一种虚拟机访问控制安全模型*

梁 彪[1]，秦中元[2,3†]，沈日胜[2,3]，陈 琦[4]，吕 游[4]，强 勇[4]，郭爱文[4]
(1. 南京三宝科技股份有限公司，南京 210049；2. 东南大学 信息科学与工程学院，南京 210096；3. 信息网络安全公安部重点实验室，上海 210204；4. 东南大学吴健雄学院，南京 210096)

摘 要：在虚拟机系统的众多安全威胁中，资源共享和数据通信所带来的内部安全问题成为了云平台下虚拟机中最为关注的问题。结合 Chinese Wall 和 BLP 模型，提出了一种适合于虚拟机的访问控制安全模型 VBAC，在 PCW 安全模型的基础上，引入了 BLP 多级安全模型，并对 BLP 进行了相应改进，该模型可对虚拟机内部资源使用、共享以及事件行为进行安全控制。实验结果显示该模型有较强的可行性及安全性。

关键词：虚拟机；中国墙；BLP；访问控制
中图分类号：TP309.2 **文献标志码**：A **文章编号**：1001-3695(2014)01-0231-05
doi:10.3969/j.issn.1001-3695.2014.01.054

Access control security model of virtual machines

LIANG Biao[1], QIN Zhong-yuan[2,3†], SHEN Ri-sheng[2,3], CHEN Qi[4], LV You[4], QIANG Yong[4], GUO Ai-wen[4]
(1. *Nanjing Sample Technology Co., LTD, Nanjing* 210049, *China*; 2. *School of Information Science & Engineering, Southeast University, Nanjing* 210096, *China*; 3. *Key Laboratory of Information Network Security, Ministry of Public Security, Shanghai* 201204, *China*; 4. *Wu Jianxiong Dept., Southeast University, Nanjing* 210096, *China*)

Abstract: Among all the security risks, the sharing of resources and data communication in virtual machines are most concerned. This paper proposed an access control security model which combined the Chinese Wall and BLP model. It introduced BLP with corresponding adoption with PCW security model. This model could be used to control the resources usage and event behaviors in virtual machines. The experimental results show its effectiveness and safety.
Key words: virtual machines(VM); Chinese Wall(CW); BLP; access control

0 引言

作为云计算的关键底层技术之一，虚拟机(VM)得到了广泛的应用，这也使得虚拟机的安全问题越来越突出[1]。和传统的计算机系统相比，虚拟机将面临更多的安全威胁，不仅包括了外部威胁，如攻击者利用 HTTP 协议的漏洞来进行恶意代码的攻击，也包括了因为虚拟机的系统特征而导致的内部威胁[2~4]，其中虚拟机内部威胁日益引起学术界和工业界的研究兴趣。虚拟机的内部威胁主要包括虚拟机之间通信攻击、拒绝服务攻击以及信息泄露。通信攻击是指攻击者通过进程、内存共享或者内存错误等来进行代码的植入和攻击，也称为隐通道威胁，这个问题目前在业界尚未得到很好地解决[5~7]。拒绝服务攻击使得计算机资源不能分配给目标机，由于 CPU、内存、磁盘、网络等资源都是共享于虚拟机环境中的，如果一个 VM 无休止地向主机申请资源，那么将导致其他 VM 不能得到足够的资源。信息泄露一方面是攻击者通过隐通道进行数据的传输，如在共享缓存的 CPU 中利用检测缓存缺失所造成的时间延迟进行数据传输，这种攻击通常都存在一些同盟的关系，双方经过协商，利用系统存在的漏洞进行关键数据的传输；另一方面则是攻击者利用不同 VM 之间共享的资源来窃取信息，如共享的内存资源或者 CPU 资源，作为虚拟机系统本身来说这种数据的泄露并不会造成系统或者数据的破坏，但是如果上层运行的 VM 是一些存有关键数据的服务器，那么这种信息的泄露则会有很大的安全隐患。

虚拟机的运行需要虚拟机管理器(virtual machine monitor, VMM)的控制。安全的 Hypervisor 可以保护虚拟机资源隔离性、数据安全性、通信安全性以及代码完整性等。这方面的代表工作是 IBM 提出的安全 Hypervisor 架构 sHype[8]，它利用安全模型，通过访问控制模块(access control module, ACM)来控制系统进程、内存的访问，实现内部资源安全隔离。但是 sHype 在使用中国墙策略防止不同虚拟机之间隐形流传递的时候，需要静态地将隶属于不同冲突类资源划分成不同的访问区域，这样的约束条件会降低资源利用率[9]。Cheng 等人[10]提出带有优先权的中国墙模型(prioritized Chinese wall, PCW)来减少虚拟机系统中隐藏信息流的风险。本文基于 sHyper 和 PCW 架构，提出了一种适用于虚拟机环境的访问控制模型，并在 Xen 中进行实现与结果验证。

本文的主要贡献有以下三点：

a)提出了一个虚拟机访问控制安全模型，在 PCW 模型基

收稿日期：2013-03-30；**修回日期**：2013-05-08 **基金项目**：国家科技支撑计划资助项目(2011BAF16B00)；信息网络安全公安部重点实验室开放课题项目(C11605)；江苏省网络与信息安全重点实验室开放课题基金项目(BM2003201)

作者简介：梁彪(1969-)，男，江苏南京人，博士，主要研究方向为信息安全；秦中元(1974-)，男(通信作者)，河南安阳人，副教授，博士，主要研究方向为信息安全(zyqin@seu.edu.cn)；沈日胜(1986-)，男，硕士研究生，主要研究方向为虚拟机安全；陈琦(1992-)，女，本科生，主要研究方向为信息安全；吕游(1992-)，男，本科生，主要研究方向为信息安全；强勇(1992-)，男，本科生，主要研究方向为信息安全；郭爱文(1992-)，男，本科生，主要研究方向为信息安全.

础上引入了 BLP 模型,将 BLP 用于控制虚拟机的内存共享,同时又保持原有 CW 模型用于控制虚拟机的启动管理。

b)对 BLP 模型作了相应的改进,针对 BLP 模型的缺陷,增加了可信主体和安全等级的控制范围。

c)对所提出的安全模型进行了仿真实验,通过三种策略的分工来共同完成安全控制策略。仿真实验证明了本文所提模型的有效性。

1 模型分析

1.1 虚拟机中的访问控制模型分析

IBM 于 2005 年提出 sHype 用于控制共享同一硬件平台的操作系统之间信息流的一个安全架构,可以实现虚拟机之间受控的资源共享。Xen 虚拟机根据 sHype 的核心思想,在虚拟层设置了一个 ACM 模块,并实施了两种安全策略[11],一种是中国墙(CW)策略[12],另一种是简单类型加强(simple type enforcement,STE)策略。其主要目标是:a)控制同一个系统中两个虚拟机之间的通信;b)控制虚拟机可以访问的硬件资源;c)控制存在利益冲突的多个虚拟机不能同时运行,从而减少隐蔽通道的发生。其中 CW 策略可以防止有利益冲突的虚拟机在同一个 VMM 上同时运行,STE 策略控制不同虚拟机之间是否可以通信以及虚拟机可以访问的硬件资源。

CW 和 STE 在资源的控制和分配上有很大的优势,这也是强制访问控制方法能在虚拟机中得到较好应用的主要原因。然而 CW 和 STE 也有很大的局限性,主要体现在不能很好地防止隐通道,攻击者很容易利用共享资源或者第三方虚拟机进行数据的传递。针对这一缺陷,Cheng 等人[10]提出 PCW 策略模型,该模型同样是基于中国墙安全策略,但与 sHype 中的 CW 策略不同的是 CW 冲突集的关系不再是静态不变的,而是伴随着信息的流向进行冲突扩展。该模型能在很大程度上防止隐通道的存在,然而也存在如下的缺陷:

a)PCW 模型过于严格,很大程度上降低了虚拟机系统的资源利用率,违背了虚拟机的设计初衷,同时还存在系统死锁的可能性,即一个虚拟机因为利益冲突的动态集合而永远无法获得运行机会,除非重新启动系统。

b)PCW 思想仅局限于对虚拟机资源分配的隔离控制,即要么能访问,要么拒绝,但这并不符合实际的环境要求。虚拟机系统相当于一个计算机系统群,这个群中虚拟机之间的冲突有两种,即资源申请和访问权限。而访问权限上的限制并不能进行严格的隔离。例如在 FTP 服务中,当主机与客户的访问权限存在冲突时,主机可以设置客户的访问权限,限制客户的读写权限从而使客户在权限范围内进行操作,但 PCW 仅仅能对这种需求完全隔离。

1.2 BLP 模型及其改进

由前述可见,CW 和 PCW 的冲突集并不能完全满足虚拟机的安全需求,为了能从整个系统的安全上解决虚拟机的系统问题,本文将引入另一种模型,即 BLP 多级安全模型。在本文提出的模型中,BLP 将用于控制虚拟机的内存共享,同时利用 CW 模型控制虚拟机的启动管理,考虑到 PCW 能够很好地防止虚拟机中网络通信和 CPU 资源共享带来的隐通道威胁。因此,CW、PCW 和 BLP 将同时存在于本文提出的模型中。为了更好地满足系统需求,本文对 BLP 进行了一些改进。

Bell-LaPadula(BLP)模型是 20 世纪 70 年代美国军方为解决分时系统的信息安全和保密问题而提出来的,该模型主要用于防止保密信息被未授权的主体访问[13]。在 BLP 模型中,设 $L(s)=l_s$ 是主体 s 的安全许可,并设 $l(o)=l_o$ 是客体 o 的敏感等级,对于所有安全密级 $l_i(i=0,1,\cdots,k-1)$,有如下三条访问控制规则:

a)自主安全特性。主体 s 可以对客体 o 存在自主访问的权限。

b)简单安全条件。主体 s 可以对客体 o 进行读操作,当且仅当 $l_o<l_s$,且 s 对 o 具有自主型读权限。

c)*-属性(星号属性:主体 s 可以对客体 o 进行写操作,当且仅当 $l_o>l_s$,且 s 对 o 具有自主型写权限。

针对虚拟机的特定环境,本文对 BLP 模型进行了如下改进:

a)增加可信主体。BLP 模型的严格机密性使得系统受到了很大的限制,例如在虚拟机环境中,任何虚拟机之间只能根据安全级进行通行,然而一些特殊的虚拟机需要在无条件下进行信息交换。因此,本文加入可信主体,它可以违背 BLP 模型的严格安全级制度而与目标客体进行信息交换。本文限制可信主体不与其他客体交换有害客体,或者可信主体不作为有害客体的信息传递媒介。

b)安全等级的范围控制。在 BLP 公理的常规实施方法中,一个主体的当前敏感标记在它的整个生命期内是固定不变的。然而系统为每个虚拟机 VM 设置一个安全等级,并不代表所有高安全等级虚拟机对低安全等级虚拟机具有数据读取的权限,因为这不满足实际服务的需求。因此,需要为安全等级设置一个区间分类,在安全区间类别中存在多级安全数据流,而不同区间类别则不存在。这种方法能根据系统需求控制不同安全级虚拟机之间的数据流。

2 模型设计和安全性分析

2.1 模型设计

模型假设:所有虚拟机的安全级是处于同一区间类别中的,同时在虚拟机系统中,虚拟机对系统分配的资源都具有自主访问权限,即系统中,主客体都满足自主访问权限。

本文设计的虚拟机安全访问模型(virtual machine based access control,VBAC)如下:

1)模型元素

a)主客体元素

主体集 $S:\{s_1,s_2,\cdots,s_m\}$,定义为发出访问请求的对象,客体集 $O:\{o_1,o_2,\cdots,o_m\}$,定义为被访问的对象,虚拟机系统中,虚拟机 VM 和系统资源都将可能成为主、客体之一。

可信主体集 $T(s):\{s|s$ 是可信的且 $s\in S\}$。

b)访问属性与请求、决择序列

访问属性 AC 包含如下几类:内存读写{read, read|write},内存传递{mem-transfer},虚拟机标签{addlabel,rmlabel},资源申请与释放{apply, release},虚拟机创建与销毁{create, destroy},虚拟机启动与关闭{start,stop},通信申请与释放{com-apply, com-release},安全级调整{level(L_u)|L_u 为新的安全级}。

虚拟机状态集:State(o){running,stop,sleep}。

系统安全级 F:安全分类,f_1、f_2 分别负责代表主、客体安全级别,f_3、f_4 代表主、客体的类别。

请求集合 $R:S^+\times O\times X$,其中 $S^+=S\cup\{\varnothing\}$,$X=AC\cup\{\varnothing\}\cup F$,请求元素记为 $R_i:\{R_i|R_i\in R,i\geq 0\}$。

安全决择集合 $D:\{$yes,no,error,?$\}$,其元素记为 $D_j:\{D_j|D_j\in D,j\geq 0\}$,为该安全模型针对请求 R_j 的决择。

请求序列 $X:R^{T}$,表示在不同时间进行请求序列集,请求元素记为 x,x_t 表示 t 时刻的请求。

决择序列 $Y:D^{T}$,安全策略针对请求序列的决择序列集,决择元素记为 y,y_t 表示 t 时刻的决择。

c)冲突集与访问矩阵

冲突类 CIS(o),与客体 o 存在利益冲突的数据集合,元素 $CIS_t(o)$ 表示客体 o 在 t 时刻的冲突关系集;冲突域 RC,凡是属于某个冲突类中的对象,都属于某一冲突域,即 $\forall s \in CIS(s), s \in RC$;若 s 不属于任何冲突集,即 $\neg \exists s' \in S, s \in CIS(s')$,即 s 不属于冲突域,那么定义 $s \notin RC$。

系统访问矩阵 $A:(S \times O)$,用于记录系统中所控制的主客体之间的访问历史,矩阵元素 $A_t(s_i,o_j)$ 表示 t 时刻系统中主客体访问矩阵,$A_t \rightarrow \{-1,0,1\}$,1 表示主体 s_i 对客体 o_j 有过访问,0 表示还未抉择,会根据抉择 $D_t(s_i,o_j,x)$ 而变化,-1 表示不允许访问。A_0 表示系统初始访问矩阵。

d)系统状态与系统

从初始状态,通过一系列的安全请求和许可之后,系统仍然是安全的,那么表示这个安全模型是安全稳定可行的。

当前访问集合:$P(S \times O \times AC)$。

系统状态 $V:P(S \times O \times AC) \times A \times F$。

状态序列 $Z:V^{T}$ 状态序列,状态元素记为 z,z_t 表示在 t 时刻的状态;认为 z_0 是初始状态,通常认为系统的初始状态是安全的。

状态转换:$W \subseteq R \times D \times V \times V$ 表示由一个状态经过主体请求和抉择后转换到另一个状态。

系统 SYM:$\sum(R,D,W,z_0) \subseteq X \times Y \times Z$,其中 $(x,y,z) \in$ SYM,当且仅当在任何 $t \in T$ 时间 $(x_t,y_t,z_t,z_{t-1}) \in W$。

如果每次请求 R 和 D 是安全的,那么每次状态转换就都是安全的;那么如果系统初始状态安全,在一系列的安装转换之后,系统将还是安全的。所以要保证整个系统的安全,就必须确保每次的请求抉择都是安全的;同理,如果每次请求抉择是安全的,那么系统也将保持安全。

2)模型规则

实现系统的安全控制,定义如下安全规则控制虚拟机系统的主客体,在下面定义的规则中,B1、B4、B5、B6、B7、B8、B9、B10 中,主、客体都是虚拟机 VM,而其他规则中主体为虚拟机,客体则是硬件资源。

规则 B1　主体添加、删除标签 $R_t(s,o,\text{addlabel}) \mid R_t(s,o,\text{rmlabel})$。

$R_t(s,o,\text{addlabel}) \mid R_t(s,o,\text{rmlabel})$ 被允许当且仅当 $s \in T(s)$,即主体是可信主体。

规则 B2　主体申请资源 $R_t(s,o,\text{apply})$。

$R_t(s,o,\text{apply})$ 被允许,当且仅当:

a)$A_t(s,o) \neq 1$,即在 t 时刻,主体 s 没有访问过 o;

b)t 时刻,不存在其他主体正在访问客体 o;

c)$\forall s' \neq s: A_t(s',o) = 1 \wedge \forall o' \in CIS_{t+1}(o): A_t(s',o') \neq 1$,访问过 o 的其他主体不能与主体 s 相冲突。

规则 B3　主体释放资源 $R_t(s,o,\text{release})$。

$R_t(s,o,\text{release})$ 被允许当且仅当 state(s) = stop,即主体所处状态为停止状态。

规则 B4　主体创建、删除客体,$R_t(s,o,\text{create} \mid \text{destroy})$。

$R_t(s,o,\text{create} \mid \text{destroy})$ 被允许,当且仅当 $s \in T(s)$。

规则 B5　主体启动客体,$R_t(s,o,\text{start})$。

$R_t(s,o,\text{start})$ 被允许,当且仅当满足如下两个条件:

a)$s \in T(s)$ 且 $\forall o', D_t(o,o',\text{apply}) = 1$,即客体虚拟机申请所有资源都得到了规则 B2 的允许;

b)$\forall s' \mid (\text{state}(s') = \text{running}) \wedge (\text{state}(s') = \text{sleep}), o \notin CIS(s')$,即客体与虚拟机系统中运行或者暂停的虚拟机不冲突。

规则 B6　主体关闭客体,$R_t(s,o,\text{stop})$。

$R_t(s,o,\text{stop})$ 被允许,当且仅当 $s \in T(s)$。

规则 B7　主体与客体通信,$R_t(s,o,\text{com-apply})$。

$R_t(s,o,\text{com-apply})$ 被允许,当且仅当:

a)$s \notin RC \parallel o \notin RC$,即主体 s 或者客体 o 不属于任何的冲突集;

b)$s \notin CIS(o) \parallel o \notin CIS(s)$,即通信的双方不在一个冲突集中。

规则 B8　主体撤销通信,$R_t(s,o,\text{com-release})$。

$R_t(s,o,\text{com-release})$ 被允许当且仅当 $D_t(s,o,\text{com-apply}) = 1$,即主体 s 申请与客体 o 通信满足规则 B7 而得到了允许,那么撤销通信总是成立。

规则 B9　主体调整安全等级,$R_t(s,o,\text{level}(L_n))$。

$R_t(s,o,\text{level}(L_n))$ 被允许,当且仅当主体 $s \in T(s) \wedge (\text{state}(o) = \text{stop})$,即 s 是可信主体且客体是非运行状态。

规则 B10　主体申请客体传递内存,$R_t(s,o,\text{mem-tranfer})$。

$R_t(s,o,\text{mem-tranfer})$ 被允许,当且仅当 $(f_1(s) > f_2(o) \wedge f_3(s) \supseteq f_4(o)) \parallel (s \in T(s))$,即主体 s 支配客体 o,或者主体是可信主体。

规则 B11　主体只读映射内存,$R_t(s,o,\text{readonly-map})$。

$R_t(s,o,\text{readonly-map})$ 被允许,当且仅当 $(f_1(s) > f_2(o) \wedge f_3(s) \supseteq f_4(o)) \parallel (s \in T(s))$,即主体 s 支配客体 o,或者主体是可信主体。

规则 B12　主体读、写映射内存,$R_t(s,o,\text{read} \mid \text{write})$。

$R_t(s,o,\text{read} \mid \text{write})$,当且仅当 $[f_3(s) \supseteq f_4(o) \parallel f_4(o) \supseteq f_3(s)]\}$,即读、写映射的双方发生在安全密级相同的主客体之间。

冲突集扩充定理:

a)主体申请资源 $R_t(s,o,\text{apply})$ 通过规则 B2 的允许,那么:

(a)若 $(\forall s_1 \in \text{SYM}, s \in CIS_t(s_1)) \wedge (\exists s_2 \in \text{SYM}, A(s_2,o) = 1)$,修改 $A(s,o) = 1$,扩充访问 s_2 的冲突集:

$$CIS(s_2) = \bigcup_{s_2 \neq s_1 \mid A(s_2,o)=1} s_1$$

即 s_1 包含在 s 冲突集中,且 o 曾经被 s_2 访问过,那么扩充 s_2 的冲突集,将 s_1 包含在其冲突集内。

(b)若 $(\forall s_1 \in \text{SYM}, s \notin CIS_t(s_1)) \wedge (\neg \exists s_2 \in \text{SYM}, A(s_2,o) = 1)$,那么只修改 $A(s,o) = 1$,即若 s 不包含在任何冲突集中且 o 没有被任何主体访问过,那么只修改访问历史矩阵 A。

b)主体请求与客体通信 $R_t(s,o,\text{com-apply})$ 得到规则 B7 的允许,那么:

(a)若 $s \notin RC \parallel o \notin RC$,即主体 s 或者客体 o 不属于任何的冲突集,那么扩充冲突集:

$$\left\{\begin{array}{l} CIS(o) = \bigcup_{s \neq o} s, s \notin RC \wedge o \in CIS(o) \\ CIS(s) = \bigcup_{o \neq s} o, o \notin RC \wedge s \in CIS(s) \\ \text{do\_nothing}, o \notin RC \wedge s \notin RC \end{array}\right\}$$

(b)$s \notin CIS(o) \parallel o \notin CIS(s)$,即通信的双方不在一个冲突集中,扩充其中任何一方:

$$CIS(s) = \bigcup_{s \neq o} CIS(o)$$

那么,若 $\exists (s_1,s_2,\cdots,s_N), (D(s_1,s_2,\text{com-apply}) = 1) \wedge (D(s_2,s_3,\text{com-apply}) = 1) \wedge \cdots \wedge (D(s_{N-1},s_N,\text{com-apply}) = 1)$,将扩展所有 $s_i (1 \leqslant i \leqslant N)$ 的冲突集:

$$\mathrm{CIS}(s_1) = \bigcup_{s_i \neq s_1}^{2 \leqslant i \leqslant N} s_i$$

通过对称性和传递性可以得到,所有经过直接通信或者间接通信的主体都将在一个冲突集中,这也就是前面提到的冲突集联盟。

2.2 VBAC 模型安全性分析

在 VBAC 模型中,规则 B1 ~ B8 采用了 CW 的冲突集核心思想,而 B9 ~ B12 则是采用了 BLP 的多级安全思想。

本文假设系统初始状态 z_0 在利益冲突与安全机密性上是安全的。规则 B2、B3 控制资源的分配问题,完全遵循冲突隔离的思想,满足 CW 简单安全特性,因此保证了资源分配上不存在信息泄露的问题。规则 B7、B8 通过 CW 思想控制虚拟机之间的事件通道,事件通道是虚拟机的一个技术核心,如同传统系统中的信号量机制。虚拟机的许多行为是基于事件通道的,如果控制过于严格,会影响虚拟机的可用性,因此,事件通道的控制需要根据系统的安全需求来设定。但是规则 B2、B3、B7、B8 都存在第三方的隐通道问题,模型中,冲突集扩充定理则是针对这一问题提出的,较好地消除了第三方安全隐患。可见,本模型在控制资源的信息隔离上是安全的。模型中规则 B9 ~ B12 是通过 BLP 思想控制虚拟机中资源共享和通信问题,规则 B10、B11、B12 满足多级安全的安全特性,因此,系统转换将受到规则 B9 ~ B12 的约束,保证了系统的机密性。

通过该模型的设置,能较好地避免系统因为资源分配而导致信息泄露的问题,同时也较好地控制了虚拟机的第三方隐通道问题。

3 模型实现与测试结果

本文设计的基于虚拟机的访问控制模型可分成三部分:主策略、第二策略、第三策略,通过这三种策略的分工来共同完成安全控制策略。

1)主策略 primary_policy 实现冲突集模型,采用 CW 核心思想,负责完成虚拟机的创建、启动、销毁等。该策略以虚拟机为粒度,遵循 VBAC 模型的 B1、B4、B5、B6 规则。

2)第二策略 second_policy 在 Xen 中,资源控制和事件通道管理采用的是 STE 模型,本文延用此安全模型,STE 同样采用了 CW 为利益冲突集思想,它负责管理虚拟硬件资源申请、资源分配、撤销,以及事件通道的管理,在本文提出的模型中,它将遵循 VBAC 模型的 B1、B2、B3、B7、B8 规则,以及冲突集扩充定理。

3)第三策略 third_policy 它控制虚拟机之间的多级安全通信,采用了 BLP 的核心思想,负责管理内存共享和数据传递等问题,遵循 VBAC 模型的 B9、B10、B11、B12 规则。

3.1 测试结果

1)内存资源管理测试

本文设置的实验机器上物理内存为 4 GB,即内存空间中存在页数为 4 GB/4 KB = 1 M 个。对于内存资源的访问控制测试情景设置如下:创建三个 VM,分别为 Dom1、Dom2、Dom3,每个 VM 的配置信息中内存配置如下,Dom1 分配内存 512 MB,Dom2 分配内存 256 MB,Dom3 分配内存 256 MB。

(1)测试方案一 测试不存在冲突关系的情况下,Dom1 和 Dom2 各自被分配的内存空间。测试步骤如下:a)Dom1、Dom2 都不进行安全类型设置,在启动 Dom1 后,检测 Xen 为其分配的内存空间;b)然后关闭 Dom1,启动 Dom2,检测 Dom2 所分配的内存空间。

针对测试方案一,通过 Xentrace 工具及 Xm 中的 debug-keys 得到 Dom1 和 Dom2 的内存分配情况,如图 1 所示。图中,纵轴表示时间轴,横轴表示页面数,实验机器中 4 GB 物理内存,因此内存页为 1 MB,即最大内存页编号为 1 048 576。在 Xen 虚拟机系统中,Xen 占据了起始的 64 MB 内存空间,如图红色所示(见电子版);Domain0 启动后,所有其他内存都由 Domain0 管理,其本身内存分配了 131 072 页,如图绿色所示;在启动 Dom1 时,所分配内存为 65 536 页,其所占页空间如图黄色所示,在关闭 Dom1 之后,收回 Dom1 的内存页;而启动 Dom2 时,Domain0 为其在内存堆中分配内存,如图黑色所示。从测试结果的数据可以看出,Dom2 和 Dom1 是可以共用内存页的。

(2)测试方案二 测试同一冲突集的虚拟机被分配的内存空间。测试步骤如下:a)将 Dom1 设置安全类型标签“A”,并将 Dom2 设置安全标签“B”,在启动 Dom1 后,检测 Xen 为其分配的内存地址段;b)关闭 Dom1,启动 Dom2,查看 Dom2 所分配的内存页状况。

图 2 是方案二所得测试结果。在方案设计中,Dom1 和 Dom2 存在利益冲突关系,因此,在 Dom1 关闭后,尽管 Domain0 收回了其内存,然而,内存历史记录数组已经记录了访问类型;当为 Dom2 分配内存页时,因为冲突关系,导致 Dom1 使用过的内存页不能被分配给 Dom2 使用。图中黄颜色表示 Dom1 使用过的内存页,亮蓝色表示 Dom2 所分配的内存页。从数据统计中可以看出,Dom1 与 Dom2 不存在共享内存页。

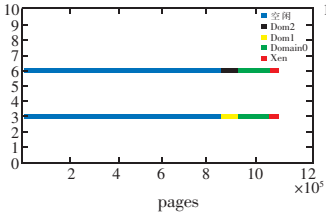

图 1 内存资源控制测试结果一

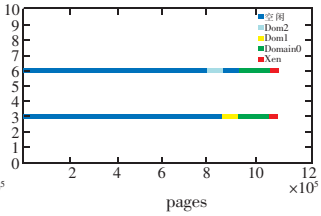

图 2 内存资源控制测试结果二

(3)测试方案三 测试冲突集扩充定理对内存分配的影响。为了能达到测试效果,将 Dom1 的内存配置设置为 512 MB,Dom2 以及 Dom3 内存设置为 256 MB。测试步骤如下:a)将内存使用历史记录数组全部置 0,将 Dom1 设置安全标签“A”,Dom2 设置安全标签“C”,在启动 Dom1 后,检测 Xen 为其分配的内存地址段;b)关闭 Dom1,启动 Dom2,测试 Dom2 所分配的内存页状况;c)创建 Dom3,设置安全标签“E”,关闭 Dom2,启动 Dom 查看 Dom3 所分配的内存页状况。

测试方案三的三种内存分配情况如图 3 所示。黄色表示 Dom1 所使用的 131 072 个内存页,Dom2 表示在 Dom1 关闭后,Domain0 为其分配的内存页;在方案中,Dom2 与 Dom3 冲突,而 Dom2 与 Dom1 共享过内存页,由冲突集扩充定理而导致 Dom1 和 Dom2 具有了相同冲突类型。因此,Dom1 使用过的内存页不能分配给 Dom3,如图亮蓝色表示 Dom3 所分配的内存。

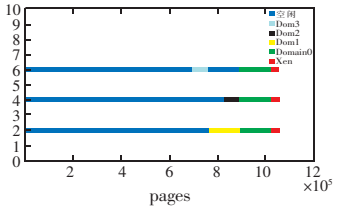

图 3 内存资源控制测试结果三

3.2　结果分析

从测试数据及结果分析中可以看出本文提出的安全模型具有如下几个特点:

a)安全性能高。在安全有效性中,本文对虚拟机管理、事件通道、资源管理以及内存共享上的访问控制进行了安全性测试。测试结果显示:虚拟机管理控制符合 CW 的利益冲突集模型要求;资源管理和事件通道的控制能得到冲突集扩充定义的有效控制;对内存共享上的控制遵循本文 VBAC 模型的多级安全控制规则。

b)空间开销少。在访问控制模块中,安全级与冲突集类型在映射后存储于数组中,策略信息缓存数组大小由安全类型数量 M 和虚拟机数量 N 决定。假设每个数组元素所占空间大小为 1 Byte,那么策略信息缓存数组所占空间为 $M \times N$ Byte。在资源控制上,内存页的安全控制空间开销最大,为了减少空间的开销,本文采用了 Bit-map 的存储方式,大大地缩减了空间开销。

c)存在一定的时间性能损耗。从测试过程中可以看出,访问控制所带来的性能损耗主要是内存分配,因为每次系统启动都将分配大量的内存页,每一页内存的分配都将产生决策信息的读取和判定,从而导致了一定的时间损耗。

4　结束语

虚拟机是云计算的底层关键技术之一,但是目前存在很多对于虚拟机的攻击。针对虚拟机的内部安全隐患,本文提出了一种适用于虚拟机环境的访问控制模型,在 PCW 安全模型的基础上,引入了 BLP 多级安全模型,并对 BLP 进行了相应改进,本文提出的模型同时使用了 CW、PCW 和 BLP,具有安全性能高、空间开销小等优点。仿真实验证明本文所提模型有较强的可行性及安全性。

参考文献:

[1] 冯登国,张敏,张妍,等. 云计算安全研究[J]. 软件学报, 2011,22(1):71-83.

[2] WANG Zhi,JIANG Xu-xian. HyperSafe:a lightweight approach to provide lifetime hypervisor control-flow integrity[C]//Proc of IEEE Symposium on Security and Privacy. 2010:380-395.

[3] SALAUM M. Practical overview of a Xen covert channel[J]. Journal in Computer Virology,2010,6(4):317-328.

[4] LIU Qian,WNAG Guan-hai,WENG Chu-liang,*et al*. A mandatory access control framework in virtual machine system with respect to multi-level security Ⅱ: implementation [J]. China Communications, 2011,8(2):86-94.

[5] RANJITH P,PRIYA C,SHALINI K. On covert channels between virtual machines[J]. Journal in Computer Virology,2012,8(3):85-97.

[6] OKAMURA K,OYAMA Y. Load-based covert channels between Xen virtual machines[C]//Proc of the 25th Annual ACM Symposium on Applied Computing. Sierre: Association for Computing Machinery, 2010:173-180.

[7] WU Jing-zheng, DING Li-ping, WANG Yong-ji, *et al*. Identification and evaluation of sharing memory covert timing channel in Xen virtual machines[C]//Proc of the 4th IEEE International Conference on Cloud Computing. Los Alamitos, CA: IEEE Computer Society, 2011: 283-291.

[8] SAILER R, VALDEZ E, JAEGER T, *et al*. sHype: secure hypervisor approach to trusted virtualized systems, RC23511[R]. [S. l.]: IBM, 2005.

[9] 程戈,金海,邹德清,等. 基于动态联盟关系的中国墙模型研究[J]. 通信学报,2009,30(11):93-100.

[10] CHENG Ge,JIN Hai-jin,ZOU De-qing,*et al*. A prioritized Chinese wall model for managing the covert information flows in virtual machine systems[C]//Proc of the 9th International Conference for Young Computer Scientists. Los Alamitos:IEEE Computer Society,2008.

[11] 石磊,邹德清,金海. Xen 虚拟化技术[M]. 武汉:华中科技大学出版社,2009.

[12] FOLEY S N. Building Chinese Walls in standard UNIX™[J]. Computers & Security,1997,16(6):551-563.

[13] ZHAO Gan-sen,CHADWICK D W. On the modeling of Bell-LaPadula security policies using RBAC[C]//Proc of the 17th IEEE Workshops on Enabling Technologies:Infrastructure for Collaborative Enterprises. Piscataway,NJ:IEEE Press,2008:257-262.

(上接第 221 页)面可以对经过的攻击数据包进行免疫处理,同时可以对可疑的潜在攻击发起路径标志,使攻击源追踪算法具有了动态性和自适应性,节省资源。此外,该路径标志技术可以进一步对通过的数据包进行标记,为之后免疫入侵检测系统针对单一攻击培养生成免疫特征检测器提供了技术基础,为在攻击路径中处理攻击数据包,避免受害主机遭受大规律攻击数据包汇聚而造成的拒绝服务情况。

实验证明,本文提出的方案在收敛效率、误报率等方面均优于传统追踪算法。虽然需要免疫响应服务器的支持,但该服务器并非仅针对路径标志而放置,其主要功能是作为免疫节点在交换网络中处理攻击数据包。

参考文献:

[1] 彭沙沙,张红梅. 计算机网络安全分析研究[J]. 现代电子技术, 2012, 35(4):109 - 116.

[2] 王大伟. 基于免疫的入侵检测系统中检测器性能研究[D]. 哈尔滨:哈尔滨理工大学, 2010.

[3] 闫巧, 雷琼钰. IP 追踪新进展[J]. 小型微型计算机系统,2012,33(9):2027 - 2032.

[4] FORREST S,PERELSON A S,ALLEN L. Self-nonself discrimination in a computer[C]//Proc of IEEE Society Symposium on Research in Security and Privacy and Privacy. 1994: 202 - 212.

[5] 李订芳. 一种提高检测率的免疫优化模型[J]. 微电子学与计算机, 2007, 23(5): 195-197.

[6] GONZALEZ F, DASGUPTA D, NINO L F. A randomized real-valued negative selection algorithm[C]//Proc of the 2nd International Conference on Artificial Immune Systems. 2003: 261 - 272.

[7] TWYCROSS J. Stochastic and deterministic multiscale models for systems biology: an auxin-transport case study[J]. BMC Systems Biology, 2010, 12(9): 29-41.

[8] 闫巧,吴建平,江勇. 网络攻击源追踪技术的分类和展望[J]. 清华大学学报:自然科学版,2005,45(4):497-500.

[9] 张婵. 一种改进的 iTrace 技术的研究[J]. 科学技术与工程, 2007,12(7): 3013-3016.

[10] SAVAGE S, WETHERALL D, KARLIN A, *et al*. Practical network support for IP traceback[C]//Proc of ACM SIGCOMM Conference. [S. l.]:ACM Press,2000:118-128.

[11] 揭摄,孙乐昌. 一种新的非重复性包标记 IP 追踪方案[J]. 计算机工程, 2007, 33(10): 105-107.

[12] 杨海松,李津生. 分布开放式的入侵检测与响应框架——IDRA[J]. 计算机学报,2003,26(9):1177-1182.

[13] 张永铮,肖军,云晓春,等. DDoS 攻击检测和控制方法[J]. 软件学报,2012,23(8):2058-2072.

[14] 宋东辉, 李丽娟. 改进的概率包标记 IP 追踪方法[J]. 计算机工程,2011,37(4):147-149.