文章编号:1001 - 9081(2008)04 - 0924 - 03

支持动态角色切换的 RBAC 模型

陈娟娟,程西军
(海军工程大学 计算机工程系,武汉 430033)
(montling@ yahoo. com. cn)

摘 要:动态角色切换是信息系统依据用户属性改变而部分或整体改变用户 - 角色指派的一种自动授权机制。将动态角色切换引入到 RBAC96 模型,论述了动态角色切换的各种形态、不同切换间的相互关系及模型实现。基于动态角色切换,系统可以自动处理触发角色切换条件而引起的用户 - 角色指派变更问题,整个过程无须人工参与,减轻了系统管理员的工作负担,提高了授权管理的效率与安全性。

关键词:角色切换;基于角色的访问控制;职责分离
中图分类号: TP309.2 **文献标志码**:A

Extended RBAC model supporting dynamic role switching

CHEN Juan-juan, CHENG Xi-jun
(*Department of Computer Engineering, Naval University of Engineering, Wuhan Hubei* 430033, *China*)

Abstract: Dynamic role switching is a kind of automatic authorization mechanism that the information system partly or wholly changes the user-role assignments based on the changes of user's attributes. The dynamic role switching was introduced into RBAC96 model. Different forms of the dynamic role switching, the relationships between the different switchings, and the realization of the extended model were presented. Based on the dynamic role switching, the information system can deal automatically with the situation of the user-role assignments triggered by the role switching condition. There is no need for the human being to participate in the role switching process. Therefor, dynamic role switching can lessen the work load of the system administrator and improve the efficiency and security of the authorization administration.

Key words: role switching; Role Based Access Control (RBAC); separation of duty

0 引言

在传统的基于角色的访问控制(RBAC)中,角色 - 权限指派(Permission Assignment, PA)和用户 - 角色指派(User Assignment, UA)均由系统管理员人工进行操作[1-3]。角色的权限根据相应工作岗位的职责而设置,比较稳定,而用户可能由于职务调动、自身变化等因素而经常更换角色,因此 UA 相对 PA 而言更具动态变化性。在下述情况中,UA 的变更尤为频繁:系统的用户具有一系列的用户属性,如商场会员卡系统中的用户具有 Name(姓名)、Sex(性别)、Birthday(出生日期)、Phone(联系电话)、Addr(通信地址)、ConsumeSum(消费金额)、NoLoginTime(未登录本系统的时间)等属性。这些属性既有固定不变的(如 Name),也有不断变化的(如 ConsumeSum)。用户的某些属性决定了用户在系统中担任的角色,如 0 < ConsumeSum < 5 000 yuan 时,用户被指派给角色 Customer;当 ConsumeSum ≥ 5 000 yuan 时,用户就应被指派给角色 VIP;当 NoLoginTime ≥ 730 days 时,原先具有角色 Customer 或 VIP 的用户就被重新指派给角色 Guest。

这种由用户属性的变化而引起的 UA 变更就是角色切换。角色切换如果仅由系统管理员人工进行,则会出现以下问题:

1)管理员的任务繁重,尤其是大型系统中管理员需要处理大量类似的 UA 变更;

2)授权管理的效率低下;

3)管理员可能有意或无意授予用户级别更高的角色,这将违背最小特权原则。

对于上述问题,Chou 在文献[4]中提出由计算机程序执行角色切换,这就是动态角色切换,但并没有进一步阐述如何进行切换。本文对 RBAC96 模型[1]进行了细化和扩展,将动态角色切换引入到用户 - 角色指派 UA 中,系统根据角色切换条件和系统安全策略自动处理由用户属性的变化而引起的 UA 变更。与人工操作相比,动态角色切换减轻了系统管理员的工作负担,提高了授权管理的效率与安全性。

1 模型描述

1.1 基本定义

定义 1 RBAC 状态。RBAC 状态可由一个三元组表示:State = (UA, PA, RH)。RBAC 基本元素 U、R、P 之间的相互关系 $UA \subseteq U \times R$、$PA \subseteq R \times P$、$RH \subseteq R \times R$ 共同决定了 RBAC 的一种状态;UA、PA、RH 三者只要其一发生改变,则 RBAC 状态随之改变。

定义 2 角色切换。假设角色 $r_1 \neq r_2$,对于 $(\forall u)$ $((u, r_1) \in UA \wedge (u, r_2) \notin UA)$,当 u 本身某种属性发生变化使得 $(u, r_1) \notin UA \wedge (u, r_2) \in UA$ 且 u 其他的角色未发生改变,则称角色 r_1 切换到 r_2,记为 $r_1 \rightarrow r_2$。r_1 是切换的源角色,r_2 是切换的目的角色。

在 UA、PA、RH 这三种关系中,UA 最易发生改变,PA 与 RH 则相对稳定。由于角色切换使一个用户从一种角色的成

收稿日期:2007 - 10 - 26。

作者简介:陈娟娟(1978 -),女,湖北枣阳人,讲师,硕士,主要研究方向:信息安全; 程西军(1973 -),男,安徽六安人,讲师,博士研究生,主要研究方向:信息安全、计算机测控。

员转变为另一种角色的成员,因此有:

定理 1　角色切换必然引起 RBAC 状态的改变。

定理 2　从 r_1 切换到 r_2 时,r_1 不能处于激活状态。

角色切换不同于角色激活:角色切换是用户获取新角色的一种途径,而角色激活则是事先已给用户指派好角色,再根据不同条件激活不同角色。

定义 3　角色切换条件。角色切换条件为 SwitchCon:$R \times R \rightarrow$Boolean。

SwitchCon(r_1, r_2) =

$$\begin{cases} \text{TRUE}, & \text{从角色 } r_1 \text{ 切换到 } r_2 \text{ 的条件为真} \\ \text{FALSE}, & \text{从角色 } r_1 \text{ 切换到 } r_2 \text{ 的条件为假} \end{cases}$$

角色切换条件 SwitchCon(r_1, r_2)是关于角色 r_1 的成员用户的属性表达式,是支持动态角色切换的 RBAC 模型中的一个关键部分。它是进行角色切换之前进行的限制性检查,只有当 r_1 的用户满足切换条件时才能成为 r_2 的用户。SwitchCon 既可以基于事件,如消费总额达到一定数目;也可以基于时间,如登录时间超过一定期限。一旦 SwitchCon 得到满足,由系统自动进行用户的角色切换,该方式是一种自触发过程。

例如,SwitchCon(Customer, VIP): ConsumeSum ≥ 5 000 yuan,对于 $\forall u \{u \mid (u, \text{Customer}) \in UA\}$,则有 SwitchCon(Customer, VIP) = TRUE⇒Customer→VIP。

定理 3　角色切换关系具有传递性。$\forall r_1, r_2, r_3 \in R$, $(r_1 \rightarrow r_2) \wedge (r_2 \rightarrow r_3) \Rightarrow r_1 \rightarrow r_3$,且 SwitchCon($r_1$, r_3) = SwitchCon(r_1, r_2) && SwitchCon(r_2, r_3)。

逻辑符"&&"的短路性质保证了先对 $r_1 \rightarrow r_2$ 的条件进行判断,仅当该条件为真时才接着对 $r_2 \rightarrow r_3$ 的条件进行判断。

不难证明,角色切换关系不具有自反性、对称性和反对称性。

定义 4　条件优先级。设 $r_1 \rightarrow r_2$ 存在两种不同的切换条件 SwitchCon1(r_1, r_2)和 SwitchCon2(r_1, r_2),若 SwitchCon1(r_1, r_2) = TRUE⇒SwitchCon2(r_1, r_2) = TRUE 成立,则称 SwitchCon1(r_1, r_2)优先于 SwitchCon2(r_1, r_2),记为 SwitchCon1(r_1, r_2) ≥ SwitchCon2(r_1, r_2)。

例如 SwitchCon1(r_1, r_2): ConsumeSum ≥ 10 000, SwitchCon2(r_1, r_2): ConsumeSum ≥ 5 000,则 SwitchCon1(r_1, r_2) ≥ SwitchCon2(r_1, r_2)。

定义 5　角色切换链。角色切换链 $r_1 \rightarrow_c r_n$($n \in N$ 且 $n > 1$)是一个角色切换序列(r_1, r_2, …, r_{n-1}, r_n),表示依次发生角色切换 $r_1 \rightarrow r_2$、$r_2 \rightarrow r_3$、…、$r_{n-1} \rightarrow r_n$。

定义 6　角色切换链条件。角色切换链 $r_1 \rightarrow_c r_n$ 的切换条件 SwitchChainCon(r_1, r_n) = SwitchCon(r_1, r_2) && SwitchCon(r_2, r_3) && … && SwitchCon(r_{n-1}, r_n)。

定理 4　$r_1 \rightarrow r_n$ 与 $r_1 \rightarrow_c r_n$ 对 RBAC 状态的影响是不相同的。

设用户 u_1、u_2 的属性分别满足 SwitchCon(r_1, r_n)和 SwitchChainCon(r_1, r_n),u_1 在 $r_1 \rightarrow r_n$、u_2 在 $r_1 \rightarrow_c r_n$ 后,(u_1, r_n) ∈ UA 且(u_2, r_n) ∈ UA。但是 u_2 在 $r_1 \rightarrow_c r_n$ 的过程中,(u_2, r_2) ∈ UA、(u_2, r_3) ∈ UA、…、(u_2, r_{n-1}) ∈ UA、(u_2, r_n) ∈ UA 依次成立。故 $r_1 \rightarrow r_n$ 与 $r_1 \rightarrow c_{r_n}$ 对 RBAC 状态的影响是不相同的。

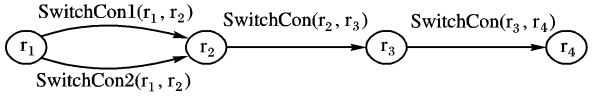


图 1　角色切换关系图

在图 1 中:

1)由角色切换的传递性可知:($r_1 \rightarrow r_2$) ∧ ($r_2 \rightarrow r_3$) ∧ ($r_3 \rightarrow r_4$) ⇒ $r_1 \rightarrow r_3$, $r_1 \rightarrow r_4$, $r_2 \rightarrow r_4$。

2)存在三个角色切换链:$r_1 \rightarrow_c r_3$、$r_1 \rightarrow_c r_4$、$r_2 \rightarrow_c r_4$,

SwitchChainCon(r_1, r_3) = SwitchCon(r_1, r_2) && SwitchCon(r_2, r_3)

SwitchChainCon(r_1, r_4) = SwitchCon(r_1, r_2) && SwitchCon(r_2, r_3) && SwitchCon(r_3, r_4)

SwitchChainCon(r_2, r_4) = SwitchCon(r_2, r_3) && SwitchCon(r_3, r_4)。

3)$r_1 \rightarrow r_2$ 存在两种不同的切换条件 SwitchCon1(r_1, r_2)和 SwitchCon2(r_1, r_2),若 SwitchCon1(r_1, r_2) ≥ SwitchCon2(r_1, r_2),则 SwitchCon1(r_1, r_2) = TRUE⇒SwitchCon2(r_1, r_2) = TRUE,反之不成立。

1.2　多重角色切换

角色切换 $r_1 \rightarrow r_2$ 的源角色 r_1 与目的角色 r_2 之间是 1∶1 的关系,事实上,r_1 与 r_2 之间还存在 1∶n、m∶1、m∶n 的多重切换关系。

定义 7　多重角色切换。假设 $R_S = \{r_{s1}, r_{s2}, \cdots, r_{sm}\}$ ($m \in N$), $RD = \{r_{d1}, r_{d2}, \cdots, r_{dn}\}$ ($n \in N$), $R_S \neq R_D$,对于 $(\forall u)$ $(\forall r_1 \in R_S)$ $(\forall r_2 \in R_D)$ $((u, r_1) \in UA \wedge (u, r_2) \notin UA)$,当 u 本身某种属性发生变化使得 $(u, r_1) \notin UA \wedge (u, r_2) \in UA$ 且 u 其他的角色未发生改变,则称角色集 R_S 切换到 R_D,记为 $R_S \rightarrow R_D$。R_S 是切换的源角色集,R_D 是切换的目的角色集。

由于 $|R_S| = m$、$|R_D| = n$,存在以下 4 种情况:1)m = n = 1;2)m = 1, n > 1;3)m > 1, n = 1;4)m > 1, n > 1。第 1 种情况属于单重切换,其余 3 种属于多重切换。

角色切换改变了用户 - 角色指派关系,并由此最终改变了指派给用户的访问控制权限。用户权限的变化分为 4 种情况:扩大、缩小、部分改变和全部改变。令:

$P_S = \{p \mid (p, r_{si}) \in PA\}$ $(1 \leqslant i \leqslant m)$

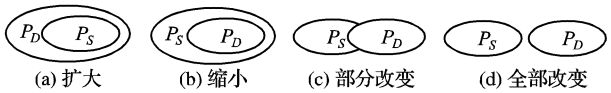
$P_D = \{p \mid (p, r_{di}) \in PA\}$ $(1 \leqslant i \leqslant n)$

1) $P_S \subset P_D$:$R_S \rightarrow R_D$ 后,用户 u 的权限范围得以扩大。尤其当 m = n = 1 时,$R_D \geqslant R_S$。

2) $P_D \subset P_S$:$R_S \rightarrow R_D$ 后,用户 u 的权限范围得以缩小。尤其当 m = n = 1 时,$R_S \geqslant R_D$。

3) $(P_S \not\subset P_D) \wedge (P_D \not\subset P_S) \wedge (P_S \not\subset P_D \neq \varnothing)$:$R_S \rightarrow R_D$ 后,用户 u 在 P_S 范围内的部分权限得以改变、部分权限保持不变。

4) $(P_S \not\subset P_D) \wedge (P_D \not\subset P_S) \wedge (P_S \cap P_D = \varnothing)$:$R_S \rightarrow R_D$ 后,用户 u 在 P_S 范围内的所有权限都发生变化。



(a) 扩大　(b) 缩小　(c) 部分改变　(d) 全部改变

图 2　角色切换后用户权限的改变

2　模型实现

2.1　职责分离安全策略

动态角色切换在实现时除了要根据角色切换条件自动处理 UA 以外,还需考虑系统的安全策略。

职责分离是一项基本的安全策略,通过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从而提高用户欺诈的风险性,故可以防止滥用权利并维护信息完整性。互斥角色是在 RBAC 中实现职责分离原则自然而有效的途径。一个用户通过拥有两个互不相同的角色而拥有足够权力破坏职责分离原则,进而威胁系统安全,则称这两个角色是互

斥角色。

文献[2,3]指出职责分离可以有静态和动态两种实现方式,分别称为静态职责分离(Static Separation of Duty,SSD)和动态职责分离(Dynamic Separation of Duty,DSD)。SSD和DSD都是通过指定角色互斥关系达到职责分离的目的,所不同的是SSD用于用户指派阶段,DSD用于角色激活阶段。SSD是指在把用户指派给角色时,如果一个用户已被指派了角色r_1,并且存在另外一个不同的角色r_2与r_1互斥,则此用户不能再被指派给角色r_2,记为$(r_1, r_2) \in SSD$。DSD是指用户可以被指派给互斥角色,但是此用户的任何会话在执行时却不能同时激活互斥角色,记为$(r_1, r_2) \in DSD$。一般而言,SSD实现简单、便于管理,但不够灵活,不能处理某些实际情况;DSD更灵活、适应实际管理需要,但实现复杂、不易管理。

定理5 $r_1 \rightarrow r_2 \Rightarrow (r_1, r_2) \notin SSD$, $r_1 \rightarrow r_2 \Rightarrow \{r \mid (u, r) \in UA\}\ (r_2, r) \notin SSD$。

假设$(r_1, r_2) \in SSD$,当用户u激活r_1执行相应权限之后$r_1 \rightarrow r_2$,则u又可激活r_2执行相应权限,这样u先后执行了互斥角色r_1、r_2的权限,从而有足够权力破坏职责分离原则,因此发生切换的两个角色不能是互斥的。同理,在系统自动进行角色切换时,需要注意指派给用户的新角色不能与他已拥有的角色互斥。

2.2 角色切换链表

用户自身属性发生变化引起相应的角色切换并导致UA变更,为了便于对UA关系进行管理及审计,本文采用链表记录每个用户的角色自动切换情况。链表节点的结构如图3所示。

u	r_i	r_j	SwitchCondition	Time	Next

图3 角色切换链表的节点结构

其中:r_i记录角色切换的源角色;r_j记录角色切换的目的角色;SwitchCondition记录角色切换条件;Time记录角色切换的发生时间;Next是指向下一个角色切换节点的指针。

系统中的每个用户都有一个角色切换链表记录该用户的角色切换历史。即使系统用户数量众多,由于该链表被访问的频率很低,可将链表直接存储在外存,因而并不占用系统宝贵的内存资源。

2.3 动态角色切换算法

由上可知,系统在进行动态角色切换时,除了满足切换条件以外,还需注意源角色的非激活状态以及系统安全策略。算法描述如下:

```
if (SwitchCon(r1, r2) == TRUE)   /* 判断是否满足切换条件 */
{
  if ((u, r1) ∉ UA)              /* 判断u是否为r1的成员用户 */
    return FALSE;
  if (r1 ISACTIVATEDROLES(u))
    /* 若u正在激活r1,则此时不可进行角色切换 */
    return FALSE;
  if ((u, r2) ∈ UA)
    /* 如果u已经是r2的成员用户,则不须进行角色切换 */
    return FALSE;
  for (r3 ∈ {r|(u, r) ∈ UA})
                          /* 判断r2是否与u已有角色互斥 */
    if ((r2, r3) ∈ SSD)
      return FALSE;
  deleteUA(u, r1);
        /* 撤消u与r1之间的指派关系,u不再是r1的成员 */
  grantUA(u, r2);                /* 将u指派给r2 */
  updateList(u, r1, r2, SwitchCon(r1, r2), CurrentTime);
  /* 更新u的角色切换链表 */
  return TRUE;
}
else
  return FALSE;
```

3 结语

将动态角色切换引入到用户-角色指派中,系统根据用户属性自动为之切换角色,授权管理的工作量得以降低、效率得以提高。需要注意的是,系统自动切换角色引起RBAC状态的改变可能与管理员手工进行的用户-角色指派之间产生某些冗余甚至是矛盾。因此,系统需要及时处理这些冗余和矛盾。

参考文献:

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

[2] FERRAIOLO D F, SANDHU R S, GAVRILA S, *et al*. Proprosed NIST standard for role-based access control[J]. ACM Transactions on Information and System Security, 2001,4(3):224-274.

[3] ANSI. American national standard for information technology-role based access control[S], 2004.

[4] CHOU S-C. An RBAC-based access control model for object-oriented systems offering dynamic aspect features[J]. IEICE Ttransactions on Information and Systems, 2005,88(9):2143-2147.

[5] KAPADIA A, AL-MUHTADI J, CAMPBELL R, *et al*. UIUCDCS-R-2000-2162, IRBAC 2000: Secure Interoperability Using Dynamic Role Translation[R]. University of Illinois, 2000.

[6] AL-MUHTADI J, KAPADIA A, CAMPBELL R, *et al*. UIUCDCS-R-2000-2163, The A-IRBAC 2000 Model: Administrative Interoperable Role-Based Access Control[R]. University of Illinois, 2000.

[7] AL-KAHTANI M, SANDHU R. A model for attribute-based user-role assignment[C]// Proceedings of the 18th Annual Computer Security Applications Conference. Washington, DC, USA: IEEE Computer Society, 2002:353-362.

[8] LI NING-HUI, TRIPUNITARA M V, BIZRI Z. On mutually exclusive roles and separation-of-duty[J]. ACM Transactions on Information and Systems Security, 2007,10(2):1-36.

[9] 廖俊国,洪帆,朱贤,等. 动态角色转换的关联优化[J]. 计算机工程与应用,2006,42(18):130-132.

(上接第923页)

[5] BRIASSOULI A, STRINTZIS M G. Locally optimum nonlinearities for DCT watermark detection[J]. IEEE Transactions on Image Processiong, 2004,13(12):1604-1617.

[6] 张辉,曹丽娜. 现代通信原理与技术[M]. 西安:西安电子科技大学出版社,2002.

[7] NIKOLAIDIS A, PITAS I. Asymptotically optimal detection for additive watermarking in the DCT and DWT domains[J]. IEEE Transactions on Image Processing, 2003,12(5):563-571.

[8] LIU X-Y, GAO KUN, CHEN WU-FAN. A blind watermarking optimal detection based on the wavelet transform domain[C]// Proceedings of the Sixth International Conference on Machine Learning and Cybernetics. Hong Kong: [s.n.], 2007:1779-1783.