文章编号:1671 - 6833(2017)02 - 0013 - 04

基于属性规则的 PRBAC 参数模型研究与实现

欧阳荣彬, 刘云峰, 龙新征

(北京大学 计算中心,北京 100871)

摘 要: PRBAC 模型可以实现细粒度的数据访问控制. 论文分析了以往有关 RBAC 数据权限的研究,总结了具体的实践探索经验,提出一种基于属性规则的 PRBAC 参数模型,以实现通用的数据权限管理. 笔者阐述了模型的设计思路,包括数据权限规则的形式组成、具体含义,还阐述了模型的实现方案,包括规则的实现形式、PRBAC 参数应用时机、规则校验的主要实现算法,以及相关的技术要点. 论文还结合该模型在北京大学 IAAA 系统的应用实践阐述了模型的优势,即数据权限规则设置具有较强的通用性,灵活而便捷,最后指出模型实现方案可以在规则冲突检验方面进一步完善.

关键词: 访问控制;数据权限;PRBAC;属性规则;参数模型

中图分类号: TP315 **文献标志码**: A doi:10.13705/j.issn.1671 - 6833.2017.02.004

0 引言

1992 年文献[1]提出了基于角色的访问控制模型(role-based access control, RBAC),之后有关 RBAC 的研究不断发展,并形成了相关标准. NIST (The National Institute of Standards and Technology,美国国家标准与技术研究院)的标准定义了三类 RBAC 模型,分别是基本模型(core RBAC)、角色分层模型(hierarchal RBAC)、角色限制模型(constraint RBAC)[2].

目前,RBAC 模型已经成为一种广泛应用的访问控制模型,其简洁、可扩展、易管理的特性被广泛认可. 在实际应用中,当一批用户具有同等的功能权限,他们被授予一个相同的角色,但是这些用户可以操作的数据对象却可能是各不相同的,即他们应当具有相应的数据权限. RBAC 标准模型并没有明确定义数据权限模型,也没有给出建议的实现策略. 虽然从理论上来说,RBAC 标准模型可以通过细粒度的操作对象(OBS)划分实现数据权限,但是在实际环境中,细粒度的划分并形成更细粒度的权限(PRMS),势必需要大量的角色,而且操作繁复.

PRBAC (parameterized RBAC) 模型是对 RBAC 基本模型的扩展,能够完成细粒度的权限访问控制,而且无需大量角色和繁复的操作. 在具体应用的实现过程中,如何设计参数才能有效地应对不同数据对象的访问控制呢? 笔者提出了一种基于属性规则的 PRBAC 参数模型,将 PRBAC 中的参数设计为一组规则集,规则集中定义了角色可以访问的数据对象,及其应当满足的属性规则. 模型和规则可以灵活地适配于各类不同的数据对象.

笔者接下来首先回顾和总结了有关 PRBAC 和数据权限的研究,然后具体阐述了基于属性规则的 PRBAC 参数模型,并给出了具体的实现算法,介绍了有关技术要点,最后介绍了该模型在北京大学的应用实践.

1 相关研究

很多研究人员在 PRBAC 的细粒度权限控制方面做了不少有益的探索和研究.

文献[3]在 PRBAC 的基础上引入面向对象的概念,提出了策略模板(policy template)和角色类(role class),即通过角色的实例化实现细粒度权限控制. 文献[4]提出参数化角色的设计,即不同用户被授予同一个角色的时候有可能携带不同的参数. 文献[5 - 6]采用 Z 语言,在 FRBAC(flat RBAC)的基础上对 PRBAC 模型进行了规范化阐述. 文献[7 - 8]对 RBAC 的 Web Service 访问控制进行了论述,其中的 Actor 概念和 PRBAC 目标

收稿日期:2016 - 10 - 31;**修订日期**:2016 - 11 - 29

基金项目:国家发改委 2011 国家信息安全专项资助项目

作者简介:欧阳荣彬(1979—),男,北京大学高级工程师,主要从事教育信息化研究,E-mail:ouyang@pku.edu.cn.

相近,也是为了实现细粒度的权限控制. 文献[9-12]引入了数据角色和数据权限规则的概念,旨在将数据权限与功能权限独立开来,以实现对数据访问的细粒度权限控制,然而数据角色的引入增加了管理的复杂性.

PRBAC是对RBAC模型的扩展,在RBAC模型的各个部件中加入了参数(包括角色、对象、权限等)以实现对权限尤其是数据权限的细粒度控制. 用户在被赋予某个角色的时候可以携带一系列的参数,并将这些参数传递给模型中的各个部件. 如此,不同参数的用户角色,其访问权限尤其是数据访问权限就会各不相同.

为了设计一种通用的参数,使其可以适配各种不同的数据对象,常见的做法是将SQL条件语句作为参数. 但是,由于数据库类型多样,SQL语句处理情况也多样,因此这种做法的通用性受到很大的限制. 笔者将PRBAC中的参数设计为一组规则集,规则集中定义了角色可以访问的数据对象,及其应当满足的属性规则,可以灵活地与各类不同的数据对象适配.

2 PRBAC参数模型

PRBAC模型中引入参数的目的是为了实现更加精细的数据权限访问控制. 为了实现参数模型的通用性,笔者将PRBAC中的参数具体设计为一组基于属性规则的数据权限规则集,规则集的主要内容正是访问控制的对象——数据对象及其属性.

2.1 数据权限规则

首先介绍数据权限规则dpr. 在形式上,dpr由元素对(data,rule_set)表示,下面是对元素对的各组成部分的描述.

data:数据对象,由数据对象名(obj_name)和数据对象类型(obj_type)组成;

rule_set:属性规则集,形式上由元素对(relation,rules,rule_sets)表示,元素对中子元素描述如下:

①relation:逻辑关系符,取值OR或AND,表示rules中组成元素以及rule_sets中组成元素之间的关系;

②rules:属性规则rule的集合,每个rule为一个简单的比较运算表达式,形式上由元素对(attr_name,comparator,value)表示,其子元素分别表示对象属性名、比较符和右值;

③rule_sets:属性规则集rule_set的集合.

基于上述描述,dpr元素对(data,rule_set)可以构成一个由比较运算表达式和逻辑关系符组成的复杂的布尔表达式,逻辑关系符为relation,每个比较表达式都由rule元素对(attr_name,comparator,value)中的子元素构成. data元素也参与比较表达式的构成,主要是用于限制元素attr_name的所属. 具体示例会在第3.1节"规则实现形式"中解释.

2.2 动态属性规则

在形式上,属性规则rule元素对(attr_name,comparator,value)中的右值value可以有两种:常量形式和变量形式. 常量形式为一具体的值,例如具体的院系代码"00082".

变量形式时,其值则与用户会话相关,如采用"{USER.deptID}"表示当前会话中用户的单位编码. 如此,PRBAC的参数就具有了用户会话的属性,产生动态效果. 形成了基于对象属性和用户会话属性的数据权限规则,使得数据权限的配置更加灵活.

3 实现方案

3.1 规则实现形式

在具体实现时,可以采用XML形式的属性规则,表1为属性规则示例. 如果某个角色被赋予了表1所示的参数,那么这个角色可以(不是只可以,因为示例中还有其他规则集,而且关系符还是OR)访问院系单位为00082的人员基本信息(类型为pku.model.PersonInfo).

3.2 参数应用的时机

按PRBAC模型所述,用户登录系统进入会话之后,激活已经赋予了参数的用户授权角色,此时的参数为一组数据权限规则集,因为一个角色可能被赋予多个数据权限规则. 同时,如果数据权限规则中的属性规则右值(value)采用了变量形式,那么在用户登录之后需要将这些变量替换成实际的用户会话属性值.

文献[4]指出,被赋予了参数的权限(parameterized privileges),形式为$(x \mid x\{a1,a2,\cdots,an\},m)$,其中:$x$为数据对象;$a1$、$a2$、…、$an$为参数;$m$为访问操作. 所以,应当是在用户进行数据访问操作的时候应用参数,校验数据权限规则,确保用户只能操作那些通过规则校验的数据,即具体的数据权限规则只对具体的用户访问操作生效,而不是对用户同类型数据的所有访问操作都生效.

表 1　属性规则示例

Tab. 1　Sample of rules base on atrributes

```
< dpr >
  < data >
    < obj_name > 人员基本信息 < /obj_name >
    < obj_type > pku. model. PersonInfo < /obj_type >
  < /data >
  < ruleset >
    < relation > OR < /relation >
    < rule >
      < attr_name > deptID < /attr_name >
      < comparator > EQUAL < /comparator >
      < value > 00082 < /value >
    < /rule >
    < ruleset > < ! —其他规则集 - - > < /ruleset
>
  < /ruleset >
< /dpr >
```

由于一个角色可能被赋予了多个数据权限规则,这些规则可能都是针对同一类数据对象. 当一个用户角色具有多个数据权限规则时,这些规则之间的关系应该是 OR 的关系.

3.3　规则校验

参数应用的过程就是数据权限规则校验的过程. 过程 FilterWithDPRs 的功能是对数据对象集合依据数据权限规则集进行过滤,其算法描述如表 2 所示. 其中参数 data_list 表示原数据对象集合,dpr_list 表示用户在当前会话中的数据权限规则集. 算法中涉及的子过程 FilterWithRS 是将数据对象集合依据单个数据权限规则的属性规则集合进行过滤,其算法表述如表 3. 其他涉及的子过程功能描述如表 4.

表 2　基于 DPRs 的过滤

Tab. 2　FilterWith DPRs

```
FilterWithDPRs( data_list, dpr_list)
BEGIN
  VAR ret;
  VAR dpr = Next( dpr_list) ;
  WHILE dpr < > NULL LOOP
    IF MatchDataType( data_list, dpr) THEN
      VAR rs = GetRuleSet( dpr) ;
      VAR tmp_list = FilterWithRS( data_list, rs) ;
      ret = Union( ret, tmp_list) ;
    END IF
    dpr = Next( dpr_list) ;
  END LOOP
  RETURN ret;
END
```

表 3　基于 RS 的过滤

Tab. 3　FilterWith RS

```
FilterWithRS( data_list, rs)
BEGIN
VAR ret;
VAR rel = GetRelation( rs) ;
IF rel = = 'AND' THEN
 ret = data_list;  /* AND 时,初始集合为全集 */
ENDIF
VAR sub_rs = NextRS( rs) ;
WHILE sub_rs < > NULL LOOP
  VAR tmp_list = FilterWithRS( data_list, sub_rs) ;
  IF rel = = 'OR' THEN
  ret = Union( ret, tmp_list) ;
  ELSE
   ret = INTERSECT( ret, tmp_list) ;
  ENDIF
  sub_rs = NextRS( rs) ;
END LOOP
VAR rule = NextRule( rs) ;
WHILE rule < > NULL LOOP
  VAR tmp_list = FilterWithRule( data_list, rule) ;
  IF rel = = 'OR' THEN
  ret = Union( ret, tmp_list) ;
  ELSE
   ret = Intersect( ret, tmp_list) ;
  ENDIF
  rule = NextRule( rs) ;
  END LOOP
  RETURN ret;
END
```

表 4　相关子过程功能描述

Tab. 4　Other functions

过程	功能描述
Next	取集合中的下一个元素
Union	取两个集合的并集
Intersect	取两个集合的交集
MatchDataType	判断 data_list 与 dpr 中的数据对象类型是否一致
GetRuleSet	获得 dpr 中的 ruleset
GetRelation	获得 ruleset 中的逻辑关系符
NextRS	获得 ruleset 中的下一个子 ruleset
NextRule	获得 dp 中的下一个属性规则 rule
FilterWithRule	遍历 data_list 中的数据对象,分别校验属性规则 rule,为真则通过过滤,否则不通过

3.4　Java 反射

在过程 FilterWithRule 中,需要校验单个属性规则. 然而,属性规则中的属性名都被配置成了文本,一般的做法是遍历所有可能属性名,然后调用相应的属性访问方法. 但是,一个数据对象一般都具有多个属性,而且数据对象的类型也不可能只有

少数几个,不同的应用系统都可以定义自己不同的数据对象类型.因此,不可能在 FilterWithRule 的具体实现中通过文本判断遍历所有的情况.

Java 语言中的反射机制能够通过文本的属性名获取到该属性的访问方法,并且可以通过程序触发运行,从而获得数据对象的属性值.实际上,Java 语言的反射机制对于本文基于属性规则的 PRBAC 参数模型的通用性有极为关键的作用,而且,其他的面向对象语言中也都有类似的反射机制.

4 应用实践

"北京大学统一身份认证和权限管理系统"(以下简称"IAAA 系统")是北京大学电子校务环境下集成了用户管理、身份认证、权限管理和日志审计管理等功能的综合安全管理系统[13].其中权限管理部分基于 PRBAC 模型设计和实现,在数据权限管理方面应用了笔者提出的 PRBAC 参数模型,实现了通用而且细粒度的数据权限管理.其简化的实现类图如图 1 所示.其中"对象约束属性"对应 2.1 节中阐述的属性规则 rule 元素对中的 attr_name.

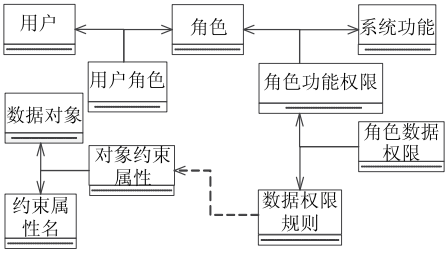

图 1 权限管理类图

Fig. 1 Class diagram of permission management

"IAAA 系统"目前共管理北京大学电子用户身份数据大约 21 万条,为 108 个应用系统提供统一身份认证服务,为 29 个应用系统提供统一权限管理服务(共有角色约 320 个),其中 10 个应用系统集成了数据权限管理服务."IAAA 系统"面向全校的应用系统提供统一而分级的权限管理服务,应用系统的权限管理均由注册的应用系统安全管理员管理和配置,无需分别在应用系统中重复设计实现和管理.

实践表明,"IAAA 系统"能够合理而有效地实现数据权限管理,应用系统集成简便,规则配置灵活,满足了应用系统的权限管理需求.

5 结论

笔者通过分析以往研究并总结具体的实践探索经验,提出了基于属性规则的 PRBAC 参数模型.阐述了该模型的设计思路和实现方案,包括数据权限规则的形式组成、具体含义,以及规则的具体实现形式、参数的应用时机、规则校验的主要实现算法和相关的实现技术要点等.规则配置时对规则冲突的检验方面本文的实现方案并未涉及,还需要进一步完善.

最后,北京大学"IAAA 系统"的应用实践表明,笔者提出的基于属性规则的 PRBAC 参数模型可以合理而有效地实现数据权限管理,规则设置具有较强通用性,灵活而且便捷,满足了应用系统的权限管理需求.

参考文献:

[1] FERRAIOLO D, KUHN R. Role-based access control [C]// Proceedings of the NIST-NSA National(USA) Computer Security Conference. Piscataway: IEEE, 1992:554 - 563.

[2] ANSI. American National Standard for information technology-role based access control[M]. ANSI INCITS 359 - 2004. New York: American National Standards Institute, Inc. 2004.

[3] EMIL L, MORRIS SM. Reconciling role based management and role based access control[C]//Proceedings of Symposium on Access Control Model and Technologies. New York: ACM. 1997:135 - 141.

[4] MEI G, SYLVIA L O. A design from parameterized roles[C]//Proceedings of Research Directions in Data and Application Security XVII. Laxenburg: IFIP, 2004:251 - 264.

[5] ETIENNE J K, ALI E A. A formal model for flat role-based access control[C]//Proceedings of ACS/IEEE International Conference on Computer Systems and Application. Piscataway: IEEE, 2003.

[6] ALI E A, ETIENNE J K. A formal model for parameterized role-based access control[C]//Proceedings of Workshop on Information Technologies and Systems. Georgia: WITS, 2005:233 - 246.

[7] XU F, LIN GY, HUANG H, et al. Role-based access control system for web services[C]//Proceedings of Conference on Computer and Information Technology. Piscataway: IEEE, 2004:357 - 362.

[8] JONATHAN K A. A service-centric approach to a parameterized RBAC service [C]//Proc. of the 5th WSEAS International Conference on Applied Computer Science. Wisconsin: WSEAS , 2006.

(下转第 40 页)