RBAC 中条件周期特性角色激活约束研究 *

殷晓玲[1], 黄海生[1,2], 王汝传[2]
(1. 池州学院 数学计算机系, 安徽 池州 247000; 2. 南京邮电大学 计算机学院, 南京 210003)

摘 要: 对 RBAC 中的角色激活特性进行研究,提出了带条件周期角色激活特性的 CP-RBAC 模型。分析了模型的一致性状态,讨论了一致性状态维护问题;并针对条件周期约束问题给出了相应算法来解决会话的状态转变问题。CP-RBAC 模型能够使系统更加安全有效,降低大规模网络应用的复杂性和安全管理的费用,为网络安全防护产生更大的经济和技术效益。

关键词: 角色; 访问控制; 条件周期; 用户激活
中图分类号: TP309.2 **文献标志码**: A **文章编号**: 1001-3695(2010)08-3034-04
doi:10.3969/j.issn.1001-3695.2010.08.059

Research on constraint of conditions and periodic in user activity in RBAC

YIN Xiao-ling[1], HUANG Hai-sheng[1,2], WANG Ru-chuan[2]
(1. *Dept. of Maths & Computer, Chizhou College, Chizhou Anhui* 247000, *China*; 2. *School of Computer Science & Technology, Nanjing University of Posts & Telecommunications, Nanjing* 210003, *China*)

Abstract: Based on the research on the constraint of user activity in RBAC, this paper described a new RBAC model with conditions and periodic in user activity character. Analyzed the consistent state of the model and discussed the problem for maintaining the consistent state. Developed some algorithms to solve the state change problem of the time-constraint and the sessions. The new model made the system be more safer and effective. Also it could reduce complexity of mass network application and cost of safety management. It could create a great deal of economic and technological benefit in safety management of Internet.

Key words: role; access control; conditions and periodic; user activity

0 引言

基于角色的访问控制(role-based access control, RBAC)是目前广泛使用的一种访问控制技术。其基本思想是用户被分配给角色,权限也被分配给角色,用户通过角色获取相应权限[1]。它的技术优势在于既可以提供传统的 DAC(自主访问控制)功能,又可以提供 MAC(强制访问控制)功能[2]。而 RBAC 的约束(constraint)规定了当权限被赋予角色时,或角色被赋予用户时,以及当用户在某一时刻激活一个角色时应遵循的强制性规则[3,4]。在 NIST 的标准 RBAC 模型中,约束包括静态约束和动态约束两类,约束与用户—权限关系共同决定了 RBAC 模型中用户的访问许可[5]。

通过对 RBAC 模型的分析和研究,发现要在实际系统中直接实现它是非常困难的,该模型没有考虑具体信息系统的特点,没有给出实现机制,因此在不同的系统中对于基于角色的访问控制的实现也是采用不同的方式。例如在实际应用环境中,当某个角色被激活时,可能需要其他用户的合作才能完成。本文从用户与角色之间的作用关系进行分析,提出了条件周期角色激活特性的 CP-RBAC(conditions and periodic role-based access control)模型。

1 用户与角色关系分析

从用户对角色的作用方式来看,用户被分为三大类,即被指派了该角色的用户(指派用户)、可以激活该角色的用户(激活用户)和与该角色不相关的用户。指派用户可能同时又是激活用户。角色在使用过程中,首先被指派用户初始化,然后由激活用户进行激活,才能使得角色进入激活状态,指派用户才能使用角色中指派的权限。

1.1 角色激活约束分类[6]

1.1.1 根据指派角色用户分类

从角色是否需要激活来看,可将用户激活约束分为两类:

a)静态激活约束。对所有指派该角色的用户均需要激活用户集中的用户进行激活(包含自身激活)。这类约束与被指派该角色的用户无关。

b)动态激活约束。根据指派该角色的用户不同,来设置不同的激活用户集。

1.1.2 根据角色激活用户集分类

根据角色的激活用户集的类型可以分为以下几类:

收稿日期: 2010-03-04; **修回日期**: 2010-04-15 **基金项目**: 国家自然科学基金资助项目(60573141,60773041);国家"863"计划资助项目(2007AA01Z404,2007AA01Z478);安徽省高校省级自然科学研究资助项目(KJ2008B116);安徽省高等学校省级优秀青年人才基金资助项目(2009SQRZ139)

作者简介:殷晓玲(1976-),女,讲师,硕士,主要研究方向为信息安全、数据库(czszyxl@tom.com);黄海生(1965-),男,副教授,主要研究方向为计算机软件技术、网络安全、信息安全;王汝传(1943-),男,教授,博导,主要研究方向为计算机软件、计算机网络、信息安全、移动代理、虚拟现实技术等.

a)单组激活。激活用户集中的 K 个用户即可激活角色。

b)分组激活。用户集是几个子集的并集,每个子集 us_i 中的 k_i 个用户即可激活角色。

c)联合激活。激活用户集是几个子集的并集,所有子集 us_i 中的 k_i 个用户组成的用户集合可以激活角色。

d)混合激活。它是第 b)c)类的组合形式。这类激活用户集是两类激活用户集,一类激活用户集实现 b)中用户激活约束,另一类实现 c)中的用户激活约束。

1.2　用户激活约束策略

角色被激活后,指派用户(会话用户)就可以使用该角色被指派的权限。但由于需要其他用户协助激活的角色,多为机密性较高的权限资源。因此角色在激活后维持的时间要有一定的限制。

根据激活状态维持的时间策略,可以将用户激活划分为以下几个类别:

a)临时激活。角色被激活后,只能被指派用户使用有限次。在使用次数达到后,该会话将被阻塞;若继续使用,则需要激活用户再次进行激活。

b)短期激活。角色被激活后,在一段有限的时间段内可以持续有效,该会话使用超过给定的时间段将被阻塞;若继续使用,则需要激活用户再次进行激活。

c)长期激活。角色被激活后,可以持续使用,直到用户终止该次会话。

在此基础上本文提出一种条件周期角色激活策略,以适应实际应用的需求。

1.3　条件周期角色激活策略

角色被激活后,在指定的条件周期下一直可以使用该角色,直到条件周期不满足或用户终止该次会话。

2　条件周期激活角色特性 RBAC(CP-RBAC)

2.1　条件周期基本语义

定义 1　时间点序列[7]

$TE=\{t_i \mid i\in N\}$,TE 是所有时间点的集合。$t_i\in TE$ 代表虚拟时间世界中的一个时间点,它并不一定要与现实时间保持一致。

$T=seq(TE)$ 表示由时间点构成的时间序列。要求序列 T 中的元素是严格递增的,即 $\forall i,j\in N,\forall t_i,t_j\in T, i<j\Leftrightarrow t_i<t_j$。

定义 2　时间区间定义

$TR=\{(t_i,t_j)\mid t_i,t_j\in T, i<j\}$,时间区间是由两个时间点所构成的区间。

$TRS=2^{TR}$ 表示由时间区间构成的集合。

定义 3　周期的表示[8]

给定历法 $C_d,C_1,C_2,\cdots,C_n$,周期 P 定义

$$p=\sum_{i=1}^{n} O_i\cdot\ C_i\rhd r\cdot\ C_d$$

这里 $O_1=\text{all}$,$O_i\in 2^{IN}\cup\{\text{all}\}$,$C_i\subseteq C_{i-1}(i=2,\cdots,n)$,$C_d\subseteq C_n$,且 $r\in IN$。其中:all 表法 C_i 的所有时间区间;$O_i\in 2^{IN}\cup\{\text{all}\}$ 是时间区间子集;符号 $\subseteq$ 代表子历法关系,即 C_i 是 C_{i-1} 的子历法;r 为时间间隔;C_d 为时间间隔的持续时间;IN 是自然数集合;符号 $\rhd$ 表示将第一部分的周期表示分离出来,即从它所代表的时间区间的开始点持续的日历长度 C_d 的数量。例如,all.year+$\{3,7\}\cdot$ months$\rhd 2\cdot$ months 代表所有年份的三月、七月,而且持续两个月时间,即每年的三、四月和七、八月。

一个周期时间表达式 P 表示了一个周期性时间区间的无限集,将这个无限集记为区间 $\prod(P)$,并有下面定义。

定义 4　函数 $\prod(\ )$[9]

设 P 是形如 $p=\sum_{i=1}^{n} O_i\cdot\ C_i\rhd r\cdot\ C_d$ 的周期时间表达式,$\prod(P)$ 是持续时间为 $r\cdot\ C_d$ 的一组时间区间,而且这组时间区间的起点集 S 为:

当 $n=1$ 时,S 包含日历 C_1 的所有时间区间的起点;

当 $n>1$,$O_n=\{n_1,\cdots,n_k\}$ 时,S 包含日历 C_n 的第 $n_1,\cdots,n_k$ 个起点,其中 C_n 在 $\prod(\sum_{i=1}^{n-1} O_i\cdot\ C_i\rhd 1\cdot\ C_{n-1})$ 中。

定义 5　函数 Sol( )[10]

设 t 表示时间,P 是一个周期表达式,begin 和 end 是日期表达式。$t\in \text{Sol}(\langle P,[\text{begin},\text{end}]\rangle)$,当且仅当存在 $\tau\in\prod(P)$,那么 $t\in\tau$ 且 $t_b\leqslant t\leqslant t_e$。其中 t_b、t_e 分别代表开始时间 begin 和结束时间 end。

定义 6　条件周期表达式[11]。具有约束条件集(简称条件集,记为 COND)限制的周期表达式称之为条件周期表达式,记为　,其定义是　= Logic(cond:COND)$\cdot$ P。其中函数 Logic(cond:COND)代表对条件集的元素进行逻辑运算,返回布尔值(Ture/False)。

根据条件周期表达的定义,可以将限制时间[begin,end]转换为条件集元素 $c_1=\text{CondFun1}([\text{begin},\text{end}],t)$。其中 t 是执行时的当前时间。时刻集合的条件周期表达式是　=$\{c_1\}\cdot$ P。当函数 Logic 执行结果为 True 时,落在周期 P 中的各种约束具有控制意义,能够为模型提供控制依据;当执行结果为 False 时,落在周期 P 中的各种约束只能为模型提供否定的控制依据。

2.2　条件周期约束模型

2.2.1　条件周期性约束模型的系统功能函数

条件周期约束模型的基本元素在文献[12]中已给出,本文只给出系统功能函数。

定义 7

(1)$\text{UN}\subseteq 2^{\text{users}}\times N$,$(us,k)$ 表示在用户集 us 上限制最小的用户数为 k。

(2)Active_UserSets:(r:Roles)$\rightarrow 2^{\text{UN}}$,返回可以激活角色 r 的用户集。

(3)Active_UserSetsU:(u:Users,r:roles)$\rightarrow 2^{\text{UN}}$,返回可以为用户 u 激活角色 r 的用户集。

(4)Active_Roles:(u:User)$\rightarrow 2^{\text{Roles}}$,返回用户可以激活的角色集合。

(5)Session_ActiveUsers:(s:Session,r:Roles)$\rightarrow 2^{\text{Users}}$,返回会话 s 中参与激活角色 r 的用户集合。

(6)Session_Holder:(s:Session)$\rightarrow$Users,返回建立会话 s 的用户。

(7)Session_Activeper:(s:Session,r:Roles)$\rightarrow P$,返回激活会话 s 中的角色 r 的激活生效周期。

(8)Session_Activecond:(s:Session,r:Roles)$\rightarrow C$,返回激活会话 s 中的角色 r 的激活有效条件。

(9)$\text{Can\_Active}\subseteq 2^{\text{users}}\times\text{Roles}$,Can_Active($us$,$r$)表示角色 r 的拥有者在使用 r 时需要用户集 us 中的所有用户进行激活。

在 RBAC 模型中,如果 $\exists(u,r)\in UA$,则用户 u 即可激活

角色 r,是最简单的用户激活形式,可以描述为 Can_avtive($\{u\}$,r)。

2.2.2 CP-RBAC 约束的构成

在文献[7]中已经对约束给出了一个形式化的描述。本文用 C 表示所有在文献[7]中的约束,用 CP_C 表示在 CP-RBAC 中的条件周期约束。

定义 8 条件周期约束谓词定义

In_Range$\subseteq C\times$ 表示约束 C 在指定的条件周期 内必须成立。

Last_Length$\subseteq C\times L$,Last_Length(c,l)表示约束 c 从激活的时间 t 开始,在某一条件周期 内最多只能激活到时间 t_0,real_time(t_0) − real_time(t) = l。

定义 9 条件周期约束(CP_C)定义

CP_C = In_Range(C,) | Last_Length(C,L)

2.2.3 角色激活的形式语言描述[6]

定义 10 静态用户激活约束。$SSA\subseteq 2^{users}\times N\times$ Rolse,该关系满足 $\forall(us,k,r)\in SSA, us\in 2^{Users}, k\in N, r\in$ Roles, $\forall t\subseteq us$,有$|t|\geqslant k\Leftrightarrow$Can_Active($t$,$r$)。

定义 11 动态用户激活约束。$DSA\subseteq 2^{users}\times N\times UA$,该关系满足 $\forall(us,k,r)\in DSA, us\in 2^{Users}, k\in N, (u,r)\in UA$, $\forall t\subseteq us$,有$|t|\geqslant k\Leftrightarrow$Can_Active($t$,$r$)。

定义 12 分组静态用户激活约束。$GSSA\subseteq 2^{users}\times N\times$ Roles,该关系满足 $\forall(us,ks,r)\in GSSA, us\in \overset{i}{\underset{j=1}{Y}} us_j, ks\in\{k_j\mid k_j\in N, j=\overline{1,j}\}, r\in$ Roles, $\forall t_j\subseteq us_j, j=\overline{1,j}$,有 Can_Active($t$,$r$)$\Leftrightarrow|t_j|\geqslant k_j, j=\overline{1,i}$。

定义 13 分组动态用户激活约束。$GDSA\subseteq 2^{users}\times 2^N\times UA$,该关系满足 $\forall(us,ks,(u,r))\in GDSA, us\in \overset{i}{\underset{j=1}{Y}} us_j, ks\in\{k_j\mid k_j\in N, j=\overline{1,j}\}, (u,r)\in UA, \forall t_j\subseteq us_j, j=\overline{1,j}$,有 Can_Active($t_i$,$r$)$\Leftrightarrow|t_j|\geqslant k_j, j=\overline{1,i}$。

定义 14 联合静态用户激活约束。$CSSA\subseteq 2^{users}\times 2^N\times$ Roles,该关系满足 $\forall(us,ks,r)\in CSSA, us\in \overset{i}{\underset{j=1}{Y}} us_j, ks\in\{k_j\mid k_j\in N, j=\overline{1,j}\}, r\in$ Roles, $\forall t_j\subseteq us_j, j=\overline{1,j}, t=\overset{i}{\underset{j=1}{Y}} t_j$,有 Can_Active($t_i$, r)$\Leftrightarrow|t_j|\geqslant k_j, j=\overline{1,i}$。

定义 15 联合动态用户激活约束。$CDSA\subseteq 2^{users}\times 2^N\times UA$,该关系满足 $\forall(us,ks,(u,r))\in CSSA, us\in \overset{i}{\underset{j=1}{Y}} us_j, ks\in\{k_j\mid k_j\in N, j=\overline{1,j}\}, (u,r)\in UA, \forall t_j\subseteq us_j, j=\overline{1,j}, t=\overset{i}{\underset{j=1}{Y}} t_j$,有 Can_Active($t_i$,$r$)$\Leftrightarrow|t_j|\geqslant k_j, j=\overline{1,i}$。

上述定义描述了用户激活约束的单组激活、分组激活和联合激活三种不同的类型。对于比较复杂的混合型的用户激活类型,可以通过上述定义得到。

对于引入条件周期事件后的 RBAC 系统,会话在对系统的条件周期特性的研究中占有重要的地位,所以必须也对会话作出相应的扩展。

定义 16 Session 的扩展

Session 由以下七元组来描述(user, roles, ua, pa, cp_c, cp_c _starttime, state_change_time, task)。其中:users、roles、ua、pa 分别是全局系统中相应集合的子集,表示该会话所允许的用户、角色、用户角色的映射关系和角色权限映射关系;cp_c 表示该会话在运行必须满足的时间约束;cp_c_startime$\subseteq$cp_c ×TE 用来记录每一个 CP_C 时间约束开始激活的时间;state_change_time 记录该会话的状态变化时间;task 表示会话需完成的一系列操作。

定义 17 函数的扩展

Currenttime:$\varnothing\rightarrow$TE:是一个全局原子函数,返回当前时间

start:TR$\rightarrow$TE:start((t_i,t_j)) = t_j

end:TR$\rightarrow$TE:end((t_i,t_j)) = t_j

in_range:TE × Sol( )$\rightarrow$|true,false|:in_range(t,Sol(〈[begin,end],P)) = $t\in$Sol(〈[begin,end],P)

not_in_range:TE × Sol( )$\rightarrow$|true,false|:not_in_range(t,Sol(〈[begin,end],P)) = $t\notin$Sol(〈[begin,end],P)

in_ranges:TE × TRS$\rightarrow$|true,false|:in_range(t,tr_i) = $\underset{tr\in tr_i}{\cap}$ in_range(t,tr)

在 CPRABC 中,函数可随用户的需要自行扩充。

定义 18 扩展后的系统状态空间

OLDS = $\{s_i\mid i\in N\}$ 表示系统中已结束会话的集合。

BLOCKS = $\{s_i\mid i\in N\}$ 表示系统中未结束但由于条件周期约束无法进行而阻塞的会话集合。

CURRENTS = $\{s_i\mid i\in N\}$ 表示系统中当前正在进行的会话集合。

ERRORS = $\{s_i\mid i\in N\}$ 表示系统中由于条件周期约束或其他原因而无法进行下去的会话集合。

其中 OLDS∩BLOCKS∩CURRENTS∩ERRORS = $\varnothing$。

本文用 state 表示一个系统状态,states 表示系统的所有状态空间:

state = $\{U,R,P,UA,PA,SA,RH$, CURRENTS, BLOCKS, OLDS, ERRORS, CP_C, CP_C_SESSIONS, currentime$\}$, states = 2^{state}

3 CP-RBAC 状态改变及其一致性讨论

3.1 CP-RABC 中的角色激活状态检测

在检测角色是否被激活时,需要检查实际参与激活用户的数量和相应的条件周期。对 SSA、DSA、GSSA 以及 GDSA 来说,就是要找出一个激活用户集,使得实际参与该次激活的用户数量满足激活约束在该用户集上的限制。

$\exists_{us'}\subseteq us, us'\subseteq$Session_ActiveUsers($s$,$r$), ($us$,$k$) $\in$ Active UserSets$U$(u,r), $\ni|us'|\geqslant k$。对于 CSSA 和 CDSA 两种方式,检测情况显得复杂一些:$\forall$ (us,k) $\in$ Active_UserSets$U$(u,r), $\exists_{us'}\subseteq us, us'\subseteq$Session_ActiveUsers($s$,$r$), $\ni|us'|\geqslant k$;在实际检测过程中,这种检测行为应该是激活动作驱动的:当一个激活用户加入到激活队列中时,系统要检测该用户参与激活后是否已经具备激活该角色的条件,可以使用算法 1。

算法 1 检测角色是否被激活

输入:u,r。

输出:角色激活的条件周期 。

a)调用 Active_UserSetsU(u,r),得到可以激活用户 u 使用角色 r 的用户集,记为(us_j,k_j),$j=\overline{1,i}$;

b)初始化计数变量 $q_j=0, j=\overline{1,j}$;

c)等到用户加入激活,新参与激活的用户记为 u',按照以下公式修改 q_j:

$$q_j\begin{cases} q_j(u'\notin us_j) \\ q_j+1(u'\in us_j) \end{cases}$$

d)判断角色的激活结果:

(a)对于 SSA、DSA、GSSA、GDSA,如果 $\exists e, \ni q_e \geqslant k_e$,则该角色将被激活。

(b)对于 CSSA、CDSA,如果 $q_j \geqslant k_j, j = \overline{1,j}$,则该角色将被激活。

e)如果用户 u 本次会话使用的角色 r 被激活,输出 r 的条件周期;否则继续执行步骤 c)。

3.2　保持条件周期约束会话的状态变化算法[12]

会话中的角色被激活后,为了确保被激活角色在允许的条件周期内使用,需要实时监测系统的条件周期,一旦发现条件不满足则该角色,将被强制转为阻塞状态,等待下一次的激活后方可再用。系统在正常状态下,会话将在 ERRORS、OLDS、CURRENTS、BLOCKS 这四个会话集合中变化。当系统创建一个会话时,将根据会话的条件周期约束是否满足而将该会话放入 CURRENTS 或 BLOCKS 中。如果会话属性不发生变化,会话可能在 CURRENTS 或 BLOCKS 两个集合中迁移零次或多次,并最终将由 CURRENTS 中正常结束而被移动到 OLDS 中,或由于无法继续运行而被移动到 ERRORS 中;如果会话属性发生变化,会话也可能从 BLOCKS 中直接迁移到 ERRORS 或 OLDS 中。

定义 19　状态变化时间函数

state_change_time:((CURRENTS ∪ BLOCKS) × CP_C) → TE,用来计算会话约束的翻转时间。

下面给出算法来计算条件周期约束的状态变化时间。本文中所有算法包含如下假定:

a)条件周期约束的定义和形式同上节定义,并且所有的条件周期集合中包含的时间区间不交。

b)当计算条件周期约束由成立到不成立的状态变化,条件周期约束必须成立;否则条件周期约束不成立。

c)算法输入参数中参数为 true 表示计算约束成立到不成立的状态变化条件周期;否则反之。

算法 2　计算会话的状态变化时间

输入:会话 s,{true,false}。

输出:会话的状态变化时间。

if ∃sessions ∈ Sessions, ∃starttimes ∈ 2^{TE}, (cp _ c, sessions, starttimes) ∈ CP_SESSIONS

if 第 2 个参数为 true//计算约束成立到不成立的状态变化时间

if s. ssc _ starttime ≠ ∅//存在已被激活的条件周期约束

在 ∏(P)中找到相应的 τ,满足

currenttime ∈ τ ∩ in_range(currenttime, Sol(<[begin,end],P)) → true

sct = end(τ)

else

{if(COND 为 true)

在 ∏(P)中找到相应的 τ,使得

currentime < start(τ) ∩ ∀τ$_i$ ∈ ∏(P),(start(τ) − currentime) ≤ (start(τ$_i$) − currenttime)

sct = start(τ)

else

sct = ∝　}

sct 即为所求得的会话状态变化时间。在计算出会话的状态变化时间之后,可将其保存在会话的 state_change_time 中。

3.3　系统一致性状态及其维护

定义 20　系统一致性状态是指满足所有条件周期约束及一般约束的系统状态。

在前述两个算法的基础上,进一步分析何时计算各类约束及会话的状态转变时间,并且在此基础上,提出系统会话调度算法。

3.3.1　状态转变时间的计算时机和系统会话调度算法

会话由运行到阻塞的状态改变时间的计算时机如下:

a)会话自身激活某条件周期约束时;

b)当会话从 blocks 移动到 currents 中时;

c)当有其他会话激活了与该会话相关的一个多会话约束时。

而会话由阻塞到运行的状态改变时间的计算时机如下:

a)当会话从 currents 移动到 blocks 中时;

b)当有一个与该会话共同受条件周期限制的其他会话中止或暂停时。

算法 3　系统会话调度算法

while(true){

if 一个会话正常结束,then 将该会话由 currents 移动到 olds 中

if 一个会话由于条件周期约束无法运行下去,then

将该会话由 currents 移动到 errors 中

if 一个会话首次激活一个 CP_C 时 then

计算该会话的状态改变时间并保存在该会话的 state_change_time 中

if 一个会话激活了一个 CP_C 或一个受某 CP_C 约束的会话中止或暂停时 then

对所有受该 CP_C 约束的会话,重新计算并保存这些会话的状态改变时间

if 一个 currents 中会话的 state_change_time 到达,then

将该会话从 currents 移动到 blocks 中,并计算、保存该会话的状态改变时间

if 一个 blocks 中的会话的 state_change_time 到达,then

将该会话从 blocks 移动到 currents 中,并计算、保存该会话的状态改变时间

}

3.3.2　状态恢复策略

在引入条件周期时间特性之前,RBAC 可在进行某个操作之前进行判断,当一致性不被满足时,操作就不被允许。在 CP_RBAC 中,是 currenttime 的递增操作引起了约束规则的状态变化。当系统发现 currenttime 越过了某条件周期约束的状态变化时间,即某条件周期约束不满足时,系统必须进行状态恢复。

3.3.3　确保安全状态策略

确保安全状态则不允许系统进入不安全状态,其基本思路为:找到状态变化时间 sct 之前的某个时间 t,满足 $t \leqslant sct \cap (sct - t) \leqslant Dt$,通过适当地选择一个很小的 Dt,可以确保当下次计算 currenttime 时,必然有 currenttime > sct,则在时间 t 预先将可能引起系统进入不安全状态的会话中止。相对而言确保安全状态实现简单。但无论怎样选择 Dt,都存在将原本可以完成的会话中止或暂停的可能性。

4　结束语

本文在 RBAC 模型约束的基础上提出了一种条件周期特性用户激活约束,以满足在实际应用中的授权管理的需求,同时对系统一致性状态的维护进行了讨论与分析。角色访问控制本身具有很多特性,如何正确地刻画这些性质,以及扩充现有模型或提出新模型以支持多种特性,在此方面仍有很多工作需要进一步研究。

参考文献:

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

[2] FERRAIOLO D F, KUHN D R, CHANDRAMOULI R. Role-based access control[M]. London: Artech House, 2003. (下转第 3052 页)

入侵或留下入侵证据。如果属于正常行为,则系统继续对用户行为进行监测。

3　仿真实验及结果分析

为评价上面提出的基于蚁群聚类的入侵检测算法,选用 KDD-99 数据集。该数据集由麻省理工学院 Lincoln 实验室仿真美国空军局域网环境建立的网络测试数据集,包含约 500 万条网络连接记录。

从 KDD-99 数据集中选取记录总数为 50、100、500、800 和 1 000 条的五个数据子集,每个子集中正常实例与异常实例之比均为 19:1,且使异常连接的种类尽可能平均,这有利于反映整个网络真实环境。针对以上测试数据集,选用 Intel Celeron 1.7 GHz CPU, 1 GB DDR2 内存,Windows XP 操作系统、MATLAB 6.5 语言编程环境,评价蚁群聚类算法和 K-means 聚类算法的两个性能指标,即检测率和误检率。检测率为检测到的入侵实例数目与检测数据子集中总的入侵实例数目之比。误检率为被误判为入侵的正常实例数目与总的正常实例数目之比。基于 KDD-99 中每个连接实例包含 42 个属性而且均已标志为正常或特定攻击类型,所以在进行聚类算法结果分析时,考虑类标志属性,得到图 2 和 3。图 2 为用蚁群聚类算法和 K-means 聚类算法测试检测率的情况,图 3 为用蚁群聚类算法和 K-means 聚类算法测试误检率的情况。

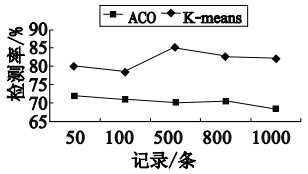

图2　检测率比较图

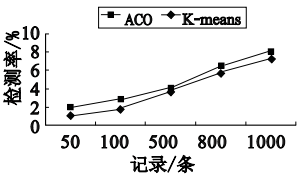

图3　误检率比较图

从图 2 可以看出,在 5% 的入侵实例比例下,蚁群聚类算法对于入侵的检测率较高,且实例数目的多少对检测率的影响不大。K-means 聚类算法的检测率偏低,是因为 K-means 聚类算法对于模式样本中个数差异较大的情况,特别是簇之间的距离不是很明显时,聚类效果较差。图 3 为蚁群聚类算法和 K-means聚类算法误检率比较图。由于一些正常实例会形成一些很小的簇,或者有的正常实例被划分到入侵实例组成的簇中,这些正常实例被误认为是入侵实例,造成误报。同时,通过实验发现随着实例总数的增加,误报率会变大,这主要是小型簇增加的结果[10]。K-means 聚类算法误检率偏大是因为模式样本出现少量的噪声和孤立点数据的情况时,K-means 聚类性能会下降,也就是 K-means 聚类算法对孤立点比较敏感,少量的该类数据能够对平均值产生较大的影响。

4　结束语

针对现有的入侵检测对未知攻击检测率和误检率方面的不足,利用蚁群聚类对未标记数据实例进行聚类,就可以判断哪些行为是合法的,哪些行为是非法的,即可以在不具备完整领域知识背景的情况下执行入侵检测功能。与其他聚类算法相比,蚁群聚类算法不需要预先知道聚类的数目,非常适合于无监督聚类的异常入侵检测。实验表明,该算法的检测率和误报率较 K-means聚类算法有明显改善,可以较好地应用于入侵检测。但由于本文所研究的是基于无监督的异常入侵检测,至于算法是否可以运用于其他种类的入侵检测,以及如何对算法作出改进以适合其他类型的入侵检测,将是下一步研究的内容。

参考文献:

[1] 马晓春,高翔,高德远. 聚类分析在入侵检测系统中的应用研究[J]. 微电子与计算机,2005,22(4):134-136.

[2] 唐正军,李建华. 入侵检测技术[M]. 北京:清华大学出版社,2004.

[3] BARBARA D, COUTO J, JAJODIA S, *et al*. ADAM: a tested for exploring the use of data mining in intrusion detection[J]. SIGMOD,2001,30(4):15-24.

[4] PORTNOY L, ESKIN E, STOLFO S J. Intrusion detection with unlabeled data using clustering[C]//Proc of ACM CSS Workshop on Data Mining Applied to Security. 2001.

[5] HUANG Xiao-fang, YANG Yi-xian, NIU Xin-xin. Towards improving ant-based clustering an chaotic ant clustering algorithm[C]//Proc of International Conference on Computational Intelligence and Security Workshops. 2007:421-424.

[6] 徐晓华,陈崚. 一种自适应的蚂蚁聚类算法[J]. 软件学报,2006,17(9):1554-1559.

[7] 张群,张利敏. 蚁群聚类组合算法的研究[J]. 武汉科技大学学报:自然科学版,2007,30(1):83-86.

[8] 吴文丽,刘玉树,赵基海. 一种新的混合聚类算法[J]. 系统仿真学报,2007,19(1):16-18.

[9] 肖立中,邵志清,钱夕元. 一种用于网络入侵检测的杂交聚类算法研究[J]. 计算机工程,2007,33(4):125-127.

[10] 余研,黄皓. 基于改进多目标遗传算法的入侵检测集成方法[J]. 软件学报,2007,18(6):1369-1378.

(上接第 3037 页)

[3] NA S Y, CHEON S. Role delegation in role-based access control[C]//Proc of the 5th ACM Workshop on Role-based Access Control. New York: ACM Press,2000:39-44.

[4] LUPU E C, MARRIOTT D A, SLOMAN M S, *et al*. A policy based role framework for access control[C]//Proc of the 1st ACM Workshop on Role-based Access Control. Gaithersbrug, Maryland: ACM Press,1996:28-29.

[5] CRAMPTON J. Specifying and enforcing constraints in role-based access control[C]//Proc of the 8th ACM Workshop on Role-based Access Control. Como: ACM Press,2003:43-50.

[6] 袁中兰,夏光升,李小标,等. RBAC 中的用户激活约束研究[C]//全国网络与信息安全技术研讨会. 2007:447-452.

[7] 黄建,卿斯汉,温红子. 带时间特性的角色访问控制[J]. 软件学报,2003,14(11): 1944-1954.

[8] BERTINO E, BETTINI C, FERRARI E, *et al*. An access control model supporting periodicity constraints and temporal reasoning[J]. ACM Trans on Database Systems, 1998,23(3):231-285.

[9] BERTINO E, BONATTI P A, FERRARI E. A temporal role-based access control model[J]. ACM Trans on Information and System Security,2001,4(3):191-223.

[10] 夏启寿,范训礼,殷晓玲. 基于时间 RBAC 授权模型的研究[J]. 西北大学学报,2008,38(6): 932-936.

[11] 欧阳凯,周敬利,董理君. RBAC 模型中条件时态的研究与设计[J]. 计算机科学, 2007,34(3):283-285.

[12] 夏启寿,殷晓玲,黄海生,等. 周期时间特性的角色访问控制[J]. 计算机应用研究,2009,26(12):4730-4734.