doi:10.11887/j.cn.201401029 http://journal.nudt.edu.cn

基于预授权的机密性和完整性动态统一模型*

张 俊,徐鲁威,孟庆德,冯昌林
(海军装备研究院,北京 100161)

摘 要:目前的访问控制模型无法对机密性、完整性和可用性做到合理的统一控制,尤其是对动态的、随机的访问请求控制不完善,使得攻击者总能找到脆弱点,也使得信息系统在实际应用中无法避免用户误操作引起的安全问题。提出了一种基于预授权的机密性和完整性访问控制模型,将BLP模型和Biba模型结合起来,通过引入预授权机制,对一些随机动态的访问活动进行合理控制。运用条件控制项,对主体执行任务的权限进行实时监控,动态地授予和取消主体执行任务的权限,实现系统机密性和完整性的统一,同时保证其具有较高的可用性,有利于信息的双向流动。给出了模型的应用实例,并对其安全性进行了证明。

关键词:机密性;完整性;任务;角色;预授权
中图分类号: TP309 **文献标志码**: A **文章编号**:1001-2486(2014)01-0167-05

Confidentiality and integrity dynamic union model based on pre-authorization mechanisms

ZHANG Jun, *XU Luwei*, *MENG Qingde*, *FENG Changlin*
(Naval Academy of Armament, Beijing 100161, China)

Abstract: With the current access control model, a reasonable unified control over confidentiality, integrity and availability cannot be achieved; especially the dynamic random access request control is far from perfect, not only always leaving some weak points open to possible attacks, but also bringing some unavoidable security problems caused by user errors in practical applications. A kind of confidentiality and integrity access control model based on the pre-authorization mechanisms is put forward. By combining BLP model and Biba model, and introducing the pre-authorization mechanisms, the reasonable control can be achieved over the dynamic random accesses activities. By making use of the condition control, the authority of subject performing the task is monitored timely, and granted or canceled dynamically. So the system's confidentiality and integrity can both be realized, while guaranteeing its high availability, which is advantageous to the two-way flow of information. Finally, the application example of the model is given and its security is proved.

Key words: confidentiality; integrity; task; role; pre-authorization

操作系统的安全性是信息系统安全运行的基础,高安全等级操作系统需要同时保证信息的机密性,完整性和可用性。

BLP(Bell&LaPadula)[1]模型是安全操作系统实现中最经典的多级安全(Multi-Level Security, MLS)策略机密性模型,模型可以简单描述为:不向上读,不向下写。该模型能有效防止敏感信息的非授权泄漏。完整性模型有Biba[2-3]模型,Lipner矩阵模型,Clark-Wilson[4-5]模型等,在实际系统实现中最常见的是Biba模型,该模型可简单描述为:不向下读,不向上写。Biba模型能有效防止敏感信息的非授权修改。

要实现高安全等级操作系统,需要同时实现机密性模型(BLP)和完整性模型(Biba)。现实中经常会出现一些临时的、随机的访问,这些访问不能同时满足BLP模型和Biba模型,但这些访问又是完成任务所必需的,这就要求系统能合理处理这些访问请求。本文提出的基于预授权的机密性和完整性访问控制模型主要是针对那些临时的、动态出现的不能同时满足BLP模型和Biba模型的访问行为实施控制,能根据主体环境状况、主体可信状态、任务执行的情况等动态地授予或撤销授权,在保证系统资源机密性和完整性的前提下,实现系统的可用性。

1 模型描述

预授权机制是一种主动的动态访问控制机制,与传统的访问控制机制有很大不同。传统的

* **收稿日期**:2013-07-15
基金项目:国家科技重大专项资助项目(2012ZX03002003)
作者简介:张俊(1976—),女,湖北武汉人,博士,E-mail:zhhu030714@163.com

访问控制模型的授权都是基于主体和客体的属性,以及请求的操作三者决定的。强制访问控制的授权是基于主体、客体安全标签和请求操作三者决定,如 BLP 模型。主体的属性是主体的机密性等级,客体的属性是客体的机密性等级,根据所请求的操作,比较主客体的安全等级,决定是否允许主体操作客体。这些安全属性是静态不变的,无论何时都是有效的。再如,自主访问控制的授权一般用三元组(s,o,p)表示,s 表示主体,o 表示客体,p 表示操作许可。若存在(s,o,p),则表明 s 可对 o 执行 p 操作许可;否则,s 不能对 o 执行 p 操作。这些三元组都是预先定义好并静态地存放在系统中,除非管理员删除某三元组,否则此三元组无论何时都是有效的。从用户的权限限制来看,这种授权是被动的、消极的。这种授权方式不适合这种动态、随机出现的访问行为,若按常规方法,管理员将该访问行为静态地写入访问控制列表中,则用户就永远具有执行该客体的权限,这样会导致用户具有过多的权限而带来安全隐患[6-8]。

文献[9]中提出了根据主体的历史标签,调整主体的安全级别,操作完成后,调整主体的历史标签控制下一次访问。这种方法不可避免的导致权限缺失,因为读了高密级的客体,主体的安全级别会越来越高,以前能写的客体因为主体安全级别提高,现在不能写了;因为写了低密级的客体,主体的安全级别则会越来越低,以前能读的客体因为安全级别降低,现在不能读了。这种方法将严重影响系统的可用性,在实际的系统中也是不可行的。以军事系统为例,参谋级别最低,司令级别最高。一次偶然的机会参谋看了司令的文件,而文献[9]对这种行为是不控制的,只要主体第一次发出的访问请求,都予以放行,只是事后对其安全等级进行调整以防止泄密。这样,参谋的级别被提高到了司令的级别,则该参谋以前能干的工作现在不能干了,比如以前能向部长汇报材料,现在不能干了,现在只能读文件,这样是能保证该参谋不泄密,但是军事系统中就少了一个能干的参谋。由此可见文献[9]提出的模型是不具有可用性的。

本文提出的预授权机制,是一个以任务为基础,对用户进行临时授权的过程[10-12]。授权在先,执行任务在后。先由任务管理员授权用户执行任务资格,对安全等级不满足 BLP 模型和 Biba 模型的用户,安全管理员对其所担当的角色进行临时调整,相当于对主体的安全等级进行临时调整,这种授权是经过任务管理员许可的。在任务执行前和执行过程中都要对主体行为及其环境进行监控,随时可以撤销授权。任务完成后即撤销授权,进行审计等相关操作。这是一个动态的过程,能在保证机密性和完整性的同时,保证系统的可用性[13-14]。下面给出该机制的形式化描述。

定义 1 S 表示主体集合,O 表示客体集合,OPS 表示操作集合,$OPS=\{r,a,w,e,c\}$,其中 r 表示只读,a 表示添加,w 表示读写,e 表示执行,c 表示创建,单个操作记为 op。

定义 2 组织机构中信息系统所涉及的所有人员称为用户(user)。单个用户记为 $user$,用户集合记为 $USERS$,它是访问请求主体的集合。

定义 3 角色(role):组织机构中业务职能与责任的抽象描述,角色通过业务职能的继承关系表述。单个角色记为 $role$,角色集合记为 $ROLES$。

定义 4 任务(task):分派给某组织机构或个人作为其部分职责,需在限定时间内按一定工作流程完成的工作称为任务。它是一个可区分的动作,规定了角色对客体所进行的操作。

定义 5 任务执行者($tactor$):任务执行者是任务运行过程中,响应会话后产生的一个动态对象,是用户以某角色执行任务的代理。单个任务执行者记为 $tactor$,执行者集合记为 $TACTORS$。$tactor=(user,role,task)$,其中,$user$ 是 $tactor$ 代理的用户,$task$ 是 $tactor$ 执行的任务,$role$ 是 $tactor$ 执行任务 $task$ 所需要的角色。

定义 6 用户角色关系(UA):用户至角色的指定关系中的多对多映射。$UA\subseteq USER\times ROLES$。对任意的用户 $user$ 和角色 $role$,若($user$,$role$) $\in UA$,则表示用户 $user$ 被指派了一个角色 $role$。

定义 7 用户任务关系(UT):用户至任务的指定关系中的多对多映射。$UT\subseteq USER\times TASKS$。对任意的用户 $user$ 和任务 $task$,若($user$,$task$) $\in UT$,则表示用户 $user$ 可以承担 $task$ 的部分工作。

定义 8 角色任务关系(TA):角色至任务的指定关系中的多对多映射。$TA\subseteq TASK\times ROLES$。对任意的任务 $task$ 和角色 $role$,若($task$,$role$) $\in TA$,则表示角色 $role$ 承担任务 $task$ 的部分工作。

定义 9 关系(OBJSA):客体至任务的指定关系中的多对多映射 $OBJSA\subseteq O\times TASKS$。对任意的客体 $o\in O$ 和任务 $task$,若(o,$task$) $\in OBJSA$,则表示完成任务 $task$ 需要使用客体 o。由于任务的完成是一动态过程,因而客体指定只表明了先决客体,即完成本任务时必需的客体。

定义 10 安全等级集合 $L=L_1\times L_2$,其中 L_1

$=C\times K, L_2=I\times K$, C 是机密性等级集合, I 是完整性等级集合, K 是类别集,机密性安全等级集合用 L_1 表示,完整性安全等级集合用 L_2 表示。$l=(l_1,l_2)=((c,k),(i,k))$, $l'=(l_1',l_2')=((c',k'),(i',k'))$,则 $l_1\geqslant l_1'$ *iff* $c\geqslant c', k\supseteq k', l_2\geqslant l_2'$ *iff* $i\geqslant i', k\supseteq k'$[15]。

定义 11　$\forall v\in V=(B\times M\times F\times H)$ 表示一个系统状态。其中 $B=S\times O\times OPS$,表示哪些主体对哪些客体有访问权限以及什么样的访问权限,它包含了那些要使用的权限。$M=S\times O\times OPS$,表示访问控制矩阵,包含自主权限的集合。因为安全等级存在差异,M 里的权限可能是不可用的。Y^X 表示函数 $f:X\to Y$。安全等级函数 $F=\{(f_s,f_c,f_o,i_c,i_o)\mid f_s\in L_1^S, f_c\in L_1^S, f_o\in L_1^O, i_c\in L_2^S, i_o\in L_2^O\}$,其中 f_s 表示主体的最高机密性等级, f_c 表示主体的当前机密性等级, f_o 表示客体的机密性等级, i_c 表示主体的完整性等级, i_o 表示客体的完整性等级。H 为客体层次[15]。

定义 12　$\sum(R,D,W,z_0)\subseteq X\times Y\times Z$ 表示系统,其中 R 表示访问请求的集合, D 表示结果集合, $W\subseteq R\times D\times V\times V$ 表示系统行为集合, z_0 表示系统初态。N 表示正整数集合, $X=R^N$ 是请求序列的集合, $Y=D^N$ 是决策序列的集合, $Z=V^N$ 是状态序列的集合。

定义 13　规则 $\rho: R\times V\to D\times V$[16]。

定义 14　函数 $taobj: TASKS\to 2^O$,每项任务至一组客体的函数映射。

定义 15　函数 $tarole: TASKS\to 2^{ROLES}$,每个任务至一组角色的函数映射。

定义 16　函数 $rtp: ROLES\times TASKS\to\{true, false, error\}$, *true* 表示角色 *role* 具有执行任务 *task* 所需的权限, *false* 表示角色 *role* 不具有执行任务 *task* 所需的权限, *error* 表示错误。

定义 17　函数 $rolech: USERS\times ROLES\to ROLES$, $rolech(user, role_1)=role_2$ 表示将用户 *user* 的担当的角色由 $role_1$ 转变为 $role_2$。

定义 18　函数 $cprecond: S\times O\times OPS\to\{true, false, error\}$ 表示 s 对 o 执行操作 *ops* 的所有前提条件控制项进行检查,所有条件控制项为真时结果为 *true*,否则为 *false*, *error* 表示错误。

定义 19　函数 $concond: S\times O\times OPS\to\{true, false, error\}$ 表示 s 对 o 执行操作 *ops* 的所有当前条件控制项进行检查,所有当前条件控制项为真时结果为 *true*,否则为 *false*, *error* 表示错误。

定义 20　函数 $prepriv: USERS\times ROLES\times ROLES\to\{true, false\}$, $prepriv(user, role_1, role_2)=true$ 表示安全管理员将用户 *user* 所担当的角色 $role_1$ 改变为角色 $role_2$, $prepriv(user, role_1, role_2)=false$ 表示安全管理员没有将用户 *user* 所担当的角色 $role_1$ 改变为角色 $role_2$。

定义 21　函数 $req: S\times O\times OPS\to\{n, y, error\}$, y 表示允许主体 s 对 o 执行操作 *ops*, n 表示不允许主体 s 对 o 执行操作 *ops*, *error* 表示错误。

安全公理 1　系统状态 (b,m,f,h) 满足安全公理 1,当且仅当 $\forall(tactor,o,op)\in b, o\in taobj(task)$,有 $op\in m[s,o]$,其中 *tactor* 是主体 s 的运行代理[16]。

安全公理 2　系统状态 (b,m,f,h) 满足安全公理 2,当且仅当 $\forall b=(tactor,o,r/e)\in B\Rightarrow(f_c(tactor)\geqslant f_o(o))\wedge(i_o(o)\geqslant i_c(tactor))\wedge(o\in taobj(task))$。

安全公理 3　系统状态 (b,m,f,h) 满足安全公理 3,当且仅当 $\forall b=(tactor,o,a)\in B\Rightarrow(f_o(o)\geqslant f_c(tactor))\wedge(i_c(tactor)\geqslant(i_o(o)))\wedge(o\in taobj(task))$。

安全公理 4　系统状态 (b,m,f,h) 满足安全公理 4,当且仅当 $\forall b=(tactor,o,w)\in B\Rightarrow(f_o(o)=f_c(tactor))\wedge(i_c(tactor)=(i_o(o)))\wedge(o\in taobj(task))$。

定义 22　如果对于所有的 $(r\times v)\in R\times V$, v 满足安全公理 1,且对请求 r 满足安全公理 2 ~ 4 中的一条,则称规则 ρ 是保持安全的。

规则 1(预授权规则)　if $(user, task)\in UT$, $(user, role_1)\in UA$, $rtp(role_1, task)=false$, $rtp(role_2, task)=true$,且 $\forall role\in tarole(task)$,都有 $role_2\leqslant role$, if $tactor=(user, role_2, task)$ 对 *task* 中的操作满足安全公理 1 ~ 4,则 $rolech(user, role_1)=role_2$ if $tactor=(user, role_2, task)$, $(o, task)\in OBJSA$, if $fcprecond(tactor, o, op)\wedge concond(tactor, o, op)=true$ $req(tactor, o, op)=y$

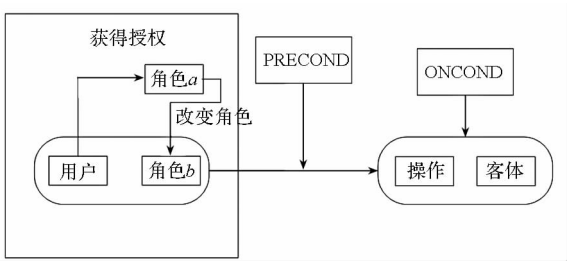

图 1　预授权机制原理图

Fig. 1　A model of preauthorization

预授权机制原理图如图 1 所示。该授权规则的含义是,如果任务的管理者指定用户 *user* 执行任务 *task*,但用户 *user* 所担当的角色 $role_1$ 不具

备执行任务 *task* 的权限,而角色 $role_2$ 是具有执行任务 *task* 的权限的最小角色,角色 $role_2$ 的安全等级、客体的安全等级及请求的操作满足安全公理1~4的要求,则安全管理员将 *user* 由角色 $role_1$ 转变为角色 $role_2$。当用户以角色 $role_2$ 请求执行任务时,首先检查 *precond* 是否得到满足,满足时,将允许用户 *user* 以角色 $role_2$ 执行任务,使得其临时具备执行任务 *task* 的权限。在用户以角色 $role_2$ 执行任务的过程中,定期对其条件 *oncond* 进行检查,一旦条件 *oncond* 不满足,则撤销用户 *user* 担当角色 $role_2$ 的权限。在用户以角色 $role_2$ 执行完任务后,撤销 *user* 担当 $role_2$ 的权限,恢复其担当角色 $role_1$ 的权限。

2 模型应用及其安全性证明

假设在安全状态 $v=(b,m,f,h)$ 下,当 $x=a\in m$, $f_c(tactor)>f_o(o)$, $i_c(tactor)\geqslant i_o(o)$ 时,有请求 $req(tactor,o,a)$,若严格执行 BLP 和 Biba,访问将被拒绝。若应用本文提出的预授权模型,建立规则2,则能在满足 BLP 和 Biba 的同时,实现对客体的临时写。

规则 2 $v=(b,m,f,h)$,当 $x=a\in m$, $f_c(tactor)>f_o(o)$, $i_c(tactor)\geqslant i_o(o)$ 时,对请求 $req(tactor,o,a)$ 作如下处理:

令 $tactor=(user,role_1,task)$, $tactor'=(user,role_2,task)$, $f_c(tactor')=f_o(o)$, $i_c(tactor')=i_c(tactor)$

if $(user,task)\in UT\wedge(o,task)\in OBJSA$

$rolech(user,role_1)=role_2$

if $prepriv(user,role_1,role_2)=true$

则有 $f_c(tactor')=f_o(o)$, $i_c(tactor')\geqslant i_o(o)$

构造 $b'=b\cup\{(tactor',o,a)\}$, $f_c'(tactor')=f_c(tactor')$, $i_c'(tactor')=i_c(tactor')$

if $cprecond(tactor,o,op)\wedge concond(tactor,o,op)=true$

then $req(tactor',o,a)=y$

进入 (b',m',f',h) 状态

else $req(tactor',o,a)=n$, $rolech(user,role_2)=role_1$

else $req(tactor,o,a)=n$

else $req(tactor,o,a)=n$

定理 1 规则2是保持安全的。

证明 规则的前半部分为授权规则,角色 $role_1$ 为用户 *user* 的原始角色,但以其安全等级来执行写客体 *o* 不满足安全公理3要求,因此为了满足系统的安全公理,需要任务管理员许可,由安全管理员对其角色进行调整,即 $rolech(user,role_1)=role_2$,调整后主体的安全等级为 $f_c(tactor')=f_o(o)$, $i_c(tactor')=i_o(tactor)$,安全状态为,$b'=b\cup\{(tactor',o,a)\}$, $m'=m\cup\{(tactor',o,a)\}$, $f_c'(tactor')=f_o(o)$, $i_c'(tactor')=i_o(tactor)$, $f_o'(o)=f_o(o)$, $i_o'(o)=i_o(o)$,即有 $f_c'(tactor')=f_o(o)$, $i_c'(tactor')\geqslant i_o(o)$。对 $x=a$,满足安全公理3的要求,但此时主体不一定能执行写操作,在 $cprecond(tactor,o,op)\wedge concond(tactor,o,op)=true$ 时,主体才可以执行写操作。

此时 (b',m',f',h) 满足安全公理3(满足 BLP 模型和 Biba 模型)且 $b'=b\cup\{(tactor',o,a)\}$ 满足安全公理1。

所以状态 (b,m,f,h) 到状态 (b',m',f',h) 的转换是保持安全的,即该规则是保持安全的。

定理 2 如果 v_o 是安全状态,系统中的每一个转换规则 ρ 都是保持安全的,则系统 $\Sigma(R,D,W(\rho),v_o)$ 是安全系统。

证明 反证法。假设 $\sum(R,D,W(\rho),v_o)$ 是不安全的。

令 $(xr,y,z)\in\sum(R,D,W(\rho),v_o)$ 中有一个非安全状态。令 t 是 T 中使得 v_t 是一个不安全状态的最小元素。因为 v_0 是安全的,因此 $t>0$,即 v_{t-1} 是安全的,根据定义 $\sum(R,D,W(\rho),v_o)$, $(xr_t,y_t,v_t,v_{t-1})\in W(\rho)$。对于 v_t,v_{t-1},有唯一规则 $\omega\in\rho$ 使得 $\omega(xr_t,v_{t-1})=(y_t,v_t)$。因为 ω 是保持安全的,且 v_{t-1} 是安全的,则 v_t 是安全的。导出矛盾,因此系统 $\sum(R,D,W(\rho),v_o)$ 是安全的。

定理 3 如果 v_o 是安全状态,则在规则2下,系统 $\sum(R,DF,W(\rho),v_o)$ 是安全系统。

证明 因为规则2是保持安全的,由定理2可得证系统 $\sum(R,D,W(\rho),v_o)$ 是安全系统。

3 结束语

本文将 BLP 和 Biba 动态地结合起来,特别是引入预授权机制对临时的、随机的访问控制行为进行合理控制。当任务执行者安全属性不具备访问客体的安全等级时,由任务管理者许可,安全管理员临时授权用户执行任务的权限。同时利用条件控制项对用户的访问行为进行严格的监控,一旦任何一个条件控制项不满足,就取消对用户的临时授权。这不同于传统的访问控制模型,权限授予后就静态不变。这符合最小特权的原则。这种临时授权机制,根据任务的需要,允许对用户的角色进行临时调整,这更加符合工作实际,为生产型信息系统中信息的双向流动、客体资源的机密

性和完整性提供了有力保障。

参考文献(References)

[1] Bell D E, LaPadula L J. Secure computer systems: mathematical foundations [R]. The MITRE Corporation, Technical Report M74 – 244, 1973.

[2] Biba K J. Integrity considerations for secure computer system[R]. ESD-76-372, PSAF Electronic System Division, Hanscom Air Force Base, 1977.

[3] Biba K J. Integrity Considerations for secure computer systems [R]. ESD-TR-76 – 372, USA Air Force Electronic System Division, 1997.

[4] Clark D D, Wilson D R. A comparison of commercial and military computer security policies[C]//Proceedings of 1987 IEEE Symposium on Research in Security and Privacy. IEEE Computer Society Press, 1987:184 – 194.

[5] Polk W. Approximating clark-wilson access triples with basic UNIX controls[C]//Proceedings of the 4th USENIX UNIX Security Symposium, 1993:145 – 154.

[6] Sandhu R S, Jaehong P. Usage control: a vision for next generation access control [C]//Proceedings of Second International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security, 2003:17 – 31.

[7] Michiharu K, Satoshi H. XML document security based on provisional authorization[C]//Proceedings of the 7th ACM Conference on Computer and Communications Security, 2000: 87 – 96.

[8] 夏启寿, 范训礼, 殷晓玲. 基于时间的 RBAC 转授权模型[J]. 西北大学学报,2008, 38(6): 932 – 936.
XIA Qishou, FAN Xunli, YIN Xiaoling. RBAC delegation model research based on time [J]. Journal of Northwest University, 2008, 38(6): 932 – 936. (in Chinese)

[9] 石文昌,孙玉芳,梁洪亮. 经典 BLP 安全公理的一种适应性标记实施方法及其正确性[J]. 计算机研究与发展,2001, 38(11): 1366 – 1372.
SHI Wenchang, SUN Yufang, LIANG Hongliang. An adaptable labeling enforcement approach and its correctness for the classical BLP security axioms [J]. Journal of Computer Research & Development, 2001, 38(11): 1366 – 1372. (in Chinese)

[10] Thomas R K, Sandhu R S. Task-based authorization controls (TBAC): a family of models for active and enterprise-oriented authorization management [C]//IFIP Conference Proceedings, 1997:166 – 181.

[11] Osborn S, Sandhu R, Munawer Q. Configuring role-based access control to enforce mandateory and discretionary access control policies [C]//Proceedings of ACM Transactions on Information and System Security, 2000: 85 – 106.

[12] Sandhu R S, Edward J C, Hal L F, et al. Role-based access control models[J]. IEEE Computer, 1996, 29(2): 38 – 47.

[13] 付松龄,谭庆平. 基于任务和角色的分布式工作流安全模型[J]. 国防科技大学学报,2004, 26(3):57 – 62.
FU Songling, TAN Qingping. Security task & role based distributed workflow model[J]. Journal of National University of Defense Technology,2004, 26(3):57 – 62. (in Chinese)

[14] 何鸿君,罗莉,曹四化,等. 基于用户意愿的文件访问控制策略[J]. 国防科技大学学报,2007, 29(6):54 – 58.
HE Hongjun, LUO Li, CAO Sihua, et al. A file access control policy based on user's intention [J]. Journal of National University of Defense Technology,2007, 29(6):54 – 58. (in Chinese)

[15] Zhang J, Yun L J, Zhou Z. Research of BLP and Biba dynamic union model based on check domain [C]// Proceedings of International Conference on Machine Learning and Cybernetics (ICMLC2008). Kunming, China, 2008: 3679 – 3683.

[16] Matt B. Computer security art and science [M]. Addison Wesley, 2002: 84 – 140,502 – 520.