ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.7, July 2009, pp.1953−1966
doi: 10.3724/SP.J.1001.2009.03295
© by *Institute of Software, the Chinese Academy of Sciences*. All rights reserved.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

安全策略模型聚合性评估方法*

蔡嘉勇[1,2,5+], 卿斯汉[1,4], 刘 伟[1,5]

[1](中国科学院 软件研究所 基础软件国家工程研究中心,北京 100190)

[2](中国科学院 软件研究所 信息安全技术工程研究中心,北京 100190)

[3](北京大学 软件与微电子学院,北京 102600)

[4](北京中科安胜信息技术有限公司,北京 100086)

[5](中国科学院 研究生院,北京 100049)

Groupability in Security Policy Models

CAI Jia-Yong[1,2,5+], QING Si-Han[1,4], LIU Wei[1,5]

[1](National Engineering Research Center for Fundamental Software, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

[2](Engineering Research Center for Information Security Technology, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

[3](School of Software and Microelectronics, Peking University, Beijing 102600, China)

[4](Beijing ZhongkeAnsheng Corporation of Information Technology, Beijing 100086, China)

[5](Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)

+ Corresponding author: E-mail: jiayong02@ios.cn

Cai JY, Qing SH, Liu W. Groupability in security policy models. *Journal of Software*, 2009,20(7):1953−1966.
http://www.jos.org.cn/1000-9825/3295.htm

Abstract: Dynamic policy supporting and authorization granularity are two key issues in access control. Present researches only compared the expressiveness of policies, but never considered the policy's structure and the granularity of authorization, which makes it difficult to support the dynamic policy and satisfy the least privilege requirement. As this paper points out that Lampson's access matrix is the most fine-grained access control model, the other security policies need to group access matrix according to their different application requirements. By defining a descriptive framework of Groupability Basing on Security Labels (GroSeLa), generic security policies can be mapped into Lampson's access matrix. GroSeLa framework consists of a set of fundamental components and an extension. The fundamental components give all policy's structure for grouping matrix, and the extension reveals all necessary administrative requirements for supporting dynamic policy completely. Based on GroSeLa, this paper proposes five grouping dimensions for evaluating security policies, including grouping factors, dynamic factors, policy scale, authorization granularity and separation of duty supporting. The paper also compares four classic

* Supported by the National Natural Science Foundation of China under Grant No.60573042 (国家自然科学基金); the National Basic Research Program of China under Grant No.G1999035802 (国家重点基础研究发展计划(973)); the Beijing Natural Science Foundation of China under Grant No.4052016 (北京市自然科学基金)

Received 2007-07-18; Revised 2007-12-28; Accepted 2008-02-27

© 中国科学院软件研究所 http://www.jos.org.cn

security policies, namely ACL (access control list), BLP (Bell LaPadula), DTE (domain and type enforcement) and RBAC (role-based access control). To the best of these knowledge, it is studied that the difference on expressiveness, usability and authorization granularity of different security policies are from the aspect of grouping access matrix.

Key words: group; security label; access matrix; dynamic policy; least privilege

摘 要: 动态策略支持与授权粒度是访问控制的关键问题.现有的研究只关注安全策略的描述能力,却忽略了对策略结构与授权粒度的分析,从而无法全面满足动态策略支持与最小授权要求.指出 Lampson 访问矩阵模型是对最细粒度访问控制的抽象,普通安全策略则根据应用安全需求对 Lampson 访问矩阵进行聚合.基于安全标签的聚合性描述框架(a descriptive framework of groupability basing on security labels,简称 GroSeLa)可将普通安全策略映射为 Lampson 访问矩阵,该框架分为基本组件与扩展两部分:前者分析用于实现矩阵聚合的安全策略结构;后者则指出实现全面动态策略支持必须支持的 7 类管理性需求.在此基础上,提出 5 项聚合性指标:聚合因子、动态因子、策略规模、授权粒度与职责隔离支持.对 4 类经典安全策略 ACL,BLP,DTE 与 RBAC 的评估,是从矩阵聚合的角度分析不同的安全策略在表达性、可用性与授权粒度上的差异.

关键词: 聚合;安全标签;访问矩阵;动态策略;最小授权

中图法分类号: TP309 **文献标识码**: A

最小授权(least privilege)是指“系统中每个程序和用户应使用足以完成任务的最小权限集合进行操作”[1],以降低黑客攻击或滥用权限造成的风险.最小授权是访问控制设计的基本原则,Schneider 认为,正确实施该原则的关键在于解释授权的度量性问题[2].已有研究主要通过改进权限的表达[3]、访问控制的实现机制[4,5]等方面来减少进程运行时拥有的权限规模,但由于缺乏相应的授权参考粒度,无法对这类改进方案所能达到的最小授权程度进行准确评估.本文认为,虽然 Lampson 访问矩阵模型[6]需要维护的策略规模过于庞大,无法在实际系统中实现,但该模型对系统每个访问事件的抽象,足以说明访问控制的最小授权粒度,而普通策略只不过是根据实际应用需求,采用安全标签对访问矩阵进行聚合,以缩小策略规模并提高策略可伸缩性的一种特殊形式.

动态策略支持(dynamic policy supporting)是近年访问控制研究的最新热点.传统的安全策略要求系统运行时,不能对策略作任何修改,属于静态策略范畴.这种方式虽然严格保证策略实施与系统安全目标的一致性,但在实际系统中并不可用[7],如军事信息系统依赖 BLP 策略实现多等级保护需求,禁止向低级别的信息流传达,但当上级命令需要向下级传达时,还是需要对信息予以降级.动态策略支持不仅要支持多策略,而且要支持这些策略的动态调节,即根据环境变化改变安全策略的能力,包括策略修改、选择、激活与禁止等[8].可见安全策略动态化的目的是增强系统可用性.研究者已针对现有策略的动态化改进[9,10]、访问控制实施体系[11]以及安全性分析[12]等多方面进行了探讨,但并未指明策略的所有动态调节对象,因而无法确保系统实现动态策略支持的完备性.本文分析了普通安全策略用于聚合访问矩阵的策略结构,进而总结了实现全面动态策略支持必须满足的 7 类管理性需求,弥补了这一方面的缺陷.

本文第 1 节根据安全策略对系统访问事件授权的共性,指出普通安全策略使用安全标签对 Lampson 访问矩阵聚合,以降低策略规模并提高策略伸缩性的本质,进而定义基于安全标签的聚合性描述框架(a descriptive framework of groupability basing on security labels,简称 GroSeLa).该框架可以建立普通安全策略到 Lampson 访问矩阵的映射,框架基本组件定义用于矩阵聚合的策略结构,框架扩展则根据策略结构指出全面动态策略支持需要实现的 7 类管理性需求.基于 GroSeLa 框架,第 2 节定义 5 项聚合性指标并给出相应的度量方法,包括聚合因子、动态因子、策略规模、授权粒度与职责隔离.应用本文设计的评估指标与方法,第 3 节综合比较 ACL(access control list),BLP(Bell LaPadula),DTE(domain and type enforcement),RBAC(role-based access control)这 4 类经典安全策略在表达性、可用性与最小授权风险上的差异.最后总结全文.

© 中国科学院软件研究所 http://www.jos.org.cn

1 框架定义

安全策略是一套用于规范组织如何管理、保护以及分发信息的法律、法规与行为准则[13],访问控制依据用户制定的安全策略对系统发生的访问事件进行控制,防止未经授权的用户访问系统资源.1971 年,Lampson 首次对信息系统的访问控制问题进行抽象,建立了访问矩阵模型[6],模型中主体(subject)代表发出访问请求的主动实体;客体(object)代表受保护的系统资源;访问矩阵(access matrix)以所有主体为行,所有客体为列,列举每个主体对每个客体的所有访问权限,以表示管理员对访问控制的具体要求,即安全策略.因此,其他安全策略也采用这种⟨主体,客体,权限⟩三元组的抽象方式,描述系统的访问请求.如 1974 年 Bell 与 LaPadula[14]建立 BLP 安全策略模型时,将系统所有访问请求定义为该三元组的集合 $b\subseteq S\times O\times A$.

Lampson 访问矩阵准确阐述了管理员对系统所有访问事件的控制要求,被公认为粒度最细的授权方式.但该策略粒度过细,难以直接表达实际安全需求;其次,系统中主、客体数量庞大造成矩阵规模不受控,导致访问控制效率低下.尤其是在安全目标发生改变时,不能提供动态策略支持[8],因此,该策略并不适合在实际系统中实现.面对不同的信息保护要求,研究者纷纷提出了各种安全策略,如 BLP[14]策略、Clark-Wilson[15]策略、DTE[16]策略以及 RBAC[17]策略,不仅是为更好地描述系统的安全需求,也是为了弥补 Lampson 模型的不可用性缺陷.访问矩阵作为二维数组,降低其规模最直接的方法就是压缩必须描述的行或者列的数量.由于主、客体在系统的唯一标识不具有安全含义,故此使用额外的符号表示主、客体的安全属性成为必然,我们称其为安全标签.使用安全标签代替主、客体标识,对访问矩阵进行聚合是普通安全策略对 Lampson 访问矩阵改进的基本方式.安全标签表示具有相同安全属性的一类主、客体,使管理员可以较为直观的方式表达系统的访问控制要求;同时,以安全标签作为矩阵坐标,不仅降低了矩阵规模,还取消了主、客体与矩阵的直接关系,提高了安全策略的伸缩性,避免了为每个访问事件制定安全策略的繁琐过程.

基于上述思想,本节定义了基于安全标签的聚合性描述框架——GroSeLa.与 Lampson 访问矩阵相比,框架的基本组件所增加的对象类、安全标签与规则部分提高了策略的描述能力,并给出聚合矩阵后安全策略的内部结构变化.框架的扩展则分析了策略所有可能的动态调节对象,并指出全面动态策略支持需要满足的 7 类管理性需求.本节还将给出框架对普通安全策略的实例化过程,建立普通安全策略到 Lampson 访问矩阵的映射.

1.1 基本组件

由上述分析可见,所有安全策略都要对系统访问事件授权,其本质都是 Lampson 访问矩阵模型,不同的是采用的矩阵聚合方式不同.GroSeLa 框架基本组件包含以下 5 个部分,用于分解普通安全策略对矩阵的聚合能力:

对象(object)**:是系统对其内部数据项的抽象表示,数据类型为 OBJ.对象拥有访问控制需要保护的信息,在访问事件中属于被动实体,通常称其为客体,系统的所有客体构成客体集 $O:\wp OBJ$.为代表用户对系统资源进行访问,一类特殊客体成为系统唯一能够主动发出访问请求的实体,称为主体,所有主体构成主体集 $S\subseteq O$.

访问接口(access interface):是指主体访问客体时使用的系统接口.在一些文献中称为“访问模式”或“访问属性”,常见的文件客体访问接口有 *read*,*write*,*execute* 等.集合 K 表示系统提供的所有访问接口.

对象类(object class):是指具有相同访问接口集合的一类对象.针对不同客体的存在形式,系统提供的访问接口集往往不同,如文件客体的访问接口集为{*read*,*write*,*execute*},而磁盘设备的访问接口集合则为{*mount*,*umount*,*ioctl*}.为表示对象到相应接口集合的映射,需要划分对象类.我们规定集合 C 为系统定义的所有对象类,对象类到访问接口集合的映射***为 $CK\subseteq C\times 2^K$.任何对象均对应唯一的对象类,映射关系为 $OC\subseteq O\times C$.主体类对象由于具有相同的访问接口集,通常被定义为单独的主体类(*subject_class*),其他非主体类客体 $O\backslash S$ 既可统一定义为客体类 *object_class*,也可根据策略需要进一步细分,如普通文件类、目录类、设备类等.

安全标签(security label):是指系统赋予对象的额外符号,用于描述其所具有的安全属性.在安全策略中,一

** 术语“object”在安全文献中大多被译为客体,本文认为只有在访问事件中成为被动实体才可以称为客体,因此区分译作对象.

*** SELinux 中将对象类到权限的映射关系称为访问向量(access vector).

© 中国科学院软件研究所 http://www.jos.org.cn

个安全标签可以表示具有相同安全属性的一类对象,系统的访问控制依据对象的安全属性对访问事件做出授权裁决.如 SELinux 策略描述语言[18]就明确指出,对访问事件的决策依赖于标签所携带的安全信息,并将访问事件中对象拥有的安全标签称为安全上下文(security context).以有限的安全标签代替无限的系统对象作为矩阵坐标,极大地降低了访问矩阵的规模.安全标签作为安全策略对一类对象的抽象表示,避免了策略与单个对象的直接关联,提高了安全策略的可伸缩性.根据具体应用需求,安全标签既可以是简单数字或字符,也可以使用复杂的数据结构,如集合、向量等.集合 SL 为安全策略给出的所有安全标签.作为对象的安全内涵,安全标签,甚至安全标签的内部分量之间可能需要表达某些特殊关系 LR.比如在第 3.2 节的 BLP 安全策略中,敏感级别间的 $\prec$ 就是一种具有明确上/下界的偏序关系.

授权规则(authorization rule):虽然安全标签可以在一定程度上缩小访问矩阵的坐标范围,但矩阵条目所列举的访问权限仍需耗费大量空间,也影响系统访问控制的实施效率.授权规则是对聚合后访问矩阵条目所列举权限分布规律的描述,主要利用 LR 构造授权时主、客体对象必须满足的安全性质、约束以及条件,系统全部授权规则构成集合 R.利用安全标签关系构造的逻辑算式称为安全标签算式:$Calculus(L,LR)$.

作为最细粒度的授权方式,Lampson 访问矩阵需对〈主体,客体,访问模式〉三元组表达的每个访问事件授权.由于没有定义主、客体对象的安全属性,因此只能直接以访问矩阵的形式表示管理员制定的安全策略.而实际操作系统中的对象数通常可达到千万的量级,需要实现的访问矩阵规模极为庞大.这不仅需要耗费大量宝贵的内核存储空间以维护安全策略,而且授权时对安全策略的查询也非常低效,极大地影响了系统的运行效率.从管理员的角度,针对每对主、客体制定访问控制要求是不可能完成的任务,尤其是在新对象加入系统时,对访问矩阵更新的代价极大,根本不能满足动态策略支持要求.

普通安全策略扩大授权粒度,针对具有相同访问控制要求的一类访问事件进行授权.为了表示不同的访问控制,安全策略给出了不同的标签集,如在第 3 节的分析中,DAC 采用用户标签、BLP 采用敏感级标签、DTE 采用型与域、RBAC 则采用角色.安全标签取消了访问矩阵与系统对象的直接关联,避免了每次新增对象都要更新访问矩阵的要求,满足了动态环境中安全策略的可伸缩性问题,也降低了安全策略制定与维护的难度.GroSeLa 框架新增的对象类组件用于区分使用不同访问接口集的系统对象,从而将访问控制的范围扩大至系统所有对象,而不仅限于传统的文件对象;新增的安全标签用于表达对象的安全属性以及聚合后的访问矩阵坐标;授权规则组件则进一步描述访问矩阵条目的分布规律,从而全面分析了普通安全策略对 Lampson 访问矩阵聚合的可能形式.

Lampson 访问矩阵授权的基本单位是〈主体,客体,访问模式〉三元组表达的访问事件.普通安全策略则对具有相同访问控制要求的访问事件授权,因此,访问模式分量相同.相同安全属性的对象用〈对象类,安全标签〉加以描述,那么〈主体类****,主体安全标签,客体类,客体安全标签,访问模式〉则描述了具有相同访问控制要求的一类访问事件.Zanin 等人[19]使用“基于型的权限(permission with respect to type)”来表示 SELinux 策略的授权,定义为 $authorize(d,prt(t,(p,c)))$,其中 $prt(t,(p,c))$表示对型 t 的 c 类客体拥有访问权限 p.由此也说明了 GroSeLa 框架 5 个基本组件划分的必要性.基于上述分析,我们给出如下关于权限的定义:

权限(permission):是指安全策略对某个访问事件的授权状态,即与安全策略对应的 Lampson 访问矩阵中每个〈主体,客体,访问模式〉三元组的策略配置状态.

表示性权限(expressive permission):是指安全策略对某类访问事件的授权状态,即对〈主体类,主体安全标签,客体类,客体安全标签,访问模式〉五元组表示的具有相同访问控制要求的一类访问事件的授权配置.定义为 $ep:caculus(SL_{subj})\rightarrow((caculus(SL_{obj})\times C_{obj})\rightarrow 2^{CK(obj)})$.

那么聚合后的访问矩阵则为表示性权限的有序集 $\overline{AM}\subseteq 2^{ep}$,一些安全策略的授权规则由于完全描述了矩阵条目的分布规律,因此不必给出聚合后的访问矩阵,即 $\overline{AM}_{BLP}=\varnothing$.从表示性权限的定义可以看出,普通安全

**** 系统所有主体通常被归为统一的主体类,因此在描述具有相同控制要求的访问事件时,主体的对象类别可以省略.

© 中国科学院软件研究所 http://www.jos.org.cn

策略的授权粒度要高于 Lampson 访问矩阵.若管理员制定安全策略时未加分析,则很可能造成对访问事件的不当授权与过分授权.因此,非常有必要避免对这部分权限的错误设置,防止由于权限滥用与误用而产生的系统安全风险,这正是最小授权原则的基本要求.

根据 GroSeLa 框架基本组件定义,普通安全策略基于安全标签的聚合方式可分为 3 类:主体聚合、客体聚合和规则聚合.前两类分别从矩阵横、纵两个坐标方向对具有相同访问属性的对象聚合,压缩访问矩阵必须描述的行或列的数量.授权规则是对矩阵条目分布规律的总结,属于高级聚合方式.$UCON_{ABC}$ 模型族[20]指出,安全策略可以从授权(authorization)、义务(obligation)、条件(condition)3 个角度及其不同阶段分别定制规则,并对系统访问控制施加影响.这里,我们将授权规则简单划分为两类:直接决定访问事件是否允许的规则称为访问规则;而用于表示合法访问事件必须满足的性质,以间接影响授权的规则称为约束规则,比如 RBAC 策略可以定制系统约束规则,以满足灵活多变的应用层职责隔离要求.一种安全策略根据具体需求,可采用 3 类聚合形式之一,或者组合构成其对 Lampson 访问矩阵的聚合能力.

图 1 给出了聚合前后安全策略结构的变化情况,在 Lampson 模型中,访问矩阵完全代表管理员制定的安全策略状态.普通安全策略由于聚合矩阵的需要,策略状态结构发生变化,由以下 7 部分构成:安全标签集 *SL*、标签关系 *LR*、主体的安全标签赋予状态 LA_{subj}、客体安全标签赋予状态 LA_{obj}、聚合后的访问矩阵状态 *AM*、授权规则状态 *AR*、约束规则状态 *CR*,即 $policy=(SL,LR,LA_{subj},LA_{obj},AM,AR,CR)$.授权时系统访问控制需要检索安全策略状态,其查询效率为对每部分结构的检索时间之和.可以看到,聚合后 *policy* 每部分结构的规模都属可控范围,降低了由于查询大规模访问矩阵对系统性能的损耗.

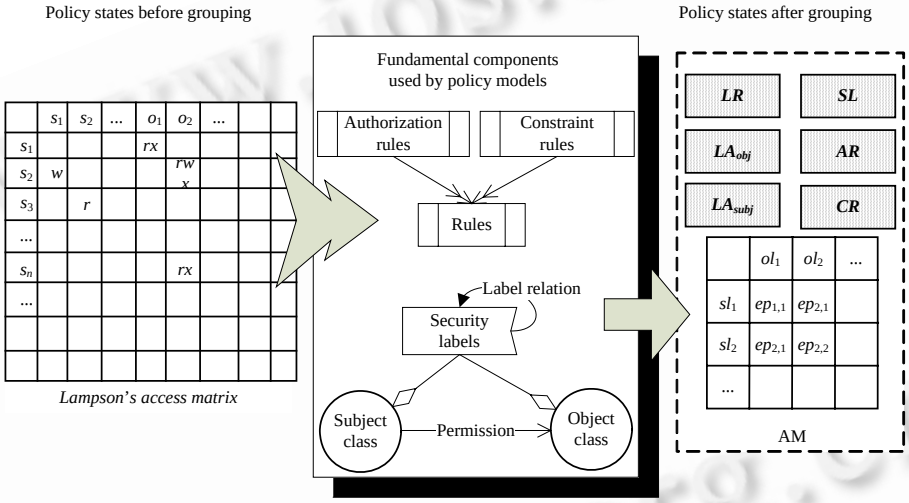

Fig.1 Transformation of access control policy structure by matrix permission grouping

图 1 聚合后访问控制策略结构变化示意图

GroSeLa 框架从聚合的角度对普通安全策略进行描述与分解,这是以往单一策略模型所不具有的描述能力.利用 GroSeLa 框架,可以实现普通安全策略到 Lampson 访问矩阵的映射.为满足应用层对访问控制时间性的要求,一些安全策略引入了时序逻辑,即向 Lampson 访问矩阵增加时间坐标,我们同样可以向 GroSeLa 框架增加相应的时间聚合组件,但这样做增加了策略分析的难度,本文将研究重点限于所有非时序安全策略范畴.

1.2 扩 展

第 1.1 节定义的 GroSeLa 基本组件,指出了因聚合访问矩阵产生的安全策略结构变化.传统的安全策略属于静态策略,要求系统运行时安全策略状态不可调整.而实际上,管理员经常需要根据时间、安全环境、系统安全需求的变化,对已实施的安全策略进行调整,使其符合新的安全目标.比如某项目采用 CVS(concurrent versions system)软件执行文档管理与版本控制,根据软件工程要求,软件项目的开发周期可分为需求定义、可行性分析、

© 中国科学院软件研究所 http://www.jos.org.cn

总体描述、系统设计、编码、调试与测试、验收与运行、维护升级 8 个阶段,静态安全策略要求管理员在 CVS 系统启动前制定整个开发周期的安全策略,而在 CVS 执行版本控制期间,安全策略不允许调整,为此,管理员不得不将整个开发周期需要的所有权限授予 CVS 用户,造成即使项目进入验收与维护阶段,小组成员仍具有修改需求定义文档的权限,而这样的权限本应在需求定义阶段完成之后就立即被禁止.因此,策略的动态支持成为近年来访问控制的研究热点.

动态策略支持即在系统运行期间允许对安全策略的状态进行有条件的修改,从而使系统在严格安全性与灵活可用性之间进行调节[8].对安全策略状态的修改,将影响系统对访问事件的授权结果,因此,我们将对安全策略状态的修改操作称为管理性操作,为实现动态策略支持而必须增加的管理性操作则称为管理性需求.已有的动态策略支持方面的研究可分为如下 3 类:

(1) 为现有安全策略增加管理模型或直接将普通安全策略升级为管理模型.如 Solworth 等人提出的基于安全属性的管理控制模型——SPBAC[7];动态 DTE 模型[21]则将管理性操作映射为普通域到型的访问操作;RBAC 模型的改进包括 ARBAC 模型[22]、SARBAC 模型[23]以及 UARBAC 模型[10]等;

(2) 管理模型的安全性分析.最早由 Harrison 等人[24]提出,主要考察执行管理操作后,系统安全性的可判定性问题.如 Sasturkar 等人[12]对 ARBAC 模型的安全性是否可判定进行了大量的分析.

(3) 动态策略支持的系统实施框架.如 SELinux 针对 Flask 体系的安全策略层次、功能组件模块化、支持权限撤销机制以及利用缓存解决性能开销[8]等方面探讨策略动态调整的问题.

然而上述研究并没有分析策略动态化的所有可调节对象,也没有给出全面动态策略支持需要实现的所有管理性需求,因而无法判断所给策略对动态策略支持的完备程度.Lampson 模型的策略状态完全由访问矩阵表示,实现动态化只需支持对访问矩阵条目的调整即可.由于对访问矩阵进行聚合,普通安全策略的状态结构分为 7 部分,需要提供的管理性需求必然比 Lampson 访问矩阵要复杂,这是在性能与复杂性之间均衡的结果.根据图 1,我们认为,用于实现聚合的 7 个组件就是策略动态化的所有可调节对象,因此,全面动态策略支持需要实现以下结构调整需求:

(1) 主体标签调整:主体代表用户执行访问操作,通常根据所代表用户的安全属性,由访问控制系统自动赋予,不允许普通用户修改,以保证系统安全的一致性.然而系统也需要允许新用户登录系统,因此主体安全标签存在调整的必要,如 *login* 进程需要根据登录用户赋予 *bash* 进程相应的安全标签,因此系统需要提供修改 LA_{subj} 的管理性操作.

(2) 客体标签调整:客体的安全标签通常在系统运行时就已确定,或者根据上下文自动生成.但新对象的产生、安全环境的变化都可能导致客体的保护需求发生变化,如在军事信息系统中,上级命令虽由高敏感级用户签发而具有高机密性,但为了使命令传达到下级部门,必须对客体降级,因此需要提供修改 LA_{obj} 的管理性操作.

(3) 标签集调整:安全策略的标签集 *SL* 可以是确定的,如 BLP 模型[14],也可以由管理员自行定义,如 ACL 策略、DTE 策略[16]以及 RBAC 策略[17]等,因此安全策略需要提供调整 *SL* 的管理性操作.

(4) 标签关系调整:规则使用安全标签算子 *Calculus*(*SL*,*LR*)表达授权条件,安全标签关系的调整也将间接影响授权结果.某些安全策略中 *LR* 是确定的,如 BLP 策略,但在大多数安全策略中,*LR* 是允许修改的.如在 RBAC 策略中,角色的层次关系可以根据实际需要进行调整,新的授权应当符合新的角色层次关系定义.因此需要提供调整标签关系 *LR* 的管理性操作.

(5) 授权规则调整:同样,在部分安全策略中,规则集 *R* 可根据实际需要作调整,如 RBAC 策略在不同时期对职责隔离的要求是不同的,因此需要提供授权规则的调整操作,以便管理员随时制定符合应用要求的安全策略.

(6) 访问矩阵调整:对于聚合后的访问矩阵 $\overline{AM}$,若不为空,则应提供管理员的调整接口.

(7) 特权接口:特定安全策略的表达能力是确定的,然而系统的访问控制要求是多变的,管理员往往需要获得比安全策略更加灵活的额外授权方式表达,以满足系统可用性要求.比如允许特权主体对任何客体写访问等.由于这类操作接口不属于安全策略的描述能力范畴,因此将其统称为特权.传统操作系统使用 *root* 身份表示特权,POSIX 标准则提出了更加灵活的权能机制[3].特权接口通常用于快速检查与撤销错误的安全策略设置,以便

© 中国科学院软件研究所 http://www.jos.org.cn

实现简单的安全策略设置、故障分析及检查等功能.

GroSeLa 框架基本组件对普通安全策略的分解,给出了安全策略的内部结构以及策略动态化的调节对象.以上我们总结了全面动态策略支持需要实现的所有管理性需求,由此可以判断普通安全策略及其动态化改进的完备程度,同时也指明了管理模型安全型分析的研究对象.与普通访问操作不同,管理性操作将导致安全策略状态发生变化,直接影响系统的安全性,对管理性操作控制尤为重要.除了防止未经授权的主体使用管理性操作以外,完整性是管理性操作控制的主要目标,比如应使用审计追究管理员的非法行为,应用职责隔离原则限制单个管理员的能力等.此外,调整后系统策略安全性的可判定性问题[12]、安全目标一致性分析[25]、动态策略有效性分析[22,23]、安全策略的冲突检测[26]、隐式授权问题[27]也是动态策略支持的研究热点.

1.3 实例化

本文指出 Lampson 访问矩阵是最细粒度的授权方式,并定义 GroSeLa 描述性框架.与单一的安全策略相比,GroSeLa 框架具有策略中立性(policy neutral)与多策略(multi-policies supporting)的描述能力,管理员可以通过给出特殊的框架组件,描述特定的安全需求,并建立普通安全策略到 Lampson 访问矩阵的映射.我们将普通安全策略中的系统实体与聚合后的策略状态称为 GroSeLa 框架实例,安全策略的实例化过程如下:

安全策略实例: $PI=\langle b,S,O,K,C,OC,CK,SL,LR,LA_{obj},LA_{subj},AR,CR,\overline{AM}\rangle$ 是安全策略实例,当且仅当

- $b\subseteq S\times O\times K$ 代表当前已发生的所有访问事件集合;
- O 为系统客体集,$S\subset O$ 为主体集,K 为安全策略保护的所有访问接口,C 是安全策略给出的对象类集,OC 与 CK 分别表示对象到对象类,以及对象类到访问接口集合的映射关系;
- $policy=(SL,LR,LA_{obj},LA_{subj},AR,CR,\overline{AM})$ 表示安全策略聚合访问矩阵后的策略状态结构.

2 聚合性指标

通过 GroSeLa 框架,普通安全策略模型可被统一映射为 Lampson 访问矩阵,在此基础上可以对不同安全策略模型的结构、系统实现难度以及模型所能实现的安全功能进行比较.我们给出以下 5 项指标及其具体度量方法.由于 5 项指标基于 GroSeLa 框架给出,因此称为聚合性指标:

- 聚合因子(grouping factor):是指 $policy=(SL,LR,LA_{subj},LA_{obj},AM,AR,CR)$ 中被普通安全策略用于聚合访问矩阵的元素,即该策略的内部结构.普通安全策略由于面对的安全需求不同,使用的聚合形式也不同,因此策略结构各异.Lampson 访问矩阵模型作为最细粒度的授权方式,我们规定其聚合因子为完全表示其安全策略的访问矩阵 AM.从理论上看,更多的聚合因子,表示安全策略具有更加灵活的聚合能力;针对同一访问控制要求,管理员可以提供多种策略制定方式,因此,应用层安全需求的表达能力也更强.
- 动态因子(dynamic factor):GroSeLa 框架扩展指出的 7 类管理性需求实际对应着对安全策略状态结构 7 个组件的调整,因此,聚合因子实际指出了安全策略实现全面动态策略支持需要实现的所有管理性需求类型.我们用动态因子表示指定特定安全策略允许调整的聚合因子,通过对比动态因子与聚合因子,安全策略模型的设计者可以清楚地了解普通安全策略对动态策略支持的完备程度.
- 策略规模(policy scale):降低策略的规模,提高访问控制的执行效率是普通安全策略模型对访问矩阵进行聚合改造的根本原因.Lampson 模型的安全策略为访问矩阵,需要针对每个访问事件进行授权,其策略规模为 $o(|S|\times|O|)$.普通安全策略需要依次查询策略所有结构,才能获得对某一访问事件的授权结果,因此,其策略规模应为聚合因子的规模之和.策略规模反映了系统为实现某一安全策略而需付出的空间与时间代价.
- 授权粒度(authorization granularity):Lampson 访问矩阵模型指出授权的最细粒度是〈主体,客体,访问模式〉表示的访问事件,普通安全策略采用不同的方式聚合矩阵权限,降低系统维护难度,造成安全策略间授权粒度的差异.在一些研究[6,16]中使用“域”抽象主体拥有的权限范围并不直观,不利于管理员比较不同安全策略间授权的差别.由于本文给出统一的授权粒度参考系——Lampson 访问矩阵,我们可以用该指标度量普通安全策略矩阵元素对比于 Lampson 访问矩阵元素的规模.如第 3.1 节中 ACL 策略的矩阵元素,表示单个客体对一类主体的

© 中国科学院软件研究所 http://www.jos.org.cn

授权,而对于每类客体而言,主体只有属主、组与其他的区别,因此其授权粒度为 $o(|S|)$.我们定义授权粒度指标的目的是反映 1 次矩阵元素授权修改对系统安全性的影响范围,以便安全管理员度量一次错误授权或者权限滥用可能造成的安全威胁.

• 职责隔离(separation of duty supporting):随着越来越多的资料被保存于信息系统,防止单个实体滥用权限造成的系统安全风险,支持职责隔离安全功能成为大多数安全操作系统的基本要求.Ferraiolo[28]将系统实现的职责隔离分为静态职责隔离(static separation of duty,简称 SSoD)、动态职责隔离(dynamic separation of duty,简称 DSoD)与操作性职责隔离(operational separation of duty,简称 OSoD)3 类.它们的差别在于实施方式,SSoD 可采用访问规则或者约束规则来实现,DSoD 与 OSoD 则只能通过约束规则来实现.安全策略的授权组件实现不同,所能实现的职责隔离类型也不同.该指标的度量进一步分为 SsoD,DsoD,OsoD 这 3 项,安全策略若能够支持某类职责隔离,则度量结果为“√”,否则为“×”.

在已有研究中,$UCON_{ABC}$[20]对访问控制问题的授权决策因素进行分解,以分析安全策略的表达性是否足以表达安全需求,但该模型没有给出策略的评估指标.DTOS(distributed trusted operating system)[29]定义了包含输入、敏感性、撤销性与传递性 4 项指标的安全策略格.Tolone 等人[30]则提出了复杂度、可理解性、易用性、适应性、协同环境支持的协同环境安全策略特性指标,但这些指标既没有定义统一的参考系,也没有给出指标的度量方法,安全策略的度量结果依赖于分析者的主观判断.Bertino 等人[31]的 ACMS(access control model schema)逻辑框架可将安全策略转化为该框架的实例,并定义模型内部结构与访问的相等/包含性、模型外部的可达与一致性指标,但该指标只反映了安全策略在表达性上的差异,而且使用 C-Datalog 语言的描述增加了对普通安全策略实例化的复杂性.

本文主要对安全策略聚合 Lampson 访问矩阵的方式进行比较,明确指出最细粒度的授权方式为 Lampson 访问矩阵,我们定义了 GroSeLa 框架,并将普通安全策略映射为 GroSeLa 框架实例,这不仅分解了安全策略的状态结构,而且建立了用于策略间比较的统一参考系.通过分析安全策略的状态结构,我们对给出的 5 项聚合性指标都给出了明确的度量方法,从而确保了策略间比较结果的客观性.聚合性指标为管理员选择系统实施的安全策略提供多方面的指导.如聚合因子反映普通安全策略授权的灵活性,并给出其实现全面动态策略支持需要提供的全部管理性操作类型;通过对比动态因子与聚合因子的差距,可以考察普通安全策略对动态策略支持的程度,反映了安全策略实施的可用性;策略规模考察安全策略实施的空间与时间复杂性;授权粒度与职责隔离指标则分别向管理员提供授权风险以及安全功能方面的参考.

3 安全策略比较

本节对 4 类经典的安全策略进行分析与比较.首先,普通安全策略模型被实例化为 GroSeLa 框架的安全策略实例,然后采用第 2 节中所定义的聚合性指标对其进行度量.表 1 最后列举了 4 类安全策略与 Lampson 访问矩阵模型的度量结果,从中可以比较不同安全策略的优劣.

Table 1 Groupability assessment and comparison of Lampson, ACL, BLP, DTE and RBAC

表 1 Lampson,ACL,BLP,DTE 和 RBAC 安全策略模型聚合性评估结果与比较

Dimension \ Policy		Lampson	ACL	BLP	DTE	RBAC
Grouping factors		$AM_{Lampson}$	(U,G), $LR(\in)$, LA_{obj}, LA_{subj}, AR_{ACL}, AM_{ACL}	SL_{BLP}, $LR(\prec)$, SL_{subj}, SL_{obj}, AR_{BLP}	$D\cup T$, LA_{subj}, LA_{obj}, AR_{DTE}, $DDT\cup DTT$	$Roles$, $LR(\lhd)$, SL_{subj}, AR_{RBAC}, CR_{RBAC}
Dynamic factors		None	(U, G), $LR(\in)$, LA_{obj}, LA_{subj}, AM_{ACL}	In BLP: None In DBLP: SL_{subj}, SL_{obj}	In DTE: None In dynamic DTE: AM_{DTE}	In RBAC: None In UARBAC: $Roles$, $LR(\lhd)$, SL_{subj}, CR_{RBAC}
Policy scale		$o(\|S\|\times\|O\|)$	$o(\|O\|+\|S\|)$	$o(\|O\|+\|S\|)$	$[o(\|S\|+\|O\|),o(\|S\|\times\|O\|)]$	$[o(\|S\|),o(\|Roles\|^{\|Roles\|})]$
Authorization granularity		1	$o(\|S\|)$	$o(\|S\|\times\|O\|)$	$[1,o(\|S\|\times\|O\|)]$	Coarser than the granularity of bottom policy
Separation of duty	SSoD	√	√	√	√	√
	DSoD	×	×	×	×	√
	OSoD	×	×	×	×	√

© 中国科学院软件研究所 http://www.jos.org.cn

3.1 ACL

自主访问控制(discretionary access control,简称 DAC)是指由客体属主决定客体访问属性的控制形式[13],是访问控制的基本形式之一.访问控制列表策略是实现该安全需求的主要方式,如常见的 UNIX 9bit 文件权限就是最简单的 ACL 安全策略模型,研究者出于提高该策略授权粒度的需要,对该模型进行了扩展.本节以 9bit 文件权限作为 ACL 策略的代表.

ACL 定义的系统对象类集为 C_{ACL}=(*subject_class*,*object_class*),分别代表主体与客体.客体类到访问接口的映射为 CK_{ACL}(*object_class*)=(*read*,*append*,*write*,*execute*),主体类到访问接口的映射为 CK_{ACL}(*subject_class*)=(*kill*).ACL 安全标签为有序二元组(*User*,*Group*),表示属主与组.*U*,*G* 为系统用户集与组集,用户与组之间存在隶属关系∈,集合 *LR*(∈)给出所有用户与组的隶属关系.所有系统对象都被赋予安全标签,ACL 利用安全标签算子对访问矩阵进行聚合,矩阵横坐标为系统所有对象,纵坐标则聚合为 3 项:$User(s)=User(o), User(s)\in Group(o), User(s)\neq User(o)\wedge User(s)\notin Group(o)$.聚合后的矩阵条目为 $ep_{ACL}:\{u,g,o\}\to(O\to 2^{CP_{ACL}(C_O)})$.授权规则中,ACL 只定义了如下访问规则:

AR_{ACL}(*subject_class*,*object_class*):如果 $\forall s:S, o:O\setminus S, perm:CK_{ACL}(object\_class)\cdot\langle s,o,perm\rangle\in b$,那么下面 3 个条件必有 1 条成立:

(1) $User(s)=User(o)$且 $perm\in ep_u(o)$,或者

(2) $User(s)\in Group(o)$且 $perm\in ep_g(o)$,或者

(3) $perm\in ep_o(o)$.

AR_{ACL}(*subject_class*,*subject_class*):如果 $\forall s_1,s_2:S, perm:CK_{ACL}(subject\_class)\cdot\langle s_1,s_2,perm\rangle\in b$,那么下面两个条件必须有 1 条成立:

(1) $User(s_1)=User(s_2)$,或者

(2) $User(s_2)\in Group(s_1)$.

对应的安全策略实例为 $PI_{ACL}=\langle b,S,O,K,C_{ACL},OC,CK_{ACL},(U,G),LR(\in),LA_{obj},LA_{subj},AR_{ACL},\varnothing,AM_{ACL}\rangle$.ACL 只对访问矩阵纵坐标聚合,但由于与矩阵列只有 3 项,规模小而且固定,因此,UNIX 通过将访问矩阵列分散存储于客体,从而避免访问控制实现规模不受控的访问矩阵的必要.

对策略状态结构分析可见,ACL 的聚合因子包括(*U*,*G*),*LR*(∈),LA_{obj},LA_{subj},AR_{ACL},AM_{ACL}.实现全面的动态策略支持需要提供除授权规则调整、特权以外的全部管理性操作,UNIX 系统用普通文件客体/etc/passwd,/etc/group 维护(*U*,*G*)与 *LR*(∈),对标签及其关系的调整可转化为主体对文件客体的访问;*setuid*(),*setgid*(),*chown*()等系统调用实现了主、客体安全标签的调整;而 *chmod*()实现了访问矩阵的调整.因此,除 AR_{ACL} 以外的聚合因子皆为动态因子,可见其动态策略支持程度较高,这也是所有操作系统都实现 ACL 策略的原因之一.由策略状态结构可见,AM_{ACL} 与 LA_{obj} 的规模均为 $o(|O|)$,LA_{subj} 为 $o(|S|)$,(*U*,*G*),*LR*(∈)规模视管理员要求而定,通常规模远小于其他组件,因此,ACL 的策略规模为 $o(|O|+|S|)$.由于只对矩阵纵坐标进行聚合,系统主体被分为 3 类,AM_{ACL} 的矩阵条目相当于聚合了主体规模的 AM_{Lampson} 条目,即授权粒度为 $o(|S|)$.策略由于缺少约束规则部分,因此只能支持 SSoD,而不支持 DSoD 与 OSoD.

3.2 BLP

强制访问控制(mandatory access control,简称 MAC)是指由安全管理员决定客体访问属性的控制形式[13].1974 年,Bell 与 LaPadula 针对军事应用安全系统面临的机密性威胁,提出了 BLP 模型[14],这是第一个经严格证明满足军事多级别安全需求的安全策略.

BLP 采用与 ACL 策略相同的对象类集、对象类到权限的映射关系.安全标签为有序二元组 L_{BLP}=⟨*sensitive*, *category*⟩,其中 *sensitive* 表示敏感级,使用整数表示,存在高低关系<;*category* 表示范畴,使用部门集合表示,存在包含关系⊆.在此基础上,定义统一的安全标签支配关系 ≺.BLP 策略将访问规则定义为简单安全属性与*-属性,

© 中国科学院软件研究所 http://www.jos.org.cn

以确保高安全等级的信息不会向低级别客体流动:

AR_{BLP}: $\forall s:S,o:O,perm:CK_{BLP}(object\_class)\cdot\langle s,o,perm\rangle\in b\triangle ABC$,那么下面的条件必须满足:

(1) 简单安全属性: $perm\in(read,write)\Rightarrow l_{BLP}(o)\prec l_{BLP}(s)$;

(2) *-属性: $perm=append\Rightarrow l_{BLP}(s)\prec l_{BLP}(o)$;或者 $perm=write\Rightarrow l_{BLP}(s)=l_{BLP}(o)$;或者 $perm=read\Rightarrow l_{BLP}(o)\prec l_{BLP}(s)$.

对应的安全策略实例为 $PI_{BLP}=\langle b,S,O,K,C_{BLP},OC,CK_{BLP},SL_{BLP},LR(\prec),LA_{obj},LA_{subj},AR_{BLP},\varnothing,\varnothing\rangle$.BLP 策略同时对访问矩阵横、纵坐标进行聚合,由于聚合后访问矩阵条目具有特殊的规律,因此访问规则的定义取消了系统访问控制实现访问矩阵的必要.

聚合因子包括 SL_{BLP}, $LR(\prec)$, LA_{subj}, LA_{obj}, AR_{BLP},根据多级别安全需求,安全标签及其关系必须固定,而授权规则只包含访问规则,因此,实现全面动态策略支持只需提供对主、客体安全标签的调整接口即可.经典 BLP 策略不允许修改主、客体安全标签,为弥补其动态性不足的缺陷,研究者提出了一些改进策略,如 DBLP(dynamic Bell LaPadula)[9].然而,由于大部分聚合因子无法调整,BLP 策略族的动态策略支持还是远逊于 ACL 策略,这也是 BLP 策略虽然提出多年,但仍只在少数安全操作系统中实现的原因.SL_{BLP}, $LR(\prec)$ 与 AR_{BLP} 的规模是一个常量,LA_{subj} 与 LA_{obj} 的规模分别为 $o(|S|)$与 $o(|O|)$,因此,策略状态的整体规模为 $o(|O|+|S|)$.AM_{BLP} 矩阵条目聚合了子矩阵规模的 AM_{Lampson} 矩阵条目,由于安全标签数量固定,因此,授权粒度为 $o(|S|\times|O|)$.该策略同样只支持 SSoD.

3.3 DTE

Badger 等人提出的 DTE 策略[16]是近年来访问控制实现研究的热点,是 SELinux 实施的三大策略之一[18].该策略要求更细的对象类划分,主体仍属于主体类(*subject_class*),但存在大量的一般客体类,如(*file_class*,*device_class*, *socket_class*,…),每类对象映射的接口集不同,这里不再一一列举,统一定义为 CK_{DTE}.主、客体使用的安全标签类型不同,主体使用"域",而客体使用"型",标签集为($D\cup T$),安全标签彼此独立,没有任何关系,因此 $LR=\varnothing$.访问矩阵定义的授权策略由域定义表(domain definition table,简称 DDT)与域转换表(domain transition table,简称 DTT)来表示,即 $AM_{\text{DTE}}=DDT\cup DTT$.DTE 定义了灵活的访问规则,由安全管理员解释安全标签的含义:

AR_{DTE}(*subject_class*,*object_class*) $\forall s:S,o:O,perm:K\setminus\{transition\}\cdot\langle s,o,perm\rangle\in b$,则必须满足:

$perm\in DDT(s,o)$.

AR_{DTE}(*subject_class*, *subject_class*): $\forall s_1,s_2:S\cdot\langle s_1,s_2,transition\rangle\in b$,则必须满足:

$transition\in DTT(s,o)$.

对应的策略实例为 $PI_{DTE}=\langle b,S,O,K,C_{DTE},OC,CK_{DTE},D\cup T,\varnothing,LA_{obj},LA_{subj},AR_{DTE},\varnothing,DDT\cup DTT\rangle$,DTE 策略从横、纵两方面对矩阵聚合,而聚合时采用的安全标签集完全由管理员根据需要定义,因此,矩阵的规模可大可小.

聚合因子包括 $D\cup T$, LA_{subj}, LA_{obj}, AR_{DTE}, AM_{DTE},实现完全动态策略支持应提供除特权与标签关系调整外的所有管理性接口.DTE 策略本身并不提供任何管理性操作接口,其改进策略——动态 DTE 策略[21]通过将 DDT 与 DTT 映射为普通文件客体实现对 AM_{DTE} 的调整,但不支持对其他聚合因子的修改,因此,该策略的动态化还有待改进.考虑极端情况下,管理员给出的安全标签集中只有一个域和型,那么 DTE 策略聚合后的访问矩阵大小则为 1;反之,若管理员对每个主、客体都赋予唯一的安全标签,那么聚合后的访问矩阵则与 Lampson 访问矩阵等同规模.因此 DTE 策略的状态规模在[$o(|S|+|O|)$, $o(|S|\times|O|)$]区间浮动.AM_{DTE} 的矩阵条目同样可聚合子矩阵规模的 Lampson 矩阵条目,但由于管理员定义安全标签集具有很大的伸缩性,因此,授权粒度在[1, $o(|S|\times|O|)$]区间浮动.DTE 策略缺乏约束规则,因此无法支持 DSoD 与 OSoD.

3.4 RBAC

RBAC 具有职责隔离、策略中立、数据抽象三大特性[17].虽然该策略是目前访问控制的热门研究对象,但在具体实现中,角色并不作为系统访问控制的直接决策因素,而是映射为底层安全策略权限,实现对系统资源的

© 中国科学院软件研究所 http://www.jos.org.cn

保护.这种保持应用逻辑与访问控制分离的方式,使得管理员可以采用角色及其层次关系抽象企业内部的管理结构,更适于表达应用层安全需求.比如在 SELinux 的 3 层策略模型架构中,RBAC 是底层 DTE 策略的顶层抽象[18].由于角色标签只赋予主体,因此角色是进一步对底层策略中的主体标签进行聚合,不失一般性,可将底层策略设计的主体标签视作确定角色 r^D,所有确定角色构成角色集 R^D.

RBAC 策略的系统对象只有主体(*subject_class*)与客体(*object_class*)之分.策略定义的安全标签为角色,即 *SL=Roles*.一个主体可同时拥有多个角色,即 *s* 的标签状态 *Roles*(*s*)是一个角色集合,客体没有安全标签.角色间可能存在继承关系 $\lhd$, $LR(\lhd)$ 为管理员对所有角色关系的定义.由于 RBAC 必须通过底层策略实现访问控制,因此对象类到接口的映射不必定义,即 $CK_{RBAC}=\varnothing$.RBAC 策略的授权规则包含访问规则与约束规则,其中约束规则可以由管理员灵活定义,访问规则则如下所示:

AR_{RBAC}: $\forall s:S,o:O,perm:K\cdot\langle s,o,perm\rangle\in b$,则下面的条件必须满足:

$\exists r_1..r_n:R\cdot(r_1\prec r_2\wedge r_2\prec r_3\wedge...\wedge r_n\prec r)\wedge r\in Roles(s)\wedge r_1\in R^D\wedge perm\in ep(r_1)$

对应的策略实例为 $PI_{RBAC}=\langle b,S,O,K,C_{RBAC},OC,\varnothing,Roles,LR(\lhd),\varnothing,LA_{subj},AR_{RBAC},CR_{RBAC},\varnothing\rangle$.需要指出的是,RBAC 虽然无须定义访问矩阵,但其实现访问控制必须获得底层策略的支持,因此总体上,系统必须实现访问矩阵取决于底层策略的设计.

RBAC 的聚合因子包括 *Roles*, $LR(\lhd)$,LA_{subj},AR_{RBAC},CR_{RBAC},这意味着若实现全面动态策略支持,RBAC 需要提供角色集、主体安全标签、角色标签关系以及约束规则的调整接口,已有的改进策略,如 ARBAC,SARBAC, UARBAC 已经提供对上述接口的支持.*Roles* 由管理员针对实际应用要求定义,因此, $LR(\lhd)$ 的规模在[0, $o(|Roles|^{|Roles|})$]区间浮动,LA_{subj} 的规模为 $o(|S|)$,约束规则的规模很小,可以忽略不计,因此,RBAC 的策略规模区间为[$o(|S|)$,$o(|Roles|^{|Roles|})$],可以看到,角色关系的复杂性直接影响系统实施的复杂程度.由于是对底层策略的主体安全标签的进一步聚合,因此 RBAC 的授权粒度要高于底层策略.而约束规则的提供,说明 RBAC 策略可以实现所有类型的职责隔离.

3.5 综合比较

表 1 详细列举了 4 类经典安全策略与 Lampson 访问矩阵的聚合性指标度量结果.可以看到,ACL 策略使用的聚合因子较多,策略的灵活性较高;而且 ACL 策略的动态因子相较于聚合因子的比例最高,说明 ACL 的动态策略支持程度最高.ACL 与 BLP 的策略规模不相上下,而 DTE 与 RBAC 则根据管理员实际指定的标签集规模在某个区间浮动,因此,授权粒度由管理员可控,具有很好的伸缩性,比较容易满足最小授权原则.但需要注意的是,在过分追求细粒度授权的情况下,这两种策略占用的系统空间与访问控制计算时间可能导致系统无法有效运行.由于多数策略授权规则不包括约束规则,因此只有 RBAC 可以实现全部 3 类职责隔离.

EARTH 是一款基于 FreeBSD 6.2-Stable 开发的安全操作系统,内核实施 Extended-ACL,MLS(multi-level security)以及 SEBSD 三大安全策略模块.其中,Extended-ACL 实现了扩展粒度的 ACL 策略,客体属主可制定任何用户/组粒度的自主访问控制策略;MLS 满足军事多级别安全需求,实施部分动态化的 BLP 策略;SEBSD 是 SELinux 在 BSD MAC 框架的移植,通过 IBAC+RBAC+DTE 三大安全策略的层次化实施结构,实现灵活粒度的强制访问控制.管理员可以通过设置/boot/loader.conf 的系统启动模块选项,选择是否在启动时加载并实施某个安全策略模块.

表 2 显示了 lmbench[32]针对 EARTH 实施不同安全策略模块的性能测试结果(处理器为 Intel Pentium(R) 4 CPU 2.93GHZ,内存 512MB DDR,单位 ms).实验结果显示:Extended-ACL 与 MLS 模块加载后,系统的性能相差无几,而 SEBSD(security enhanced BSD)虽然利用访问向量缓冲机制[11](access vector cache,简称 AVC)解决大规模策略查询效率低下的问题,但性能仍逊于前两者.实验结果不仅说明了表 1 对策略规模指标估计的正确性,也说明了本文所提出的聚合性指标的客观性.

© 中国科学院软件研究所 http://www.jos.org.cn

Table 2 Groupability assessment of Lampson, ACL, BLP, DTE and RBAC

表 2 NFS-ARK 实施不同安全策略模块的性能比较

Secure module in NFS-ARK	Policy enforced	Null call	Stat()	Open()+close()	Fork()
None loaded	None	0.47	4.33	6.18	262.81
Extended-ACL only	ACL	0.48	8.08	8.21	263.04
MLS only	BLP	0.49	7.92	7.91	267.11
SEBSD only	IBAC+RBAC+DTE	0.63	19.83	30.23	296.56
All modules loaded	All	0.47	24.1	37.43	347.13

4 总 结

根据应用需求,安全策略针对系统访问事件给出不同的授权结果.Lampson 访问矩阵模型精确描述了访问控制对系统每个访问事件的授权情况,因此被公认为粒度最细的安全策略.本文通过定义 GroSeLa 框架,建立了普通安全策略到 Lampson 访问矩阵的映射,不仅给出不同安全策略比较的参考系,还指出了普通安全策略为提高系统可用性而对 Lampson 访问矩阵聚合的内部状态结构.

本文将 GroSeLa 框架的定义分为基本组件与扩展两部分,前者给出普通安全策略可用于聚合矩阵的组件,后者则总结了安全策略实现全面动态化必须提供的 7 类管理性操作接口,从而为判断普通安全策略动态化完备程度与普通安全策略实现全面动态化改进指明了方向.基于 GroSeLa 对普通安全策略状态内部结构的分解,本文定义的 5 项安全策略聚合性指标,从聚合矩阵的角度,准确而客观地揭示了不同安全策略在聚合灵活性、动态策略支持程度、系统实施复杂性、授权粒度以及安全功能上的差异.

管理员选择系统访问控制实施的安全策略,不仅需要考虑策略对应用安全需求的描述能力,还必须分析策略在系统实施的可行性.已有安全策略的比较性研究主要探讨策略表达性是否足以表达实际应用安全需求,而本文给出的聚合性指标不仅进一步分析了不同安全策略的描述能力(聚合因子指标与职责隔离指标),而且还研究了安全策略在系统可用性(动态因子指标、策略规模指标)与最小授权风险(授权粒度指标)方面的差异,从而为管理员选择系统实施的安全策略提供更全面的指导.

动态策略支持与最小授权是访问控制研究的两大关键.本文给出了安全策略实现全面动态化支持的必要条件,但尚未给出实现动态策略支持的基本方法,研究者需要针对现有安全策略动态化不足的缺陷进一步改进,提高安全策略实施的可用性.同样,本文也客观地给出不同安全策略的授权粒度,但在设计新的策略描述语言、访问控制机制等方面还有待研究者进一步加以研究.

References:

[1] Saltzer J H, Schroeder M D. The protection of information in computer systems. Proc. of the IEEE, 1975,63(9):1278−1308.

[2] Schneider F B. Least privilege and more. IEEE Security & Privacy, 2003,1(5):55−59.

[3] Portable Applications Standards Committee of IEEE Computer Society, Standards Project. Draft Standard for Information Technology-Portable Operating System Interface (POSIX), PSSG Draft 17. New York: IEEE, Inc., 1997.

[4] Ji QG, Qing SH, He YP. A new formal model for privilege control with supporting POSIX capability mechanism. Science in China (Series F), 2005,48(1):46−66.

[5] Liang B. Research on trusted process mechanism and related problems [Ph.D. Thesis]. Beijing: Institute of Software, the Chinese Academy of Sciences, 2004 (in Chinese with English abstract).

[6] Lampson BW. Protection. Operating Systems Review, 1974,8(1):18−24.

[7] Solworth JA, Sloan RH. Security property based administrative controls. In: Samarati P, Ryan P, Gollmann D, Molva R, eds. Proc. of the 9th European Symp. on Research in Computer Security. LNCS 3193, Sophia Antipolis: Springer-Verlag, 2004. 244−259.

[8] Wu YJ. Research on key technologies of dynamic policy support in secure operating system [Ph.D. Thesis]. Beijing: Institute of Software, the Chinese Academy of Sciences, 2006 (in Chinese with English abstract).

[9] Ji QG, Qing SH, He YP. An improved dynamically modified confidentiality policies models. Journal of Software, 2004,15(10): 1547−1557 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/15/1547.htm

© 中国科学院软件研究所 http://www.jos.org.cn

[10] Li N, Mao Z. Administration in role-based access control. In: Deng R, Samarati P, eds. Proc. of the 2nd ACM Symp. on Information, Computer and Communications Security, Conf. on Computer and Communications Security, SESSION: Access Control. Singapore: ACM, 2007. 127−138.

[11] Spencer R, Smalley S, Loscocco P, Hibler M, Andersen D, Lepreau J. The flask security architecture: System support for diverse security policies. In: Proc. of the 8th USENIX Security Symp. Washington: USENIX Association, 1999. 123−139. http://unix.hensa.ac.uk/sites/ftp.wiretapped.net/pub/security/operating-systems/selinux/papers/the-flask-security-architecture.pdf

[12] Sasturkar A, Yang P, Stoller SD, Ramakrishnan CR. Policy analysis for administrative role based access control. In: Proc. of the 19th IEEE Workshop on Computer Security Foundations. Washington: IEEE Computer Society, 2006. 124−138. http://www.cs.binghamton.edu/~pyang/csfw-2006-TR.pdf

[13] Department of Defense. Trusted Computer System Evaluation Criteria. National Computer Security Center, 1985.

[14] Bell DE, LaPadula LJ. Secure computer systems: A mathematical model. Technical Report, ESD-TR-73-278, Vol. 2, ESD/AFSC, 1973.

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

[16] Lee B, Sterne DF, Sherman DL, Walker KM, Haghighat SA. A domain and type enforcement UNIX prototype. In: Proc. of the 5th USENIX UNIX Security Symp. Salt Lake City: USENIX Association, 1996. 127−140. https://www.usenix.org/publications/library/proceedings/security95/full_pape rs/badger.pdf

[17] Sandhu RS, Coyne EJ, Feinstein HL, Younan CE. Role-Based access control models. Computer, 1996,29(2):38−47.

[18] Smalley S. Configuring the SELinux policy. NAI Technical Report, 02-007, NAI Labs., 2002.

[19] Zanin G, Mancini LV. Towards a formal model for security policies specification and validation in the SELinux system. In: Proc. of the 9th ACM Symp. on Access Control Models and Technologies. Yorktown Heights, New York: ACM, 2004. 136−145. http://infosecurity.org.cn/content/secbase/sec_poli_spec_valid.pdf

[20] Park J, Sandhu R. The $UCON_{ABC}$ usage control model. ACM Trans. on Information and Systems Security, 2004,7(1):128−174.

[21] Tidswell J, Potter J. An approach to dynamic domain and type enforcement. In: Varadharajan V, Pieprzyk J, Mu Y, eds. Proc. of the 2nd Australasian Conf. on Information Security and Privacy. LNCS 1270, London: Springer-Verlag, 1997. 26−37.

[22] Sandhu RS, Bahamidipati V, Coyne E, Ganta S, Youman C. The ARBAC97 model for role-based administration of roles: Preliminary description and outline. In: Proc. of the 2nd ACM Workshop on Role-Based Access Control. Fairfax: ACM, 1997. 41−50. http://eprints.kfupm.edu.sa/68894/1/68894.pdf

[23] Crampton J, Loizou G. Administrative scope and role hierarchy operations. In: Proc. of the 7th ACM Symp. on Access Control Models and Technologies. Monterey: ACM, 2002. 145−154. http://www.isg.rhul.ac.uk/~jason/Pubs/sacmat02.pdf

[24] Harrison MA, Ruzzo WL, Ullman JD. Protection in operating systems. Communications of the ACM, 1976,19(8):461−471.

[25] Li N, Tripunitara MV. On safety in discretionary access control. In: Proc. of the 2005 IEEE Symp. on Security and Privacy. Washing: IEEE Computer Society, 2005. 96−109. https://www.cerias.purdue.edu/tools_and_resources/bibtex_archive/archive/2005-20.pdf

[26] Jaeger T, Zhang XL. Policy management using access control spaces. ACM Trans. on Information and System Security, 2003,6(3):327−364.

[27] Eβmayr W, Kastner F, Pernul G, Preishaber S, Tjoa AM. Authorization and access control in IRO-DB. In: Su SYW, ed. Proc. of the 12th Int'l Conf. on Data Engineering. Washington: IEEE Computer Society, 1996. 40−47.

[28] Ferraiolo DF, Cugini J, Kuhn DR. Role-Based access control (RBAC): Features and motivations. In: Proc. of the Computer Security Applications Conf. New Orleans: IEEE Computer Society Press, 1995. 241−248. http://www.csrc.nist.gov/groups/SNS/rbac/documents/ferraiolo-cugini-kuhn-95.pdf

[29] Secure Computing Corporation. DTOS generalized security policy specification. CDRL Sequence No.A019, Secure Computing Corporation, 1997.

[30] Tolone W, Ahn GJ, Pai T, Hong SP. Access control in collaborative systems. ACM Computing Surveys, 2005,37(1):29−41.

© 中国科学院软件研究所 http://www.jos.org.cn

[31] Bertino E, Catania B, Ferrari E, Perlasca P. A logical framework for reasoning about access control models. ACM Trans. on Information and System Security, 2003,6(1):71−127.

[32] McVoy L, Staelin C. lmbench: Portable tools for performance analysis. In: Proc. of the Annual Technical Conf. on USENIX 1996 Annual Technical Conf. San Diego: USENIX Association, 1996. 279−284. http://www.usenix.org/publications/library/proceedings/sd96/full_papers/mcvo y.ps

附中文参考文献:

[5] 梁彬.可信进程机制及相关问题研究[博士学位论文].北京:中国科学院软件研究所,2004.

[8] 武延军.安全操作系统动态策略支持的关键技术研究[博士学位论文].北京:中国科学院软件研究所,2006.

[9] 季庆光,卿斯汉,贺也平.一个改进的可动态调节的机密性策略模型.软件学报,2004,15(10):1547−1557. http://www.jos.org.cn/1000-9825/15/1547.htm

蔡嘉勇(1978－),男,福建莆田人,博士生,主要研究领域为信息系统安全理论和技术.

卿斯汉(1939－),男,研究员,博士生导师,CCF 高级会员,主要研究领域为信息系统安全理论和技术.

刘伟(1979－),男,博士生,主要研究领域为操作系统安全,网络安全.

© 中国科学院软件研究所 http://www.jos.org.cn