第36卷 第18期 计 算 机 工 程 2010年9月
Vol.36 No.18 Computer Engineering September 2010
·安全技术· 文章编号：1000—3428(2010)18—0135—03 文献标识码：A 中图分类号：TP393.08

RBAC策略冲突及其检测算法的研究

程相然，陈性元，张 斌，杨 艳
(解放军信息工程大学电子技术学院，郑州 450004)

摘 要：针对RBAC模型在实施职责分离、最小特权等安全原则时引起的冲突问题，形式化定义5种RBAC策略冲突类型，分析策略冲突产生的原因，提出一种完整的策略冲突检测算法并进行仿真测试。结果表明，该算法能够有效检测定义的各类策略冲突，为RBAC策略冲突检测实施提供基础。

关键词：RBAC策略；策略冲突；检测算法

Research on RBAC Policy Conflict and Its Detection Algorithm

CHENG Xiang-ran, CHEN Xing-yuan, ZHANG Bin, YANG Yan
(Electric Technology Institute, PLA Information Engineering University, Zhengzhou 450004, China)

【Abstract】With respect to conflict problems raised when implementing security principals such as separation of duty, least privilege in RBAC, this paper formalizes five RBAC policy conflict types, discusses causing reasons, and proposes a conflict detecting algorithm as well as simulation results, which can effectively detect conflicts defined in this paper. The work in this paper provides the basis for implementation for RBAC conflict detection.

【Key words】RBAC policy; policy conflict; detection algorithm

1 概述

基于角色的访问控制(Role-based Access Control, RBAC)[1-3]在实施最小特权、职责分离等安全原则时可能引起策略冲突。目前，针对RBAC模型的策略冲突研究集中在冲突约束的描述方法上[4-6]，冲突类型定义尚不完备，且缺乏对冲突检测算法的深入研究。本文对RBAC安全约束和策略冲突进行了形式化定义，设计了策略冲突检测算法并进行了仿真测试。

2 RBAC模型及其策略冲突

2.1 RBAC模型

参考标准RBAC模型[1-3]，基本元素定义如下：

(1) $USERS, ROLES, OPS, OBS, SESSIONS$ 分别表示用户、角色、操作、客体、会话的集合。

(2) $PRMS = 2^{(OPS\times OBS)}$ 为权限集合。

(3) $UA \subseteq USES \times ROLES$ 为用户到角色之间多对多的映射关系。

(4) $PA \subseteq PRMS \times ROLES$ 为权限和角色之间多对多的分配关系。

(5) $assigned\_users : (r : ROLES) \rightarrow 2^{USERS}$，给定角色 r 到用户集合 $USERS$ 的映射关系，返回分配给角色 r 的用户集合，形式化表示为：

$$assigned\_users(r) = \{u \in USERS \mid (u, r) \in UA\}$$

(6) $assigned\_permissions(r: ROLES) \rightarrow 2^{PRMS}$，角色到权限集合的映射关系，形式化表示为：

$$assigned\_permissions(r) = \{p \in PRMS \mid (p, r) \in PA\}$$

(7) $session\_users(s : SESSIONS) \rightarrow USERS$，会话 s 到相应用户的映射关系。

(8) $session\_roles(s : SESSIONS) \rightarrow 2^{ROLES}$，会话 s 到角色的映射关系，形式化表示为：

$$session\_roles(s : SESSIONS) \subseteq \{r \in ROLES \mid (session\_users(s), r) \in UA\}$$

(9) $RH \in ROLES \times ROLES$，为角色上的偏序关系，称为继承关系，记为 $\succeq$。$r_1 \succeq r_2$ 当且仅当 r_2 的权限均为 r_1 的权限，r_1 的用户均为 r_2 的用户，角色继承具有自反、传递和反对称性。RH 通常为直接角色继承关系的集合，根据传递性和自反性可计算出完整的继承关系集合 RH^+。

(10) $authorized\_users(r: ROLES) \rightarrow 2^{USERS}$，在角色继承下，角色 r 到用户集合的映射，形式化表示为：

$$authorized\_users(r) = \{u \in USERS \mid r' \succeq r,\ (u, r') \in UA\}$$

(11) $authorized\_roles(u : USERS) \rightarrow 2^{ROLES}$，在角色继承下，用户 u 到角色集合的映射，形式化表示为：

$$authorized\_roles(u) = \{r \in ROLES \mid r' \succeq r,\ (u, r') \in UA\}$$

(12) $authorized\_persmissions(r : ROLES) \rightarrow 2^{PRMS}$，在角色继承下，角色 r 到权限集合的映射，形式化表示为：

$$authorized\_permissions(r) = \{p \in PRMS \mid r \succeq r', (p, r') \in PA\}$$

(13) $actived\_roles(s) : SESSIONS \rightarrow 2^{ROLES}$，表示用户在会话中活动角色的集合，在角色继承下，当角色在会话中被激活时，其下层角色自动被激活成为活动角色。

(14) $senior\_roles(r : ROLES) \rightarrow 2^{ROLES}$，在角色继承下，$r$ 到其上层角色集的映射，形式化表示为：

$$senior\_roles(r) = \{r' \in ROLES \mid r' \succeq r\}$$

基金项目：国家“863”计划基金资助项目(2006AA01Z457, 2009AA01Z438)

作者简介：程相然(1984－)，男，硕士研究生，主研方向：网络安全；陈性元，教授、博士、博士生导师；张 斌，副教授、博士；杨 艳，讲师、硕士

收稿日期：2010-03-30 **E-mail：**chengxr0916@yahoo.com.cn

(15) $junior\_roles(r:ROLES)\rightarrow 2^{ROLES}$，在角色继承下，$r$ 到其下层角色集的映射，形式化表示为：

$$junior\_roles(r)=\{r'\in ROLES \mid r\succeq r'\}$$

2.2 RBAC 策略冲突类型

目前，RBAC 安全约束和冲突类型的定义尚不完备，本文在文献[1-6]的研究基础上扩展提出 5 类冲突，定义如下：

定义 1(冲突权限约束、权限冲突 $PRMS\_CF$) 冲突权限约束 $cp=(ps,n)$，其中，$ps=\{prms_1,prms_2,\cdots,prms_{|ps|}\}\subseteq PRMS$，$|ps|\geq n\geq 2$，表示最多只能将 ps 中 $n-1$ 个权限赋予同一角色。当 $n=2$ 时，ps 中权限两两互斥。冲突权限约束集合记为 $CP=\{cp_1,cp_2,\cdots,cp_{|CP|}\}$。

当同一角色直接或间接地获得了 ps 中多于 $n-1$ 个权限，就发生了权限冲突。

定义 2(静态冲突角色约束、静态角色冲突 SR_CF) 静态冲突角色约束 $scr=(rs,n)$，其中，$cr=(rs,n),\ rs=\{r_1,r_2,\cdots,r_{|rs|}\}\subseteq ROLES,\ |rs|\geq n\geq 2$，一个用户最多只能拥有 rs 中 $n-1$ 个角色。记 $SCR=\{scr_1,scr_2,\cdots,scr_{|SCR|}\}$ 为静态冲突角色约束集。

对于静态冲突角色约束 $scr=(rs,n)$，当 rs 中多于 $n-1$ 个角色直接或间接地赋予同一用户时，就引发了静态角色冲突。

由静态冲突角色约束的定义可知给定静态冲突角色约束 $scr=(rs,n)$，任一角色 r 最多继承 rs 中 $n-1$ 个角色。由于角色继承具有自反性，当 $r\in rs$ 时，r 最多继承 rs 中的 $n-2$ 个角色。

定义 3(动态冲突角色约束、动态角色冲突 DR_CF) 动态冲突角色约束 $dcr=(rs,n)$，其中，$rs=\{r_1,r_2,\cdots,r_{|rs|}\}\subseteq ROLES,\ |rs|\geq n\geq 2$，表示同一会话最多激活 rs 中 $n-1$ 个角色。当 $n=2$ 时，rs 中角色两两动态互斥。动态冲突角色约束集合记为 $DCR=\{dcr_1,dcr_2,\cdots,dcr_{|DCR|}\}$。

对于动态冲突角色约束 $dcr=(rs,n)$，当 rs 中不少于 n 个角色被同一会话所激活时，就产生了动态角色冲突。

定义 4(冲突用户约束、用户冲突 $USER\_CF$) 冲突用户约束 $cu=(us,n,r)$，其中，$us=\{u_1,u_2,\cdots,u_{|us|}\}\subseteq USERS, r\in ROLES,\ |us|\geq n\geq 2$，表示角色 r 最多只能分配给集合 us 中的 $n-1$ 个用户。当 $us=USERS$ 时，冲突用户约束也可用来表示基数约束[5]。冲突用户集的集合记为：

$$CU=\{cu_1,cu_2,\cdots,cu_{|CU|}\}$$

对于任意冲突用户集 $cu=(us,n,r)$，当 us 中大于 $n-1$ 个用户获得角色 r 时，就发生了用户冲突。

定义 5(角色继承路径、角色继承环路冲突 RIC_CF) 根据继承的传递性，角色继承路径 $r_i\succeq r_{k,1}\succeq r_{k,2}\succeq\cdots\succeq r_{k,n}\succeq r_j\ (n\geq 1)$ 表示角色 r_i 到角色 r_j 的传递继承关系。若存在角色继承路径 $r\succeq r_{k,1}\succeq r_{k,2}\succeq\cdots\succeq r_{k,n}\succeq r$，则产生角色继承环路冲突。

定义 6(系统状态 $status$) RBAC 基本元素集和安全约束集称之为系统状态，形式化表示为

$$status=<USERS,ROLES,PRMS,SESSIONS,UA,PA,RH,CU,CP,SCR,DCR>$$

3 策略冲突检测

在 RBAC 中，及时检测并消除策略冲突问题是实施各种安全原则的保证。而冲突检测的时机一般发生在导致冲突发生的操作时，这样可以让管理员及时消除冲突，提高系统的安全性。

3.1 策略冲突产生

策略冲突一般发生在对 RBAC 基本元素或其关系进行管理操作时，以下 4 条常用操作会引起冲突的发生。

(1) 用户分配，格式为 $AssignUser(user\in USERS, role\in ROLES)$，将角色 $role$ 分配给 $user$。

(2) 权限分配，格式为 $GrantPermission(object\in OBS, operation\in OPS, role\in ROLES)$，该操作将 $object$ 的 $operation$ 权限分配给 $role$。

(3) 增加角色继承，格式为 $AddInheritance(r\_asc\in ROLES,\ r\_desc\in ROLES)$，该操作建立直接角色继承关系 $r\_asc\succeq r\_desc$，称 r_asc 为高级角色，r_desc 为低级角色。

(4) 角色激活，格式为 $AddActiveRole(user\in USERS, session\in SESSIONS, role\in ROLES)$，该操作将角色 $role$ 设置为用户 $user$ 在会话 $session$ 的活动角色。

图 1 列出了以上操作可能引起的冲突类型及实例，其中，$cp=(\{p_x,p_y\},2)$；$cu=(\{u_x,u_y\},r_z,2)$；$scr=(\{r_x,r_y\},2)$；$dcr=(\{r_{x'},r_{y'}\},2)$。

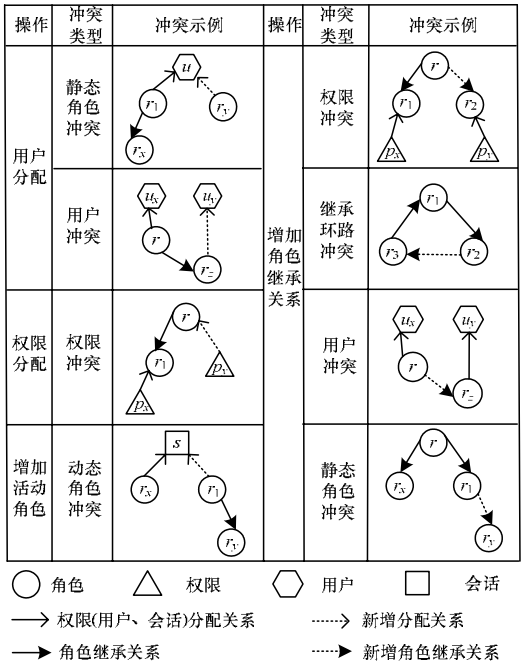

图 1 引起冲突的操作与相应冲突类型及实例

3.2 策略冲突检测算法

完整的策略冲突检测算法如下：

算法 RBAC 策略冲突检测算法

输入 *status*//当前系统状态

输出 *resultSet*//若存在冲突，返回类型 *type*、元素 *element*、约束 *constraint* 构成的集合，否则为 NULL

```
1 begin
2 resultSet=NULL
3 for each r∈ROLES
4   Pt=authorized_permissions(r)
5     for each cp∈CP
6 if | Pt∩cp.ps|≥cp.n //发生权限冲突
7 add (PRMS_CF, r, cp) to resultSet
8 for each cu∈CU
9     Ut= authorized_users(cu.r)
10       if | Ut |≥cu.n //发生用户冲突
11         add (USER_CF, Ut, cu) to resultSet
```

```
12 for each scr∈SCR
13   { for each u∈USERS
14        Rt = authorized_roles(u)
15        if | Rt ∩scr.rs |≥scr.n //因用户分配发生静态角色冲突
16           add (SR_CF, u, scr) to resultSet
17     for each r∈ROLES
18        Rt = junior_roles(r)
19        if | Rt ∩scr.rs |≥scr.n //因角色继承引发静态角色冲突
20              add (SR_CF, r, scr) to resultSet
21   }
22 for each dcr∈DCR
23    for each s∈ SESSIONS
24        Rt = actived_roles(s)
25        if | Rt ∩dcr.rs |≥dcr.n //发生动态角色冲突
26           add (DR_CF, s, dcr) to resultSet
27 for each ri∈ ROLES
28    for each rj∈ ROLES
29       if (ri, rj) ∈ RH
30          M (ri , rj ) =1 //构建直接角色继承矩阵
31       else M (ri, rj) =0
32 M+ =Warshall(M) //计算 M 矩阵的传递闭包
33 R'=NULL //初始化环路继承冲突角色集合
34 for each r∈ ROLES
35   if M (r, r) ==1
36      R' = R'∪{r}
37 if R'! =NULL //发生角色继承环路冲突
38   add (RIC_CF, R', NULL) to resultSet
39 return resultSet
40 end
```

算法的输入为系统的当前状态 *status*，输出为若干冲突类型 *type*、发生冲突的元素 *element*、违反的冲突约束 *constraint* 构成的集合 *resultSet*。

其中，第 3 行~第 7 行检测是否存在权限冲突；第 8 行~第 11 行检测是否发生用户冲突；第 12 行~第 21 行检测是否发生静态角色冲突；第 22 行~第 26 行检测是否发生动态角色冲突；第 27 行~第 38 行检测是否发生角色环路继承冲突，其中，第 32 行调用 *Warshall* 算法[6]计算矩阵的传递闭包。冲突检测的时机通常是在可能引发冲突的操作发生时，若检测到该操作导致冲突发生，则采取回滚或其他操作加以消除，例如发生权限冲突时，可以通过回滚或撤销角色已有权限来消除。

4 仿真测试

某办公自动化环境用户个数为 51，细分为 126 个权限，设置有 10 个角色，其层次结构如图 2 所示。由于动态角色冲突、用户冲突等与实际应用密切相关，该实验主要测试权限冲突与静态角色冲突的发生情况。

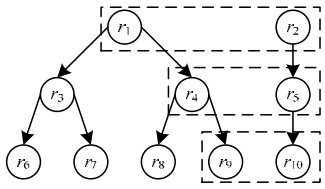

图 2　测试环境角色层次图

图 3 中的实验用来分析权限分配时引起冲突的情况。分别构建了冲突权限约束数目为 1、2、3、5、10、20、30、40、50 共 9 种情况，针对每种情况分别测试了随机进行 100、200、300、500、700 次权限分配时权限冲突发生的平均概率。可以看出，随着权限分配次数的增多，角色自身获得的权限越来越多，执行权限分配操作造成冲突的概率逐渐增大；权限分配次数相同时，冲突约束数量的增多也会导致冲突发生概率的增大。

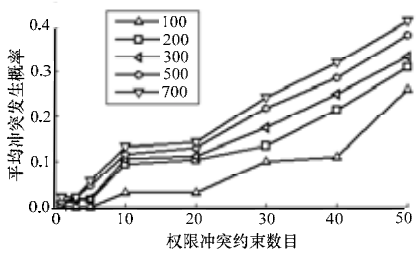

图 3　权限冲突发生概率与冲突约束的数目关系

图 4 中的实验分析为用户分配角色时静态角色冲突发生的情况。分别构建了 3 种静态冲突角色约束的情况(图 2 中虚线框所示)，根据角色继承的传递性可知，其实际静态冲突角色约束数目分别为 1、4、9。分别测试了 3 种情况下随机进行 200 次权限分配的平均引起静态角色冲突发生次数的变化情况。可见，随着用户分配次数的增多，用户拥有的授权角色越来越多，在静态冲突角色约束数目一定的情况下，造成冲突的总数快速增加；同时，在用户分配次数相同的情况下，静态冲突角色约束数目增加也会导致用户分配时冲突发生次数显著增多。

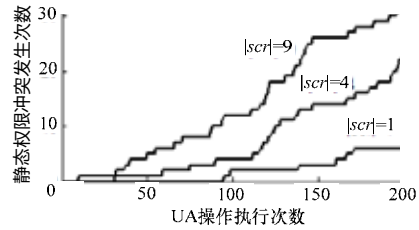

图 4　静态角色冲突发生次数与用户分配总数关系

5 结束语

本文研究了 RBAC 策略冲突及其检测问题，在对 RBAC 模型各种安全约束形式化描述的基础上定义了 5 种冲突类型。设计了冲突检测算法并进行了仿真测试分析，为 RBAC 系统冲突检测实施提供了基础。

参考文献

[1] Sandhu R, Coyne E, Feinstein H, et a1. Role-based Access Control Model[J]. IEEE Computer, 1996, 29(2): 38-47.

[2] Ferraiolo D, Sandhu R, Gavfila S, et al. Proposed NIST Standard for Role-based Access Control: Towards a Unified Standard[J]. ACM Trans. on Information and System Security, 2001, 4(3): 224-274.

[3] ANSI, INCITS. ANSI/INCITS 359-2004 Information Technology——Role Based Access Control[S]. 2004.

[4] Ahn G J, Sandhu R. Role-based Authorization Constraints Specification[J]. ACM Trans. on Information and System Security, 2000, 3(4): 207-226.

[5] 袁春阳, 贺也平, 何建波, 等. 具有冲突约束的 RBAC 模型的形式化规范与证明[J]. 计算机研究与发展, 2006, 43(增刊): 498-508.

[6] 吴　迪, 朱淼良, 陈溪源, 等. 分布式环境下基于 RBAC 互操作的安全检测[J]. 浙江大学学报: 工学版, 2007, 41(9): 1552-1556.

编辑　任吉慧