基于RBAC的软件自主实验平台设计

石晓耀[1],张维成[2],江 颉[2]
(1.浙江省教育装备和勤工俭学管理中心,浙江 杭州 310008;
2.浙江工业大学 软件学院,浙江 杭州 310014)

摘 要:RBAC技术以角色为访问主体,将访问权限与角色相联系,通过给用户分配适合的角色,使角色成为访问控制的主体,可以提高管理的效率,减少授权管理的复杂性。设计了一个基于Web方式的软件自主实验平台,并且利用RBAC技术实现了对自主实验的访问控制管理。介绍了基于RBAC的软件自主实验平台的总体设计、部分关键技术及实现。

关键词:基于角色的访问控制;软件自主实验;Web方式
中图分类号:TP302.1 **文献标识码**:A **文章编号**:1673-629X(2007)10-0230-04

Design of Software Self-Guidance Experiment Platform Based on RBAC

SHI Xiao-yao[1], ZHANG Wei-cheng[2], JIANG Jie[2]
(1. Zhejiang Education Equipment and Work-Study Programme Management Center, Hangzhou 310008, China;
2. College of Software, Zhejiang University of Technology, Hangzhou 310014, China)

Abstract: RBAC takes the role as the access main body, relates access privilege and role, through assigning to the user the suitable role, makes the role become the main body of access control, enhances the management efficiently, and reduces the complexity of authorized management. In the paper, the Web-based software self-guidance experiment platform is designed, and the role-based authorization access control technique is used to implement the access to this platform. The architectural design, the essential technique and implementation is introduced in it.

Key words: RBAC; software self-guidance experiment; Web

0 引 言

随着软件技术的发展和社会对软件工程专业培养应用型人才的需求,越来越多的企事业单位都要求学生在进入生产实习期前就能拥有运用软件工程系列知识的能力,这对学生而言往往需要独立的在课外通过个人的自学和实验才能获得。文中提出的软件自主实验平台设计采用Web方式,让学生综合利用所学知识,选择一个较为复杂的软件项目,以项目小组形式,在合理进行软件项目开发和维护的一般过程中,通过分析、设计、编程、测试等阶段加深理论知识的深度,拓展知识面,提高综合运用知识能力,达到实践能力、自学创新能力、交流表达能力、团队合作能力的培养与增强,为更深入地学习和今后从事软件工程实践打下良好的基础。在实验的设计过程中发现,为了使师生合理、便利地使用该实验平台,需要对人们访问该平台的能力进行访问权限的控制。

访问控制服务一直是网络安全的一项重要内容。传统的访问控制技术如DAC和MAC在本实验平台的使用中有其明显的不足。在DAC中用户拥有修改部分访问权限的权力,这使得管理员难以确定哪些用户对哪些资源有访问权限,不利于实现统一的全局访问控制。而MAC过于偏重保密性,对其他方面如系统连续工作能力、授权的可管理性等考虑不足。这也是导致近几年来人们普遍感到DAC和MAC不能满足大量存在的商业和政府部门系统的安全需求。

基于角色的访问控制RBAC[1](Role Based Access Control)技术以角色为访问主体,将访问权限与角色相联系,通过给用户分配适合的角色,使角色成为访问控制的主体,大大提高了管理的效率,减少授权管理的复杂性,可以非常好地适合本实验平台的设计和使用。

因此,文中主要研究软件自主实验平台的设计以

收稿日期:2006-12-29
基金项目:浙江省科技计划项目(2004C31108)
作者简介:石晓耀(1973-),男,浙江诸暨人,工程师,主要研究领域为多媒体、信息管理系统、网络安全。

及其中关键的基于 Web 方式的 RBAC 访问控制的设计、应用实现以及部署等。

1　软件自主实验平台总体设计

1.1　基于角色的访问控制技术

基于角色的访问控制模型(RBAC)概念早在 20 世纪 70 年代就随着多用户、多应用在线系统的出现而产生,但真正形成一套完整理论的则是由 Sandhu[2]等人于 1996 年在 IEEE 上发表的文章——基于角色的访问控制模型,习惯上称之为 RBAC96 模型。

RBAC 的核心思想是将访问权限与角色相联系,通过给用户分配适合的角色,让用户和访问权限相联系。角色是根据企业内为完成各种不同的任务需要而设置的,根据用户在企业中的职权和责任来设定他们的角色,用户可以在角色间进行转换,系统可以添加、删除角色,还可以对角色的权限进行添加、删除,这样通过应用 RBAC 可以将安全性放在一个接近组织结构的自然层面上进行管理。

目前,国外 RBAC 研究机构主要是美国 NIST 和 George Mansion Univ. LIST 实验室(Prof. Ravi. Sandhu)。NIST 主要是进行 RBAC 及其相关模型的标准化工作,LIST 侧重于对 RBAC、RBDM 及其扩展模型的创建、形式化描述、评价分析,以及在 Web 中的应用等。国内主要是中国科学院软件研究所和华中科技大学计算机科学与工程系,他们正在对 RBAC 模型扩展和应用方面进行深入的研究。

迄今,RBAC 研究已应用于 PMI、CORBA、CSCW 等体系架构中,并在电子商务、大型信息系统中也得到广泛应用。它是目前应用最广、效果良好的访问控制策略与模型。RBAC 只是一个模型,所以当今对 RBAC 的应用研究重点之一是如何在特定应用系统中使用这种访问控制技术。

文中主要以开放源码 Open PERMIS[3]为原型系统,探讨在论文设计的实验平台上的基于角色的访问控制技术的设计与实现。

1.2　实验平台内容总体框架

文中的基于 Web 的应用系统是学生软件自主实验平台,主要用途是提供一个虚拟的实验环境,可以让教师布置软件实验要求后,学生可以通过浏览器客户端在线(或离线)完成任务,摆脱了以往那种只能在实验室等地方做实验的空间束缚。整个软件自主实验平台主要如图 1 所示。

学生自主实验平台按用户类型的不同分成三大模块:学生模块、教师模块和管理员模块。

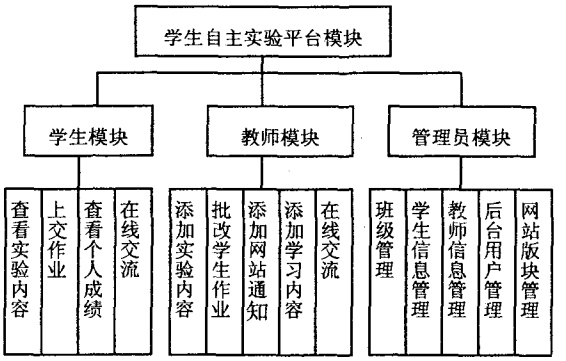

图 1　软件自主实验平台总体框架

1.3　基于角色的访问控制在自主实验平台上部署

在一般 B/S 结构的应用系统中,客户端直接与 Web 服务器进行交互,客户端发送一个 http 请求到服务器端,服务器接受请求并进行处理后将结果返回给客户端。访问控制都在 Web 服务器端实现,原先的学生自主实验平台的权限判断按照如下流程进行:

1) 在用户输入用户名和密码提交给服务器验证其身份合法后,服务器端在 session 里设置一个标志。

2)权限的判断仅仅是在需要特定权限才能访问的页面上加入类似代码进行具体的判定。这样的设计存在如下的缺陷:

a. 访问控制 Web 服务器端实现,功能混杂。

b. 安全性不够,假如账号和密码被窃取(这样的概率很大),那么在该网站上所有权限都可被使用。

c. 权限的概念不明确,角色与权限的关系也不明确。

d. 重复编码,这样需要在每张页面中写上面的代码。

针对前面所述,采用 RBAC 访问控制技术,文中的设计方案是在客户端浏览器和 Web 服务器之间加入一个 Web 代理 AEF 模块,实现访问控制执行和决策功能,将基于 Web 的 RBAC 系统集成并且将其部署到学生自主实验平台中。如图 2 所示。

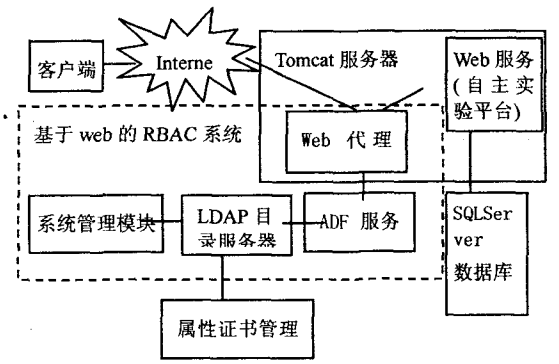

图 2　基于 Web 的 RBAC 软件自主实验平台部署图

其中 Web 代理拦截所有到 Web 服务器的 http 请

求,这些 http 请求都是以 URL 的方式传递给 Web 代理。Web 代理 AEF 的作用是截取用户的访问请求,把请求信息组合成访问控制决策模块 ADF 可以接收的决策请求参数,如用户的 LDAP DN、请求的目标和请求的操作信息,也可以包括一些环境变量。Web 代理提供了一个对用户认证的接口,并从该接口中获取用户的 LDAP DN。

2 关键技术设计与实现

2.1 软件自主实验平台主要类图

本系统三类角色:学生、教师、管理员,都实现了 Person 接口,描述如图 3 所示:接口 Person 包括了上述三类角色的公共属性和操作;学生类(Student)、教师类(Teacher)等实现 Person 类,并对接口进行了扩展。

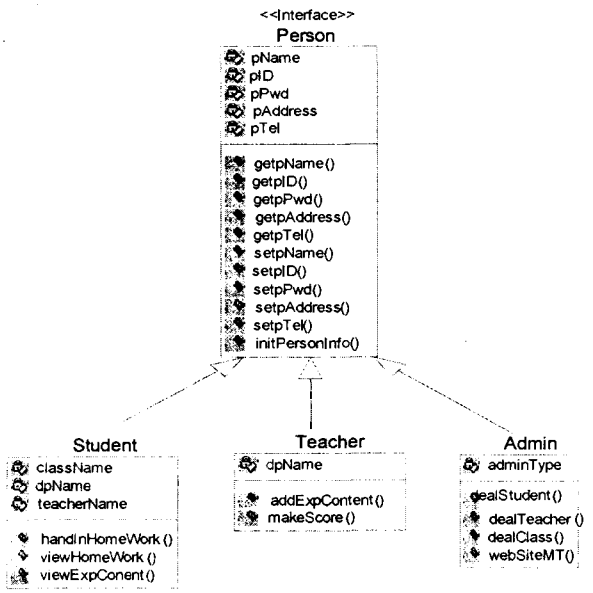

图 3 自主实验平台的主要类图

基于 RBAC 的自主实验平台定义了四种角色:管理员(Admin)、教师(Teacher)、学生(Student)、游客(Guest)。在策略证书中表示为:

```
<RoleHierarchyPolicy>
<RoleSpec OID="1.2.826.0.1.3344810.1.1.14"
Type="permisRole">
<SupRole Value="Guest" />
<SupRole Value="Admin">
<SubRole Value="Guest" />
</SupRole>
<SupRole Value="Teacher">
<SubRole Value="Guest" />
</SupRole>
<SupRole Value="Student">
<SubRole Value="Guest" />
</SupRole>
</RoleSpec>
</RoleHierarchyPolicy>
```

2.2 Servlet Filter 设计

由于大部分的网络应用涉及到 Web 服务器的资源访问,因此这里访问控制执行模块 AEF 设计为一个 Web 代理。Web 代理拦截所有到 Web 服务器的 http 请求,这些 http 请求都是以 URL 的方式传递给 Web 代理。

文中将根据 Servlet 规范,采用过滤器机制 Filter 来实现 Web 代理。过滤器是 Servlet 2.3 中增加的一个新功能,过滤器可以改变一个请求或者是修改响应。过滤器不是一个 Servlet,它只是一个 Servlet 接受到请求前的预处理器。就是说,用户发送一个请求给 Servlet 时,在 Servlet 处理之前,这个过滤器首先执行,然后才是 Servlet 的执行。Java 的扩展包中定义了过滤器接口,通过编写过滤器接口的实现类,并在配置文件中正确设置过滤器的属性,就可以将过滤器部署到 Web 服务器上。过滤器将拦截所有的 Web 请求,并调用过滤器的 doFilter 方法来对请求进行预处理。在 doFilter 方法中对请求进行处理。

2.3 AEF 设计

Web 代理 AEF 的作用是截取用户的访问请求,把请求信息组合成访问控制决策模块 ADF 可以接收的决策请求参数,如用户的 LDAP DN、请求的目标和请求的操作信息,也可以包括一些环境变量。Web 代理提供了一个对用户认证的接口,并从该接口中获取用户的 LDAP DN。同时,Web 代理对用户发出的 http 请求进行分析处理,获取请求的目标和请求的操作参数。然后,Web 代理把相关参数包括用户的 LDAP DN、请求的目标和请求的操作信息发送到访问控制决策模块 ADF 策略服务器,并执行 ADF 的决策结果。

Web 代理 AEF 从策略服务器 ADF 获得信息并做出断言的步骤设计如下所示:

首先启动策略服务器,进行初始化,从 LDAP 服务器上的策略证书中读取策略,然后等待从 AEF 发送的断言请求。

1)用户发出访问请求,被 Web 代理截取,先对用户认证,认证通过后,获取用户的 LDAP DN。

2)Web 代理从用户的请求信息中获取请求的目标资源和请求的操作,和用户的 LDAP DN 一起发送给策略服务器。

3)策略服务器根据策略和用户角色分配证书包含的角色,对用户的访问请求作出同意或拒绝的决定,把决策结果发送给 Web 代理。

4)Web 代理执行决策结果,如果允许访问,则让用

户进行访问操作;如果禁止访问,则发送禁止访问的信息给用户。

2.4 策略服务器 ADF 设计

访问控制决策模块 ADF 的作用是根据 AEF 所传送过来的决策请求参数,以及授权策略来决定发起者是否对目标资源具有相应的操作权限。在文中,基于角色的访问控制模型,用户的角色存储在角色分配证书中,授权策略存储在策略证书中。因此,访问控制决策模块需要从策略证书中获取授权策略,并解析 XML 编码的授权策略,从角色分配证书获取用户的角色,最后作出决策。

策略获取的主要步骤设计如下:

1) 访问控制决策模块 ADF 通过配置文件指定的 LDAP[4]目录服务器地址定位存放策略证书的 LDAP 目录服务器。通常配置文件指定了 LDAP 目录服务器地址列表,可以包括一个以上的 LDAP 目录服务器地址,其中策略证书存放在 LDAP 目录服务器地址列表中的第一个 LDAP 目录服务器上。因此,ADF 只需要到 LDAP 目录服务器地址列表中的第一个地址检索策略证书。

2) 访问控制决策模块 ADF 在指定的 LDAP 目录服务器上根据配置文件指定的权威源 SOA 的 LDAP DN,检索策略证书,策略证书存放在权威源 SOA 的目录项的一个属性值中。

3) 判断策略证书的有效性,验证证书的签名是否有效、证书是否过期等。

4) 证书验证有效后,从策略证书中提取 XML 编码的策略,从中读取策略标识符 OID,跟配置文件中指定的策略 OID 进行比较,判断该策略是否是应用需要的访问控制策略。

5) 策略判断有效后,将策略文件表达成内部形式,产生策略规则,包括角色分配和权限分配等策略规则。

2.5 流程示例

当用户在客户端输入一个访问目标(TargetURL),比如 http://172.30.164.198:8080 /lab/index.jsp,这个 http 请求首先被 Filter 拦截,Filter 在 tomcat 的 web.xml 文件里,配置如下:

```
<filter>
<filter-name>AEF</filter-name>
<filter-class>permis.AEF</filter-class>
</filter>
<filter-mapping>
<filter-name>AEF</filter-name>
<url-pattern>*.jsp</url-pattern>
</filter-mapping>
```

这个 Filter 是一个 AEF,它获得了 Target,然后去 Target-Action-Mapping.xml 映射表中查找这个 URI 对应的操作(action),Target-Action-Mapping.xml 的文件格式如下:

```
<ParamList>
<TARGET-ACTION-MAPPING>
  <TARGET>/lab/index.jsp</TARGET>
  <ACTION>CommonRequest</ACTION>
</TARGET-ACTION-MAPPING>
</ParamList>
```

从这个 Target 到 Action 的映射中,可以知道"/lab/index.jsp"这个 URI 对应的 action 是 CommonRequest。

接下来获取用户 DN。如果是游客(Guest,即未登录用户),则系统分配一个公共的用户 DN:cn=guest1,ou=role,o=permis,c=gb;如果是一个已登录的用户,则从用户的从 session 中获取 DN。

最后把 TargetURL,action 和 userDN 传递给 ADF。

3 总　结

所开发的学生自主实验平台采用 MVC 三层构架,即把一个应用的输入、输出、处理流程按照 Model、View、Controller 的方式进行分离。经过文中设计实现的基于 Web 方式的 RBAC 系统已经作为独立的访问控制模块在自主实验平台上部署成功,经试用表现出较好的实用性。文中实现的 RBAC 访问控制模块还可以用于其他基于 Web 的应用系统进行访问控制。笔者将在后续的工作中研究在自主实验中加入上下文受限的 RBAC[5]访问控制,以期获得更好的动态访问控制效果。

参考文献:

[1] 尹　刚,王怀民,滕　猛. 基于角色的访问控制[J]. 计算机科学,2002,29(3):69-71.

[2] Sandhu R, Conyne E J, Lfeinstein H, et al. Role based access control models[J]. IEEE Computer, 1996,29(2):38-47.

[3] The PERMIS Consortium, Modular PERMIS Project[EB/OL]. 2005-05. http://sec.cs.kent.ac.uk/permis.

[4] Wahl M, Kille S, Howes T. RFC2253: Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names[S/OL]. IETF. 1997-12. http://www.ietf.org/rfc/rfc2253.txt.

[5] Crampton J. Specifying and enforcing constraints in role-based access control[C]// Proceedings of the eighth ACM symposium on Access control models and technologies. New York:ACM Press,2003:43-50.