ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.8, August 2009, pp.2298−2306
doi: 10.3724/SP.J.1001.2009.03300

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

行为驱动的基于角色的信任管理*

李 澜[1,2+], 范 磊[1,2], 回 红[1,2]

[1](上海交通大学 信息安全工程学院,上海 200240)

[2](上海信息安全综合管理技术研究重点实验室,上海 200240)

Behavior-Driven Role-Based Trust Management

LI Lan[1,2+], FAN Lei[1,2], HUI Hong[1,2]

[1](School of Information Security Engineering, Shanghai Jiaotong University, Shanghai 200240, China)

[2](Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Shanghai 200240, China)

+ Corresponding author: E-mail: lanli@sjtu.edu.cn

Li L, Fan L, Hui H. Behavior-Driven role-based trust management. *Journal of Software*, 2009,20(8): 2298−2306. http://www.jos.org.cn/1000-9825/3300.htm

Abstract: RT_B, the language describing behavior-driven trust management, is given. Variables are introduced into roles to maintain the cumulate behavior status of the users. Behavior-Driven credentials modify users' assigned local roles in a trust domain according to the behaviors enforced by the users. Combined credentials improve the efficiency of trust determination. Trust policy update credentials allow trust domains to update trust policies automatically when the system statuses are changed. Implementation framework of behavior-driven trust management is described. Several optimization mechanisms of realization are discussed.

Key words: trust management; behavior-driven; role-based; trust credential; trust delegation

摘 要: 给出了描述行为驱动的信任管理语言 RT_B.将变量引入到角色中可以记录用户的累积行为状态;行为驱动的信任规则根据用户已发生的行为调整其在本信任域中被分配的角色;组合规则提高了信任判定的效率;信任策略更新规则允许信任域在系统状态发生变化时自动调整信任策略.描述了行为驱动的信任管理的实现框架,并讨论了优化实现的几种机制.

关键词: 信任管理;行为驱动;基于角色;信任规则;信任委托

中图法分类号: TP393 **文献标识码**: A

在规模较大的分布式环境中,如何快捷而有效地对用户进行授权及访问控制管理是较难解决的问题.多用户和机构协调工作的系统中很可能没有一个全局控制中心,或者同时有多个控制中心.而资源的数据控制者与访问者之间的关系是动态变化和异构的,因此,使用传统的授权与访问控制方法不能处理这种复杂的关系.大规模分布式系统用户数量众多,数据资源种类丰富,控制中心必须能够根据安全策略快速地处理用户的访问请求,

* Supported by the National Natural Science Foundation of China under Grant No.60803145 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant Nos.2008AA01Z409, 2007AA01Z473 (国家高技术研究发展计划(863))

Received 2007-10-17; Accepted 2008-02-21

同时也要实现灵活与适当的安全策略.

传统的访问控制模型,包括自主访问控制(discretionary access control,简称 DAC)和强制访问控制(mandatory access control,简称 MAC),都是根据用户的标识来进行授权与访问控制,这在分布式环境中并不是非常合适,因为在这样的环境中,控制者并不能方便地了解访问请求者的详细信息.基于角色的访问控制(role-based access control,简称 RBAC)[1]已被证明在权限管理方面是成功的.尽管 RBAC 可通过分布式管理的方式来处理大量角色,但是由于多控制中心的用户、角色以及权限之间的分配关系难以管理,纯粹的 RBAC 也不能解决大规模分布式环境中的授权与访问控制问题.

分布式环境下的权限管理有很多不确定性,而信任(trust)是处理不确定性时的一个较好的选择.信任关系发生在信任者(trustor)与被信任者(trustee)之间,信任者提供资源和服务,而被信任者需要访问资源或者使用服务.被信任者可能是单个主体,也可能是某些主体的集合,比如角色.RT 是一种基于角色的信任管理语言族[2],它把角色的概念引入到信任管理中.属性证书用来表示证书所有者在证书发布域中被分配的角色,而信任策略则根据角色来判断用户的被信任度,并得到相应的权限.RT 也为现实情况中复杂的权限委托(delegation)提供了基本的操作.信任规则(credentials)在很多信任管理语言中用来处理权限委托,比如 KeyNote[3]和 SPKI/SDSI,每条信任规则指明了权限从某个角色委托到另一个角色.RT_0 是 RT 语言族中最核心的语言,包括了 RT 中最关键的部分.下面这个例子描述了 RT_0 的主要特性.

例 1:假设一个在线电子书店与某大学合作,允许大学图书馆网站的注册用户免费在线阅读图书.而只有该大学的老师和学生才能申请成为图书馆网站的注册用户.信任规则如图 1 所示.

(1) $OStore.Free \leftarrow ULib.Member$
(2) $ULib.Member \leftarrow U.Teacher \cap ULib.Applied$
(3) $ULib.Member \leftarrow U.Student \cap ULib.Applied$
(4) $U.Teacher \leftarrow T$
(5) $U.Student \leftarrow S$

Fig.1 Example of an online electronic book store's credentials

图 1 在线电子书店的信任规则举例

RT_0 语言的最基本的语句形如 $A.r \leftarrow B$,说明 B 是信任域 A 定义的角色 r 中的成员.而权限委托则是形如 $A.r \leftarrow B.s$ 的语句,说明信任域 B 中角色 s 的所有成员也是信任域 A 中角色 r 的成员.RT_0 语言不仅描述了信任域本地的信任管理策略,而且也可以描述不同信任域之间的信任委托关系,为分布式环境下多控制中心的信任管理提供了一种有效的方法.但是,RT_0 主要是描述了静态信任策略,虽然文献[4]提出了与上下文相关的动态语句,增加了包括时间、权责分离等方面的限制,但是仍然没有考虑到被信任者的累积行为对信任关系的影响.由于大规模分布式环境中的信任关系是动态发展的,信任域可能会因为用户执行了某些操作之后改变对用户的信任状态,或者因为某个外部角色的大量用户的行为改变对该外部角色的信任委托状态,这样就可以对用户和角色实施更加灵活的授权及访问控制,也更符合大规模分布式环境的实际情况.我们可以通过下面这个例子来讨论访问控制的动态性.

例 2:假设在线电子书店与某大学合作举行促销活动,允许大学图书馆网站的注册用户免费下载 3 本电子图书,一旦该大学师生购买电子图书数量达到 5 000 本,则免费下载数将从 3 本提高到 5 本.信任规则如图 2 所示.

与例 1 相比,例 2 在几个方面作了扩展.首先,给角色增加了变量,比如 $OStore.Free(3)$.其次,例 2 增加了(i)、(ii)两条规则,它们将根据被分配了该角色的用户的行为来改变用户角色中的变量值,甚至改变用户被分配的角色.最后,例 2 增加(a)这条规则,它是对规则本身的控制,一旦该大学师生在本书店的购买总数超过 5 000 本,则用左边这条规则替代右边这条规则.由此可以看出,本文对 RT 语言族扩展的主要贡献包括 3 个方面:(1) 对角色进行了扩展,控制端可以为自己的角色增加变量,其值由信任规则或委托规则来决定.(2) 扩展后的语言允许信任域的控制中心根据用户的行为来改变其被信任的状态,添加了行为驱动的角色转换规则.(3) 扩展后的语言可以描述更加宏观的信任状态的变化,允许根据大量用户的行为统计来改变原先对这类用户(即角色)的信任规则

或者委托规则,也就是制定规则之上的规则.

(i) $OStore.noFree \xleftarrow{Download\ \ one\ \ book} OStore.Free(1)$

(ii) $OStore.Free(i-1) \xleftarrow{Download\ \ one\ \ book} OStore.Free(i)$ while $i>1$

(1) $OStore.Free(3) \leftarrow ULib.Member$

(2) $ULib.Member \leftarrow U.Teacher \cap ULib.Applied$

(3) $ULib.Member \leftarrow U.Student \cap ULib.Applied$

(4) $U.Teacher \leftarrow T$

(5) $U.Student \leftarrow S$

(a) $(OStore.Free(5) \leftarrow ULib.Member) \xleftarrow{ULib.totalbuy>5000} (OStore.Free(3) \leftarrow ULib.Member)$

Fig.2 Example of an online electronic book store's credentials driven by behaviors

图 2 在线电子书店行为驱动的信任规则举例

1 相关的工作

由于传统的访问控制模型和机制不能满足很多分布式环境的需求,因此,基于信任的安全成为很多研究者关注的热点.Blaze 等人[5]首先把信任管理单独提出来作为通信系统安全的重要组成部分,通过简单的语言来表明信任的行为和信任关系,并描述了一个信任管理原型——PolicyMaker.文献[6]描述了为虚拟组织(virtual organization,简称 VO)设计的基于信任的访问控制机制,支持全局信任关系.文献[7]则使用交叉证书来描述客户与信任域的关系,使用用户-角色证书来标识用户,使用角色继承证书来向其他信任域传递角色的继承关系,从而实现跨域的大规模访问控制.文献[8]提出了 P2P 文件共享系统中基于信任的访问控制框架.该框架通过计算直接与间接的信任值、直接与间接的贡献值来判断用户的请求是否应该被满足.在 Ad-hoc 这样的动态协作环境中,文献[9]给出了以节点为中心的信任管理系统,可以提供根据对方的行为来设置多个访问层次的能力.然而,这些研究成果若能发挥作用,则都必须先对本地信任策略进行有效的管理,以实现安全且灵活的授权策略.

为了减少分布式环境下的信任管理的复杂性,信任可以与基于角色的管理结合起来.文献[10]在 RBAC 的基础上扩展了信任级别,提出了一个基于信任的访问控制模型 TrustRBAC.文献[11]则在基于 XML 的 RBAC 模型(X-RBAC)上扩展了上下文相关的访问控制,通过给用户分配 TM 证书的方法在模型中引入了信任域.RT 语言族[2]将角色引入到信任管理中,极在地减小了分布式环境信任管理的复杂度,同时支持多信任域之间的信任委托.文献[4]在 RT 中引入了动态的信任规则,从而可以实现只有在特定的上下文环境下信任规则才会发生作用,这些环境可能包括时间、地点以及其他与上下文相关的状态,但是不包括主体实施的行为.文献[12,13]则描述了如何控制信任委托的深度,包括使用整数值或者信任度的阈值.该文献虽然可以帮助 RT 语言提供比较细粒度的权限委托,但是也没有对运行过程中的信任状态变化进行描述.

2 行为驱动的基于角色的信任管理语言RT_B

我们在 RT 语言中引入了动态的元素——行为(behavior),使得 RT 语言的规则从对静态信任状态的描述扩展成为对动态信任变化的描述.信任者可以指明被信任者的行为对其被信任程度的影响,既可能增加也可能减少信任度.而且可以根据某个角色的大量用户的行为统计来改变对角色或信任域的信任,也就是改变 RT 语言的信任规则或者信任委托规则.在定义行为驱动的信任规则之前,我们先对角色进行扩展.

2.1 带变量的角色

为了对用户的行为引起的信任变化结果进行描述,我们允许在角色中创建变量.这些变量与参数化角色的参数有所区别,角色的参数会根据某个用户的属性值来设置具体的值,而我们引入的角色变量则不然,变量的值要么是在某条规则中就被确定,要么是由某条规则来改变,与用户提交的属性值无关.

定义 1. 一个带变量的角色具有如下形式: $Dom.Role(V_1,V_2,...,V_n)$,其中 *Dom* 是角色所在的信任域,*Role* 是角色的名称,V_1 到 V_n 是 *Role* 的 *n* 个独立的变量.

每个用户获得角色后,变量的值都已经被明确地分配好了.角色的变量值表明了担当该角色的用户当前的行为累积状态.相同名称的角色可能拥有不同的变量值,表示行为累计的不同状态,但对于权限控制模块来说仍然是同一个角色,信任域在制定本地授权及访问控制策略时,只需要针对角色名称,而不需要考虑具体的变量值.因此,尽管带变量的角色增加了对用户行为累积状态的描述,但是没有增加授权及访问控制策略的复杂度.由于角色的变量值记录的是用户的行为累积状态,变量值不会因为用户提交的属性值而改变.一般来说,信任域在制定规则时,将会有一条规则来对用户担当角色的变量赋值,然后会有一条或多条规则根据用户已经实施过的行为来修订变量的值.比如图 2 的规则(1)中,*ULib.Member* 的成员同时也是 *OStore.Free*(3) 的成员,*OStore.Free* 角色的变量值在规则中就被赋值为 3,与用户的其他属性无关.又如规则(i),如果 *OStoreFree*(*i*) 的成员下载了一本书,则角色的变量值将会被减去 1,用于记录用户的行为累积状态.角色中的变量一般都是有值域的,这个值域中的不同值只是说明了角色的行为累计,当累计到一定程度时,可能会达到值域的极端值,此时某些行为将引起角色的质变.这也说明了信任是灵活的,具有较好的伸缩范围.

带变量的角色一般由信任域自身来控制,因为变量的变化实际上是本地信任策略的一种表示.但是,由于授权与访问控制策略都是实施在角色级别上的,而不会考虑角色中变量的具体值,因此,信任域在制定本地信任委托策略时,我们并不需要关心外部角色的变量值.这表明用户在某个信任域的行为不需要让其他信任域了解,从而保持各信任域之间的独立性.

2.2 行为驱动的信任规则

对角色进行扩展后,我们可以在RT语言中引入用户行为驱动的信任规则,根据用户发生的行为来改变用户担当角色中变量的值,甚至改变用户担当的角色.

定义 2. 一条基于行为的信任规则是具有如下形式的规则: $R_1 \xleftarrow{B} R_2 \quad C$.此规则说明在条件 C 下,如果实施过行为 B,本来应该被分配角色 R_2 的用户现在只能被分配角色 R_1.

(1) R_2 是规则使用前的角色,称为规则的输入角色.

(2) R_1 是规则使用后的角色,称为规则的输出角色.R_1 和 R_2 可以不同,也可以是变量值不同的同一角色.

(3) B 是驱动规则发生作用的行为.行为是由信任域定义的,应该是一个操作的序列.这个序列是一个整体,类似于数据库管理系统中的事务,要么都执行,要么都不执行.应用系统层应该保证行为的原子性和系统的一致性,具体如何实现则需要根据每个应用系统自身的业务流程来决定.

(4) C 是规则适用的条件.条件是可选项,即在某些规则中可以不设置条件.然而一旦设置了条件,就说明只有满足该条件时才可以使用这条行为驱动的规则.定义 3 对条件进行了详细的定义.

用户如何得到某信任域角色是由该信任域的信任策略,也就是信任规则或者信任委托规则来决定的.这些规则有可能会使用其他信任域定义的角色,信任域需要根据用户提交的属性证书来确定其被分配的角色.在分布式环境下,各个信任域都独立地控制着本域角色分配的条件和限制,因此在制定授权策略时,为了保证自己的安全策略能够得到正确的实施,每个信任域都只会把权限分配给自己定义的角色.其他域的角色将会通过一系列信任规则或者信任委托规则映射为某个本地角色.经过映射后,用户获得了本地角色的权限.这时我们就可以根据用户的行为来继续判定用户的被信任度.为此每个信任域都需要维护用户在本地已经实施的行为.一般来说,信任域不必考虑用户在其他信任域发生的行为,而且也不需要维护那些发生在本地但是对信任状态没有影响的行为,所以,每个信任域只需记录用户的一小部分行为.由于需要针对每个用户记录具体行为,这将使得存储和信任判定的代价变高,在第 3 节我们将会讨论较优化的实现方法.

因为信任域只根据本地的行为来制定信任状态的变换策略,所以行为驱动的信任规则都将作用于本地的角色,这些规则的输入角色与输出角色必须是在本信任域中定义过的带变量的角色.其中,角色变量的值可以是明确的,也可以通过某种方式指定一个作用域,比如在后面的条件中指定.定义 3 给出了基于行为的信任规则中条件的描述.

定义 3. 行为驱动的信任规则的条件具有如下形式:When θ.其中,$\theta = (\theta_0) \mid \theta \vee \theta \mid \theta \wedge \theta \mid \neg\theta, \theta_0$ 是不包含与、

或、非操作的简单条件.

从θ的递归定义可以看出,信任规则中的条件是由多个简单条件通过与、或、非操作联合起来的复合条件.简单条件可能与某个环境相关,比如时间、访问地址等,也可能与输入角色中的变量值相关,用于限制规则中变量的取值范围,比如例 2 中的规则(i),只有当输入角色 *OStore.Free* 的变量值大于 1 时,才能应用这条规则.通过在条件中增加对变量值的限制,我们可以间接地定义变量的值域.从例 2 可以看出,通过在规则中增加约束,使得我们可以更灵活地定义行为对用户被信任度的影响,实现更细粒度的信任策略.

2.3 组合规则

在行为驱动的规则中,每条规则所涉及的行为都是一个与实际业务紧密相关的操作序列.系统将把用户已经发生的行为记录下来,作为调整其被信任度的依据.系统在判断用户的被信任程度时,将会根据已经实施的一个或多个行为,应用一条或多条行为驱动的规则来得到用户最终被分配的角色.然而,为每个行为都去查找并使用一次规则将会耗费比较多的资源和时间.为了提高效率,我们可以将用户已经实施的多个行为合并成一个大的行为,同时将多条基于行为的规则合并成一条这个合并行为驱动的组合规则.比如例 2 的规则(ii),用户每下载一本图书都会被记录一次,并在判断用户被信任度时应用一次规则.如果用户下载了两本书,则记录两次并应用两次规则.这时,我们可以增加一条组合规则:

$$\text{(ii) } OStore.Free(i-2) \xleftarrow{Download\ \ 2\ \ book} OStore.Free(i) \text{ while } i>2,$$

同时把用户实施的两个“*Download* 1 *book*”行为合并成为一个“*Download* 2 *book*”,这样,只需记录 1 个行为,应用 1 条规则就可以了.下面我们对组合规则进行定义.

定义 4. 组合规则是两条或两条以上的规则组合起来的规则,有如下形式: $R_1 \xleftarrow{B} R_2 \quad C$.说明在条件 C 下,被分配了 R_2 角色的用户实施了行为 B 之后,将不再允许使用 R_2 角色,而只能转换成角色 R_1.

(1) R_2 是规则的输入角色,也是参与组合的规则中最早的一条规则的输入角色.

(2) R_1 是规则的输出角色,也时参与组合的规则中最晚的一条规则的输出角色.

(3) 在 R_2 和 R_1 之间存在 N–1 个中间角色(其中,N 是参与组合的规则数),每个中间角色既是一条规则的输出角色,也是另一条规则的输入角色,通过这些中间角色,多条规则可以组合成一条组合规则.

(4) B 是参与组合的规则对应的行为按规则组合的顺序形成的行为序列.

(5) C 是所有参与组合的规则条件的与.

组合规则的引入既可以减少用户行为的记录条数,也可以减少行为驱动的信任规则被使用的次数,但是需要系统根据自己的策略来制定组合,也需要增加信任规则的数目.然而相对于庞大的用户行为数目来说,增加一些组合规则带来的开销远远小于它能节省的开销,因此组合规则是有价值的.必须说明的是,组合规则不能替代参与组合的这些规则,因为组合规则中的行为不是原子的,而是由原子行为组合成的序列,在实际情况下,用户可能只会执行其中部分行为,也可能没有按照序列的顺序实施行为,这时我们就不能使用组合规则,而仍然必须使用原来的规则.

2.4 信任策略更新规则

信任域根据本地信任策略制定信任规则,这些规则反映了信任域如何去信任本地或者其他信任域的角色,也规定了用户的行为将怎样影响其在信任域的被信任度.一旦制定了这些信任规则,规则所反映的信任策略也就被固定下来.然而,在大型分布式环境中,信任策略本身也应该是动态的.随着时间的推移,用户的行为、资源以及环境的变化可能迫使我们改变信任策略.比如,在线书店允许某大学的师生免费在线阅读图书,但是由于带宽的限制,书店需要限制同一时间内免费在线阅读的人数.因此,一旦在某个时间内该大学师生免费在线阅读的人数达到上限,系统就不再允许该大学其他师生阅读.而当人数重新低于上限时,系统又将允许其他师生阅读.因此,在动态信任策略中,我们还要根据大量用户的行为、资源以及环境的变化动态地调整对某个角色的信任.这个角色可以是本地的,也可以是其他信任域的.对角色信任的调整意味着对信任规则的修改,也就是对信任策略的更新.为了能够动态地调整信任规则,我们引入了规则之上的规则——信任策略更新规则.在讨论信任策略更

新规则之前,我们先定义状态变量.

定义 5. 状态变量是由信任域维护一组变量,用于记录行为统计、资源或者环境信息.

每个信任域都可能自行选择一些状态变量,用于反映资源和环境的状态,也可以记录大量用户行为的统计信息,这些状态变量的值将可能影响本地的信任策略.比如在例 2 中,在线书店定义了状态变量 *ULib.totalbuy*,记录了该大学师生已购买的图书数目.一旦该数目达到了 5 000 本,规则(a)就会将该大学师生的免费下载图书数从 3 本提高到 5 本.定义 6 描述了信任策略更新规则的能力.

定义 6. 信任策略更新规则是具有如下形式的规则: $L_1 \xleftarrow{SC} L_2$.该规则说明,如果状态变量发生了变化,使得 *SC* 条件从不满足变成可以满足,那么信任规则 L_2 将被替换成另一条信任规则 L_1.

(1) L_2 是更新规则的输入规则.

(2) L_1 是更新规则的输出规则,L_1 和 L_2 必须是不同的两条规则.

(3) *SC* 是与一个或多个状态变量相关的条件.

信任策略更新规则将在 *SC* 从不满足变成满足的临界点时发生作用,从信任规则中删掉输入规则,同时将输出规则添加进去.很多情况下,输入规则和输出规则的输入角色是相同的,说明达到一定条件后,系统将用另一个策略来信任该角色.但是,这并不是必须的,输入规则和输出规则的输入角色也可以是完全不同的角色,这时系统使用一个全新的信任策略来取代原来那个信任策略.此外,由于 *SC* 与一个或多个状态变量相关,随着系统的动态发展,*SC* 可能从不满足的状态变成可满足的状态,同样,也有可能从已满足的状态又变成不满足的状态.这时,系统可能设置另一条更新规则,当 *SC* 从满足的状态变成不满足的状态时,恢复前面的更新规则所作的改变.于是这两条更新规则互为反规则.

定义 7. 两条信任策略更新规则具有如下形式时,互为反规则:

(1) $L_1 \xleftarrow{SC_1} L_2$;

(2) $L_2 \xleftarrow{SC_2} L_1$.

1. 规则(1)的输入规则是规则(2)的输出规则.
2. 规则(2)的输入规则是规则(1)的输出规则.
3. 规则(1)的条件 SC_1 和规则(2)的条件 SC_2 互为反条件,即 $SC_1 = \neg SC_2$.

信任域在定义一条信任策略更新规则时,可以让系统自动添加新更新规则的反规则.但是,不是所有的信任策略更新规则都需要反规则.存在这样的情况,状态变量只会朝一个方向发展,比如,在线书店的状态变量 *ULib.totalbuy* 只会递增,不会减少.如果一个条件与单向变化的状态变量相关,那么该条件的满足状态也只会朝一个方向发展,因此不需要提供对应的反规则.

第 2.2 节中我们定义了组合规则来提高信任判断的效率,当信任策略更新规则的输入规则是参与组合的其中一条规则时,在更新规则发生作用时,输入规则将被输出规则替代.由于输出规则和输入规则的输入、输出角色不尽相同,在信任策略中删除输入规则,增加输出规则将使得原有的一些组合规则不再有效.因此,在使用信任策略更新规则时,需要将那些与输入规则有关的组合规则删除,同时为输出规则创建新的组合规则.

3 行为驱动的信任管理的实现机制

为了更好地描述大规模分布式环境下的信任的动态性,我们的 RT_B 语言引入了行为驱动的信任规则和信任策略更新规则,增强了 RT 语言的描述能力,但是也增加了实现的复杂度,特别是行为驱动的信任规则的实现.普通的信任规则和信任委托规则只与角色有关,而行为驱动的信任规则除了与角色有关之外,还与具体的行为相关.由于被分配同一角色的每个用户已实施的具体行为各不相同,我们必须把用户和他们所做的行为联系起来.但是属性证书不适合保存用户的具体行为,因此只能在信任域本地保存用户已经发生的行为.由于大规模分布式系统中有大量的用户,每个用户又可能实施很多行为,因此需要通过多种手段优化行为驱动的信任管理的实现机制,使得信任域能够有效地处理大量用户的信任判定.

尽管用户在系统中会实施很多动作,但是行为驱动的信任管理不会关注所有的行为,而只考虑那些会影响

用户被信任度的行为.在大部分比较完善的应用系统中,都会有日志来存放发生的事件,作为了解系统运行状况的依据.因此,信任判断模块完全可以与日志模块共享用户在系统中实施的行为.通过对行为进行良好的定义,使得行为的粒度适中,既可以作为改变信任度的最小单位,又不至于太琐碎.行为驱动的信任管理使得我们在信任判断的时候必须去检索用户已经发生的行为.大规模分布式环境中的信任域需要与其他很多信任域以及大量用户进行交互,使得本地数据库可能会保存大量的行为记录,从而造成查找效率低下.因此,我们必须采用多种优化方法来提高行为检索的效率.首先,我们可以把不同信任域的用户行为分开存储,在相对较小的记录集中检索所需的信息更加高效.其次,由于大部分情况下正常用户的行为规律都是有章可循的,因此可以将用户的行为进行组合,并设置组合规则来进行判断,这将成倍地减少检索和信任判断的时间.再其次,可以将信任判断的结果暂时保存下来,只要用户没有实施影响信任的行为,下次登录就可以重用上次信任判断的结果.最后,信任策略本身也是动态的,大量用户的行为可能会引起信任域调整对角色或者其他信任域的信任策略.一旦对信任策略进行了调整,用户之前实施的很多行为就不再会被作为信任判断的依据,因此系统可以删除这些过时的行为,从而提高查找行为的效率.

虽然行为驱动的信任策略与单个用户实施的行为相关,但是它与基于身份的信任有本质的区别.信任域的信任与安全策略仍然是实施在角色上,并不会针对单个用户指定独立的策略.而且同一角色的用户行为是有规律可循的,行为驱动的信任策略根据这些规律化的行为对同一角色的大量用户进行细分,并给予不同的信任度.由于细分数量有限,引入基于行为的信任规则既可以实现灵活的信任策略,又不至于增加过多的负担.图 3 给出了行为驱动的信任及授权管理的基本实现框架.如果信任域 1 与信任域 2 有直接或者间接的信任委托关系,则用户可以向信任域 1 提交信任域 2 发布的属性证书,包含用户在信任域 2 中被分配的角色.信任域 1 收到属性证书后,根据已有规则,得到用户在本地对应的角色.这时,我们还不能根据映射后的角色进行授权,必须再调用行为驱动的信任判断模块,根据用户已经实施的行为,应用行为驱动的信任规则,最终获得用户实际被分配的本地角色.这时,系统就可以根据基于角色的授权策略处理用户请求.处理完毕之后,如果用户在系统中实施了影响信任的行为,该行为则被记录下来.如果可能,新行为和原来的行为可以合并成组合行为,代替原先的多条行为.此外,如果用户的行为改变了某个状态变量的值,而且又触发了某条信任策略更新规则,则更新信任规则库,也包括行为驱动的信任规则.如果行为驱动的信任规则发生变化,则可以删除那些不可能再触发规则的行为,从而提高系统的执行效率.

4 结 论

基于角色的信任管理语言族 RT 可以有效地处理大规模分布式环境下的信任及授权问题,同时也描述了不同信任域之间的信任委托.然而,由于信任和授权只是针对用户在信任域中被分配的角色,并不能真正反映分布式环境下信任的动态性,包括用户的行为对其被信任程度的影响,以及大量用户的行为对信任域之间信任度的影响等,这使得信任域无法实施灵活的授权策略.行为驱动的信任管理将能提供更为强大的描述能力和更为灵活的信任策略.本文首先定义了带变量的角色,每个变量的值都表明了某些行为的积累.然后在 RT 语言族的基础上,扩展了行为驱动的信任描述能力,形成了行为驱动的基于角色的信任管理语言 RT_B.扩展部分包括:(1) 引入了行为驱动的信任规则,根据用户已经实施过的行为调整其在本地被分配的角色,使得实施不同行为的同一角色用户有不同的被信任度;(2) 定义了组合规则,将用户多次行为合并成组合行为,把多条基于行为的规则合并成组合规则,降低了行为检索和信任判断的代价;(3) 作为规则之上的规则,信任策略更新规则允许信任域根据预定义的状态变量来动态地调整本地的信任策略,一旦某个状态变量的值达到预先设定的值,就触发信任策略更新规则,用新的信任规则替代原有的信任规则;(4) 本文讨论了如何高效地实现行为驱动的信任管理,通过多种优化的机制可以提高信任判断的效率,并给出了信任判断及请求处理的具体流程.RT_B 保留了 RT 语言族的优点,只对角色授权避免了对大量用户授权及访问控制的复杂性,而行为驱动的信任规则和信任策略更新规则又能帮助信任域根据用户的行为动态地调整对用户和角色的信任,从而可以实现更加灵活的信任及授权策略,满足系统的实际需要.

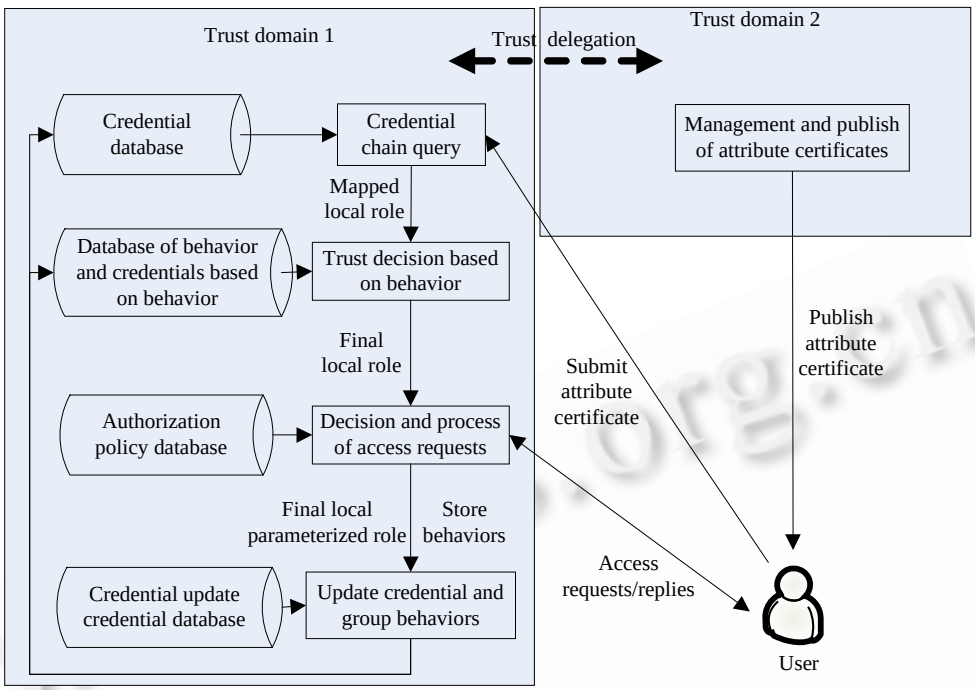

Fig.3 Implement framework of behavior-driven trust management

图 3 行为驱动的信任管理实现框架

References:

[1] Ferraiolo DF, Sandhu R, Gavrila S, Kuhn DR, Chandramouli R. Proposed NIST standard for role-based access control. ACM Trans. on Information and System Security, 2001,4(3):224−274.

[2] Li NH, Mitchell JC, Winsborough WH. Design of a role-based trust management framework. In: Heather H, ed. Proc. of the IEEE Symp. on Security and Privacy. Washington: IEEE Computer Society Press, 2002. 114−130.

[3] Blaze M, Feigenbaum J, Ioannidis J, Keromytis A. The KeyNote trust-management system version 2. IETF RFC 2704, 1999. http://www.apps.ietf.org/rfc/rfc2704.html

[4] Gorla D, Hennessy M, Sassone V. Inferring dynamic credentials for role-based trust management. In: Bossi A, Maher MJ, eds. Proc. of the 8th ACM SIGPLAN Symp. on Principles and Practice of Declarative Programming. New York: ACM Press, 2006. 213−224.

[5] Blaze M, Feigenbaum J, Lacy J. Decentralized trust management. In: Proc. of the 17th Symp. on Security and Privacy. Oakland: IEEE Computer Society Press, 1996. 164−173.

[6] Lin A, Vullings E, Dalziel J. A trust-based access control model for virtual organizations. In: Proc. of the 5th Int'l Conf. on Grid and Cooperative Computing Workshops. IEEE Computer Society Press, 2006. 557−564.

[7] Denker G, Millen J, Miyake Y. Cross-Domain access control via PKI. In: Michael JB, ed. Proc. of the 3rd Int'l Workshop on Policies for Distributed Systems and Networks (POLICY 2002). Washington: IEEE Press, 2002. 202−205.

[8] Tran H, Hitchens M, Varadharajan V, Watters P. A trust based access control framework for P2P file-sharing systems. In: Sprague RH, ed. Proc. of the 38th Annual Hawaii Int'l Conf. on System Science. Washington: IEEE Computer Society Press, 2005.

[9] Adams WJ, Davis NJ. IV, TMS: A trust management system for access control in dynamic collaborative environments. In: Proc. of the 25th IEEE Int'l Performance Computing and Communication Conf. Washington: IEEE Computer Society Press, 2006. 143−150.

[10] Chakraborty S, Ray I. TrustBAC-Integrating trust relationships into the RBAC model for access control in open systems. In: Proc. of the 11th ACM Symp. on Access Control Models and Technologies. New York: ACM Press, 2006. 49−58.

[11] Bhatti R, Bertino E, Ghafoor A. A trust-based context-aware access control model for Web-services. In: Zhang LJ, ed. Proc. of the IEEE Int'l Conf. on Web Services (ICWS 2004). Washington: IEEE Computer Society Press, 2004. 184−191.

[12] Hong F, Zhu X, Wang SB. Delegation depth control in trust-management system. In: Proc. of the 19th Int'l Conf. on Advanced Information Networking and Applications (AINA 2005). Washington: IEEE Computer Society, 2005. 411−414.

[13] Zhai ZD, Feng DG, Xu Z. Fine-Grained controllable delegation authorization model based on trustworthiness. Journal of Software, 2007,18(8):2002−2015 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/18/2002.htm

附中文参考文献:

[13] 翟征德,冯登国,徐震.细粒度的基于信任度的可控委托授权模型.软件学报,2007,18(8):2002−2015. http://www.jos.org.cn/1000-9825/18/2002.htm

李斓(1977－),男,江西南昌人,博士,讲师,CCF 会员,主要研究领域为系统与网络安全.

范磊(1975－),男,博士,副教授,主要研究领域为网络安全,密码学.

回红(1969－),女,博士,副教授,主要研究领域为网络安全,生物识别.