Xen混合多策略模型的设计与形式化验证

祝现威 朱智强 孙 磊
(中国人民解放军信息工程大学密码工程学院 郑州 450004)

摘 要 Xen作为一种虚拟化工具因开源、高效等特点而受到越来越多的关注。作为Xen安全的基础,XSM决定了其安全性。原生XSM没有对系统资源进行安全分级,并且以虚拟机为管理对象使得Dom0作为一个唯一管理域不符合最小特权,文中设计了一种混合多策略模型SV_HMPMD。在该模型中,针对BLP引入多级安全标签,从而增加BLP的实用性,并通过DTE和RBAC对特权进行更细致的划分,从而对Dom0特权进行合理限制。提出了一种分层模型,利用该模型对混合模型进行形式化的描述。运用系统不变量构造访问规则的安全属性需求,通过Isabelle/HOL对模型设计与安全需求的一致性进行验证。

关键词 SV_HMPMD,语义模型,形式化证明,Isabelle/HOL定理证明
中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.10.026

Design and Formal Verification of Xen Hybrid Multi-police Model

ZHU Xian-wei ZHU Zhi-qiang SUN Lei
(School of Cryptography Engineering, PLA Information Engineering University, Zhengzhou 450004, China)

Abstract As a popular open-source virtualization tools, XEN has attracted more and more attention. XSM, as a Xen security model, determins its security. Native XSM does not carry on safe differentiated control design to system source and uses Dom0 as unique virtual machines administrative domain that does not meet minimum privileges. According to these questions, we designed a hybrid multi-police model named SV_HMPMD. In order to improve BLP's practicability, the model introduces multi-level security labels. In order to divide the privilege in detail, we combined DTE with RBAC. We designed a hierarchical model that describes SV_HMPMD by formal methods for xsm to verify the consistency between achievements and security requirements by the tools named Isabelle/HOL.

Keywords SV_HMPMD, Semantic model, Formal proof, Theorem proving of Isabelle/HOL

1 引言

云计算作为一种新型的计算模式,以开源、高效、灵活的特点迅速成为计算机技术研究的热点。随着云计算的广泛应用,云安全问题越来越被学术界和工业界所关注。在众多的云安全问题中,作为云计算基础支撑的虚拟化平台的安全问题成为了研究的热点。

Xen是剑桥大学开发的一个开源的虚拟化平台,在现有云计算环境中得到了广泛应用。为了增强其安全性,一些研究人员通过借鉴安全操作系统的理论和技术提出了针对Xen虚拟化平台的安全模型。

Xen的访问控制模型ACM/sHype[1]提供了一种虚拟机之间的分布式访问控制,但是其只有一些简单的安全策略,如STE和中国墙。为了提高安全策略的适用性,Xen的安全模型借鉴了SELinux中RBAC和TE的混合模型对Dom0和DomU进行管理,以虚拟机作为管理单元进行了简单的角色划分,实现了多种访问控制策略,但是由于对Dom0和DomU使用了同样的管理粒度,没有对Dom0中的特权操作进行细粒度的划分,因此造成Dom0特权集中,不符合系统安全的最小特权原则。针对最小特权,美国海军实验室和Joanna Rutkowska分别提出了Xenon[2]和Qubes[3-4]将Dom0分为多个虚拟设备域,构建了访问控制模型,实现了Dom0的细粒度管理,但是其仍以虚拟机或虚拟机组作为控制单元,且未对系统资源进行安全分级,不能满足分级系统的需求。BLP是一种公认的多级安全模型,通过对信息流的控制实现了信息"不上读,不下写"。Wen等[5]将BLP运用于虚拟机系统,以虚拟机作为访问控制对象并定义了访问规则,为在Xen中使用BLP模型提供了可能。马萌等人[6]提出了基于条件随机场的BLP,提高了模型的有效性和准确性。Wen等使用的是传统BLP模型,该模型具有*-特性过于严格的缺点,并且没有实现高效的权限管理。Kuhn[7]提出了一种在多级安全系统中使用RBAC的方案,该方案利用组合算法将范畴映射为尽可能多的角色,由于映射过程没有使用密级概念,因此该方法可以适用于BLP。虽然上述方法解决了Xen的部分安全问题,

到稿日期:2016-09-22 返修日期:2016-12-21 本文受国家重点研发计划项目:协同精密定位技术(2016YFB0501900),国家重点基础研究发展计划("973"计划)基金(2012CB315900)资助。
祝现威(1991-),男,硕士生,主要研究方向为复杂的系统建模与仿真、信息安全,E-mail:1056670972@qq.com;朱智强(1961-),男,博士,教授,主要研究方向为信息安全、软件可靠性;孙 磊(1973-),男,博士,研究员,主要研究方向为云计算基础设施可信增强、可信虚拟化技术。

但是没有给出形式化证明以证明其自身的安全性。危美林等[8]提出了一种宏观和微观相结合的形式化建模方法。钱振江[9]提出了一种以集合论为基础的操作系统形式化描述方法。以上方法均具有借鉴意义，但作者没有将其引入到安全模型的证明中，因此本文尝试使用 Isabelle/HOL 对安全模型进行形式化的证明。

本文给出了一种 Xen 虚拟化环境下的混合多策略模型 SV_HMPMD（Security Virtualtization_Hybrid Multi-Policy Model），其特点如下：

（1）通过 RBAC 和 DTE 对 Dom0 的特权和访问权限进行了细粒度的划分，将 Dom0 单个主体完成的操作分解为多个主体共同完成的操作。

（2）对 BLP 模型进行改进，形成多安全标签的 BLP。与原始的 BLP 为每个主客体都赋予当前安全标签不同，SV_HMPMD模型为主客体赋予了（w_min，r_max），（L_min_o，L_max_o）两个安全标签范围，提高了 BLP 的灵活性。

（3）对可信主体进行划分，使用多个可信主体代替 Dom0 的单个可信主体。加入了特权域和特权状态，对特权范围和特权动态变化进行了管理。

（4）对 SV_HMPMD 模型进行了形式化的安全验证，提出了一种以类型论和高阶逻辑为理论基础的安全模型对象语义模型。通过不动点得来的安全属性证明其安全性。证明 SV_HMPMD 满足国标 GB17859 的安全模型要求。

本文第 2 节对 SV_HMPMD 模型的框架设计进行总体论述；第 3 节阐述了 SV_HMPMD 的具体实现和关键问题的解决方案；第 4 节给出形式化模型框架的描述并通过该模型对安全模型进行描述；第 5 节将安全模型转换成 Isabelle/HOL 语言进行描述，并对其安全性进行形式化证明；最后总结全文并对未来工作进行展望。

2　SV_HMPMD 模型

2.1　SV_HMPMD 的目标

SV_HMPMD 的安全目标是在虚拟化多用户共享信息的环境下防止用户对信息进行非法存储，有效地保护系统信息不被非授权用户使用，并对 Dom0 作为可信主体进行特权的划分，实现细粒度的管理。因此需要将 BLP，RBAC 和 DTE 模型进行有机的集合，实现如下目标：

（1）由于虚拟化平台的多用户性，管理员的行为权限应与用户不同。为实现该目标，需对管理域 Dom0 和用户域 DomU 进行粒度区分。

（2）实现安全域隔离。模型能够提供一种隔离机制来限制非可信域的操作，可保证其他用户的信息不会被窃取。

（3）拥有较强的实用性。传统 BLP 对信息流进行了严格的限制，其实用性不强使得其只能在单机上使用，限制了其在多用户环境下的使用。

（4）模型本身是安全的。模型应具有可验证性，能够被进行形式化建模和验证。

（5）能够限制可信主体的特权，因为可信主体可以执行违反安全约束的操作，如果可信主体特权可执行的特权操作过多，则会威胁系统的机密性。

2.2　SV_HMPMD 的设计

通过上述目标的分析，结合 BLP，RBAC，DTE 3 种访问控制模型来解决以上问题，其设计思路如下：对系统主体进行划分，将主体划分为普通主体和可信主体来区分管理域和用户域的操作。普通主体是在 BLP 标签范围内的主体，可信主体可以负责安全相关的操作。对于普通主体，使用 BLP 和 DTE 来实现其安全需求的一致性；对于可信主体，使用 RBAC 和 DTE 来实现最小特权的访问控制。

SV_HMPMD 模型的框架：DTE 提供一个易实现的访问框架，该框架实现主、客体间的访问，并在保证系统完整性的同时利用 BLP 安全模型保证虚拟域自身与其他域之间访问的机密性。通过可信主体来规定管理域 Dom0 违反 BLP 安全特性但是执行系统操作允许的动作。通过限制域间的转换和特权继承来保证可信主体和普通主体的隔离性，可信主体与可信主体间的特权赋予通过 RBAC 来实现。在 SV_HMPMD 中 RBAC 的主要功能是实现安全标示的分配和管理，普通主体与特权主体访问控制的具体实施由 DTE 模型实现。

3　SV_HMPMD 的具体实现

从上节可知，SV_HMPMD 通过改进的 BLP 模型来实现系统的访问控制，并规定了其安全规则以保证系统的机密性。通过 RBAC 实现对管理域 Dom0 的特权划分以及不同用户 DomU 的静态权限划分。将其与 DTE 相结合作为一种灵活的访问控制模型，实现了系统的完整性和特权受控。接下来将对以上特性的原理进行阐述。

3.1　多策略混合模型的改进 BLP

使用了一种多标签的 BLP 模型，该模型对原始 BLP 模型进行了改进：使用标签范围代替原有的标签，在不降低原有模型机密性的同时提高了模型的灵活性。SV_HMPMD 使用（w_min，r_max）的安全标签范围替代原有模型中严格的当前安全标签，使其在一个相对可靠的范围内实现安全访问。其中，w_min 是主体可以对客体进行写操作的最小安全标签，r_max 是主体可以对客体读的最大安全标签。在此范围内的输出函数为 $ok(s)$。相似的 SV_HMPMD 为客体也赋予了一个标签范围（L_min_0，L_max_0），其中 L_min_0 为客体最小访问安全标签，L_max_0 为客体最大访问安全标签。客体安全级别在此范围内的输出函数为 $ok(o)$。

在使用新的安全标签后，生成的 BLP 安全公理如下。

1）简单安全性：状态 $v\in V$ 满足简单安全性，主、客体动作三元组 $(s,o,x)\in b$，一定存在主体安全等级 $L(s)$ 大于客体最大安全等级 $L\_max_0(o)$。

2）$*$-安全性：

$(s,o,a)\in b\Rightarrow L\_max_0\leqslant r\_max_0$

$(s,o,r)\in b\Rightarrow L\_min_0(o)\geqslant w\_min_s(s)$

$(s,o,w)\in b\Rightarrow ok(o)\subseteq ok(s)$

在 SV_HMPMD 中将主体划分为可信主体与普通主体，普通主体的定义与原 BLP 模型一样，$w\_min(s)=r\_max(s)$，部分可信主体 $w\_min(s)\neq r\_max(s)$，这部分主体只能访问其安全权限达到的客体。对于可信主体，其不受安全策略约束，其最小特权需要被限制。

3.2　可信主体的划分

为了防止因 Dom0 的权限过大而导致其控制整个系统的

情况发生，将 Dom0 的操作划分成若干个可信子集，并通过 RBAC 赋予这些子集相应的“角色”。将这些“角色”分给完成特权操作的进程，用可信用户代替原来的单可信用户。

3.3 完整性的实现

由于可信主体的可违反安全规则特性的存在，因此对安全信息进行不恰当的修改使得可信主体自身的完整性访问控制受到了威胁。DTE 的访问控制关系的不可传递性和域间的可转换性为模型提供了更好的细粒度访问操作。其实现的思想是：将具有相同完整性安全需求的客体设置为同一个型(type)，并使用域交互表(DTT)规定主体对客体的访问规则，防止安全信息泄露给未授权的主体，限制 BLP 中权限过大的可信主体的特权范围，以保证可信主体的最小特权。域之间的信息隔离通过 DTE 模型的 DIT 表来实现，主体间的隔离可防止可信主体通过域间信息流动发生完整性损害。完整性访问控制规则如下：

(1)域对型的访问规则；

(2)域对域的交互规则；

(3)定义可进行域间转换的操作，规定访问客体后能发生某域间的转换，保证系统中 Domain 和通道可以在各自特定的域中。

3.4 特权访问控制

目前，模型对特权操作的管理较少，大多数的管理还只是针对非特权的访问控制，但实际上特权访问的操作严重影响了系统的安全性，比如特权的滥用、特权非隔离的传播都会导致严重的系统安全问题，因此很多病毒和恶意代码通过特权获取来实现恶意传播。这要求模型设计之初既要考虑非特权的访问控制，又要保证特权访问的安全使用，在特权访问带来便利的同时限制其特权范围以保证系统的机密性和完整性。

在 SV_HMPMD 中需要管理的特权操作对象主要是 Domain、共享内存和进程。在实际的特权操作中，Dom0 通过进程实现对共享内存和 DomU 的管理和访问。因此特权访问管理的核心问题是进程的特权管理。而对于 DomU 的特权管理，则通过 RBAC 对其赋予角色来进行全局管理。共享内存的特权执行范围赋予对其执行的进程，特征传递是在进程执行 exec 共享内存之后产生的，这也会引起进程的特权变化。

本文从特权角色检查和特权计算两方面来解决特权范围过大和特权动态变化的问题。对于特权范围的问题，SV_HMPMD 首先通过对同一用户进行静态的角色划分来实现同一角色的行为隔离，然后将特权进程与特定的特权域进行绑定。这些特权域具有特定的特权属性，即每个特权域规定了该域的特权主体所能执行的特权功能，进一步限制了特权的生效范围。对于特权的动态变化，模型引入了特权状态，并通过 DTE 中的域动态转换机制将特权进程的特权状态时刻限制在当前角色和特权域的范围内，以防止特权的动态传播。

特权状态是特权进程变化的标志，从上文特权管理规则可知特权进程是由进程 Domain 的角色 P_r、进程所在的特权域 P_d 以及特权进程所要执行的共享内存共同决定的。本文用(P_i，P_p，P_e)和(f_i，f_e)分别表示特权的状态和共享内存的特权状态，其描述如表 1 所列。

表 1 特权进程和共享内存的特权集合

符号	特权集	对象	描述
P_i	可继承特权集	特权进程	在允许访问的条件下，进程可以将特权传递给后继进程
P_P	许可特权集	特权进程	进程可以执行的特权操作的最大特权集合
P_e	执行特权集	特权进程	当前进程可以执行的特权集合
f_i	可继承特权集	特权进程	对共享内存执行特权操作能获得的特全集合
f_e	执行特权集	特权进程	执行共享内存时，共享内存对其特权进程的特权映射

本文参照 Dragovic[14] 遗传算法，使用遗传算法进行特权计算：

$$p_i' = p_i \wedge f_i$$

$$p_p' = (f_i \vee (p_i' \wedge p_p)) \wedge p_r \wedge p_d$$

$$p_e' = p_p' \wedge f_e$$

其中，(p_p'，p_i'，p_e')为进程执行后的特权状态集。上述公式表示可继承特权状态的特权为当前状态下进程特权状态和共享内存特权状态下的最小特权状态。P_d 为进程所在的域所允许的最大特权集合，表示允许执行的特权集合由当前角色和特权域以及特权进程的上个状态的特权共同决定，最大程度地限制了特权集合的范围，也实现了用户间的特权隔离。

3.5 整体框架

在 SV_HMPMD 中，3 种访问控制模型在功能的完整性、机密性上相互配合。本节将在一个整体的框架上对 SV_HMPMD访问过程进行描述。

作为 BLP 机密性的补充，对特权访问的判断将贯穿整个访问判断过程。特权访问又分为补充型特权访问和管理型特权访问。补充型特权访问是其他访问策略的补充，其特权是有条件的特权，是满足访问规则下的特权。管理型特权访问违反特定访问控制模型的访问规则，例如 MAC_OVERRID 和 DAC_OVERRID，前者可以违反 MAC 访问控制，后者是对 DAC 访问控制的违反。第一种特权访问与其他访问规则共同作用，只有访问符合了访问控制模型的要求后特权访问才能生效。第二种以 Dom0 中所执行的特权访问最为常见，只要满足此特权访问与其他特权访问中的一个便能使访问生效。整个访问控制过程如图 1 所示。

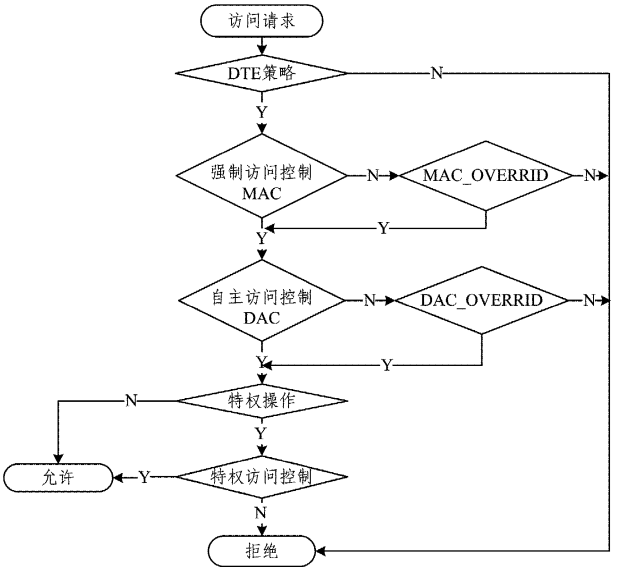

图 1 访问控制过程

在主体发出访问请求后，首先通过 DTE，访问规则判断主体所在的域是否允许执行该操作，若允许，则通过 MAC 和 DAC 进行二次判断；若不允许，则检查其是否符合相应的违反规则特权。如果都不符合，则拒绝访问。

4　混合策略模型验证框架

安全模型是否正确并且能否达到功能需求的一致性？需要对安全模型进行形式化的验证。对于 SV_HMPMD 模型的验证工作，需要防止其状态爆炸并保证其证明强度，因此提出了一种利用 Isabelle/HOL 语言和定理证明来验证安全策略与安全需求的一致性的方法。此方法既有一定的易用性，也能保证安全证明强度。

通过建立安全模型的状态机对混合安全模型进行形式化描述，在状态机模型中系统状态集由多个安全状态组成。安全策略是由约束和安全状态迁移规则来定义的。

定义 1　安全混合策略模型 M 是一个状态自动机，其中 $M=\{V,V_0,L,R,D,T,C\}$。V 为状态集合；V_0 为初始状态集合，$V_0\subseteq V$；L 为安全标签集合；R 为当前状态向下一个状态转变所发出的请求集合；D 为状态间转换是否合法的判断；T 为迁移状态集合 $T\subseteq(v\times L\times R)\rightarrow V\times D$；$C$ 为系统状态 V 和迁移状态 T 的约束及安全策略。

定义 2　设安全模型的安全需求为 R，安全策略模型满足安全需求的条件为：

$(V,C)\models R,(V_0,C)\models R$

安全框架的验证分为 3 部分：

(1)将混合安全策略模型转化为模型的形式化描述，状态机表示状态与状态的转换，通常一个状态机依附于一个类，用于将一类状态的定理证明元素转化成形式化的描述的对象，将安全状态定义为模型的不变量，用迁移规则来定义主客体间的映射。

(2)给出描述的形式化语法，将转化后的模型译成 Isabelle/HOL 能够识别的语言，以便对安全需求进行验证。

(3)使用 Isabelle/HOL 来验证安全模型是否满足安全需求，此步需要使用定理证明其可识别的语言对待验证的安全需求进行形式化的描述，本文通过信息流的方式对安全需求进行描述。

4.1　语义模型框架分析

不同于一般的以安全状态机的方式建模，该模型使用通过主、客体划分的分层模型建模。模型采用运行轨迹来表示状态的变化，因此模型可分为访问规则层、标签层、实现层。访问规则层表示模型的主、客体对象和访问基本规则的描述。本文总共有 get_access，execute，release_access，add_user，delete_user，create_object，authorize_role，revoke_role，authorize_dmd，authorize_D，authorize_executable_domains 11 条访问规则，由于篇幅原因，只对其中 5 条进行说明和形式化证明。

4.1.1　访问规则层

(1)对象集合 M_1

主体行为层对象集合 M_1 包含以下对象集合：

1)用户集合 U。

2)主体集合 $S=\{S_t,S_u,P,AR\}$，其中，S_t 表示可信主体可以某种方式违反可信规则；S_u 为普通主体，除了可信主体外的主体都为普通主体；P 为特权，每种特权依据主体而定；AR 为主体访问操作方式，其状态有 $\{READ,WRITE,APPEND,CREATE,EXECUTE,AUTO\}$。

3)客体集合 $O=\{O_t,O_u,M(O),H(O)\}$，O_t 为可信客体集合，通过执行可信客体产生主体映象；O_u 为普通客体；$M(O)$ 表示主体对客体具有的访问权限；$H(O)$ 表示客体集到客体集幂集的函数。若 $O_1\neq O_2$，则 $H(O_1)\cap H(O_2)=\emptyset$。

4)域型访问关系 $DTT:=D\times T\times A$，表示域对型的访问集合。域 $D=\{d_1,d_2,d_3,\cdots,d_n\}$，$T$ 为对应的型，A 是访问方式，$A\subseteq AR$。

5)模型状态输出 $S=\{subject\_P,output\}$。$subject\_P$ 为主体进程与特权所具有的映射；$output$ 决策输出其取值为 YES 或 NO。

6)模型当前访问状态 $B=\{b|b\in(S,O,AR)\}$。

(2)访问规则描述

定义 3(主体对客体的读写访问)　在访问规则层主体发出对客体的读写访问请求时，通过 DTE 模型的 DTT 表判断访问方式是否恰当。当满足 DTT 访问权限许可或者有该主体对此客体的特权时，决策输出 YES。将执行的系统访问状态添加到现有的访问状态中。

假设在状态 a 下主体 S 向客体 O 发出访问请求，在 w 状态下完成，其访问规则用逻辑公式表示为：

$get\_access\ S\ O$

$w.output=$YES

$w.B.b=s.B.b\cup(s,o,AR)$

if $a.s.(d,t,p)\in DTT\wedge a.Subject.P=M(o)\vee a.Subject.p\in subject\_prive(s)$

invariable other

定义 4(主体 S 对客体 O 直接执行)　当主体的执行域在客体的运行域时，通过 DTT 表判断主体是否可以直接对客体访问或者是否对该客体具有特权。如果允许则决策输出为 YES。

在状态 a 下，主体 S 直接对客体 O 进行访问，在状态 w 下完成，逻辑表示为：

$cxecute\ S\ O$

$w.output=$YES

$w.B.b=s.B.b\cup\{s,o,e\}$

if $a.(d,t,e)\in DTT\vee a.Subject.p\in subject\_prive(s)$

invariable other

定义 5(撤销访问许可)　主体对以某种方式 $x(x\in AR)$ 访问客体的行文发出终止请求，并将撤销的访问状态从现有的访问状态集合中删除。

在状态 a 下主体发出终止访问请求，在状态 w 下完成，逻辑表示为：

$release\ S\ O$

$w.otput=$YES

$w.B.b=a.B.b-\{s,o,x\}$

定义 6(创建客体)　在域 D 中主体 S 请求创建客体 O，其在状态 a 下创建客体并在状态 w 下完成功效。

create object $S\ O$

$w.output=$YES

$w.B.b=a.B.b\cup\{s,o,CREATE\}$

$w.O=a.O\cup\{o\}$

$w.H=H\cup\{o_0,o\}$

if $a.(d,t,CREATE)\in DTT\wedge\{o\}\notin a.O$

4.1.2 标签层

该层主要实现对 BLP 模型的描述，在访问规则层的基础上对主、客体对象添加了安全等级标签。

在设计混合模型时，为了满足其机密性，使用目前应用最为广泛的 BLP 模型。BLP 的核心安全思想是通过主、客体的安全标签来判断是否能发生读写访问，做到“不上读，不下写”。

(1)对象集合 M_2

1)访问规则层对象集合 M_1。

2)安全标签 $LEVEL=\{G,C,\geqq\}$，安全级别 G 表示主体、客体的安全级别；安全范畴 C 表示主、客体可执行的安全范围；$\geqq$ 表示安全标签支配关系，在安全范畴为包含关系时，高等级安全标签可以对低等级安全标签进行支配。

3)标签函数 $f=\{f_s,w\_min,r\_max,L\_min_0,L\_max_0\}$，$f_s$ 表示主体最大安全标签函数；w_min 表示主体最小可写标签；r_max 表示主体最大可读标签；L_min_0 表示客体最小可写标签；L_max_0 表示客体最大可读标签。

(2)规则语义描述

在标签层主要考虑安全等级标签对该行为条件判断的影响或对安全标签最终状态的改变。

1)主体对客体的直接访问

在状态 a 下主体 S 向客体 O 发出的访问请求在状态 w 下完成。

$a.S.AR:=READ\wedge a.f.L\_max_0(o)\leqslant a.f.r\_max_s(s)\vee a.S.AR:=APPEND\wedge a.f.L\_min_0\geqslant a\_min_s(s)\vee a.S.AR:=WRITE\wedge ok(o)\subseteq ok(s)$

2)主体 S 对客体 O 直接执行

如果可以执行，则要求客体的最大可读权限小于主体的最大可写权限。

$L\_max_0(0)\leqslant r\_max(s)$

3)创建客体

在状态 a 下创建客体前先判断其安全标签的权限，在状态 w 下完成功效并对其安全标签进行更新。

$(a.Level.C=L\_max(o)=L\_min(0)\wedge\exists o_0,o\in H(o_0)\wedge(s,o_0,w)\in b\wedge a.Level,G\geqslant f_o(o))\cup a.f.L\_max_o(o)\geqslant a.f.L\_min_o(o)\wedge a.f.r\_max(s)\geqslant a.f.L\_max_o(o)\wedge a.f.L\_min_o(s)\geqslant a.f.w\_min(s)$

当创建客体完成后对新生成的客体标签进行更新。

$w.f=a.f\cup\{(o,Level)\vee(o,L\_min_o,L\_max_o)\}$

4.1.3 实现层

实现层主要实现基础访问规则所涉及的其他行为语义及数据类型。

针对特权变化引入了特权状态，主体执行客体时会引起特权状态的变化，为实现主体间的隔离性引入了互斥关系，防止产生不安全的信息流。通过域之间的转换提高访问的灵活性。

(1)对象集合 M_3

1)M_3 包括标签层对象集合 M_2。

2)实现层的主体集合在访问规则层中的主体集合的基础上增加了 $subject\_p:S\rightarrow P$ 主体与特权之间的对应关系、$subject\_role:S\rightarrow R$ 主体和角色的指派关系。动态互斥域 $DMD:D\rightarrow D,(d_1,d_2)\in DMD$，表示不能同时将 d_1,d_2 赋予同一用户或同一角色。

3)实现层的客体集合在访问规则层主体集合的基础上增加了 $executable\_p:O\rightarrow P$，可执行客体与其特权之间的对应关系。

(2)规则语义描述

在状态 a 下主体 S 拥有访问特权，可直接访问客体 O 并在状态 w 下完成。在实现层分情况描述：

①当主体 S 拥有特权并可直接访问客体 O，且主、客体在同一个域内时，不发生域间的转换。特权状态变化如下：

$w.subject\_p(s).P_i=a.subject\_p(s).P_i\cap a.executable\_p(o).f_i$

$w.subject\_p(s).P_p=(a.executable\_p(o).f_i\vee(w.subject\_p(s).P_i\wedge a.p\_role(subject\_role(s)).P_p\wedge a.p\_domain(subject\_role(s)))$

$w.subject\_p(s).P_e=a.subject\_p(s).P_p\wedge executable\_p(o).f_e$

if $s\in S_T$

②当主、客体不在同一个域中时，需要进行域转换才能完成操作。

$(w.S_t=a.S_t\cup\{s\})\wedge(w.S_u=a.S_u-\{s\})\wedge w.subject\_p(s).P_i=subject\_p(s).P_i\cap execute\_p(o).f_i$

$w.subject\_p(s).P_p=(executable\_p(o).f_i\vee(w.subject\_p(s).P_i\wedge a.p\_role(subject\_role(s))\wedge a.prive\_domain(subject\_role(s))$

$w.subject\_p(s).P_e=w.subject\_p(s).P_p\wedge executable\_p(o).f_e$

if $(d_s,d_o)\in DMD$

$o\in domains\_executables(d_o)$

$role\_domains(d_o)\subset role\_domain(d_s)$

5 SV_HMPMD 的 Isabelle/HOL 验证

5.1 SV_HMPMD 的 Isabelle/HOL 符号表示

本节主要使用 Isabelle/HOL 符号来表示模型，Isabelle/HOL[11-12]是一种形式化验证工具语言，可以实现模型的形式化描述和验证，采用与函数式编程类似的语法规范，支持归纳演算、Lambda 演算以及经典逻辑证明，拥有强大的类型表达能力。Isabelle[13]使用 typedectl a 来引入新的定义，可以用 $x::a$ 进行结构体类型的定义；通过 $simp,auto$ 对目标进行归纳化简；利用 lemma a 进行定理的定义。

首先声明系统的状态变量：

$typedecl\ S$

$typedecl\ O$

$typedecl\ Class$

以上是对主体客体和客体类型的声明。

$record\ Security\ Level=C::“Class\ Set”$

$G::nat$

datatype AccessModel = Read | APPEND | WRITE | CONTROL

datatype Request=get_access|release_access|create_object

types AccessTriple="Subject×Object×AccessBode"

其中，*Security Level* 是安全标签，其包含 *C* 安全范畴和 *G* 安全等级，*AccessModel* 为访问类型说明，*Request* 是访问请求，*AccessTriple* 是访问控制集合。

安全标签定义如下：

types fs="subject⇒SecurityLevel"

wmin="subject⇒SecurityLevel"

rmax="subject⇒SecurityLevel"

Lmin="object⇒SecurityLevel"

Lmax="object⇒SecurityLevel"

Hierarchy="(att×string,att,object)"

对系统状态 *State* 中元素的定义如下：

record State=

S::"subject set"

O::"object set"

CAT::"AccessTriple set"

AM::"AccessTriple set"

w_min::wmin

r_max::rmax

L_min_0*::Lmin*

L_max_0*::Lmax*

其中，*CAT* 是状态的访问集合，集合元素对应模型中的当前访问状态 *b*；*AM* 为自主访问控制矩阵；*w_min* 为最小写权限；*r_max* 为最大读权限；L_min_0 为客体最小可写标签；L_max_0 为客体最大可读标签。

接下来对安全标签的支配关系进行定义：

constdefs

dominates::"SecurityLevel⇒

SecurityLevel⇒bool"

(infixr"@@"65)

"Level1@@Level2≡(if(Senstive Level1≥Senstitive Level2)∧(Category Level1⊇Category Level2) then True elseFalse)"

constdefs

equals::"SecurityLevel⇒SecurityLevel⇒bool"

(infixr"@==@"65)

"Level1@==@Level2≡

(if(Sensitive Level = Sensitive Level2)∧(CategoryLevel1=CategoryLevel2)then True else False)"

以上是 *dominates* 支配关系和 *equals* 相等关系，当标签 1 的安全等级高于标签 2 且标签 1 的安全范畴包含标签 2 时，其关系为支配关系；反之，如果安全等级一样且安全范畴相等时，其关系为相等关系。

5.2　形式化安全状态属性

首先设立不变量，不变量是模型为了保证其安全性而需要遵守的规则。通过证明访问规则和不变量的一致性来说明模型的安全性。本文通过安全状态属性来表示系统的不变量。

敏感级控制属性：

constdefs

LevelControl::"States⇒bool"

"LevelControl v≡(∀Sub.(Sub∈S v)→(f_s v)sub@@(r_max v)Sub)∧(v_max v)Sub@@(w_min v)sub)∧(∀Obj.(Obj∈O v))→(L_max_0 *v)Obj@@(L_min v)Obj)"*

该条安全属性说明主体的安全级别高于主体最小可读和最大可写的安全级别。任何客体都有一个客体最大、最小安全级别组成的安全范围。

通用安全属性：

constdefs

NormalSecurityAUX::"AccessTriple⇒fs⇒AL_*max_0⇒bool"*

"NomalSecurity x Slevel Omax≡(let(s,o,am)=x in (am=read|write)→SLevel s@@omax o)"

constdefs

NormalSecurity::"State⇒bool"

"SimpleSecurity v≡(∀Sub Obj am.(Sub∈S v∧Obj∈O v∧(Sub,Obj,am)∈CAT v)→NormalSecurityAUX(Sub,Obj,am)(f_s v)(L_max v)"

该条安全属性说明主客体进行访问时主体的安全级别大于客体的最大安全级别，这是主、客体访问的基础。

*-安全属性：

constdefs

StarSecurityAUX :: "AccessTriple ⇒ wmin ⇒ rmaxLmin⇒Lmax⇒bool"

"StartSecurity x w_min r_max L_min L_max≡

(let(s,o,am)= x in "

(case am of read⇒r_max@@ L_max_o*)*

|append⇒L_min_o*@@w_min*

|write⇒(r_max@@L_max_o*)∧(w_min@@L_min)*

|control⇒(r_max@@L_max_o*)∧(w_min@@L_min)*

constdefs

StarSecurity::"States⇒bool"

"StarSecurity v≡(∀Sub Obj am.(Sub∈S v∧Obj∈O v∧(Sub,Obj,am)∈CAT v)→StarSecurity(Sub,Obj,am)(w_min v)(r_max v)(L_min_0 *v)(*L_max_0 *v))"*

主要通过规定访问的安全级策略来防止高安全级的信息向低安全级泄露。

访问控制安全属性：

constdefs

ObjectAndCAT::"State⇒bool"

"ObjectAndCAT v≡(∀Sub Obj am. Obj∉O v→(Sub,Obj,am)∈CAT v)"

执行访问控制时访问集中的客体集和系统状态集中的客体集相统一。

5.3　形式化安全证明

初始化系统状态，使系统状态满足安全需求，保证其初始状态是可信的。

theorem InitialSecure:"SecureState Initial_state

apply(simp add:Initial_state_def)

apply(auto simp add:SecureState_def NormalSecurity_def NormalSecurity_def StarSecurity_def StartSecurity_def DiscerrtionarySecurity_def LevelControl_def ObjectAndCAT_def)"

初始化时,使用 *simp* 方法对系统安全状态的单步满足性进行证明。通过 *auto* 进行目标化简。

定理 1 主体对客体的读写访问满足安全属性:

theorem get_accessSecure:"∀ *a. v. Security*→*SecureState (a. send(get_access v rq Sub Obj am))*"

证明:通过构造读写访问函数 *get_access* 进行单步执行,在执行中加入所要满足的安全状态属性来进行逆向求精,当满足当前安全属性时才会进行下一条安全属性检查。当满足所有的安全属性时,证明其访问规则与安全属性一致。

其证明过程如下:

apply(simp add:get_access_def)

apply auto

apply(simp add:SecureState_def NormalSecurityAux_def NormalSecurity _ def StarSecurity _ def StarSecurityAux_def LevelControl_def ObjectAndCAT_def)

apply auto

done

使用 *simp* 方法对 *get _access* 函数进行单步执行,通过 *auto* 进行目标化简,并通过已有的结论进行证明。

类似地,可以证明定理 2,即主体对客体直接执时行满足安全属性。

定理 2 主体对客体直接执行时满足安全属性:

theorem excuteSecure:"∀ *v. SecureState v*→*SecureState (send(excute v rq Sub Obj am))*"

证明过程如下:

apply (auto simp add:execute_def)

apply auto

apply (auto simp add:SecureState_def NormalSecurityAUX_def NormalSecurity_def StarSecurityAUX _def LevelControl_def)

apply auto

done

定理 3 创建客体时满足安全属性:

theorem createobjectSecure:"∀ *w. SecureState* → *SecureState(send(createobject v rq Sub Obj L _min$_o$ L _max$_o$ Obj New InsObj))*"

证明:构造创建客体和更新系统状态的单步执行,首先需系统安全状态为安全,且满足通用最大安全属性和最小安全属性,在此基础上敏感级安全属性需满足访问控制安全属性。

apply(simp add:createobject_def fun_upd_def)

apply auto

apply(simp add:SecureState_def NormalSecurityAUX_def NormalSecurity_def StarSecurity_def LevlControl_def ObjectAndCAT_def)

apply(rule conjI)

apply clarify

apply(rule conjI)

apply simp

apply clarify

apply simp

apply clarify

apply(erule disjE)

apply simp

apply clarify

apply(erule disjE)

apply simp_all

done

其中,*rule* 是利用已有的规则和结论进行验证;*erule disjE* 和 *simp* 则利用 *disjE* 规则进行消除的反向推导;最后使用 *Simp_all* 对所产生的子目标进行化简。

结束语 本文根据 Xen 的隔离性和完整性以及特权控制的安全需求,提出了一种混合模型 SV_HMPMD。SV_HMPMD 通过改进 BLP 来保证系统的机密性,通过 RBAC 和 DTE 来保证系统的完整性和隔离性,在此基础上实现了对系统的特权管理。同时对模型进行分析,定义了模型的迁移规则和不变量,并给出形式化的描述,通过 Isabelle/HOL 语言进行安全需求一致性的证明,使其满足《GB/T 20272—2006》中安全操作系统的形式化要求,即安全操作系统具有可形式化的安全策略模型。

未来工作将对模型进行改进,提高模型的集成度,减少模型在实际使用中的性能开支;将其与现有的安全框架(如 Flask)相结合来提高其应用的普及性。同时,研究模型如何识别隐通道,以防止隐通道对数据的泄露。引入 Isabelle/HOL 进行全自动证明,以降低其形式化的门槛,使管理员进行安全策略配置时可以进行形式化的验证。

参 考 文 献

[1] SAILER R, VALDEZ E, JAEGER T, et al. sHype: Secure hypervisor approach to trusted virtualized systems[J]. IBM Research Report Rc, 2005, 119(2): 331-352.

[2] REITAS L, MCDERMOTT J. Formal methods for security in the Xenon hypervisor[J]. International Journal on Software Tools for Technology Transfer, 2011, 13(5): 463-489.

[3] RUTKOWSKA J, WOJTCZUK R. Qubes OS Architecture. Version 0.3[OL]. http://www. qubes-os. org/attachment/wiki/QubesArchtecture/arch-spec-0. 3. pdf.

[4] RUTKOWSKA J, WOJTCZUK R. Qubes OS Architecture[OL]. http://burrough. org/Documents/Qubes_OS_Architecture. pdf.

[5] WENG C L, LUO Y, LI M, et al. A BLP-Based Access Control Mechanism for the Virtual Machine System[C]// International Conference for Young Computer Scientists (Icycs 2008). Zhangjiajie, Hunan, China, 2008: 2278-2282.

[6] MA M, TANG Z, LI R F, et al. Improved BLP Model Based on CRFs[J]. Computer Science, 2015, 42(8): 138-144. (in Chinese)
马萌,唐卓,李仁发,等. 基于条件随机场的改进型 BLP 访问控制模型[J]. 计算机科学, 2015, 42(8): 138-144.

[7] KUHN R. Role Based Access Control on MLS Systems without Kernel Changed[C]// Proceedings of the ACM Workshop on

Role Based Access Control. ACM，1998.

[8] WEI M L，ZHANG M Q，TANG J，et al. Formal Modeling of Complex Network Security Based on MAS[J]. Computer Science，2015，42(3)：102-105. (in Chinese)
危美林，张明清，唐俊，等. 基于 MAS 的复杂网络安全形式化建模[J]. 计算机科学，2015，42(3)：102-105.

[9] QIAN Z J，LIU W，HUANG H. OSOSM：Operating System Object Semantics Model and Formal Verification[J]. Journal of Computer Research and Development，2012，49(12)：2702-2712. (in Chinese)
钱振江，刘苇，黄皓. 操作系统对象语义模型(OSOSM)及形式化验证[J]. 计算机研究与发展，2012，49(12)：2702-2712.

[10] 屈延文. 形式语义学基础与形式说明[M]. 北京：科学出版社，2009.

[11] KUNČAR O，POPESCU A. A Consistent Foundation for Isabelle/HOL[C]// ITP. 2015：234-252.

[12] ELPHINSTONE K，HEIESER G. From L3 to seL4 what have we learnt in 20 years of L4 microkernels? [C]// Twenty-Fourth ACM Symposium on Operating Systems Principles. 2013：133-150.

[13] NIPKOW T，PAULSON L C，WENZEL M. Isabelle/HOL：a proof assistant for higher-order logic[M]. Springer Science & Business Media，2002.

[14] KINGHT G. LinSec-Linux Security Protection System [EB/OL]. http://www.linsec.org/doc/final.pdf.

[15] 佩莱德. 软件可靠性方法[M]. 北京：机械工业出版社，2012.

[16] CALZAVARA S，RABITTI A，BUGLIESI M. Formal Verification of Liferay RBAC[M]// Engineering Secure Software and Systems. Springer International Publishing，2015：1-16.

(上接第 112 页)

location based on outdated channel state information in wireless multi hop networks [J]. Journal of Electronics and Information Science，2014，36(11)：2750-2755. (in Chinese)
冯维，冯穗力，丁跃华，等. 无线多跳网络下基于过时信道状态信息的跨层资源分配[J]. 电子与信息学报，2014，36(11)：2750-2755.

[2] ZHAO N，WU M H，XIONG W，et al. Research and Simulation of Multi-channel spectrum resource allocation and optimization [J]. Computer Simulation，2016，33(1)：209-213. (in Chinese)
赵楠，武明虎，熊炜，等. 多信道频谱资源优化分配仿真研究[J]. 计算机仿真，2016，33(1)：209-213.

[3] SHAO J J. Research on Key Technologies of channel access and resource allocation in wireless body area network [D]. Hangzhou：Zhejiang University of Technology，2015. (in Chinese)
邵剑集. 无线体域网的信道接入和资源分配关键技术研究[D]. 杭州：浙江工业大学，2015.

[4] LIAO S B，TAN Y M. The convergence analysis of the algorithm for maximizing the utility of wireless networks [J]. Journal of Beijing Institute of Technology，2014，34(8)：807-812. (in Chinese)
廖盛斌，谭运猛. 无线网络效用最大化算法的收敛性分析[J]. 北京理工大学学报，2014，34(8)：807-812.

[5] LI Q，HE D Z，GUAN Y F，et al. A suitable resource allocation method for digital television channel uplink[J]. TV Technology，2015，39(11)：94-98. (in Chinese)
李青，何大治，管云峰，等. 一种适合数字电视上行信道的资源分配方法[J]. 电视技术，2015，39(11)：94-98.

[6] LIU Q，NIU H，XU W，et al. A service-oriented spectrum allocation algorithm using enhanced PSO for cognitive wireless networks[J]. Computer Networks，2014，74：81-91.

[7] GOUDARZI P. Scalable video transmission over multi-hop wireless networks with enhanced quality of experience using swarm intelligence[J]. Signal Processing Image Communication，2012，27(7)：722-736.

[8] XU J C. Multi period mobile communication channel assignment equilibrium model [J]. Science Bulletin，2015，31(10)：145-147. (in Chinese)
许健才. 移动通信中多时段信道分配均衡模型[J]. 科技通报，2015，31(10)：145-147.

[9] FENG W J，LI J J，WANG P. Parameter optimization and sensitivity analysis of cognitive radio based on particle swarm optimization [J]. Computer Science，2011，38(10)：87-90. (in Chinese)
冯文江，李俊建，王品. 基于粒子群算法的认知无线电参数优化及敏感度分析[J]. 计算机科学，2011，38(10)：87-90.

[10] OUYANG D T，HE J S，BAI H T. A constrained particle swarm optimization algorithm for wireless sensor network node localization algorithm [J]. Computer Science，2011，38(7)：46-50. (in Chinese)
欧阳丹彤，何金胜，白洪涛. 一种约束粒子群优化的无线传感器网络节点定位算法[J]. 计算机科学，2011，38(7)：46-50.

[11] LIU L，JIN T，FU L，et al. An improved call admission control and resource allocation for multimedia in wireless networks[J]. Signal Processing，2007，23(3)：343-347. (in Chinese)
刘莉，荆涛，付立，等. 一种优化无线多媒体业务接入允许控制和资源分配算法[J]. 信号处理，2007，23(3)：343-347.

[12] ZHU Y，TANG C，SONG L，et al. Analytical and comparative investigation of 60 GHz wireless channels[J]. Telecommunication Systems，2015，60(1)：179-186.

[13] ZHAO C X，CHEN F L，WANG R C，et al. Research and development of multi objective gateway deployment of wireless Mesh network with integrated channel assignment [J]. Computer Research and Development，2015，52(8)：1831-1841. (in Chinese)
赵传信，陈付龙，王汝传，等. 融合信道分配的无线 Mesh 网络多目标网关部署[J]. 计算机研究与发展，2015，52(8)：1831-1841.

[14] ZHENG P Y，HE S B，ZHANG X Y，et al. A Game-based Channel Assignment for Wireless Mesh Networks [J]. Journal of Chongqing University of Technology(Natural Science)，2013，27(4)：90-95. (in Chinese)
郑鹏宇，何世彪，张馨月，等. 一种基于博弈论的无线网状网络信道分配算法[J]. 重庆理工大学学报(自然科学)，2013，27(4)：90-95.