

# 基于角色的访问控制(RBAC)研究与应用

张艳华、王新华、夏明俊

上海远程教育集团

**摘要:** 基于角色的访问控制 RBAC 是一种方便、安全、高效的访问控制机制。本文针对教育资源用户信息量庞大、种类多等特点,运用基于角色的用户权限管理参考模型,详细研究和分析了该模型的构成和相关技术,给出了其在上海教育资源库项目中的应用。

**关键词:** 角色 权限 用户 教育资源库

## 1. 引言

随着计算机应用的日益普及,特别是网络的迅速发展,如何在发展和推广网络应用的同时进一步提高访问控制的安全与效率,已经成为目前网络界所必须研究和解决的课题。针对不同的应用,需要根据项目的实际情况和具体架构,在维护性、灵活性、完整性等 N 多个方案之间比较权衡,选择符合的方案。对于在企业环境中的访问控制方法,一般有三种:

- 1、自主型访问控制方法。目前在我国的的大多数的信息系统中的访问控制模块中基本是借助于自主型访问控制方法中的访问控制列表(ACLs)。
- 2、强制型访问控制方法。用于多层次安全级别的军事应用。
- 3、基于角色的访问控制方法 (RBAC)。是目前公认的解决大型企业的统一资源访问控制的有效方法。其显著的两大特征是: 1.减小授权管理的复杂性,降低管理开销。2.灵活地支持企业的安全策略,并对企业的变化有很大的伸缩性。

## 2. 上海教育资源库项目用户角色管理概述

### 2.1 总体描述

上海教育资源库项目根据其项目的特点,选用了基于角色的用户访问控制(RBAC)。访问控制系统可以为应用系统建立一个高安全强度,更易维护管理,扩展能力极强的访问控制环境,并能够有效的控制管理的复杂性,提供标准化和可以不断延伸授权平台。访问控制系统的体系结构设计保证了用户能按照应用规模和数量快速地建立访问控制体系。完全基于策略思想设计的权限管理控制系统作为构建访问控制体系的基石,它能使用户已建立的访问控制体系不断满足演化的应用权限需求,如支持新应用、支持新的权限、增加用户数量、增加用户类型、增加策略数量等。

上海教育资源库系统在用户访问控制管理中引入了角色的概念,即在系统中提供若干个角色,每个角色的权限可以任意定制,在每一个角色下可以包含若干个用户,用户只能够使用本角色允许使用的系统功能,对用户无权使用的系统功能可以设置其状态为不可见或隐藏。管理员可以首先分配好各个角色的权限,然后将系统的用户分配到各个不同角色中,即可以完成用户权限的设定,而不用逐个的去设定用户的使用权限。

### 2.2 角色的概念

在现实生活中经常提到某人扮演了什么角色。在基于用户角色的用户权限管理中,角色与实际的角色概念有所不同。在这里角色可以看作是一组操作的集合,不同的角色具有不同的操作集,这些操作有系统管理员分配给角色。

用户的授权是通过授予用户一个角色来实现的,即赋予用户一个角色,一个用户可以承担不同的角色,从而实现授权的灵活性。只要某用户属于某个角色那么他就具备这个角色的

所有操作许可，即该角色所拥有的权限。用户与角色是多对多的关系。即一个用户可以属于多个角色之中，一个角色可以包括多个用户。

### 2.3 RBAC 模型构件分析

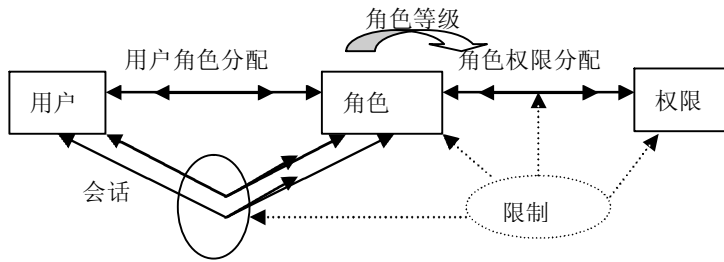


图 3.3 RBAC 模型

我们对该模型定义如下：

- U, R, P, S: 用户；角色，权限，会话；
- $PA \subseteq P * R$ : 权限分配，多对多的关系；
- $UA \subseteq U * R$ : 用户分配，多对多关系；
- User:  $S \rightarrow U$ , 每一个会话 s 对应单一用户 user(s) 的映射；
- Roles: 会话 s 到角色集合  $role(s) \subseteq \{r | (user(s), r) \in PA\}$

易见：该模型由三个实体组成，分别是：用户（U）、角色（R）、权限（P）。其中用户指自然人；角色就是组织内部一件工作的功能或工作的头衔，表示该角色成员所授予的职责的许可，系统中拥有权限的用户可以执行相应的操作。

用户与角色之间以及角色与权限之间用双双箭头相连表示用户角色分配 UA 和角色权限分配 PA 关系都是多对多的关系，即一个用户可以拥有多个角色，一个角色也可被多个用户所拥有。同样的，一个角色拥有多个权限，一个权限能被多个角色所拥有。用户建立会话从而对资源进行存取，每个会话 S 将一个用户与他所对应的角色集中的一部分建立映射关系，这个角色会话子集称为会话激活的角色集。于是，在这次会话中，用户可以执行的操作就是该会话激活的角色集对应的权限所允许的操作。

## 3. RBAC 特点及应用优势

### 3.1 RBAC 几大特点

(1) 访问权限与角色相关联，不同的角色有不同的权。用户以什么样的角色对资源进行访问，决定了用户拥有的权限以及可执行何种操作。

(2) 角色继承。角色之间可能有互相重叠的职责和权力，属于不同角色的用户可能需要执行一些相同的操作。RBAC 采用角色继承的概念，如角色 2 继承角色 1，那么管理员在定义角色 2 时就可以只设定不同于角色 1 的属性及访问权限，避免了重复定义。

(3) 最小权限原则，即指用户所拥有的权力不能超过他执行工作时所需的权限。实现最小特权原则，需要分清用户的工作职责，确定完成该工作的最小权限集，然后把用户限制在这个权限结合的范围之内。一定的角色就确定了其工作职责，而角色所能完成的事物蕴涵了其完成工作所需的最小权限。用户要访问信息首先必须具有相应的角色，用户无法绕过角色直接访问信息。

(4) 职责分离。一般职责分离有两种方式：静态和动态。

(5) 角色容量。在一个特定的时间段内，有一些角色只能有一定人数的用户占用。在创建新的角色时应该指定角色的容量。

### 3.2 RBAC 应用优势

最突出的优点在于系统管理员能够按照部门、学校、企业的安全政策划分不同的角色，执行特定的任务。一个系统建立起来后主要的管理工作即为授权或取消用户的角色。用户的职责变化时只需要改变角色即可改变其权限；当组织功能变化或演进时，则只需删除角色的旧功能，增加新功能，或定义新角色，而不必更新每一个用户的权限设置。这极大的简化了授权管理，使对信息资源的访问控制能更好地适应特定单位的安全策略。

RBAC 另一优势体现在为系统管理员提供了一种比较抽象的、与企业通常业务管理性类似的访问控制层次。通过定义、建立不同的角色、角色的继承关系、角色之间的联系以及相应的限制、管理员可动态或静态地规范用户的行为。

## 4. RBAC 实现

### 4.1 RBAC 通信过程

在网络系统中使用 RBAC，其实是对网络服务器而言的，对于用户浏览器没有选择。而且在浏览器与服务器的通信方式有很多中，如 RBAC/Web。当然使用 RBAC 的前提是有效而且正确的身份验证和保密地传输数据，这些包括使用用户号/口令识别法、字段加密、安全套接层（SSL）和安全 HTTP(HTTPS)以及私有信息技术协议（PCT）等。

RBAC/Web 的整个通信过程如下：当终端用户想要对数据库进行某一操作时，他首先向 RBAC Server 提出建立 RBAC 对话的请求，Server 显示该终端用户可以激活那些角色 (ARS)。这些角色的选择满足限制条件和角色等级限制以及 Server 割据用户的要求建立会话，这时用户就可以对 RBAC 数据库提出操作请求，由 Server 进行相应的处理。最后用户工作完毕，Logout 系统，结束这次会话。在整个过程中，ABS 保持被激活的状态，这些 ABS 构成了 RBAC 会话的基本特征。

### 4.2 资源库权限分配的应用实例

我们用一个上海教育资源库功能模块来举例子，显示 RBAC 在上海教育资源库中的应用：每当用户登陆系统，系统会找到该用户所属的角色，根据角色得到相应的权限，将该用户所拥有的权限保存员工权限的 String 数组中。当判断该用户是否有一特定权限，如上传资源的权限，可以对比当前员工的权限和给这个上传资源的权限分配的“功能 ID”判断当前用户是否有使用这项功能的权限。如果保存员工权限的 String 数组中这个 ID，那这个功能的操作按钮就不会显示，如果员工的 String 数组中有此功能的 ID，那这个功能的操作按钮就会显示。

以下代码显示资源库系统中权限与身份验证部分代码

```
boolean bolResManage = false;
boolean bolUserManage = false;
boolean bolSysConfig = false;//系统设置
boolean bolFeeConfig = false;//记费设置
boolean bolUpload = false;
boolean bolLogined = false;
boolean bolGuest = false;
String strUserName = "";
try{
    HttpSession session1 = request.getSession();
    strUserName = (String)session1.getAttribute("UserName");
    if(strUserName.length()==0 ||strLogout.length()>0){ //未登陆，默认为客人
        bolLogined = false;
        //默认为客人登陆
        request.setAttribute("userName","guest");
        request.setAttribute("userPwd","guest");
        strUserName="guest";
        boolean bool=LoginInfo.userLogin(request,response);
```

```

    }else{
        bolLogined    = true;
    }
    String[] arr = null;
    arr = (String[]) session1.getAttribute("resmanage"); //判断资源管理的权限
    if(arr != null&&arr[2].equals("1")){
        bolResManage  = true;
    }
    arr = (String[]) session1.getAttribute("usermanage");//判断用户管理的权限
    if(arr!=null&&arr[2].equals("1")){
        bolUserManage = true;
    }
    arr = (String[]) session1.getAttribute("config");    //判断系统设置的权限
    if(arr!=null&&arr[2].equals("1")){
        bolSysConfig  = true;
    }
    arr = (String[]) session1.getAttribute("feeconfig"); //判断计费设置的权限
    if(arr!=null&&arr[2].equals("1")){
        bolFeeConfig  = true;
    }
    if(strUserName.length()>0 && (!strUserName.equalsIgnoreCase("GUEST")))
        bolUpload    = true;
    arr = (String[]) session1.getAttribute("upload");    //判断资源上传的权限
    if(arr!=null&&arr[2].equals("1")){
        bolUpload    = true;
    }
    if(strUserName.equals("GUEST")){//客人
        bolGuest     = true;
    }
} catch(Exception e){
    out.println("Right init error!");
}
}

```

## 5. 结束语

一个安全的网络需要可靠的访问控制作保障。在网络规模变大、用户增多、需求更复杂的情况下，传统的访问控制已经不能满足许多企业或组织的安全需要，基于角色的访问控制（RBAC）便明显地显示出其优越性。基于角色的访问控制可以很好的解决资源库项目的访问控制，为系统开发提供了一套有力的工具，还为用户评估系统提供了标准。

## 参考文献:

- 1、曹天杰，张永平。管理信息系统中基于角色的访问控制（J）.计算机应用，2001，21（8）：21-23
- 2、汪厚祥、李卉。基于角色的访问控制研究.计算机应用研究，2005（4）：125-127
- 3、上海教育资源库详细设计
- 4、Ahn GH,Arvisandhu.Role-based Authorization Constrans Specification[J].ACM Transcations on Information and System Security,2002,(3):207-226
- 5Sylvia Osborn,Yuxia Guo.Modeling Users in Role-based Access Control[C].Berlin:Processdings of the 5th ACM Workshop on Rolebased Access Control(RBAC-00),2000.26-27