

云环境下的用户-角色分配优化算法

王静宇, 陈 丽

(内蒙古科技大学 信息工程学院, 内蒙古自治区 包头 014010)

摘 要: 针对云环境下基于角色访问控制模型中用户角色指派不精确及效率低的问题, 设计了基于属性和信任的角色访问控制模型, 提出基于此模型的用户-角色分配优化算法, 该算法融合了最小角色集合查找与基于属性的角色筛选, 通过删除混杂角色层次结构中的角色激活和角色继承关系, 得到最小角色集合, 通过用户属性、环境属性、信任值筛选得到最佳角色集合, 并精确匹配给用户。仿真实验表明, 此算法能有效减少系统中的角色数, 缩短用户角色的指派时间, 提高用户-角色匹配的精确度。

关键词: 用户-角色指派; 角色筛选; 最佳角色集合; 访问控制

中图分类号: TP309

文献标识码: A

文章编号: 1000—7180(2018)08—0087—06

User-role Assignment Optimization Algorithm in Cloud Environment

WANG Jing-yu, CHEN Li

(School of Information Engineering, Inner Mongolia University of Science and Technology,
Baotou 014010, China)

Abstract: Aiming at the problem of imprecise and inefficient user role assignment in the role-based access control model in the cloud environment, the role-based access control model based on attribute and trust is designed. and A user-role assignment optimization algorithm based on this model is proposed, the algorithm combines the minimum role set searching and the attribute-based role filtering, by deleting the role activation in the hybrid role hierarchy and Role inheritance, get the smallest role set, through the user attributes, environmental attributes, trust value filter to get the best role set, and accurate match to the user. Simulation results show that this algorithm can effectively reduce the number of roles in the system, shorten the assignment time of user roles and improve the accuracy of user-role matching in the presence of a large number of user role assignments in the cloud environment.

Key words: user-role assignment; role filter; the optional role set; access control

1 引言

随着云计算的快速发展, 如何对开放式云环境进行安全有效的管理是当前云计算信息安全领域的一个研究热点^[1]. 访问控制机制^[2]可以限制对关键资源的访问, 防止非法用户进入系统及合法用户对系统资源的非法使用, 从而保护信息系统中存储和处理信息的安全^[3]. 目前多数的访问控制模型主要在传统的模型上进行改进和扩展^[4]. 例如, Qi H^[5]等人提出结合 RBAC 和 ABAC 的复合访问控制模

型, 此模型既有高效的管理机制又有细粒度的访问控制, 但其用户角色分配速度和效率较低. 熊厚仁^[6]等人提出了以属性为基础的角色权限缩减机制, 为基于属性的角色筛选提供了思路.

针对以上访问控制模型约束条件不足及现有用户角色分配算法存在的问题, 本文设计了基于属性和信任的角色访问控制模型(AT-RBAC), 并提出基于 AT-RBAC 模型的用户-角色分配优化算法(User-Role Assignment Optimization Algorithm, URAOA), 解决了用户角色匹配不精确的问题, 减

少分配给用户的角色数.

2 AT-RBAC 模型设计

本文在传统访问控制模型的基础上设计了 AT-RBAC 模型,如图 1 所示.

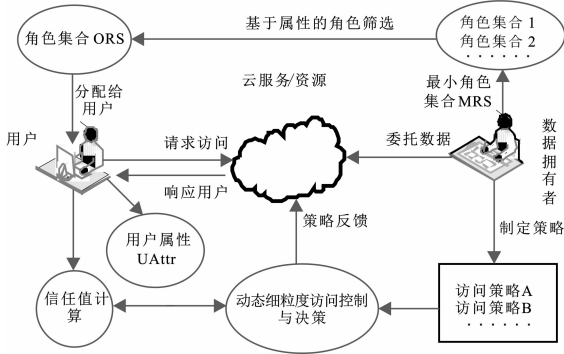


图 1 AT-RBAC 模型

AT-RBAC 模型的创新主要包括两个方面:最小角色集合的查找和最佳角色集合的筛选.当用户需要访问某一资源时,通过角色分配在数据拥有者处获取访问控制策略从而得到访问权限. AT-RBAC 模型在保留标准 RBAC 模型高效管理的基础上,弥补了用户数量增多时“角色爆炸”的缺点,并且动态全面的判定标准更加适合现有的云环境.

2.1 相关定义

定义 1: U, O, OP, P, R, DO 分别代表用户、客体、授权、操作、角色、数据拥有者;

定义 2: $P = O \times OP$, 其中 O 是资源客体的集合, OP 是读、写、执行等访问;

$PRA^T = P \times T \times R$ 多对多的访问规则到角色指派关系;

$RA = R \times R$ 角色到角色的多到多指派关系;

$URA^T = U \times T \times R$ 用户到角色的多到多指派关系;

定义 3: 访问控制策

$POLICY = \langle U, R, DO, O, OP, P, T, URA^T, PRA^T, RA, UR \rangle$;

定义 4: 访问请求 (AReq): 访问者在用户端发送访问请求给数据拥有者, 请求记录为 AReq, $AReq = \langle UR, DO, request\ index, data\ block\ index, P \rangle$

其中 $request\ ndex$ 表示用户的访问请求, P 表示的请求操作, $data\ block\ index$ 表示用户请求访问的

数据块.

定义 5: 属性: att ATT 是具有指定数据类型和值域的变量, 各自表达属性所属实体, 包括用户 $user$ USER、资源 Res RES 和环境 s SESSION.

定义 6: 属性表达式: 是与属性条件有关的一个判断表达式, 其对属性做详尽描述或制约.

定义 7: 信任值 T 的计算由三部分组成, 直接、间接以及综合信任值计算. 其中综合信任值是直接和间接信任值的综合值.

2.2 AT-RBAC 模型访问过程

AT-RBAC 模型的访问过程下:

- 1) 用户申请访问资源时, 需要提交自己的基本信息;
- 2) 数据拥有者根据自身数据属性的特点, 查找最小角色集合 (Minimum Role Set, MRS);
- 3) 依据用户的基本信息, 环境因素等计算其信任值;
- 4) 依据用户的属性, 环境属性, 信任值筛选出最佳角色集合 (Optimal Role Set, ORS);
- 5) 把 ORS 分配给用户, 从而得到访问权限.

云环境中存在巨量的数据资源, 在授权时会生成很多角色, 保持访问者的角色指派关系需要耗费巨量的计算和存储资源. 针对这一问题第三章提出一种 URAOA 算法, 使得在云环境中能依据访问者的权限需求, 得出符合其访问资源权限的最佳角色集合.

3 基于 AT-RBAC 模型的 URAOA 算法

URAOA 算法融合了最小角色集合查找和基于属性的角色筛选两种方法, 最小角色集合查找主要是对混杂角色层次树中的角色进行组合, 使角色间不存在激活和继承关系. 基于属性的角色筛选算法主要对 MRS 进一步约束, 从而得到最佳角色集合 ORS.

3.1 最小角色集合查找

定义 8: MRS 集合 定义 $S = (R, F)$ 是 AT-RBAC 模型中的角色层次树. 其中, R 是角色集合, F 是节点间的关系集合. $MRS(S) = \{R_1, R_2, R_3, \dots, R_m\}$, 其中, $\emptyset \in Ri \subseteq R, i \in \{1, 2, 3, \dots, m\}$ 在 S 的所有角色集合里. R_i 是最小且唯一的角色组合, 当且仅当: $\forall R_i, R_j \in MRS(S), PS(R_i) \neq$

$PS(R_j)$, 其中 $i, j \in Z, i \neq j$;

2) 对于 $\forall Z \subseteq R_s, tZ \notin MRS(S)$, 若存在 $R' \in MRS(S), PS(R') = PS(Z)$, 则有 $|R'| < |Z|$ 成立; 其中, $PS(R)$ 表示角色集合 R 的权限, $|R'|$ 是 R' 中元素的个数.

根节点存在于角色层次树中. 定义运算符如下:

$$N_1 \otimes N_2 = \{\{x_1 \cup x_2\} \mid x_1 \in N_1, x_2 \in N_2\}$$

$$N_1 \otimes N_2 \otimes \dots \otimes N_m = \{\{x_1 \cup x_2 \cup \dots \cup x_m\} \mid x_1 \in N_1, x_2 \in N_2, \dots, x_m \in N_m\}$$

假设 $M = \{N_1, N_2, \dots, N_m\}$ 有

$$: \$M = N_1 \otimes N_2 \otimes \dots \otimes N_m$$

算法 1: Get_MRS 获取最小角色集合 */

Read : S /* 输入一个混合角色层次树 */

Output : MRS /* 输出混合角色层次树 S 的所有最小角色集合 MRS */

Searchtemp $\leftarrow \emptyset$ /* 初始化所有被搜寻到的临时角色集 Searchtemp 为空 */

Subtree $\leftarrow \emptyset$ /* 初始化角色层次树 S 的所有子树 Subtree 为空 */

Roleset $\leftarrow TreeNode(S)$ /* 将所有角色层次关系 S 中的角色赋值给角色集 Roleset */

if ($s \in Roleset$)

then

If ($s \in P$) /* 如果节点与双亲节点存在激活关系 */

Then ($Subtree[i] \leftarrow TreeNode(S, s)$) /* 输出以此节点为根节点的子树 */

Roleset $\leftarrow Roleset - Subtree[i]$

return Subtree

end

/* 第一阶段首先输出混合角色层次树 S 的不含激活关系的所有子树 Subtree */

tree $\leftarrow \emptyset$ /* 初始化角色层次树 S 的所有不含继承关系的子树 tree 为空 */

Subtree $\leftarrow Get_Subtree(S)$

if ($Sub \in 2^{Subtree}$) /* Sub 是 Subtree 的幂集中的一个子集 */

then

If ($|Sub| == 1$) /* Sub 中的元素个数为 1 */

Then ($tree \leftarrow tree \cup IH$) /* 把这个元素当做一个子集加入到 tree 中 */

else

if (集合 Sub 中的角色存在继承关系)

Sub $\leftarrow 2^{Subtree-Sub}$

else

tree $\leftarrow tree \cup Sub$

return tree

end

/* 第二阶段在第一阶段的基础上输出混合角色层次树 S 的所有不含继承关系的子树 tree */

$MRS \leftarrow \emptyset$ /* 初始化角色层次树 S 的所有不含继承关系的子树 MRS 为空 */

Subtree $\leftarrow Get_Subtree(S)$

tree $\leftarrow Get_tree(S)$

if ($R \in 2^{tree}$) /* R 是 tree 的幂集中的一个子集 */

then

If ($|R| == 1$) /* R 中的元素个数为 1 */

if ($r \in R$) then

else

$MRS \leftarrow MRS \cup \{r\}$

Sub $\leftarrow 2^{Subtree-Sub}$

else

$MRS \leftarrow MRS \cup \$R$

Return MRS

end

/* 第三阶段输出混合角色层次树 S 的所有最小角色集合 MRS */

角色混杂层次树如图 2 所示, 图中的节点为角色, 角色之间的虚线代表其激活关系 (a-hierarchy); 实线代表其继承关系 (i-hierarchy); 单箭头表示指派给角色权限; 双箭头表示混杂关系 (hybrid hierarchy), 其表示的是同时具有继承和激活关系的角色层次. AT-RBAC 的角色分为就绪、失活、激活三种状态^[7], 且能依据访问权限互相转换.

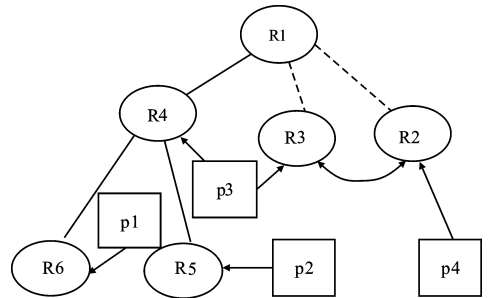


图 2 AT-RBAC 中的角色层次树

最小角色集合可以通过以下两个步骤获得:

(1) 首先在图 2 的角色层次树中通过清除角色间的激活关系得到 Subtree. 如图 3 所示, 混杂角色层次树被分为 3 棵子树, 三棵子树间及子树内均无角色激活关系.

(2) 其次除去 Subtree 子树中角色间的继承关

系,得到角色层次树中不含继承关系的子树 tree. 并通过角色间的组合得到 MRS. 如图 3 所示的 Sub-tree 中,子树 1 中不含继承关系的角色集合有 $\{R1\}$ 、 $\{R4\}$ 、 $\{R5\}$ 、 $\{R6\}$ 、 $\{R5, R6\}$. 以此方法计算所有子树中不含继承关系的角色集合,并任意组合这些角色集合,这些角色集合构成了混杂角色层次树的 MRS 集合.

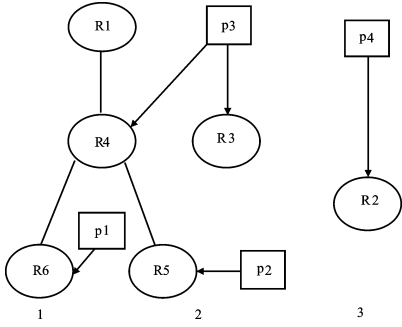


图 3 AT-RBAC 中角色层次树的分解

在 AT-RBAC 模型中,通过给用户指派角色获得用户的访问权限. 云环境中存在着巨量的资源,对于用户的访问控制权限,则有很多有差异性的角色匹配方法. 本章提出的 MRS 规定在角色集合中,角色之间不含有激活和继承关系,从而避免角色间的权限冗余.

3.2 基于属性的角色筛选

其基本原理如图 4 中所示. 其核心是在 AT-RBAC 模型的用户角色指派关系中,得到最小角色集合 MRS 并筛选出最适合的角色集合 ORS.

通过引入筛选函数和映射函数完成基于属性的角色筛选,首先对 MRS 中的角色集合进行筛选函数的匹配,其过程需通过映射函数来完成,满足映射函数设定的资源属性要求时,才为其分配筛选函数,对于 MRS 中的每个角色集合均进行此操作;其次,设定每个筛选函数均返回一个布尔类型的值,通过验证用户的静态属性和信任值、环境属性是否满足预设定的属性值,若满足则返回 true,反之返回 false,若这些筛选函数中有任何一个函数的值不是 true,则表示这个角色集合中有某个角色集合是不满足需求的,并从最小角色集合 MRS 中过滤掉;最终可得到满足要求的角色集合 ORS.

函数介绍如下:

(1)Filter 筛选函数

设计基于 AT-RBAC 模型的访问控制系统时,安全设计师就要给出相关的筛选函数 $Filter = \{F_1, F_2, F_3, \dots, F_n\}$. 每个筛选函数都是一个布尔表

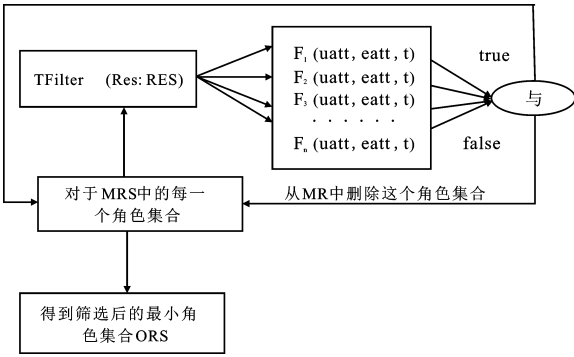


图 4 基于属性的角色筛选算法基本原理

达式,这个布尔表达式是根据用户属性、环境属性信任值来确定的,即对于任意的 $F_i: UATT \times EATT \times Trust \rightarrow \{true, false\}$.

(2)TFilter 映射函数

TFilter 函数用于实现每个资源到一个筛选函数子集的映射,并用属性表达式实现此关系,其用于判断每个筛选函数是否适用.

通过如下所示的三个步骤进行配置角色筛选策略:

角色筛选函数 Filter 的定义:

第 1 步定义角色筛选函数 F :描述 Filter 中的每个筛选函数 $F_i(uatt:UATT, eatt:EATT, t:Trust)$

第 2 步定义布尔条件 $cond$:针对每一个筛选函数 $FiFilter$,判断它是否使用布尔条件 $cond$

第 3 步定义映射函数 TFilter:将每个资源映射到其可以使用的筛选函数集合中,如下所示:

```
TFilter(res:RES);
filter: ← {}
cond1: filter: ← filter ∪ F1
cond2: filter: ← filter ∪ F2
...
condn: ← filter: filter ∪ Fn
return filter;
```

第 1 步为角色筛选函数的定义. 制定访问控制系统时,需要依据用户及资源属性以及用户的信任值定义每一个 Fi ,并选取应用于每个资源的筛选函数子集. 筛选函数的选择通过 Filter 函数完成. Filter 函数需要借助资源属性来为每一个 Fi 描述 $cond$.

第 2 步为布尔条件的定义. 对于每个过滤函数 Fi ,都有一个 $cond$,通过这些布尔条件判断哪些 Filter 能够被使用.

第 3 步为 TFilter 函数的定义和描述. Filter 函

数依据资源属性对每个 cond 进行判定,判断 F_i 可用与否,以此为给定的资源匹配一个可用的筛选函数子集.

算法 2 主要对算法 1 中得到的 MRS 进行角色筛选,即通过用户属性、环境属性、信任值筛选出最佳的角色集合 ORS 分配给用户.

算法 2 Get_ORS 获取最佳的角色集合 ORS * /
Input : S /* 输入一个混合角色层次树 */
Output : ORS /* 输出混合角色层次树 S 的所有最佳角色集合 ORS */
 $ORS = \leftarrow \emptyset$ /* 初始化筛选后的角色集合 ORS 为空 */
Fliter $\leftarrow \{F_1, F_2, F_3, \dots, F_n\}$ /* 筛选函数 */
TFilter(res: RES); /* 函数将每个资源映射到其可用的筛选函数集合 */
 $F_i \leftarrow F_i(uatt; UATT, eatt; EATT, t; Trust)$
If($ORS_i \in MRS$)
then if(判断 ORS 中的用户的属性,环境属性和信任值与预设定的是否符合)
 $f_i \leftarrow \{f_1, f_2, f_3, \dots, f_n\}$ /* 所有筛选函数布尔值的与集 */
if($f_i == true$);
Return ORS
end

3.3 算法分析

算法 1 中可知,第一阶段的时间复杂度主要由求解集合的幂集 $\$$ 决定,其复杂度为 $O(2xpm)$, m 为第二阶段中拆分成的不含继承和激活关系的子树的数量.最好的境况是混杂角色层次树中不含任何激活关系,第二阶段返回的有且只有一棵子树,此时时间复杂度为 $O(2xpn)$,其由角色数的深度优先遍历的复杂度决定,其中 n 是角色数量.最糟糕的境况是角色层次结构中不存在任何的角色继承,故第二阶段所得到的子树个数便是全部的角色数,时间复杂度为 $O(2xpn)$, n 为角色数量.但不含任何继承关系的系统是不存在的,因此最坏的境况出现的概率很低,由并行计算原理可知,算法 2 的复杂度取决于角色集合的个数,即为 $O(p)$, p 是 MRS 中的角色集合的个数,大部分境况下的时间复杂度为 $O(2xpn)$ 或 $O(2xpm)$.由于角色间的激活关系在系统中占的比重较小,所以 $m < n$,故此算法在 AT-RBAC 模型中是可行的.

4 仿真实验

本章节在自主研发的 AT-RBAC 模型系统上对 URAOA 算法进行了仿真实验.实验中模拟用户

不断增加对系统的访问请求,在同一平台上将 URAOA 算法与文献[8]提出的最小角色 MUR 算法进行对比,分析在匹配角色数量、运行时间、以及用户-角色匹配的精确度方面的优劣.实验规定:用户请求的权限数量为 100-1 000.

实验环境:实验以 Trust Builder2 作为仿真平台,分别使用相同配置的两台 8G 内存、Intel(R) Core(TM) i3-3240 CPU @3.40GHz,安装有 windows7 操作系统的 PC 机,并且数据部分使用 origin pro9 做数据绘图.

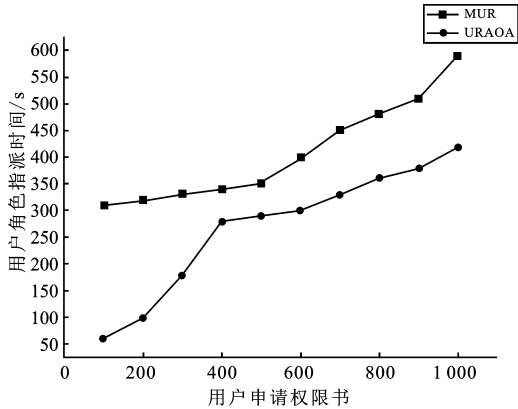


图 5 用户-角色分配数量对比

图 5 给出的是用户-角色分配的数量关系对比,从实验结果可以看出,当用户申请某个特定量的权限时, MUR 算法得到的角色数增加很快而 URAOA 算法得到的角色数目起伏不大,由此可看出 URAOA 算法明显优于 MUR 算法.

随着权限数的增加,URAOA 算法与 MUR 算法所用的时间呈增长趋势,但 URAOA 的增长速度明显小于 MUR 算法.

随着申请权限数的增多,URAOA 算法的匹配精确度高,而 MUR 算法的匹配精确度很低,综上所述可以得出结论,URAOA 算法的各项性能指标均优于 MUR 算法.

5 结束语

本文自主研发了 AT-RBAC 模型,与传统模型相比较,此模型结合基于属性的细粒度访问策略与基于信任值的动态判定标准,这些优点使 AT-RBAC 模型更加的适用于现实中的云环境,保证了访问控制的安全性;本文在 AT-RBAC 模型的基础上提出了 URAOA 算法,该算法有以下优点:在 AT-RBAC 模型中,角色间不存在激活和继承关系,避免权限冗余;系统根据用户的访问需求,通过此算

法筛选出 ORS 指派给用户,用户授权时间减少,准确匹配精确度提高,节省系统资源.

参考文献:

[1] Almulla S A, Chan Y Y. Cloud computing security management[C] Second International Conference on Engineering Systems Management and ITS Applications. IEEE, 2010;1-7.

[2] Majhi S K, Dhal S K. A Study on Security Vulnerability on Cloud Platforms [J]. Procedia Computer Science, 2016, 78:55-60.

[3] Curran K, Carlin S. Cloud computing security[M]// Cloud Computing Basics. Springer New York, 2014: 14-19.

[4] LI Fenghua, SHI Guozhen, MA Jianfeng, et al. Research status and development trends of access control model[J]. Acta Electronica Sinica, 2012, 40(4): 805-813.

[5] Qi H, Luo X, Di X, et al. Access Control Model Based on Role and Attribute and Its Implementation

[C]// International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. IEEE, 2017;66-71.

[6] 熊厚仁,陈性元,费晓飞等. 基于属性的 RBAC 的混合扩展访问控制模型[J]. 计算机应用研究,2016,33(7): 2167-2169.

[7] Joshi J B D, Bertino E, Latif U, et al. A Generalized Temporal Role-Based Access Control Model[J]. IEEE Transactions on Knowledge & Data Engineering, 2005, 17(1):4-23.

[8] Zhang L, Zhang H L, Han D J, et al. Theory and Algorithm for Roles Minimization Problem in RBAC Based on Concept Lattice[J]. Acta Electronica Sinica, 2014, 42(12):2371-2378.

作者简介:

王静宇 男,(1976-),博士,副教授. 研究方向为云计算、信息安全.

陈 丽(通讯作者) 女,(1992-),硕士研究生. 研究方向为云计算与信息安全. E-mail: 15039084237@163.com.