SeVMM: VMM-based Security Control Model

Chen Wen-zhi
Computer Science College, Zhejiang University
chenwz@zju.edu.cn

Zhu Hong-wei
Computer Science College, Zhejiang University
griffithzhw@hotmail.com

Huang Wei
Computer Science College, Zhejiang University
jeffrey-h@163.com

Abstract

The security problem became more severe since the security requirement of different applications may conflict with the others in distributed application or grid computing. Virtualization technology can improve the system's security, but did not satisfy the requirement of virtual resource sharing and inner-domain communication in distributed services and grid computing. By dividing the virtual resources into sharing virtual resources and normal ones, SeVMM provided secure mechanism for inter-domain communication control, which formed the base of multi-level security control model for virtual machine monitors, operating systems and applications. Case study and application showed that SeVMM improved the system's security without causing significant performance penalty.

1. Introduction

With the development of Web applications, the distributed network[1] security technology has become very important. One case for distributed environment is to deploy different applications on different platforms, while another case is that, with virtualization technology, different applications on the same SeVMM which should provide strong isolation[2] between applications. VMM has to provide applications with communication and resource sharing control mechanisms[3]. The traditional operating system and the VMM with arbitrary access control strategy (DAC)[4] can not solve the threat caused by the infection of virus. SeLinux [5][6] provides a mandatory access control (MAC) [7][8] in Linux, which made up part of the security flaws. Somehow, it still can not meet the security requirements of different applications, which makes the security of system very fragile from another aspect. Therefore, a distributed virtualization environment is need to provide both isolation and sharing of constrained resources.

Based on the isolation capability of VKernel[9] in Pcanel[10], inner-domain communications [11] [12] and security control mechanism of virtual resource[13][14] in virtualization layer was implemented to provide a controllable way of solving above problems.

2. Background

2.1 Virtualization background

Independent Virtual Machine Monitor (VMM) is a sort of general software layer running on the hardware directly and controlling the actual hardware resources, while the guest operating systems(Virtual Machine, VM) running above the VMM control the virtual resources. VMs will have lower privilege than the VMM but higher than the applications on the VMs.

The sharing of the virtual resources and inter-domain communications need to be monitored in distributed applications. The virtual resources can be divided into two categories: shared resources like virtual network, virtual disk, etc, and non-shared ones such as virtual processors, virtual memory. According to the requirements of specific application, we should implement the secure control mechanisms of sharing virtual resources and inter-communication.

2.2 Coalition of VMs

Among distributed applications, data communications between them is very frequent. A coalition is a set of VMs that have close communication, as shown in Figure 1. In order to improve the security of the whole system, VMM Coalition should provide not only the corresponding isolation, but also sharing of limited resources to achieve data communications.

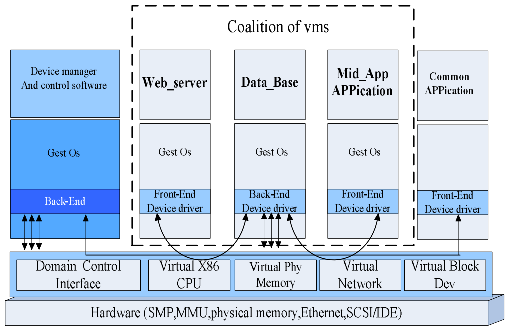


Figure 1 coalition of VMs

VMM directly control hardware resources to which all access by VMs should be monitored by VMM. Accordingly, the sharing of resources and inter-domain communications should be controlled by the corresponding strategy of VMM. Common methods of control are DAC, MAC and role-based access control, which has its own characteristic. So the VMM control all inter-domains data communications.

2.3 SeVMM

There are several security objectives of SeVMM:

- Strong isolation between the multiple VMs.
- Control of the sharing resources between VMs.
- Control of inter-domain communication.
- Management and control of the virtual resource.
- Security services.

Although MAC has been implemented in the Linux, the access to kernel data structure will cause a complex control strategy, which results in the absence of inter-process isolation in SeLinux. Various applications have different security requirements and they should be isolated according to their security levels.

3. Design of SeVMM

3.1 Design concept

SeVMM provided many kinds of services, such as security service, resource monitoring, inter-domain communication control, isolation of virtual resources, etc. Providing the resource monitoring service, SeVMM guarantees the interaction between the security service applications. For instance, controlling the virtual memory usage can deal with DOS attacks. At the mean time, SeVMM uses formal methods to define the security policy of data flow between control domains. In SeVMM, we use Flask framework to configure security strategy.

SeVMM has the following characteristics: First, the performance of the entire system is greatly affected by the performance of SeVMM. Second, SeVMM can automatically deploy security policies, thus reduces management costs. Third, SeVMM support multiple security policies, which enhance security flexibility.

3.2 Overall architecture

In SeVMM, the key of access control of virtual resources is the reference monitor. SeVMM supports a variety of security policies such as the BLP, CW, TE, to ensure the integrity of the system and data flow between the domains. SeVMM monitors have the following three characteristics: First, deal with all security-related calls. Second, SeVMM itself is tamper-proofed. Third, it runs simply and rapid. The SeVMM framework is shown in Figure 2. Security policy management domain manages specific security policy; security policy enforcement module is in SeVMM, implementing the control of access to virtual resources.

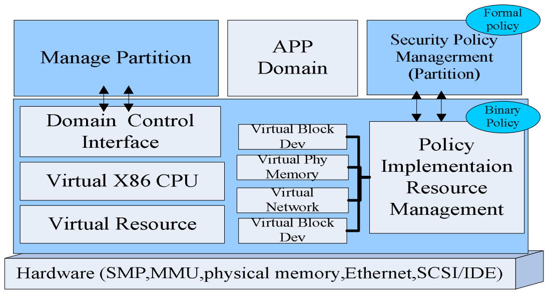


Figure 2 SeVMM architect

SeVMM domain can control the data flow based on the inter-domain isolation. Thus it guarantees the communication and sharing within coalitions while preventing communication and sharing outside coalitions.

4 Implementation of SeVMM

SeVMM consists of three modules: security policy management module, safety hook and security policy enforcement module, as shown in detail in Figure 3.

Safety hook is used to control the sharing of virtual resources between VMs. Security policy management module manages all security policies and protect the update and modification of security policy in the security policy enforcement module. Security policy enforcement module can make decision based on the security information and policy transferred by the hook. The security control process of SeVMM is as follows:

First, collect access control relevant information (define the domain security attributes, attributes of virtual resources, type of access operation).

Second, call the security access control module to obtain security policy results.

Third, execute the security policy.

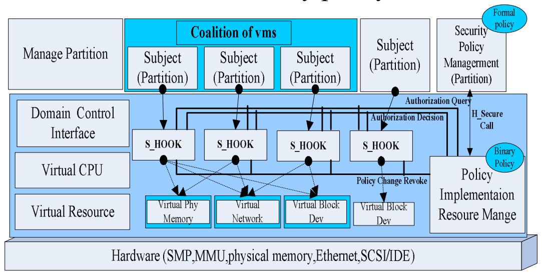


Figure 3 **SeVMM details architect**

4.1 Management of the secure policy

In SeVMM, function call H_SecureCall is used to achieve the update of security policy. When SeVMM changes security policy, Strategy management domain will call H_SecureCall to update the XML description of the security policy into the binary form, and delivers it to the security enforcement module. If the domain has not right to access the shared resources, SeVMM retrieves the shared resources of the domain. The rules of security decision-making are as follows.

$\delta : Vms$ secure lable

γ: VResource secure lable

ω: access operation

$\Delta cw = ((\delta a, \delta b, \ldots), (\delta x, \delta y, \ldots), \ldots)$

$\delta x : CreateDomain, \delta a$、$\delta b$ exist

$CreateDomainControl = \exists A \in (\Delta cw \times \Delta cw) \&\& ((\delta a \times \delta b \times \delta x) \subseteq A)$

$\Delta te = ((\delta a, \delta b, \delta c, \ldots), (\delta x, \delta y, \ldots), \ldots)$

(δx, γa, ω) :

$VResourceControl = ((\gamma a \in \delta a) \&\& (\exists A \in \Delta te) \&\& ((\delta x, \delta a) \subseteq A))$

In SeVMM, there is a security attribute binding with each virtual resource and security domain. At the initialization of each resource or domain, SeVMM allocate security attribute according to the policy in the security policy management module. Access control is implemented based on this stored information. ChineseWall and Type-Enforcement Strategy are implemented in SeVMM.

4.2 Access control hook

SeVMM security control hook function achieves the security check of the access to the virtual resource by domain. The safety hook function needs the corresponding security parameters as follows: attribute of VM, attribute of virtual resource, the type of operation, as shown in Figure 4. Safety hook function transfers the security information to security implementation module, and security implementation module determines whether they have access to the resources according to the security parameters and security policy.

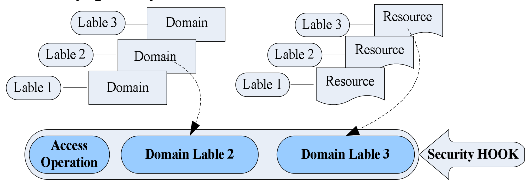


Figure 4 secure hook

We place safety hook in the following places: First, the domain management operations; Second, the event channel operation; Third, the shared memory operation and Fourth, the virtual resources access operation.

In SeVMM, secure hooks are placed in the code path where the domain binds the virtual resources, decisions will be make when the domain binds resources and the communication mechanism is newly established. While the domain accesses the virtual resources, decision-making is not needed. Such mechanisms require the use of virtual resources be clearly security decided, and when necessary, be able to make decision again. This method can significantly improve the operating performance of the whole system. At the same time, security policy cache mechanism in SeVMM stores security policy that is frequently used in the cache to improve the performance of their access.

4.3 Access control enforcement module

The enforcement module of SeVMM stores the current binary security policy to which SeVMM makes decision. Meanwhile, the callback functions will do the safety checks of the allocated virtual resources when security policy changes.

The Access_init function is used to initialize the attribute of the virtual resource or domain in accordance with binary security policy. And the Secure_authorize function call is use security attributes transferred by secure hook to implement security control, as shown in Figure 5.

When a domain access to binding resources, security hook function checks the security attributes of virtual resources the domain, the virtual resources safety attributes, together with the operation type. According to the security strategy, decisions will be make to decide whether the domain can get control of the virtual resources. Privileged call H_SecureCall for policy management is only allowed by policy management domain.

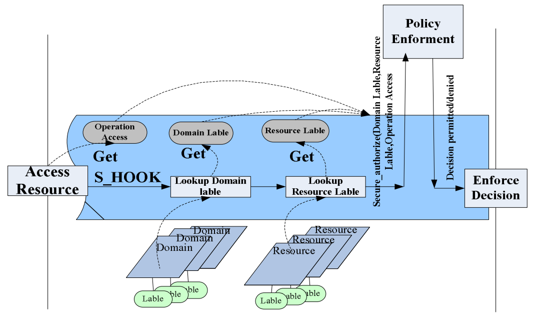
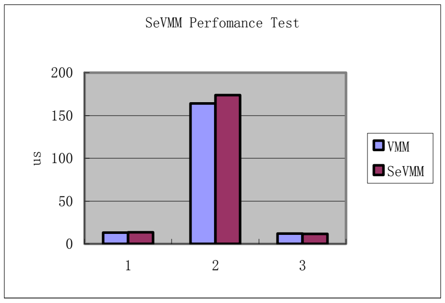


Figure 5 access control process

5 Performance Test

We make related test about SeVMM, first is a benchmark test of the system call delay, the second benchmark is FIFO communication delay, the third is the application process between the two application communication costs. Test data as shown in Figure 6. The overall performance penalty caused by SeVMM is at most around 5 percent, which is tolerable since it significantly improved the security of the system.



1：syscall 2：FIFO 3：PIPE
Figure 6 SeVMM performance test

6 Future work

Based on the VKernel core in Pcanel platform, SeVMM adds control of the inter-domain communication and security control of virtual resources. While ensuring the domain isolation, it provides a secure control mechanism to control the communications and sharing of virtual resources between domains. In future work, we will address its overall structure for further research and analysis according to distributed virtual environment.

References:

[1] T. J. Gibson. Architecture for flexible, high assurance, multi-security domain networks. In *Network and Distributed System Security Symposium*, San Diego, CA, February 2001.

[2] S. E. Madnick and J. J. Donovan. Application and analysis of the virtual machine approach to information system security and isolation. In *Proc. of the Workshop on Virtual Computer Systems*, pages 210–224, March 1973.

[3] J. M. McCune, S. Berger, R. Caceres, T. Jaeger, and R. Sailer. Shamon-a system for distributed mandatory access control. In *Proc. 22nd Annual Computer Security Applications Conference (ACSAC),* Miami Beach, Florida, December 2006.

[4] National Security Agency. Security-Enhanced Linux. URL: *http://www.nsa.gov/selinux/*.

[5] C. J. PeBenito, F. Mayer, and K. MacMillan. Reference policy for security enhanced Linux. In *Proc. of the 2006 Security Enhanced Linux Symposium*, March 2006.

[6] T. Jaeger, R. Sailer, and X. Zhang. Analyzing integrity protection in the SELinux example policy. In *Proc. of the 12th USENIX Security Symposium*, August 2003.

[7] T. R. Jaeger, S. Hallyn, and J. Latten. Leveraging IPsec for mandatory access control of Linux network communications. In *Proc. of ACSAC*, 2005.

[8] E. Bertino, B. Catania, E. Ferrari. A logical framework for reasoning about access control models. In *Proc. of the sixth ACM symposium on Access control models and technologies,* pages 41-52, 2001.

[9] Chen Wenzhi，Huang Wei, Xie Cheng, et al. Trusted Computing Base Using Virtualization Platform. *Zhejiang University (Engineering Edition),* 2008. Accepted, to be published.

[10] Wenzhi Chen, Cheng Xie, Jiaoying Shi. A Precise Control Core for Component based Embedded Operation System. *Journal of Computers (China)*. page 867-874, June 2006.(EI)

[11] S. Bellovin. Virtual machines, virtual security. *Communications of the ACM*, 49(10):104, October 2006.

[12] T. Garfinkel and M. Rosenblum. A virtual machine introspection based architecture for intrusion detection. In *Proc. of the Network and Distributed Systems Security Symposium*, February 2003.

[13] R. Sailer, T. Jaeger, E. Valdez, et al. Building a MAC-based Security Architecture for the Xen Opensource Hypervisor. In *Proc. of the 21st ACSAC*, December 2005.

[14] P. A. Karger, M. E. Zurko, D. W. Bonin, A. H. Mason, and C. E. Kahn. A retrospective on the VAX VMM security kernel. *IEEE Trans. on Software Engineering*, 17(11):1147 – 1165, November 1991.