processes.

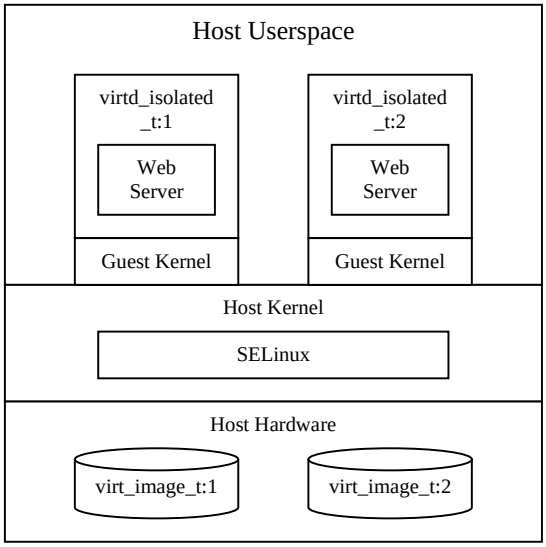

Figure 4. structure of secure server virtualization

During implementation, a plug-in structure sVirt is added to libvirt in RHEL6, namely making SELinux supported in libvirt. We use svirt_t and svirt_image_t to respectively label the VM process and image file. SELinux policy dictates that process with svirt_t can read/write the files and devices with svirt_image_t.

Change VM security context from three-section "user:role:type" to four-section "user:role:type:MCS", where a sensitive level s0 [8] and data category (c0.c1023) are defined in MCS. Libvirt can dynamically distribute a different random MCS label to each VM and the associated image file, thus ensuring uniqueness of MCS field which can also be manually specified. It's provided in SELinux that a VM process only has the access to the image file with the same MCS field, thus ensuring that VMs would not attack each other. As shown in figure 5, two VMs xp and xp3 are created on the server, and the security context of the VM processes and image files are shown in figure 6, figure 7 and figure 8.

```
[root@yuySELinux ~]# virsh list
 Id 名称                 状态
----------------------------------
  5 xp                   running
  6 xp3                  running

[root@yuySELinux ~]#
```

Figure 5. VM status of the server

```
[root@yuySELinux ~]# ps -eZ |grep qemu
system_u:system_r:svirt_t:s0:c221,c631 3006 ?  00:20:46 qemu-kvm
system_u:system_r:svirt_t:s0:c10,c100 3042 ?   00:20:39 qemu-kvm
[root@yuySELinux ~]#
```

Figure 6. security context of VM processes

```
[root@yuySELinux ~]# ls -Z /var/lib/libvirt/images/xp.img
-rw-------. root root system_u:object_r:svirt_image_t:s0:c221,c631 /var/lib/libv
irt/images/xp.img
[root@yuySELinux ~]#
```

Figure 7. security context of the image file of xp

```
[root@yuySELinux ~]# ls -Z xp3.img
-rw-r--r--. root root system_u:object_r:svirt_image_t:s0:c10,c100 xp3.img
[root@yuySELinux ~]#
```

Figure 8. security context of the image file of xp3

The MCS field is dynamically generated in xp and manually specified in xp3, shown in figure 9.

It can be seen that, for the two VMs, the security context including type label and MCS field of the VM process and image file are both consistent with each other. In addition, MCS currently support about 500000 labels, that is, MCS fields of 500000 VMs can be different from each other, which is sufficient to meet the practical needs.

```
[root@yuySELinux ~]# vi xp3.xml
    <sound model='ich6'>
      <alias name='sound0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/
>
    </sound>
    <video>
      <model type='vga' vram='9216' heads='1'/>
      <alias name='video0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/
>
    </video>
    <memballoon model='virtio'>
      <alias name='balloon0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/
>
    </memballoon>
  </devices>
  <seclabel type='static' model='selinux' relabel='yes'>
    <label>system_u:system_r:svirt_t:s0:c10,c100</label>
    <imagelabel>system_u:object_r:svirt_image_t:s0:c10,c100</imagelabel>
  </seclabel>
```

Figure 9. manually specified MCS filed of xp3

IV. CONCLUSION AND PROSPECT

The access control of server virtualization is security-enhanced with SELinux in this paper. Setting different security contexts to different VMs and giving the same MCS field to the process and image file of the same VM, each VM can be separated from the server and other VMs, thus to ensure the security of the server. However, since only TE model and single-sensitivity MLS model of SELinux are used in this paper, to further strengthen the security of server virtualization together with RBAC model and multi-sensitivity MLS model is the focus of future research.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China (61073188), China Postdoctoral Science Foundation (20100471355).

REFERENCES

[1] Sehgal N K, Ganguli M. Applications of virtualization for server management and security.Proceedings of IEEE International Conference on Industrial Technology. Mumbai, India, 2006: 2752-2755.

[2] Jing Fang, Hao Wu, Songlin Bai. Virtualization Security Issues of Cloud Computing. Telecommunications Science, 2012, 2(04): 135-140.

[3] Mayer, Frank, Karl. SELinux by Example. USA: Addison-Wesley, 2005: 157-198.

[4] Wenjie Tu, Haibing Guan, Caiying Bai. Access Control Features of Enhanced Linux File System. Computer Applications And Software, 2006, 23(2): 117-119.

[5] Zhe Ma, Xi Yu, Ao Yuan, Xiaolei Yi, Qingwen Shen. Analysis of Xen Security Mechanism. Information Network Security, 2011, 10(11): 31-35.

[6] Hua Jiang. Research of Role-based Access Control Technology in SELinux[D]. Wuhan: Huazhong University of Science and Technology, 2009.

[7] Jie Zeng. Research of Linux-based Least Privilege Management Mechanism and Automation of Security Test. Beijing: Beijing Jiaotong University, 2006.

[8] Yang Zhang. Analysis Method of SELinux Security Policy Information Flow with Sensitive Label. Chinese Journal of Computers, 2009, 4(4): 709-720.