应用角色和任务访问控制的工作流动态授权模型

杨天怡[1]，董红林[1]，黄 勤[1]，刘益良[2]
（1. 重庆大学 自动化学院，重庆 400044；2. 重庆理工大学 电子信息与自动化学院，重庆 400044）

摘 要：为实现职责分离和最小权限约束，在传统基于角色和任务访问控制模型的基础上，提出一种应用角色和任务访问控制的工作流动态授权模型。该模型主要包含：①引入了工作流上下文信息来加强职责分离约束；②把权限最小化到任务状态层次；③根据工作流的变化和执行任务所处的状态进行动态地授权。最后以驾驶员培训系统为例，说明了该模型怎样动态实现最小权限约束、职责分离和动态授权，以此说明该模型能够满足工作流动态变化频繁的复杂系统访问控制的需要。

关键词：职责分离；访问控制；最小权限；动态授权
中图分类号：TP393.08 **文献标志码**：A **文章编号**：1001-3695(2010)04-1511-03
doi:10.3969/j.issn.1001-3695.2010.04.086

Workflow dynamic authorization model with task-role-based access control

YANG Tian-yi[1], DONG Hong-lin[1], HUANG Qin[1], LIU Yi-liang[2]
(1. *School of Automation, Chongqing University, Chongqing* 400044, *China*; 2. *School of Electronic Information & Automation, Chongqing University of Technology, Chongqing* 400044, *China*)

Abstract: According to the traditional research on task-role-based access control model, the paper proposed a workflow dynamic authorization model with task-role-based access control to achieve the separation of duties and least privilege. The model main contain: ①strengthening separation of duties by introducing the context information of workflow; ②achieving the least privilege to task-status level; ③performing dynamic authorization according to the changes of workflow and the status of task. Finally, gave an example of driver training management system to indicate how to implement the least privilege, separation of duty and dynamic authorization in the model, which could satisfy the requirements of frequent changes of workflow in the complex access control system.

Key words: separation of duty; access control; least privilege; dynamic authorization

工作流技术在企业信息化中有着广泛的应用，为防止用户进行越权存取而对信息的完整性造成破坏，需要一个合适的访问控制机制。一个适合工作流的访问控制机制必须具备两个条件：a）最小权限原则，即用户在执行任务具体的实例时只能访问该任务所允许的操作和操作客体；b）职责分离原则，将职责和权力分散于多个用户之中而不是仅仅集中在一个用户身上，这样来降低用户欺诈行为。

传统的访问控制（DAC 和 MAC）以用户为基本对象授予权限，这使得大型复杂系统的授权管理繁杂，容易出错，且对职责分离原则和最小权限原则考虑较少。Sandhu 等人[1]提出了基于角色的访问控制参考模型，通过引入角色概念实现了用户与权限的逻辑分离，极大地方便了权限的管理。然而在目前大多数的基于该模型的实际应用中，笔者发现其存在两点不足：a）权限是预先定义的，缺乏灵活性。在授权时预先把这些数据从相关业务表手工抽取出来，以静态形式存储于权限表中，授权时再通过查表方法来确定用户所拥有的权限；b）授权是静态的，权限最小化到角色层次。一旦给某角色指派了权限就将无条件一直拥有，没有考虑当前应用环境因素对权限的制约，无法对权限动态地进行回收或限制。这两点不足都不能很好地满足职责分离原则和最小权限约束。后来人们又提出了基于任务的访问控制（TBAC）模型[2,3]。TBAC 模型是一种以任务为中心的，并采用动态授权的主动安全模型，但它的主要不足之处在于它不适合大型企业的应用，权限与任务相关联，当工作流管理系统应用于大型企业的流程自动化管理时，该系统的访问控制就会不可避免地牵涉到许多任务以及用户的权限分配问题，将会引起配置过于繁琐的缺点，不利于人员的安全控管。

为了解决以上所述的问题，本文提出了一种应用角色和任务访问控制的工作流动态授权模型（WDAM）。该模型为在工作流系统中实现职责分离约束和最小权限约束，对基于角色和任务的访问控制（TRBAC）模型[4]进行了改进：引入了上下文信息从而加强了职责分离的约束；在文献[5]的基础上，把权限最小化到任务状态层次并使其互相关，根据工作流的变化和执行任务所处的状态对权限拥有者进行动态地授权。

1 WDAM 模型定义及描述

WDAM 模型如图 1 所示。

其基本思想是：根据现代工作流管理系统访问安全的需要，即在一个任务的执行过程中，根据任务状态的不同授予不

收稿日期：2009-06-30；**修回日期**：2009-09-07

作者简介：杨天怡（1952-），男，江苏镇江人，教授，博士，主要研究方向为计算机控制技术；董红林（1985-），女，湖北宜昌人，硕士研究生，主要研究方向为信息系统及安全（20071302062@cqu.edu）；黄勤（1960-），女，重庆人，教授，主要研究方向为智能仪器远程技术；刘益良（1949-），男，四川人，副教授，主要研究方向为信息及安全.

同的权限,使任务状态与最小权限相关联。在原有的 TRBAC 模型的基础上,使权限和任务不直接挂钩,而是根据任务状态的变化授予该任务不同的权限。某用户在工作流运行到某一任务实例时请求授权,首先应对用户指派合适的角色,接着判断该角色是否有能力执行当前任务;最后根据当前任务所处状态的不同授予用户不同的执行权限,当该任务状态结束,立即收回相应的权限。

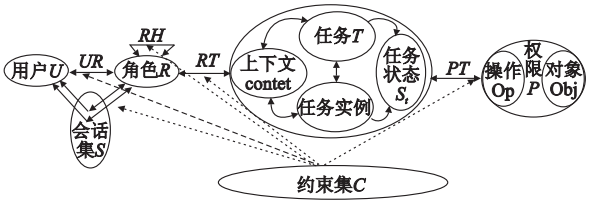

图1　WDAM模型

以下给出几个将要用到的概念:

a)用户集 $U=\{u_1,u_2,\cdots,u_n\}$,其中 $u_i(1\leqslant i\leqslant n)$ 为系统中的某一任务的执行者。

b)角色集 $R=\{r_1,r_2,\cdots,r_n\}$,其中 $r_i(1\leqslant i\leqslant n)$ 表示系统中存在的角色。

c)任务集 $T=\{t_1,t_2,\cdots,t_n\}$,其中 $t_i(1\leqslant i\leqslant n)$ 是工作流引擎调度的最小逻辑单位,若干个任务组成一个工作流程。

d)任务状态集 $T_s=\{t_{s1},t_{s2},\cdots,t_{sn}\}$,其中 $t_{si}(1\leqslant i\leqslant n)$ 是某个任务所处的状态。在任务的生命期内,任务可能处于不同的状态:存在、失败、运行、等待、完成等。本文只考虑任务的初始态、执行态、提交态。一个任务在某一时刻只能对应一种状态,不同的任务可以对应同一种状态。

e)权限集 $P=\{p_1,p_2,\cdots,p_n\}$,其中 $p_i(1\leqslant i\leqslant n)$ 是用户所具备的对任务进行操作的能力。

f)用户角色分配映射关系 $UR\subseteq U\times R$,由管理员对用户和角色作出的分配。

g)角色任务映射关系 $RT\subseteq R\times T$,角色和任务之间多对多的关系。

h)权限任务映射关系 $PT\subseteq P\times T$,权限与任务之间多对多的关系。

i)用户角色激活关系 $UA\subseteq UR$,是在会话中用户和被激活的角色之间的多对多的关系。

j)上下文信息(context)是用来描述工作流所处状态的信息。本文用工作流历史信息对上下文信息进行描述,用 *HIS* 表示工作流历史信息的集合,*his* 表示一条历史信息($his\in HIS$),它被定义为一个四元组$\langle u_i,r_j,t_n,n\rangle$,表示在一个工作流运行过程中,用户 u_i 曾经以角色 r_j 的身份执行了该工作流中第 n 个任务 t_n。

k)约束(constraint)在工作流环境中,许可、角色、活动和用户都存在发生冲突的可能性,就要相应的策略机制规则进行控制。职责分离原则(separation of duty, SoD)是一种为了减少欺诈犯罪潜在危险的约束,它分为以下两种:

(a)静态职责分离(static separation of duty,S-SoD)一个用户或者是一对冲突用户(conflict users, CU)不能被指派两个互斥角色(mutually exclusive roles, MER)。其中,冲突用户指的是拥有足够权力并极有可能合伙欺诈的两个用户,一般而言,有亲戚、亲密朋友或利益关系等有可能合伙欺诈的用户均是冲突用户。在系统管理中,应将冲突用户当成单个用户来看待,并防止互斥角色授予冲突用户,即

$$\forall(u_1,r_1),(u_2,r_2)\in UR\wedge(r_1,r_2)\in MER\Rightarrow u_1\neq u_2\wedge(u_1,u_2)\notin CU$$

(b)动态职责分离(dynamic separation of duty,D-SoD)一个用户或者是一对冲突用户在一个会话中不能同时激活两个互斥角色,即

$$\forall(u_1,r_1),(u_2,r_2)\in UA\wedge(r_1,r_2)\in MER\Rightarrow u_1\neq u_2\wedge u_1\notin\{u\mid(u,u_2)\in CU\}$$

考虑了上下文信息以后,S-SoD 和 D-SoD 的形式化描述分别如下:

$$(r_i,r_j),\in MER\wedge\langle u_1,r_i,t_i,i\rangle\in HIS\wedge\langle u_2,r_j,t_j,j\rangle\in HIS\wedge i\neq j\Rightarrow u_1\neq u_2\wedge u_1\notin\{u\mid(u,u_2)\in CU\}$$

和

$$(r_i,r_i')\in MER\wedge\langle u_1,r_i,t_i,i\rangle\in HIS\wedge\langle u_2,r_i',t_i,i\rangle\in HIS\Rightarrow u_1\neq u_2\wedge u_1\notin\{u\mid(u,u_2)\in CU\}$$

2　动态授权的实现

本文对文献[5]中的权限分配模型进行了改进,除了引用原有的角色管理器(role manager)、任务管理器(task manager)、状态管理器(status manager)、权限管理器(privilege manager),还引入了验证管理器(verification manager)、策略加强点服务器(PEP server)、策略决策点服务器(PDP server)、工作流上下文信息集(context information)和约束条件集(constraint policies)。

2.1　运行中动态授权的实现

WDAM 中动态授权的实现框架如图 2 所示。

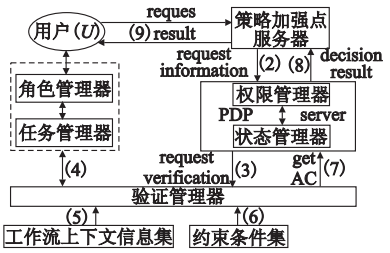

图2　WDAM的实现框架

验证管理器的作用:

a)若在一个会话中,用户激活的任意两个角色违背了动态职责分离原则,那么用户不能同时激活它们。

b)查询工作流历史信息,若某用户在前面的工作流任务中已经激活了某些角色,那么该用户在后续的工作流任务中不能激活与这些角色互斥的角色,否则将违背静态职责分离原则。

PEP(policy enforcement point)是策略强制点,在访问目标服务器处实现对资源访问的控制。PDP (policy decision point)是策略决策点,依据验证管理器返回的授权证书决定用户的请求是否被通过。若通过,则根据已经过验证的角色和当前请求的任务与请求用户的对应关系,查询任务状态管理器和权限管理器,并授予用户在该状态下的权限操作。

WDAM 中动态授权流程如下:

(1)用户 u_i 发送授权请求给服务方 PEP server;

(2)PEP server 将用户的资源授权请求信息传送给 PDP server;

(3)PDP server 请求验证管理器对授权请求进行验证;

(4)~(7)Verification manager 查询①在工作流运行过程中,依据用户角色指派(*UR*)和角色任务指派(*RT*)得到的用户 u_i、角色 r_j、任务 t_k 三者间的对应关系,②工作流上下文信息以及③所需约束条件,动态地对用户激活的角色和用户以角色的身份执行的任务进行职责分离约束的验证,若满足所需的限制,即返回授权证书 AC(authorization certificate)则可执行(8);

否则,拒绝用户 u_i 的授权请求;

(8)PDP server 查询状态管理器 status manager 中任务 t_k 此时的任务状态 t_{sm},并根据此时的任务状态 t_{sm} 和权限 - 任务状态对应列表查询用户所被允许的操作 op_h,并将授权决策结果及允许的权限返回给 PEP server。授权决策结果可能是 permit、deny、undetermined。其中:deny 表示验证条件不满足;undetermined 表示根据当前的授权策略和工作流的上下文环境还不能做出决策结果。

(9)PEP server 返回授权决策结果给用户,若用户请求被允许,则显示出相应的操作界面,使用户 u_i 在允许权限内进行操作;若被拒绝,显示出无权访问的界面;若果结果是 undetermined,则根据授权系统的披露显示,引导用户披露进一步的信息。执行(8),若查询 status manager 中任务状态 t_{sm} 结束,则立即收回相应的权限,并使任务状态进入下一状态;直至下一状态为提交态时结束对此任务 t_k 的操作。

2.2　系统性能的分析

1)同步和最小权限约束　在 WDAM 中,随着工作流推进到不同的任务阶段,对于每个任务下不同的任务状态,角色被赋予的权限也发生变化,把授权最小化到任务状态这一层次,从而实现了最小权限约束原则。授权定义为在某一任务单元某一任务状态的执行期间可进行授权,在不同任务单元的不同任务状态下对应着不同的操作权限。当工作流推进到某一任务单元的某一任务状态阶段时,相应的权限才能被激活,根据对应的用户—角色—任务对应关系授予用户相应的权限;当此任务状态结束后,权限立即回收,立刻取消与该任务状态相关的授权,任务状态进入下一状态,再进行相关的授权,直至任务状态为提交态,才结束与该任务单元相关的授权,因而满足了工作流管理系统中的信息存取要求。

2)职责分离约束　职责分离是现代企业的一大特点,可以防止企业活动中的欺骗和越权行为, WDAM 模型在 TRBAC 模型授权约束的基础上,加入了验证环节。通过工作流上下文信息和约束条件对用户—角色—任务的对应关系进行职责分离的验证,从而实现了权责分离原则,减轻了管理员面对庞大的用户数目和信息资源所承担的负担。

3)动态授权　在工作流运行过程中,验证管理器把验证结果传送给 PDP server,PDP server 可以通过授权证书查询任务状态管理器和权限管理器,动态地决定是否可以授权给用户;当任务状态发生变化,及时收回相应的权限,并判断下一状态的授权,从而满足工作流动态运行的需要。

3　实例应用

为了更清楚地描述 WDAM 的应用方式,本文以《机动车驾驶员培训管理系统》中的工作流程为例,如图 3 所示,说明本文模型的应用。

图3　驾驶员培训资料工作流程

在该系统中,这里对一些用户进行描述,例如,用户包括办公室王主任,操作人员小张、小李、小黄、小甘、小周、小唐和小孙,其中,小李与小张是亲戚关系。根据业务流程,可以有如下角色:报名人、审核人、收费人、教练、考官、制证人、核证人和发证人,其中报名人与收费人,制证人与核证人为互斥角色。相应的任务有:报名、资料审核、收费、教学、考试、制证、核证和发证。每个任务都有三种任务状态,即初始态、执行态和提交态。

3.1　职责分离约束的实现

根据该系统的描述,$MER = \{(报名人,收费人),(制证人,核证人)\}$,$CU = \{(小李,小张)\}$。

对于某一学员的工作流系统,假设小张申请报名人的权限,则系统经过验证后允许他执行报名的任务,即(小张,报名人,报名,1) $\in HIS$;当工作流运行到资料审核时,根据上下文信息及约束关系,则所有用户申请该任务的权限时都可以被授予相应的权限,若此时还是小张申请了该任务,则(小张,审核人,资料审核,2) $\in HIS$;到当前任务为收费任务时,根据上下文信息和约束条件(即(小李,小张) $\in CU$,(报名人,收费人) $\in MER$),此时系统将会阻止小张和小李对此任务权限的申请,因为对于某一个学员信息系统,小张先前角色是报名人,在此就不能为收费人的角色,而小张和小李为冲突用户,小李也不能对该学员进行收费工作。若小黄申请了该任务的权限,则(小黄,收费人,收费,3) $\in HIS$;这样依此类推就可以得出后续任务的执行情况,对于该学员的工作流系统下面给出它其中一种执行情况:

(小张,报名人,报名,1)→(小张,审核人,资料审核,2)→(小黄,收费人,收费,3)→(小甘,教练,教学,4)→(小甘,考官,考核,5)→(小黄,制证人,制证,6)→(小李,核证人,核证,7)→(小张,发证人,发证,8)。

3.2　最小权限约束的实现

限于篇幅,在此只以收费任务为例进行描述,其他任务的实现情况与此相似。当进入收费任务的初始态时,小黄被授予查阅权限,即能查看学员的报名、资料审核以及缴费情况。当可以进行收费时,进入执行态,系统立即收回小黄的查阅权限,授予小黄写的操作权限,即可写入或更改学员的缴费情况,当小黄完成此操作,可以提交,提交后系统自动进入提交态,他的写权限就立即被收回,不能再对学员的缴费情况进行修改,只具有查看权限。当该任务结束后,系统自动地进入下一任务,直至工作流程结束。由此可看出该模型把权限最小化到任务状态层次,真正地实现了最小权限约束。

4　结束语

本文对以往的访问控制模型及动态授权在工作流管理系统中的应用进行了研究,提出了一个新的动态授权模型 WDAM,该模型能满足职责分离约束和最小权限约束,能随着任务状态的变化动态地授予权限并随时监管控制权限的授予与收回,能够满足工作流动态变化频繁的复杂系统访问控制的需要。

参考文献:

[1] SANDHU R, COYNE E, FEINSTEIN H, *et al*. Role-based access control models[J]. IEEE Computer, 1996,29(2):38-47.

[2] TAN K, CRAMPTON J, *et al*. The consistency of task-based authorization constraints in workflow systems[C]// Proc of the 17th IEEE Computer Security Foundations Workshop. Washington DC: IEEE Computer Society, 2004:155-166.

[3] 洪帆,赵晓雯.基于任务的访问控制模型及其实现[J].华中科技大学学报,2002,30(1):17-19.

[4] OH S, PARK S. Task-role-based access control model [J]. Information System, 2003,28(6): 533-562.

[5] 陈传波,黄俊华.基于工作流任务状态的访问权限分配模型[J].计算机工程与科学,2006,28(7):87-90.