基于可信级别的多级安全策略及其状态机模型*

谢 钧[1,2+], 许 峰[1], 黄 皓[1]

[1](南京大学 计算机软件新技术国家重点实验室,江苏 南京 210093)

[2](解放军理工大学 指挥自动化学院,江苏 南京 210007)

Trust Degree Based Multilevel Security Policy and Its Model of State Machine

XIE Jun[1,2+], XU Feng[1], HUANG Hao[1]

[1](State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

[2](Institute of Command Automation, PLA University of Science and Technology, Nanjing 210007, China)

+ Corresponding author: Phn: +86-25-84630021, E-mail: yulu_mail@263.net, http://www.nju.edu.cn

Received 2003-05-21; Accepted 2004-01-06

Xie J, Xu F, Huang H. Trust degree based multilevel security policy and its model of state machine. *Journal of Software*, 2004,15(11):1700~1708.

http://www.jos.org.cn/1000-9825/15/1700.htm

Abstract: MLS (multilevel security) is being widely applied in many security critical systems, but it can't implement many important security policies such as 'channel-control'. In this paper, the concept of trust degree is introduced into the MLS to implement policies like 'channel-control' conveniently. An access control state machine model which enforces the trust degree based multilevel security policy is established, and is proved to be secure for this policy. It is also proved that this model can enforce all static information flow policies. An extension of the model is also offered to support the dynamic change of storage objects' security labels. The model avoids the disadvantage of MLS' not being able to resolve the problem of secure downgrading and not taking integrity into consideration, and at the same time it retains the advantage of easy understanding and use enjoyed by the traditional classified policy models.

Key words: security policy; multilevel security; access control model; information flow model

摘 要: 虽然 MLS(multilevel security)被广泛应用于各种安全系统,但是它不能实现信道控制等重要的安全策略.将可信级别的概念引入到 MLS 中,使其可以方便地实现各种信道控制策略.建立了一个实现这种基于可信级别的多级安全策略的访问控制状态机模型,并证明其对定义的策略是安全的,而且可以实现所有静态信息流策略.另外,还扩展了该模型,使其可以支持存储对象安全属性的动态改变.该模型克服了 MLS 不能解决安全降级问题以及不考虑完整性的缺点,同时又保留了传统分级策略模型易理解、易使用的优点.

关键词: 安全策略;多级安全;访问控制模型;信息流模型

中图法分类号: TP309 **文献标识码**: A

* Supported by the Natural Science Foundation of Jangsu Province of China under Grant No.BK2002073 (江苏省自然科学基金)

作者简介: **谢钧**(1973－),男,四川成都人,博士生,讲师,主要研究领域为信息安全,计算机网络;**许峰**(1970－),男,博士生,讲师,主要研究领域为网络安全;**黄皓**(1957－),男,博士,教授,博士生导师,主要研究领域为信息安全,计算机网络.

MLS(multilevel security)安全策略是目前各种安全系统应用最为广泛的一类安全策略.在 MLS 系统中,主体不能读取高于其密级的客体,不能修改低于其密级的客体,其目的是防止高密级信息泄漏给低密级的用户.在这样的系统中,即使存在特洛伊木马并且其骗取了对高密级信息的访问权,也不可能将其读到的秘密信息泄漏给未授权用户.在 MLS 中,任何主体都是不可信任的,即使它是处理高密级信息的用户.因此,在 MLS 系统中,一个用户不能同时处理多种密级的信息.

虽然 MLS 能够很好地防止信息的非授权泄漏,保护信息的机密性,但存在一些明显的缺点.一个缺点是,它不允许主体同时处理多种密级的信息,这使得不违背 MLS 很多功能无法实现.如,文件系统软件若不违背 MLS 的规则,就不能完成其正常的文件管理功能.一些实现 MLS 的安全系统解决这类问题的方法是,将这些主体排除在 MLS 控制的范围以外,认为它们是可信的,给它们一些特权,以绕过访问控制机制访问各种资源.而这些能绕过访问控制机制的特权往往比实际需要的权限更大,因而成为安全的隐患.MLS 的另一缺点是,它没有考虑信息和操作的完整性.

Biba 模型[1]采用与 MLS 相似的方法来维护信息的完整性要求,而 Assured Pipelines 模型[2]和 Clark-Wilson 模型[3]则强调了信息处理过程的完整性.Biba 模型在数学上与 MLS 等价,都是基于格的信息流策略,并且我们可以将这两个格叠加起来形成一个统一的格,同时实现系统的机密性策略和完整性策略.但由于这样的流策略仍然是基于格的,因此不能实现 Assured Pipelines 以及解决多安全级别信息的处理问题.其本质原因在于,基于格的信息流策略是传递的,也就是说,若 $A\sim B,B\sim C$,则 $A\sim C$,其中~表示信息的流向.John Rushby 证明了 MLS 的信息流策略与传递的信息流策略是等价的[4],而传递的信息流策略不能实现“信道控制”[5]等重要的安全问题.信道控制可以用一个有向图来表示,边的方向表示允许的信息流向.比如,一个防火墙系统实际上是一个信道控制系统.我们用图 1 来描述一个简单防火墙系统的信道控制功能.防火墙的外连模块不能和内连模块直接通信,而必须经过防火墙的访问控制模块,在该模块的控制下,外连模块和内连模块进行通信.但是,这样的信道控制模型不能用 MLS 实现.

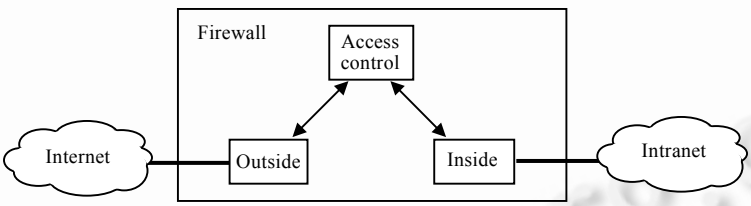

Fig.1 Channel control of a firewall system

图 1 防火墙系统的信道控制功能

从信息处理的角度来看,一些信息处理程序必须同时处理多种安全等级的信息.比如,一个信息管理系统可以综合多种密级的信息,提供给用户一个综合信息(如平均值),但不允许用户知道具体的各项秘密信息.例如,系统允许某用户查询客户总数,但不能查询具体客户的姓名.

有时信息的密级需要降低.在一个加密系统中,高密级的信息被加密程序加密以后,包含秘密信息的密文应该是可以降低密级的,以便通过低密级安全域.这其实是加密的一个重要目的,但却违反 MLS 策略.

因此,在一个 MLS 系统中,为了完成这些功能,必须将安全核外的一些应用进程视为可信进程,使它们可以绕过系统的访问控制机制,如 HP-UX/CMW[6]等.这必将扩大安全防线的范围,但这些可信进程往往获得了比实际需要大得多的特权,因而成为系统的安全隐患.

1 可信级别

虽然 MLS 来源于现实世界中军方的文档处理,但实际上很多涉密信息并不总是从下往上流的,如命令的下达是从上往下流的,而作战命令显然是有密级的.在现实世界中,很多访问秘密客体的主体实际上具有一种可信度的安全属性.比如,一个拥有中等机密信任度的主体应允许读低于其密级的客体,写高于其密级的客体,同时还应被信任不会将其读取的秘密信息泄漏到低密级的客体中,但它不能读取高于其密级的客体.但在 MLS 中,

所有主体都是不可信任的,认为它们都有可能是一个特洛伊木马.实际上,MLS 中的安全级别是一种安全敏感级别,它与可信度是不同的概念.在现实世界中,这两种概念都是存在的,并且不能互相替代.比如,抄写秘密文档的记录员应具有一定的安全敏感级别,而军队指挥官具有的应是相当的可信度.因此,我们可以将可信度的概念引入到 MLS 策略中,为某些主体设置一定的可信级别,以表达更符合实际的安全策略.

由于信息的机密性和完整性都是信息安全中非常重要的内容,我们在基于可信级别的多级安全 TBMLS(trust degree based multilevel security)策略中将信息的机密性和完整性进行了综合考虑.每个主、客体都有一个安全属性.每个安全属性包括 3 个部分:机密性级别 Se、完整性级别 In 和访问类别集 Ca.在主体 S 中有些主体是可信主体,它们的安全属性是它们的可信级别,可信主体集合 $T\subseteq S$.如果用 $s\mathbf{R}o$ 表示允许主体 s 读客体 o,$s\mathbf{W}o$ 表示允许主体 s 写客体 o,则 TBMLS 的访问控制规则可以表示为:

(1) $s\notin T\rightarrow(s\mathbf{R}o\leftrightarrow(Se(s)\geq Se(o)\wedge In(s)\leq In(o)\wedge Ca(s)\supseteq Ca(o)))$;

(2) $s\notin T\rightarrow(s\mathbf{W}o\leftrightarrow(Se(s)\leq Se(o)\wedge In(s)\geq In(o)\wedge Ca(s)\subseteq Ca(o)))$;

(3) $s\in T\rightarrow(s\mathbf{R}o\leftrightarrow(Se(s)\geq Se(o)\wedge Ca(s)\supseteq Ca(o)))$;

(4) $s\in T\rightarrow(s\mathbf{W}o\leftrightarrow(In(s)\geq In(o)\wedge Ca(s)\cap Ca(o)\neq\varnothing))$.

规则 1 表示非可信主体不能窃取高密级信息,不能读取低完整性的信息,以防破坏其完整性,也不能非法获取无关信息.规则 2 表示非可信主体不能将高密级信息泄漏到低密级客体中,不能破坏高完整性客体的信息完整性,也不能将信息泄漏给无关客体.规则 3 表示低机密信任等级的可信主体不能窃取高密级信息和无关信息,但能读取低完整性的信息,并被信任有能力保证该信息不会破坏其完整性.规则 4 表示低完整性信任级别的可信主体不能破坏高完整性客体的信息完整性,但能写低密级客体,并被信任有能力保证不会将高密级信息泄漏到低密级客体中,也不会将信息泄漏给无关客体.

很明显,TBMLS 可以很方便地实现客体安全级别降级及处理多种安全级别的问题.后面我们会证明,该策略模型也能实现各种信道控制策略,并具有很强的策略表达能力.

由于在 MLS 策略中引入了可信级别,使得可信主体被纳入到访问控制策略中,使最小特权原则得到较好的体现,从而不让可信主体因有特权绕过访问控制机制而获得过多的权限,成为系统的安全隐患.Type Enforcement[2]与 DTE(domain and type enforcement)[7]虽然都声称不需要可信主体就能实现包括信道控制策略在内的各种访问控制策略,然而,在 DTE 中所有的主体都隐含地具有某种可信度,但无法为其设定明确的级别.因此,在 DTE 中没有统一可遵循的级别概念来指导其安全策略的设计,对所有流必须单独进行考虑.但是,如果不意识到其主体的这种可信度的存在,实际上是危险的,某些关键域因为具有比较高的可信度,应该得到更多的验证和测试.同样,在文献[8]中也声称其不需要使用可信进程.该安全模型将主体的读安全属性和写安全属性分离,从而可以实现各种跨多种安全级别的访问控制策略.但是,一个主体的读安全属性和写安全属性之所以可以不同,实际上是因为该主体存在某种级别的可信度.

2 一个状态机模型

为了考虑该策略的访问控制机制的实现,我们建立了一个访问控制机制的状态机模型,并证明了该模型的一些特性.由于 John Rushby 的非传递不干扰模型为我们研究信道控制策略提供了一个很好的方法,因此,在我们的系统模型中引入了非传递不干扰信息流理论及相关结论.

2.1 非传递不干扰模型[4]

John Rushby 的非传递不干扰模型用一个有限状态自动机来描述系统,并给出了系统安全的定义.

定义 1. 系统 M 包括以下集合和函数:

- 系统状态集合 S,初始状态 $s_0\in S$;
- 系统安全域集合 D;
- 系统操作集合 A,包括系统执行的输入、输出、命令、指令等;
- 系统输出集合 O;

- 单步状态转换函数 $step$:$S\times A\to S$;
- 系统运行函数 run:$S\times A^*\to S$,$run(s,\Lambda)=s$,$run(s,a\circ\alpha)=run(step(s,a),\alpha)$,其中$\Lambda$表示空串;
- 输出函数 $output$:$S\times A\to O$;
- 函数 dom:$A\to D$ 表示系统每个操作的执行域.

在非传递不干扰模型中,$\rightsquigarrow$为 D 上的自反关系,表示安全域间的直接干扰关系.一个系统的信息流安全策略可以用关系$\rightsquigarrow$来表达.非传递不干扰模型给出了系统 M 对由关系$\rightsquigarrow$表达的信息流策略(简称为策略$\rightsquigarrow$)安全的条件.

定义 2. 先定义两个辅助函数:

- 函数 $sources$:$A^*\times D\to P(D)$,$P(D)$表示 D 的幂集.

$$sources(\Lambda,u)=\{u\}.$$

$$sources(a\circ\alpha,u)=\begin{cases}sources(\alpha,u)\cup\{dom(a)\} & \text{if } \exists v: v\in sources(\alpha,u)\wedge dom(a)\rightsquigarrow v\\ sources(\alpha,u) & \text{otherwise}\end{cases}.$$

- 函数 $purge$:$A^*\times D\to A^*$.

$$purge(\Lambda,u)=\Lambda$$

$$purge(a\circ\alpha,u)=\begin{cases}a\circ purge(\alpha,u) & \text{if } dom(a)\in sources(a\circ\alpha,u)\\ purge(\alpha,u) & \text{otherwise}\end{cases}.$$

系统 M 对策略$\rightsquigarrow$是安全的,当 $output(run(s_0,\alpha),a)=output(run(s_0,purge(\alpha,dom(a))),a)$.

在定义 2 中,$v\in sources(\alpha,u)$表示 $v=u$,或者存在一个α的子序列,该序列由域 $w_1,w_2,\ldots,w_n$ 执行的操作组成,并且 $w_1\rightsquigarrow w_2\rightsquigarrow\ldots\rightsquigarrow w_n$,其中 $v=w_1$,$u=w_n$. $purge(\alpha,u)$表示从α中删除了所有不能干扰 u 的域的操作的一个子序列(包括直接干扰与间接干扰).该安全定义表示,若在系统的执行序列中去除所有不能干扰 u 的域的操作后,对 u 执行的操作不产生任何影响,则系统对该非干扰策略是安全的.

由于该定义需要考虑系统的状态与执行序列,不便于系统的验证,因此 John Rushby 给出并证明了一个只涉及单步状态的系统安全的展开条件.

定理 1. 如果系统 M 存在某等价关系$\sim^u$,满足下列条件,则 M 对信息流策略$\rightsquigarrow$是安全的:

(1) $s\sim^{dom(a)}t\to output(s,a)=output(t,a)$;

(2) $s\sim^u t\wedge s\sim^{dom(a)}t\to step(s,a)\sim^u step(t,a)$;

(3) $\neg(dom(a)\rightsquigarrow u)\to s\sim^u step(s,a)$.

2.2 TBMLS访问控制模型

下面在定义 1 的基础上定义一个实现 TBMLS 的访问控制模型,并证明它是安全的.

定义 3. 在定义 1 定义的系统 M 的基础上增加以下集合和函数的定义:

- 可信安全域集合 T,$T\subseteq D$;
- 系统存储对象名称集合 N;
- 系统存储对象取值集合 V;
- 存储对象取值函数 $contents$:$S\times N\to V$;
- 函数 $observe$:$D\to P(N)$表示安全域可以观察到的存储对象集合;
- 函数 $alter$:$D\to P(N)$表示安全域可以修改的存储对象集合;
- 机密性级别属性函数 Se:$D\cup N\to$N,N 表示自然数集合;
- 完整性级别属性函数 In:$D\cup N\to$N;
- 系统访问类别集合 C;
- 访问类别集属性函数 Ca:$D\cup N\to P(C)$.

系统 M 的访问监控器满足如下假设:

(1) $n\in observe(u)\leftrightarrow(u\notin T\wedge Se(u)\geq Se(n)\wedge In(u)\leq In(n)\wedge Ca(u)\supseteq Ca(n))\vee(u\in T\wedge Se(u)\geq Se(n)\wedge Ca(u)\supseteq Ca(n))$;

(2) $n\in alter(u)\leftrightarrow(u\notin T\wedge Se(u)\leq Se(n)\wedge In(u)\geq In(n)\wedge Ca(u)\subseteq Ca(n))\vee(u\in T\wedge In(u)\geq In(n)\wedge Ca(u)\cap Ca(n)\neq\varnothing)$;

(3) $s\sim^{dom(a)}t\rightarrow output(s,a)=output(t,a),a\in A$,其中$\sim^u$为系统状态集$S$上的等价关系:$s\sim^u t\leftrightarrow(\forall n\in observe(u):contents(s,n)=contents(t,n)),s,t\in S$;

(4) $\forall n\in N:s\sim^{dom(a)}t\wedge(contents(s,n)=contents(t,n))\rightarrow contents(step(s,a),n)=contents(step(t,a),n)$;

(5) $contents(step(s,a),n)\neq contents(s,n)\rightarrow n\in alter(dom(a))$;

(6) $u\rightsquigarrow v\leftrightarrow\exists n\in N:n\in alter(u)\wedge n\in observer(v),\forall u,v\in D$.

访问监控器假设(3)表示输出结果只与执行域所能观察到的对象值有关.假设(4)表示执行域对系统中对象值的改变只依赖于对象原值及执行域观察的结果,因此,假设(4)允许执行域修改不可见对象,并支持"*append*"和"*rewrite*"操作.文献[4]中,访问监控器假设(4)不支持"*append*"操作.在文献[4]的模型中,执行域若不能观察某对象,则修改后的该对象值与原对象值无关.为了支持广泛需要的"*append*"操作,将此条件放宽了.我们证明模型仍然是安全的.假设(5)表示执行域只能改变可修改对象的值.假设(6)表示信息只能通过存储对象,从一个安全域流向另一安全域.

定理 2. 定义 3 定义的系统 M 对信息流策略$\rightsquigarrow$是安全的.

证明:分别证明系统 M 满足定理 1 的 3 个条件:

1. $s\sim^{dom(a)}t\rightarrow output(s,a)=output(t,a)$.

由访问监控器假设(3)直接得证.

2. $s\sim^u t\wedge s\sim^{dom(a)}t\rightarrow step(s,a)\sim^u step(t,a)$.

$s\sim^u t\wedge s\sim^{dom(a)}t\rightarrow\forall n\in observe(u):s\sim^{dom(a)}t\wedge(contents(s,n)=contents(t,n))$ (访问监控器假设(3))

$\forall n\in observe(u):s\sim^{dom(a)}t\wedge(contents(s,n)=contents(t,n))\rightarrow\forall n\in observe(u):contents(step(s,a),n)=contents(step(t,a),n)$ (访问监控器假设(4))

$\forall n\in observe(u):contents(step(s,a),n)=contents(step(t,a),n)\rightarrow step(s,a)\sim^u step(t,a)$ (访问监控器假设(3))

由此得证.

3. $\neg(dom(a)\rightsquigarrow u)\rightarrow s\sim^u step(s,a)$.

即证明:$\exists n\in observe(u):contents(s,n)\neq contents(step(s),n)\rightarrow dom(a)\rightsquigarrow u$.

$\exists n\in observe(u):contents(s,n)\neq contents(step(s),n)\rightarrow n\in alter(dom(a))\wedge n\in observe(u)$ (访问监控器假设(5))

$n\in alter(dom(a))\wedge n\in observe(u)\rightarrow dom(a)\rightsquigarrow u$ (访问监控器假设(6))

由此得证. □

定理 3. 定义 3 定义的系统 M 能够实现所有安全域间的静态信息流策略$\rightsquigarrow$.

证明:只要存在某算法对任意域间的静态信息流策略$\rightsquigarrow$为系统M找到一组安全属性配置,即得证.算法如下:

首先设置初始值,对$\forall u\in T,\forall n\in N:Se(u)=se_1,Se(n)=se_2,se_1>se_2;In(u)=in_1,In(n)=in_2,in_1>in_2,Ca(u)=\{u\}$.

若策略中存在$u\rightsquigarrow v$,(1) 将u,v设置为可信域,即$u,v\in T$;(2) 设置一个新存储对象$n\in N$,并使$Ca(n)=\{un,n,nv\}$,并将un添加到$Ca(u)$中,使$un\in Ca(u)$;将nv添加到$Ca(v)$中,使$nv\in Ca(v)$.由于对象名和域名是唯一的,则对$\forall e\in D\cup N:e\neq u\rightarrow un\notin Ca(e),e\neq v\rightarrow nv\notin Ca(e),e\neq n\rightarrow n\notin Ca(e)$,即$e\neq u\wedge e\neq v\wedge e\neq n\rightarrow Ca(n)\cap Ca(e)=\varnothing$.

根据访问监控器假设(1)、(2)、(6),$n\in alter(u)\wedge n\in observe(v)$,即$u\rightsquigarrow v$.且$e\neq u\wedge e\neq v\rightarrow n\notin alter(e)\wedge n\notin observe(e)$,即$n$不会影响除了$u,v$之间以外的其他任何域之间的信息流关系$\rightsquigarrow$. □

虽然定理 3 的证明中所有域都是可信域,但实际的安全策略的实现并不需要这样.可信域往往用于信道控制、受控降级等非传递流策略,而对于传递流策略,如基于格的流策略,则完全可以用非可信域实现.在后面的防火墙系统实例中可以清楚地看到这一点.

2.3 支持存储对象安全属性的动态改变

以上定义的系统模型不允许存储对象安全属性的动态改变,但在很多应用中,为了提高系统性能,在处理数据时都尽量避免存储单元的复制,因此在不同安全级别的域之间传递数据时,需要对存储对象的安全属性动态地加以改变.例如,在防火墙系统中,希望某些存储单元直接从外连模块通过访问控制模块传递到内连模块,而

不进行内存复制,从而提高防火墙的性能.

定义 4. 为了支持存储对象安全属性的动态改变,修改定义 3 的某些定义如下:

- 函数 $observe$:$S\times D\to P(N)$,安全域可以观察到的存储对象集合与系统状态相关;
- 函数 $alter$:$S\times D\to P(N)$,安全域可以修改的存储对象集合与系统状态相关;
- 安全域机密性级别属性函数 Sd:$D\to$N,N 表示自然数集合;
- 安全域完整性级别属性函数 Id:$D\to$N;
- 安全域访问类别集属性函数 Cd:$D\to P(C)$;
- 存储对象机密性级别属性函数 Sn:$S\times N\to$N;
- 存储对象完整性级别属性函数 In:$S\times N\to$N;
- 存储对象访问类别集属性函数 Cn:$S\times N\to P(C)$;
- 存储对象安全属性函数 $Classification$:$S\times N\to$N$\times$N$\times P(C)$.

系统 M 的访问控制访问监控器假设重写如下:

(1) $n\in observe(s,u)\leftrightarrow(u\notin T\wedge Sd(u)\geq Sn(s,n)\wedge Id(u)\leq In(s,n)\wedge Cd(u)\supseteq Cn(s,n))\vee(u\in T\wedge Sd(u)\geq Sn(s,n)\wedge Cd(u)\supseteq Cn(s,n))$;

(2) $n\in alter(s,u)\leftrightarrow(u\notin T\wedge Sd(u)\leq Sn(s,n)\wedge Id(u)\geq In(s,n)\wedge Cd(u)\subseteq Cn(s,n))\vee(u\in T\wedge Id(u)\geq In(s,n)\wedge Cd(u)\cap Cn(s,n)\neq\varnothing)$;

(3) $s\sim^{u}t\leftrightarrow(observe(s,u)=observe(t,u)\wedge alter(s,u)=alter(t,u)\wedge(\forall n\in observe(s,u):contents(s,n)=contents(t,n))),s,t\in S$,且:$s\sim^{dom(a)}t\to output(s,a)=output(t,a),a\in A$;

(4) $\forall n\in N:s\sim^{dom(a)}t\wedge(contents(s,n)=contents(t,n))\to contents(step(s,a),n)=contents(step(t,a),n)$;

(5) $\forall u\in D:s\sim^{dom(a)}t\wedge(observe(s,u)=observe(t,u))\to observe(step(s,a),u)=observe(step(t,a),u)$;

(6) $\forall u\in D:s\sim^{dom(a)}t\wedge(alter(s,u)=alter(t,u))\to alter(step(s,a),u)=alter(step(t,a),u)$;

(7) $contents(step(s,a),n)\neq contents(s,n)\to n\in alter(s,dom(a))$.

访问监控器假设(3)表示输出结果不仅与可观察对象的值有关,还与可修改对象集合有关,因此对不能修改的对象执行写操作,访问监控器可以返回一致的错误信息而不会产生隐蔽信道.若输出结果与可修改对象集合无关,如对不能修改的对象或不存在的对象执行写操作,访问监控器则返回不同的错误信息,会产生隐蔽信道(通过有规律地创建和删除对象可将信息传递给对该对象有修改权但无观察权的域).假设(4)、(5)、(6)表示执行域对系统中对象值、可观察对象集合及可修改对象集合的改变只依赖于对象原值原集合及执行域观察的结果.

定理 4. 定义 4 定义的系统 M 若对$\forall s\in S$ 满足以下条件,则对信息流策略$\sim$是安全的:

(1) $observe(step(s,a),u)\neq observe(s,u)\to dom(a)\sim u$;

(2) $alter(step(s,a),u)\neq alter(s,u)\to dom(a)\sim u$;

(3) $n\notin observe(s,u)\wedge n\in observe(step(s,a),u)\to n\in observe(s,dom(a))$;

(4) $\exists n\in N:n\in alter(s,u)\wedge n\in observe(s,v)\to u\sim v$.

证明:分别证明系统 M 满足定理 1 的 3 个条件:

1. $s\sim^{dom(a)}t\to output(s,a)=output(t,a)$.

由访问监控器假设(3)直接得证.

2. $s\sim^{u}t\wedge s\sim^{dom(a)}t\to step(s,a)\sim^{u}step(t,a)$.

即证明:

$s\sim^{u}t\wedge s\sim^{dom(a)}t\to(observe(step(s,a),u)=observe(step(t,a),u)\wedge alter(step(s,a),u)=alter(step(t,a),u)\wedge(\forall n\in observe(step(s,a),u):contents(step(s,a),n)=contents(step(t,a),n)))$ (访问监控器假设(3))①

$s\sim^{u}t\wedge s\sim^{dom(a)}t\to s\sim^{dom(a)}t\wedge(observe(s,u)=observe(t,u))$ (访问监控器假设(3))

$s\sim^{dom(a)}t\wedge(observe(s,u)=observe(t,u))\to observe(step(s,a),u)=observe(step(t,a),u)$ (访问监控器假设(5))

即 $s\sim^{u}t\wedge s\sim^{dom(a)}t\to observe(step(s,a),u)=observe(step(t,a),u)$ ②

同理,由访问监控器假设(6)可证:$s\sim^{u}t\wedge s\sim^{dom(a)}t\to alter(step(s,a),u)=alter(step(t,a),u)$ ③

由②③:$s\sim^{u}t\wedge s\sim^{dom(a)}t\to(observe(step(s,a),u)=observe(step(t,a),u)\wedge alter(step(s,a),u)=alter(step(t,a),u)$ ④

对于$\forall n\in observe(step(s,a),u)$,分情况加以讨论:

(1) 若 $n\notin observe(s,u)\wedge n\in observe(step(s,a),u)$,根据条件(3),有 $n\in observe(s,dom(a))$,则:

$s\sim^{dom(a)}t\rightarrow contents(s,n)=contents(t,n)$ (访问监控器假设(3))

$s\sim^{u}t\wedge s\sim^{dom(a)}t\rightarrow contents(step(s,a),n)=contents(step(t,a),n))$ (访问监控器假设(4))

(2) 若 $n\in observe(s,u)\wedge n\in observe(step(s,a),u)$,根据 $s\sim^{u}t\rightarrow contents(s,n)=contents(t,n)$,有

$s\sim^{u}t\wedge s\sim^{dom(a)}t\rightarrow contents(step(s,a),n)=contents(step(t,a),n))$ (访问监控器假设(4))

由(1)(2):$s\sim^{u}t\wedge s\sim^{dom(a)}t\rightarrow\forall n\in observe(step(s,a),u):contents(step(s,a),n)=contents(step(t,a),n))$ ⑤

由④⑤,等价式①得证.

3. $\neg(dom(a)\sim u)\rightarrow s\sim^{u}step(s,a)$,即证:$\neg(s\sim^{u}step(s,a))\rightarrow dom(a)\sim u$.

分情况讨论:

(1) 若 $observe(step(s,a),u)\neq observe(s,u)\vee alter(step(s,a),u)\neq alter(s,u)$,则由条件(1)和条件(2),有 $dom(a)\sim u$;

(2) 若 $observe(step(s,a),u)=observe(s,u)\wedge alter(step(s,a),u)=alter(s,u)$,则:

$\neg(s\sim^{u}step(s,a))\rightarrow\exists n\in observe(s,u):contents(s,n)\neq contents(step(s,a),n)$ (访问监控器假设(3))

$\exists n\in observe(s,u):contents(s,n)\neq contents(step(s,a),n)\rightarrow\exists n\in N:n\in alter(s,dom(a))\wedge n\in observe(s,u)$

(访问监控器假设(7))

$\exists n\in N:n\in alter(s,dom(a))\wedge n\in observe(s,u)\rightarrow dom(a)\sim u$ (条件(4))

由(1)(2)得证. □

定理 4 的条件(3)表示,只有在执行域可观察存储对象内容的情况下才能改变其安全属性,使得原来不能观察到它的域可以观察到它.定理 4 表明,若系统存储对象的安全属性可以通过执行域的操作改变,则系统不仅可能通过存储对象的值来传递信息,还可能通过可观察对象集合及可修改对象集合来传递信息.安全策略必须考虑到这样的信息流,否则会导致隐蔽信道的出现.

定理 5. 访问监控器满足以下条件的系统 M(定义 4)对信息流策略$\sim$是安全的.

(1) $(Classification(s,n)\neq Classification(step(s,a),n))\rightarrow(dom(a)\in T\wedge Sd(dom(a))\geq Sn(s,n)\wedge Id(dom(a))\geq In(s,n)\wedge Cd(dom(a))\supseteq Cn(s,n)\wedge Id(dom(a))\geq In(step(s,a),n)\wedge Cd(dom(a))\cap Cn(step(s,a),n)\neq\varnothing)$;

(2) $u\sim v\leftrightarrow(u\in T\wedge v\in T\wedge Ca(u)\cap Ca(v)\neq\varnothing)\vee(u\in T\wedge v\notin T\wedge In(u)\geq In(v)\wedge Ca(u)\cap Ca(v)\neq\varnothing)\vee(u\notin T\wedge v\in T\wedge Se(u)\leq Se(v)\wedge Ca(u)\subseteq Ca(v))\vee(u\notin T\wedge v\notin T\wedge Se(u)\leq Se(v)\wedge In(u)\geq In(v)\wedge Ca(u)\subseteq Ca(v))$.

证明:分别证明系统 M 满足定理 4 的 4 个条件:

1. $observe(step(s,a),u)\neq observe(s,u))\rightarrow dom(a)\sim u$.

$(observe(step(s,a),u)\neq observe(s,u))\rightarrow\exists n:(n\notin observe(s,u)\wedge n\in observe(step(s,a),u))\vee(n\in observe(s,u)\wedge n\notin observe(step(s,a),u))$

由于 u 的安全属性不变,因此,$Classification(s,n)\neq Classification(step(s,a),n),dom(a)\in T$.

分情况讨论:

(1) $n\notin observe(s,u)\wedge n\in observe(step(s,a),u)$.

若 $u\notin T$,则:

$Sd(u)\geq Sn(step(s,a),n)\wedge Id(u)\leq In(step(s,a),n)\wedge Cd(u)\supseteq Cn(step(s,a),n)\wedge(Sd(u)<Sn(s,n)\vee Id(u)>In(s,n)\vee\neg(Cd(u)\supseteq Cn(s,n)))$

$Id(dom(a))\geq Id(u)\wedge Cd(dom(a))\cap Cd(u)\neq\varnothing\wedge dom(a)\in T\wedge u\notin T$ (条件 1)

$n\notin observe(s,u)\wedge n\in observe(step(s,a),u)\wedge u\notin T\rightarrow dom(a)\sim u$ (条件 2)①

若 $u\in T$ 则:

$Sd(u)\geq Sn(step(s,a),n)\wedge Cd(u)\supseteq Cn(step(s,a),n)\wedge(Sd(u)<Sn(s,n)\vee\neg(Cd(u)\supseteq Cn(s,n)))$

$Cd(dom(a))\cap Cd(u)\neq\varnothing\wedge dom(a)\in T\wedge u\in T$ (条件 1)

$n\notin observe(s,u)\wedge n\in observe(step(s,a),u)\wedge u\in T\rightarrow dom(a)\sim u$ (条件 2)②

由①②:$n\in observe(s,u)\wedge n\notin observe(step(s,a),u)\rightarrow dom(a)\sim u$.

(2) $n\in observe(s,u)\wedge n\notin observe(step(s,a),u)$.

与(1)类似,容易证明 $n\in observe(s,u)\wedge n\notin observe(step(s,a),u)\rightarrow dom(a)\rightsquigarrow u$.

由(1)(2)得证.

2. $alter(step(s,a),u)\neq alter(s,u)\rightarrow dom(a)\rightsquigarrow u$.

分情况讨论 $n\notin alter(s,u)\wedge n\in alter(step(s,a),u)$和 $n\in alter(s,u)\wedge n\notin alter(step(s,a),u)$,与第 1 种情况类似,容易证明 $dom(a)\rightsquigarrow u$.

3. $n\notin observe(s,u)\wedge n\in observe(step(s,a),u)\rightarrow n\in observe(s,dom(a))$.

由于 u 的安全属性不变,

$$n\notin observe(s,u)\wedge n\in observe(step(s,a),u)\rightarrow Classification(s,n)\neq Classification(step(s,a),n)$$

根据访问监控器假设(1)、(2)和条件(1)可证:$n\notin observe(s,u)\wedge n\in observe(step(s,a),u)\rightarrow n\in observe(s,dom(a))$.

4. $n\in alter(s,u)\wedge n\in observe(s,v)\rightarrow u\rightsquigarrow v$.

由访问监控器假设(1)、(2)和条件(2)容易证明. □

定理 5 的条件(1)使得可信域只能将自己既能观察又能修改的对象移入到它能修改的对象集合中,这样可以使得改变存储对象安全属性的信息流与原信息流策略一致起来.由条件(2)可以看出,信息流策略只与安全域的安全属性相关.定义 4 还不支持存储对象的创建与删除,但很容易扩充定义 4 来支持存储对象的创建与删除.

3 防火墙系统实例

为了说明 TBMLS 在实际应用中的策略表达能力,我们设计了一个简单的防火墙系统模型,如图 2 所示.该防火墙系统的安全策略要求从外网进入系统的数据必须通过访问控制模块的检查才能进入内网,反之亦然.在模型中,该策略的解释为:所有从 Outside 模块流入 Inside 模块的信息必须经过 Access control 模块.除此之外,系统还要求所有模块可以读取系统配置信息,但不能对其修改;所有模块可以追加日志信息,但不能读取该信息.为了实现以上信息流策略,我们为每个模块及相关存储对象设置相应的安全属性.安全属性(s,i,c)表示实体的机密性级别为 s,其完整性级别为 i,而访问类别集为 c,$c\in P(\{O,I\})$.为了实现访问控制模块的信道控制功能,我们将 Access control 模块设置为可信主体.可以从图中看出,信息流除了通过访问控制模块能向下流入内外模块以外,其他所有信息流都只能从下向上流,显然,该信息流策略是违反 MLS 的.从该实例中我们可以看到,虽然访问控制模块是可信实体,但由于其密级低于日志而不能读取日志信息,其完整性级别低于配置文件而不能篡改配置信息,因此其行为仍然受到访问控制机制的严格控制,并在其职责范围内完成信息降密的任务.

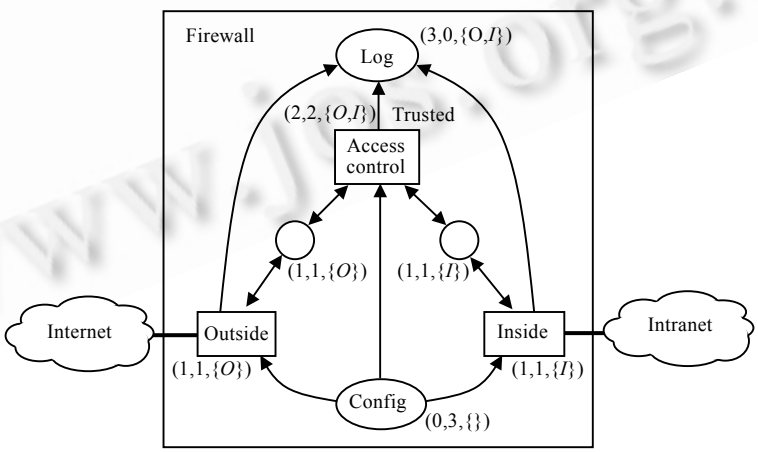

Fig.2 An instance of firewall system

图 2 一个防火墙系统实例

4 结束语

基于网格的流策略虽然具有优美的结构,但它只是一种对安全策略简单理想的设想.基于这种思想的 MLS 策略确实能够有效地解决特洛伊木马问题,但同时也限制了其应用范围.为了解决实际的一些应用问题,将所有必须违反 MLS 策略的进程都视为可信进程,使其获得能够绕过访问控制机制的特权,这无形中扩大了系统可信计算基的大小,违反了最小特权原则,也增加了系统验证的难度.TBMLS 综合考虑了信息的机密性和完整性,并引入可信度的概念,从而将可信进程纳入到安全策略中来,体现了最小特权原则,降低了系统的安全风险.TBMLS 可以很方便地实现客体安全级别降级和处理多种安全级别等问题.同时我们证明了该策略模型也能实现各种信道控制策略,并具有很强的策略表达能力.Type Enforcement[2]与 DTE[7]以及文献[8]虽然都声称不需要可信主体就能实现包括信道控制策略在内的各种访问控制策略,但实际上,在这些策略模型中都隐含有可信度的问题,只是没有显式地表示出来.TBMLS 在 MLS 中引入可信级别,保留了传统分级策略模型易理解和易使用的特点,同时,其明确的可信级别概念有利于安全策略的设计.

References:

[1] Sandhu RS. Lattice-Based access control models. IEEE Computer, 1993,26(11):9~19.

[2] Thomsen DJ, Haigh JT. A comparison of type enforcement and Unix setuid implementation of well-formed transactions. In: Proc. of the 6th Annual Computer Security Applications Conf. Tucson: IEEE Computer Society Press, 1990. 304~312.

[3] Clark DD, Wilson DR. A comparison of commercial and military computer security policies. In: Proc. of the 1987 IEEE Symp. on Security and Privacy. Oakland: IEEE Computer Society Press, 1987. 184~194.

[4] Rushby J. Noninterference, transitivity, and channel-control security policies. Technical Report, CSL-92-02, Menlo Park: Stanford Research Institute, 1992.

[5] Rushby J. Design and verification of secure systems. In: Proc. of the 8th ACM Symp. on Operating System Principles. Pacific Grove: ACM Press, 1981. 12~21.

[6] Zhong Q, Edwards N. Security risk control of COTS-based applications. Technical Report, HPL-97-108, Bristol: HP Laboratories, 1997.

[7] Walker KM, Sterne DF, Badger LM, Petkac MJ, Sherman DL, Oostendorp KA. Confining root programs with domain and type enforcement (DTE). In: Proc. of the 6th USENIX Security Symp. San Jose: USENIX Association, 1996. 21~36.

[8] Schellhorn G, Reif W, Schairer A, Karger P, Austel V, Toll D. Verification of a formal security model for multiapplicative smart cards. In: Proc. of the 6th European Symp. on Research in Computer Security. Toulouse: Springer-Verlag, 2000. 17~36.