Design of Xen Hybrid Multiple Police Model

Lei Sun, Renhao Lin and Xianwei Zhu
Zhengzhou Information Science and Technology Institute, Zhengzhou, China
Email: zhjuya_boy@163.com, zhu13939055330@163.com, 1056670972@163.com

Abstract. Virtualization Technology has attracted more and more attention. As a popular open-source virtualization tools, XEN is used more and more frequently. Xsm, XEN security model, has also been widespread concern. The safety status classification has not been established in the XSM, and it uses the virtual machine as a managed object to make Dom0 a unique administrative domain that does not meet the minimum privilege. According to these questions, we design a Hybrid multiple police model named SV_HMPMD that organically integrates multiple single security policy models include DTE,RBAC, BLP. It can fulfill the requirement of confidentiality and integrity for security model and use different particle size to different domain. In order to improve BLP’s practicability, the model introduce multi-level security labels. In order to divide the privilege in detail, we combine DTE with RBAC. In order to oversize privilege, we limit the privilege of domain0.

1. Introduction
All In recent years, cloud computing technology enjoys rapid development. With the method of renting calculating resources provided by the cloud service provider, the user can get easy and quick access to software and hardware resources. Virtualization technology[1] is a good way to achieve these benefit that promote the usage of the hardware while decrease the cost of the power.

Considering the issue of security, the direct access to the physical hardware resource is not usually granted to VMs in the virtual machine system. In order to enhance its security, some research scholars have proposed the security model for Xen virtualization platform by referring to the theory and technology of safe operating system.

However, currently so many applications need their multiple parts to communication with each other, which may run on different virtual machines. Consequently, the secure resource sharing manner is needed between related virtual machines, that is to say, the appropriate access control mechanism is needed to enforce the access control between VMs.

In this paper, we introduce a hybrid multiple security police model for Xen named SV_HMPMD(Security Virtualization_Hybrid Multi-Policy Model). There are some advantage as follow: It achieve the minimum privilege to management Dom0 by RBAC and DTE. The single subject of operation of Dom0 is divided into several subjects. In the second place, We improve the BLP model by multi security label. Unlike the original BLP which give each subject and object the current security label, object of SV_HMPMD model are given security label range of (w_min, r_max) and (L_mino, L_maxo) to improve the flexibility of the BLP. The trusted subject is divided to prevent the whole system from the occurrence of a single privileged entity. A number of trusted subjects replace the single trusted subject of Dom0. Privileged domain and privilege state are added to manage the privilege and privilege dynamics.

The rest of this paper is organized as follows. Section 2 discuss related work, and the framework design of SV_HMPMD model is discussed in Section 3. In section 4, we discuss the concrete implementation of SV_HMPMD and the resolution of key issues.
2. Related Knowledge Introduction

In the field of secure operating system, domestic and foreign scholars have done a lot of research on access model: Xen's access control model ACM / sKype [2]provides a distributed access control between virtual machines, but it has only a few simple security policies such as STE and the Chinese wall. In order to improve the applicability of the security policy, Xen's security model draws on the mixed model of RBAC and TE in SELinux to manage Dom0 and DomU, and uses the virtual machine as the management unit to carry on the simple role division, it realize the multiple access control strategy but The same management granularity is used for Dom0 and DomU. Ease of Use. But there is no fine-grained division of privilege operations in Dom0 that causes Dom0 privilege to be centralized and violates the principle of system security of minimum privilege. In order to solve this problem, Xenon[3] proposed by the US Navy Laboratory and Qubes[4] proposed by Joanna Rutkowska's divide Dom0 into multiple virtual device domains, Constructing Access Control Model to Realize Dom0 ‘s Fine-grained Management. It can not meet the needs of the classification system. Because it use virtual machine or virtual machine group as a control unit and does not safely classify the system resources. Chuliang wen[1] used BLP on a virtual machine system, using virtual machines as access control objects and defining access rules. He use the traditional BLP model, which has * - strict characteristics, did not achieve efficient rights management.

In this paper, we present a hybrid multi-strategy model SV_HMPMD in the face of Xen virtualized environment. Its characteristics are as follows:

1) We use RBAC and DTE to achieve fine-grained the privileges and access rights of Dom0. The operation of single subject of Dom0 is divided into multiple entities together to complete.

2) We change the BLP model into a multi-security label BLP. It is different from the original BLP that given the current security label to each subject and object, SV_HMPMD model gives the subject and object with \((w_{\text{min}}, r_{\text{max}}), (L_{\text{min}}, L_{\text{max}})\) two security label range to improve the flexibility of BLP.

3) We divide the trusted subject to prevent the occurrence of a single privileged subject to control the entire system. Multiple trusted principals replace Dom0's single trusted subject. Multiple trusted subject replace Dom0's single trusted subject.

3. SV_HMPMD Model

3.1 SV_HMPMD goals

SV_HMPMD security goal is to avoid user of illegal storage of information in Virtualize multi-user sharing information and effective protection system information from by unauthorized users. And the privilege of Dom0 as a trusted subject is divided to achieve fine-grained management.

1) Due to the multi-user nature of the virtualization platform, the administrator has different permissions from the user’s behavior. In order to achieve this goal, particle-size partition is made between the management domain Dom0 and the user domain DomU.

2) Implement security domain isolation. The model can provide an isolation mechanism that limits the operation of non-trusted domains to ensure that other users' information is not stolen.

3) It is can restrict the privilege of the trusted subject, because the trusted subject can perform operations that violate the security constraints, and if the trusted privilege can perform too much privilege. it will threaten the confidentiality of the system.

3.2 SV_HMPMD Design

First, DTE provides an easy-to-implement access framework that implements access between subject and object and ensures system integrity. It use the BLP security model to ensure the confidentiality of the virtual domain itself and other domains. Through the trusted subject to specify the Dom0, management domain, violates the BLP security features but the implementation of the system operation allows the action. The privilege interference between the trusted subject and the trusted subject, and the granting of its privilege are achieved by RBAC. RBAC main function is the security label allocation and management in SV_HMPMD. The the access control of the general subject and the privileged subject is realized by the DTE model.
4. SV_HMPMD Specific Implementation

From the previous chapter we can see SV_HMPMD through the improved BLP model to achieve the system access control. It achieve the management area Dom0 privilege division and the different users DomU static permissions division though RBAC. Combining it with DTE as a flexible access control model achieve a system integrity and privileged controlled. The principle of the above characteristics will be described later.

4.1 BLP implementation

A multi-tagged BLP model was used to improve the original BLP model as follows:

Using the label range instead of the original label, it do not reduce the confidentiality of the original model while improving the flexibility of the model. SV_HMPMD uses the security tag range that is $(w_{\text{min}}, r_{\text{max}})$. $w_{\text{min}}$ is the minimum security label that the subject can write to the object. $r_{\text{max}}$ is the maximum security label that the subject can read on the object. In this range is expressed as ok ($s$).

SV_HMPMD also gives a tag range $(L_{\text{min}}, L_{\text{max}})$ for the object, which is the minimum access security label, the object's maximum access security label. Object security level within this range is ok ($o$).

BLP safe kilometers are as follows:

a) If a subject can access an object in a read or write manner, the current security level of the subject must control the maximum security level of the object.

\[ \forall o \in O, s \in S. (s,o,r \vee a) \in b \Rightarrow f_r(s) \geq L_{\text{max}}(o) \]

b)*-property

\[ (s,o,a) \in b \Rightarrow L_{\text{max}}(o) \leq r_{\text{max}} \]

\[ (s,o,r) \in b \Rightarrow L_{\text{min}}(s) \geq w_{\text{min}}(s) \]

\[ (s,o,w) \in b \Rightarrow ok(o) \equiv ok(s) \]

c) SV_HMPMD divides the subject into a trusted subject and a general subject. The definition of the general subject is the same as the original BLP model. Part of the trusted subject condition $w_{\text{min}}(s) \neq r_{\text{max}}(s)$, this part of the subject can only access the object of its security rights. Trusted subject are not restrict to security policy constraints and limit their minimum privileges.

4.2 The Division of the Trusted Subject

In order to limit the authority of Dom0 and prevent Dom0 from controlling the entire system, the Dom0 operations are subdivided into several trusted subsets and assigned to the corresponding "roles" through RBAC. We assign these "roles" to the process of completing the privileged operation and replace the original single trusted user with a trusted user.

4.3 Integrity Implementation

Due to the existence of a credible subject it can violate the security rules. Improper modification of the security information and integrity of the trusted body itself is threatened. The non-passability and inter-domain transformability of DTE's access control relationships provide a better fine-grained access to the model. The idea of its realization is:

The object with the same integrity security requirement is set to the same type. And it uses the Domain Interaction Table (DTT) to specify the access rules of the subject to the object, to prevent the disclosure of the security information to the unauthorized subject, and to limit the privilege of the trusted body with excessive authority in the BLP to ensure the minimum privilege of the trusted body. The isolation of information between domains is achieved through the DIT table of the DTE model, the separation between subject prevent the credibility of subject through the flow of information between the integrity of damage. Integrity access control rules:

a) domain-type access rules

b) domain-domain interaction rules

c) Defines the inter-domain conversion operation, the provisions of the object can occur after the conversion between those domains. The system Domain and channel can be in their specific domain.
4.4 Privileged Access Control

The current model for the management of privileged operations is still very small. Most of the management is non-privileged access control, but the actual operation of privileged access seriously affects the security of the system, such as the abuse of privilege. Privileged non-isolated propagation can lead to serious system security issues. So many viruses and malicious code achieve malicious spread by the acquisition of privilege. This paper deals with the problem of dynamic change from privilege scope and privilege. We solve the problem of privilege role checking and privilege calculation. SV_HMPMD first isolates the same role by static role partitioning on the same user, and binds the privileged process to a specific privileged domain. These privilege fields have a specific privilege attribute, that is, each privilege domain specifies the privilege function that the privileged subject of the domain can perform. It limit the scope of the franchise. The privileged state is a sign of the privilege process change. From the above privilege management rules we can see that the privileged process is affected by the role of the process Domain Pr and the privilege field Pd of the process and the shared memory to be executed by the privileged process.

In this paper, \( P_i, P_p, P_e \) indicates the state of privilege. \( (f, f_e) \) represents the privileged state of shared memory. It is described in Table 1.

This article uses genetic algorithm that improved by Dragovice[5] to privilege calculations.

\[
\begin{align*}
P_i' &= P_i \land f_i \\
(P_i' \lor P_p') &= P_i \lor P_p \\
(P_i' \land P_p') &= P_i \land P_p \\
(P_i', P_p', P_e') &= (f, f_e)
\end{align*}
\]

\( P_i', P_p', P_e' \) is the set of privilege states that the process is executed after. The formula indicates that the privileged state of the inheritable privileged state is the minimum privileged state under the current state of the process privilege state and the shared memory privilege state. \( P_d \) is the largest privilege set allowed by the domain where the process is located. The set of privileges that are allowed to execute is determined by the privileges of the current role and the privileged domain and the state of the privileged process. Thus it limit the scope of the privilege set. On the other hand it also achieved the isolation of users of privilege.

| symbol | Privilege set | object | description |
|--------|--------------|--------|-------------|
| \( P_i \) | inherit the privilege set | Privileged process | a process can pass privileges to subsequent processes |
| \( P_p \) | Permission privilege set | Privileged process | A maximum privilege set for a privileged operation that a process can perform |
| \( P_e \) | Execute privilege set | Privileged process | A collection of privileges that the current process can perform |
| \( f_i \) | inherit the privilege set | Privileged process | A special collection of privileged operations on shared memory. |
| \( f_e \) | Execute privilege set | Privileged process | When the shared memory is executed, the shared memory maps the privileges of its privileged processes. |

4.5 Overall Frame

In SV_HMPMD, the three access control models cooperate with each other in the integrity and confidentiality of functions. This section describes the SV_HMPMD access procedure on a framework. The judgment of privileged as a supplement to BLP confidentiality access runs through the entire access judgment process. Privileged access is divided into supplementary privileged access and managed privileged access. Supplemental privileged access is a supplement to other access policies that are privileged to meet access rules. Managed privileged access is privileged access that violates the access rules of a particular access control model. Such as MAC OVERRIDE and DAC OVERRIDE. The former can violate MAC access control, The latter is a violation of DAC access control. For the first
privileged access, it work with other access rules, so Privileged access takes effect only if the access meets the requirements of the access control model. The second privileged access is common in Dom0, As long as you meet this privilege or other privileged access it will able to take effect. The whole process is shown in Figure 2.

When the subject place an access request, it is first determined whether the domain where the subject is located is allowed to perform the operation by the DTE. If access is allowed, it is judged by MAC and DAC. If it is allowed to access, subject access the object. If it do not allow access, it check whether it meets the appropriate violation of the rule privilege. If it does not allowed to access, it deny to visit.

![Figure 1 Access control process.](image)

5. Model Rules

5.1 Model Element

User set U; Subject set S={St, SU, P, AR}. S_t is a trusted subject, S_u is general subject and S_u=S-S_t; P is a privilege, each privilege depends on the subject. AR is the Subject of access to the operation of its state are {READ, WRITE, APPEND, CREATE, EXECUTE, AUTO};

Object set O={O_t, O_u, M(O), H(O)}; O_t is trusted object, It generate the subject image by executing a trusted object; O_u is general object; M(O) represents the subject's access to the object; H(O) represents the function that the object set maps to the set of set powers. If O_t≠O_u, then H(O_t) ∩ H(O_u) = ∅.

Domain-type access relationship: DTT := D × T × A; Domain is D={d_1, d_2, d_3…d_n}; T is the type; A is the way of access A \subseteq AR;

Model status output S={subject_P, output}; Subject_P: subject process and privilege with mapping; Output (decision output) value is YES, NO.

The model currently accesses the state, B = {b | b ∈ (S, O, AR)};
safety label $\text{LEVEL} = \{G, C, \Xi\}$; The security level $G$ represents the security level of the subject object. The security category $C$ represents the security scope that the subject and object can perform. $\Xi$ safety label control relationship, High-level security labels can control low-level security labels when security is included.

Label function $f = \{f_s, w_{\min}, r_{\max}, L_{\min_0}, L_{\max_0}\}$: $f_s$ is the maximum security label function of subject. $w_{\min}$ is subject minimum writeable label; $r_{\max}$ is subject maximum readable label; $L_{\min_0}$ is object minimum writeable label; $L_{\max_0}$ object maximum readable label;

5.2 Migration Rules
In order to meet the needs of the actual system development, we give 11 migration rules that the model must meet. These are get_access, execute, release_access, add_user, delete_user, create_object, authorize_role, revoke_role, authorize_dm, authorize_D, authorize_executable_domains. Due to space reasons only three of them to explain and formal proof.

**Role 1.** Domain read and write access $\text{get\_access}(s, d, o, t, p)$

If the requested access type is not appropriate, the next state remains unchanged and the decision output is deny; otherwise, check the following conditions:

1. $(d, t, p) \in DTT$
2. $(s, p) \in M(o)$
3. $((p = r) \land (L_{\min_o} \leq v_{\max_s}))$
   $\lor (((p = a) \land (L_{\min_o} \geq a_{\min_s}))$
   $\lor ((p = w) \land (\text{run}(o) \subseteq \text{run}(s)))$

4. $\exists \text{priv} \in P, \text{priv} \in \text{subject \_ privs}(s)$, You can get Domain access to p of resource. If the condition 1-3 or the condition 4 is satisfied, the decision output is yes and the next state changes as follows:

   $b' = b \cup \{s, o, p\}$, Other state variables remain unchanged.

**Role 2** Domain is executed directly on object O

There are two cases:

a) $d = d'$, If the type of access requested is not appropriate, The next state remains unchanged. Decision output UNDEFINED; Otherwise, check the following conditions:

1. $(d, t, e) \in DTT$
2. $L_{\max_s} \leq v_{\max_s}$
3. $\exists \text{priv} \in P, \text{priv} \in \text{subject \_ privs}(s)$

If the condition 1-2 is satisfied or the condition 3 is satisfied, the decision output is YES, The next state changes to follow state:

   $b' = b \cup \{s, o, e\}$

   $\text{subject \_ images}(s') = o$

   if $s \in S$, then

   $\text{subject \_ p}(s), P = \text{subject \_ p}(s), P \land$

   $\text{executeable\_ p}(o), f$

   $\text{subject \_ p}(s), P' = (\text{executeable\_ p}(o), f) \lor$

   $(w, \text{subject \_ p}(s), P) \land$

   $p_{\_ role}(\text{subject \_ role}(s)), P \land$

   $p_{\_ domain}(\text{subject \_ role}(s))$

   $\text{subject \_ p}(s), P' = \text{subject \_ p}(s), P \land$

   $\text{executeable\_ p}(o), f$

If the above conditions are not met, the decision output is the NO, the next state remains the same.
b) When the subject and the object are not in the same domain, they need to be domain conversion to complete the operation. If the type of access requested is not appropriate, the next state remains unchanged. Decision output UNDEFINED; otherwise, check the following conditions:

1. \( o \in \text{domains}_{executables}(d') \)
2. \((d,t,e) \in \text{DTT}\)
3. \((d,d) \in \text{DMD}\)
4. \(\text{role}_\text{domains}(d') \subseteq \text{role}_\text{domains}(d)\)
5. \(L_{\text{max}}(o) \leq v_{\text{max}}(s)\)
6. \(\exists \text{priv} \in P, \text{priv} \in \text{subject}_\text{privs}(s)\)

If the condition 1-5 is satisfied or the condition 6 is satisfied, the decision output is YES, The next state changes to follow state:

\[
b' = b \cup \{s,o,e\}
\]

\(\text{subject}_\text{images}(s^*) = o\)

\(\text{subject}_\text{domains}(s^*) = d\)

If \(s \in S_{d}\), then

\[(S_{s} = S_{s} \cup \{s\}) \land (S_{o} = S_{o} - \{s\}) \land\]

\(\text{subject}_\text{p}(s).P_{s} = \text{subject}_\text{p}(s).P_{s} \land\)

\(\text{execute}_\text{p}(o).f_{e}\)

\(w.\text{subject}_\text{p}(s).P_{w} = (\text{executable}_\text{p}(o).f_{e} \lor (\text{subject}_\text{p}(s).P_{s} \land \text{role}(\text{subject}_\text{role}(s)) \land\)

\(\text{priv}_\text{domain}(\text{subject}_\text{role}(s))\)

\(\text{subject}_\text{p}(s).P_{p} = \text{subject}_\text{p}(s).P_{p} \land\)

\(\text{executable}_\text{p}(o).f_{e}\)

if \((d_{d},d_{d}) \in \text{DMD}\)

\(o \in \text{domains}_{executables}(d_{d})\)

\(\text{role}_\text{domains}(d_{d}) \subseteq \text{role}_\text{domain}(d_{d})\)

If the above conditions are not met, the decision output is the NO, the next state remains the same.

Role3. Add Domain

A user who has a role r in domain d requests a domain to be added to the system. If the type of access requested is not appropriate, the next state remains unchanged. Decision output UNDEFINED; otherwise, check the following conditions:

1. Add the user's privileges to r
2. d allows users to add privileges and privileges take effect
3. \(u \notin U\)

If the condition 1-3 is satisfied, the decision output is YES, The next state changes to follow state:

\[U' = U \cup \{u\}\]

If the above conditions are not met, the decision output is the NO, the next state remains the same.

6. Conclusion

In this paper, according to Xen's isolation, integrity, privilege control, we introduce the hybrid model SV_HMPMD. It guarantees the confidentiality of the system with improved BLP . Through the RBAC and DTE to ensure the integrity of the system and isolation and the privilege management of the system is realized on this basis. Future work will improve the model and improve the integration of the model to reduce the performance expenditure of the model in actual use.
7. References

[1] Sailer, Reiner, E. Valdez, and T. Jaeger. "sHype : Secure Hypervisor Approach to Trusted Virtualized Systems." IBM Research Report Re119.2(2005):331--352.

[2] Freitas, Leo, and J. McDermott. "Formal methods for security in the Xenon hypervisor." International Journal on Software Tools for Technology Transfer 13.5(2013):463-489.

[3] I Lu, Qi, Z. X. Cao, and Z. Xue. "Xen-based Qubes OS Technical Architecture and Safety Performance Analysis." Information Security & Technology (2014).

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[8] Electronic Publication: Digital Object Identifiers (DOIs):

[9] Article in a journal:

[10] D. Kornack and P. Rakic, “Cell Proliferation without Neurogenesis in Adult Primate Neocortex,” Science, vol. 294, Dec. 2001, pp. 2127-2130, doi:10.1126/science.1065467.

[11] Article in a conference proceedings: H. Goto, Y. Hasegawa, and M. Tanaka, “Efficient Scheduling Focusing on the Duality of MPL Representatives,” Proc. IEEE Symp. Computational Intelligence in Scheduling (SCIS 07), IEEE Press, Dec. 2007, pp. 57-64, doi:10.1109/SCIS.2007.357670.