RBAC 扩展 J2EE/JAAS 安全机制的设计与实现*

陈 阳, 佘 堃, 周明天
(电子科技大学 计算机学院 卫士通信息安全实验室, 四川 成都 610054)

摘 要: 提出了一种拓展 J2EE 平台安全性的设计,将基于角色的访问控制应用到 Web 应用中。设计方案将 Internet 上的各种资源抽象成 URI,采用 JAAS 和 Filter 技术集中管理,将开发阶段需要考虑的安全问题转移到了部署阶段,从而实现应用逻辑与安全逻辑分离的目的。归纳了面临的问题和解决办法,最后给出了典型环境下的应用。

关键词: Java 认证与授权服务; J2EE; Filter; 基于角色访问控制; Web 安全
中图法分类号: TP393.08 **文献标识码**: A **文章编号**: 1001-3695(2005)01-0114-03

Research and Implementation of RBAC-based Security Architecture of J2EE

CHEN Yang, SHE Kun, ZHOU Ming-tian
(Information Secure United Laboratory of UESTC-Westone, School of Compuer Science & Engineering, University of Electronic Science & Technology of China, Chengdu Sichuan 610054, China)

Abstract: Here gives a design of expending security of J2EE architecture, which applied role-based access control to Web application on Internet. Web resources are considered as entitys identifiered by URI. An additional tire is applied to Web container to adapt 3th security products. The security issue considered in development phase can be moved to deployment.
Key words: JAAS; J2EE; Filter; RBAC; Web Security

电子政务逐渐兴起, J2EE 成为服务器端主流平台。面对国外厂商垄断 J2EE 产品的现状, 需要一种方案能够在 J2EE 中融合第三方的安全产品,提供完全自主的安全控制。基于角色的访问控制是当前备受关注的安全概念,在发展过程中它相对独立, 对资源的定义也很抽象。在实际应用中, 资源的概念必须具体化, 并希望能够在统一的框架中使用和管理。本文提出的方案将安全问题归结到一个中间层, 成为分布式系统运行期的安全服务, 向上提供应用程序的接口, 向下为第三方安全产品提供扩展。

1 面临的问题

在继续后面的表述之前约定部分术语: 基于角色大访问控制[1,2] (Role Based Access Control, RBAC) ; Java 认证与授权服务[3,4] (Java Authentication and Authorization Service, JAAS) ; 敏感资源, 即根据访问者权限进行分配的的资源。

RBAC 没有规定受保护资源的具体形式, 将 RBAC 应用于保护 J2EE 的各种资源, 将面临以下几个问题:

(1) 如何抽象 Web 应用中的资源。RBAC 是相对独立的概念, 其中关于资源的定义非常抽象。Internet 环境中存在多种具体资源, 简单地将 RBAC 拼合到应用中会产生冲突。统一资源标志符(Uniform Resource Identifier) 是 W3C 组织制定的国际标准, 标准的目的是通过 URI[5,6] 引用 Internet 上的各种资源, 包括文档、页面、图片、多媒体, 甚至是服务。基于 HTTP 协议的 Web 应用适合采用 URI 的资源表示方式[7], 设计中资源被抽象成 URI, 用户每次请求的 URI 是授权模块的参数。

(2) 如何扩展 J2EE 规范中的安全策略。应用服务器对于安全问题采用了开放但是很简单的策略。J2EE 规范中定义的“角色”仅仅是用户组的概念, 无法适应继承、互斥等复杂策略。J2EE 体系目前不支持 RBAC, 可通过在 Web 应用与 Web 容器之间增加一个中间层, 形成一个通用平台, 扩展安全接口。

(3) J2EE 环境中身份的认证需要扩展, 使其能够支持第三方的安全设备。

(4) 如何实现声明安全(配置安全) 。网络上已存在大量 Web 应用, 理想的情况下, 用户希望尽可能保持原有逻辑, 通过统一配置而不是逐个修改代码, 将 RBAC 应用到现存的 Web 应用上。这需要将应用逻辑与安全逻辑分离, 同时支持编程安全和声明安全。应用开发者可以把注意力集中在最关心的应用逻辑上, 而把安全配置放在部署阶段。底层安全设备与安全策略的变动不会影响上层的应用。

2 模块设计与实现

2.1 结构

整个设计建立在 J2EE 平台上, 核心内容是认证模块和授权模块。通常情况下, Web 应用运行在 J2EE 容器中, 只能得到容器提供的安全保护。如果加入第三方的 RBAC 工具和加密设备, 需要在 Web 应用与容器之间增加安全中间层, 包括认证与授权模块。中间层与 Web 应用一起部署到容器中, Web 应用通过 API 访问认证与授权模块提供的服务, 外部的 RBAC 工具和加密设备通过 SPI 挂接到 J2EE 环境(图 1) 中。

收稿日期: 2003-12-24; 修返日期: 2004-03-16
基金项目: 总装备部基金资助项目(51415010101DZ02)

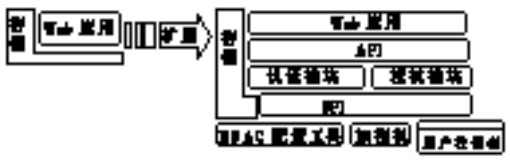

图 1　通过 RBAC 扩展 J2EE 应用前后

2.2　资源的抽象

在典型的 Internet 环境中, 应用程序或者用户通过 URI 访问远程服务器上的资源。本文将 Web 容器中的所有资源被抽象为 URI, 包括静态页面(HTML) 、动态页面(JSP/Servlet) 和多媒体数据。因此监视所有 HTTP 请求, 并从中提取被访问资源的 URI, 成为安全控制的第一步。

2.3　模块设计

系统主要的组件包括: Filter, LoginServlet, JAAS, API, SPI。

(1) Filter 组件能够拦截所有来自客户端的请求, 在请求得到处理之前增加一些处理。在此需仔细分离各种请求, 既要保证安全的缜密又要确保效率。这里使用了 Servlet 2. 3 提供的 Filter 新特性[8]。用户可通过配置文件改变 Filter 的工作属性, 其中一个重要的参数是 LoginPage。当系统发现此次请求超出了权限, 就会自动导向 LoginPage 制定的登录页面, Filter 的主要任务包括截获所有请求, 提取访问资源的 URL; 判断被访问资源是否是敏感、分流; 询问认证系统是否已经认证; 询问授权系统是否允许操作; 在声明安全中, 保存触发安全的 URL。

(2) LoginServlet 组件是一个通用的登录机制, 接收用户提交的认证信息, 包装成对象后交给下层。它有三个参数可以配置: Login_success_page 制定登录成功后自动转向何处; Login_failed_page 登录失败后自动转向何处; Session_time 会话的有效时间。客户通过自定义的登录页面提交认证信息, 因此登录页面需要遵循一定的规范。通常登录页面具有一定的通用性, 数量不会很多。LoginServlet 主要任务包括提取所有认证信息; 初始化认证系统; 执行登录方法 login( ) ; 创建会话对象 Session; 绑定 Subject 对象与 Session 对象。如果保存了 reqURL, 则转向 reqURL, 否则转向默认页。

(3) JAAS 模块提供了一个认证和授权服务的框架, 支持诸如多重认证模块等特性。设计中我们只采用它的一个子集。由于认证与授权都与下层实现有关, 如认证采用不同的算法, 不同的加密设备, 授权采用 RBAC 不同的实现版本, 应此需要 SPI 作为适配器。JAAS 在主要任务包括可插拔登录模块; 保存登录环境; 用户多次登录, 拥有多重身份; 实现与 RBAC 的协作。

(4) API 模块是中间的安全服务的接口, 用于支持编程安全。编程安全不能够完全避免, 有时资源不能全部提供, 也不能进一步细分, 需要在运行期根据访问者角色来选择部分或全部资源。API 主要任务为编程安全提供服务; 能够帮助应用程序判断当前调用者是否通过认证; 能够帮助应用程序完成登录, 在同一会话内保持有效; 封装本系统的实现, 使应用程序不能直接访问 LoginContext, Subject 以及 Session 等对象。

(5) SPI 模块为下层的安全实现提供一定的兼容性。尽管用户使用安全服务的方式变化不大, 底层的实现方式因环境而异。针对不同的认证系统和授权系统需要一定的代码开发量。SPI 主要任务包括分离认证逻辑与用户注册系统, 支持多种注册系统(File, OS, DB, LDAP) , 同时减少工作量。分离访问控制逻辑和访问控制策略, 即使 RBAC 向前发展仍能保持兼容, 也不需要改动 Web 应用。

2.4　流程设计(图 2)

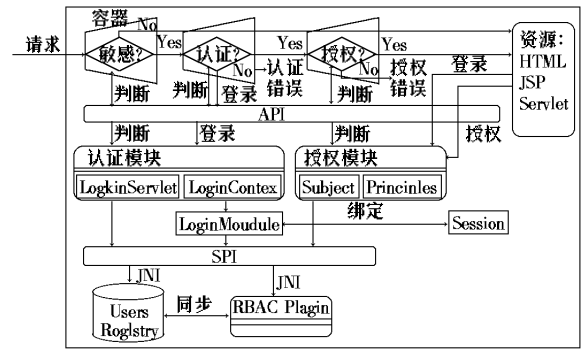

图 2　工作流程设计

请求进入 J2EE 应用服务器后, 要经过三个级联 Filter 进行筛选。第一 Filter 的工作是判断被访问的资源是否是敏感的。如果不是敏感资源就跳过认证与授权 Filter, 将控制权交给容器。由于认证与授权 Filter 开销比较大, 这样的设计能够显著提高效率。请求到达认证 Filter 后, 需要根据上下文判断当前会话是否通过了认证, 如果没用则强制转向认证页面。只有通过认证的请求(会话) 才能到达授权 Filter。授权 Filter 将用户信息和资源构成一个值对, 通过 API 传递给授权模块, API 的返回值决定这次访问是否合法。API 封装了下层的具体实现, 应用程序只需要传递必要的认证信息, 不必关心外部加密设备的类型和工作方式。过程描述如下:

```
访问请求被认证 Filter 截获;
flag1 = 访问认证模块, 资源是否受保护?;
//受到保护返回 true, 不受保护返回 false
if( flage1) {
flag2 = 判断用户是否登录?
//已登录返回 true, 没有登录返回 false
if( flag2) {
传递给授权 Filter;
flag3 = 访问授权模块, 用户是否有访问资源的权限?;
//具有权限返回 true, 否则返回 false
if( flag3) {
允许访问;
} else{
返回登录页面 login. jsp 要求用户登录, 或返回授权错误页面;
}
else{
返回登录页面 login. jsp;
      }
else{
允许访问;
}
```

序列图(图 3) 描述了整个过程中不同模块之间的调用关系。

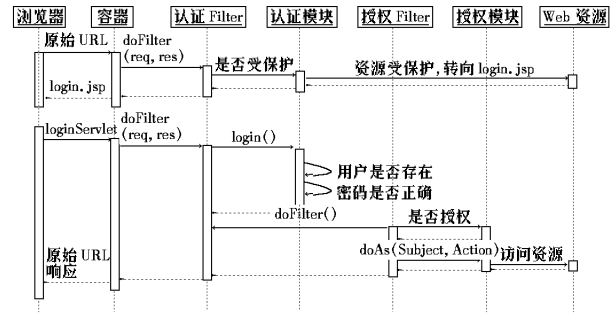

图 3　系统工作流程序列图

3　应用示例

以下将以一个网站的安全管理(图 4) 为例, 逐步阐述完成

一次客户访问的安全环节。这个试验性的网站建立在 Websphere 应用服务器上,安装了加密设备,提供了 Web Services,本系统采用的环境是用户信息服务器(LDAP),通过用户名和密码认证,授权策略通过 RBAC 工具集中管理。

(1) 服务器的安全管理员在 J2EE 服务器上部署安全平台,配置保存用户注册信息的 LDAP 服务器,配置是否使用加密机。

(2) 安全管理员通过用户管理工具增加、修改、删除用户注册信息。

(3) 安全管理员通过角色管理工具(RBAC 工具)增加、修改、删除角色,定义每个角色所具有的权限。如果此时系统已经拥有了注册用户,通过角色管理工作为用户指派需要的角色。所有设置工作被 RBAC 系统记录在配置文件中。

(4) 客户端通过浏览器发出 HTTP/SOAP 请求,到达应用服务的所有请求被安全平台捕获。以下流程进入安全平台内部。

(5) 安全平台询问 RBAC 系统被访问的资源是否受到安全保护。一般请求被放行,敏感的请求被截获。系统返回登录页面,要求用户提供认证信息。用户只需要登录一次,以后的访问系统可以自动识别。

(6) 用户可以选择是否采用加密机加密登录信息。如果采用,登录信息加密后才发送到 Internet 上。

(7) 安全平台获得用户登录信息后,通过调用 JAAS 提供的认证服务完成全部认证过程,包括调用加密机接口。认证的结果与会话绑定,用户下次访问不再需要登录。

(8) 用户通过认证后,安全平台调用 JAAS 提供的授权服务,判断用户是否有权访问指定资源。其中的实现细节将进入 RBAC 系统内部。如果授权成功,请求将被放行,否则返回授权错误。

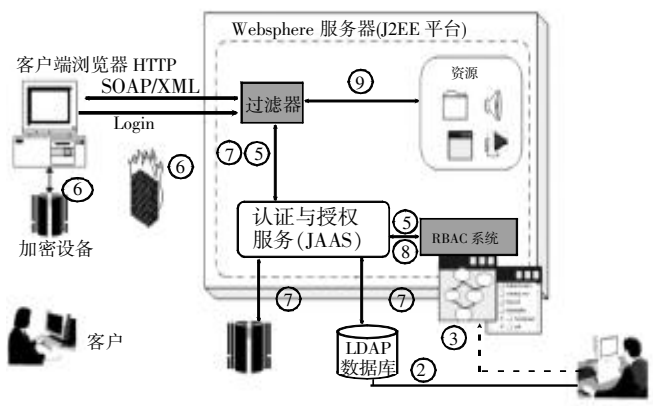

图 4 网站安全管理示意图

4 未来的工作

安全平台对所有的请求施加安全判断无疑会牺牲一定的效率,特别是在多个应用服务采用一个公共的认证授权服务器的情况下。为了避免安全成为性能的瓶颈,在集中式安全管理环境中授权模块可以采用分层策略,根据资源类别分层,也可以根据资源的敏感程度分层,甚至可以将 Filter 的工作提取出来,放在应用服务器之前形成安全网关。Web Services 的安全随着电子政务兴起成为关注的问题。Web Services 中的 SOAP 协议运行在 HTTP 协议之上,因此采用目前的设计保护 Web Services 在理论上是可行的。为了提供更好的访问控制粒度,需要将敏感资源进一步对象化,兼容不同应用的请求。

参考文献:

[1] Ferraiolo, *et al*. Role-based Access Control: Features and Motivations[C]. Annual Computer Security Applications Conference, IEEE Computer Society Press, 1995.

[2] R Sandhu, *et al*. Role-based Access Control Models[J]. IEEE Computer, 1996, 29(2).

[3] SUN Microsystems, Inc. Java™ Authentication and Authorization Service (JAAS) 1.0 Developers Guide[EB/OL]. http://java.sun.com/security/jaas/doc/api.html.

[4] Charlie Lai, Seema Malkani. Implementing Security Using JAAS and Java GSS. Worldwide Java Developer Conference[EB/OL]. http://java.sun.com/security/javaone/2003/2236-JAASJGSS.pdf.

[5] Berners-Lee, *et al*. Uniform Resource Locators (URL). RFC 1738 12/20/1994[EB/OL]. http://sunsite.icm.edu.pl/pub/doc/rfc/rfc1738.txt.

[6] Fielding. Relative Uniform Resource Locators. RFC 1808 06/14/1995[EB/OL]. http://sunsite.icm.edu.pl/pub/doc/rfc/rfc1808.txt.

[7] T Berners-Lee. Universal Resource Identifiers in WWW: An Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in W-W-W. RFC 1630 06/09/1994[EB/OL]. http://sunsite.icm.edu.pl/pub/doc/rfc/rfc1630.txt.

[8] SUN Microsystems, Inc. Java™ Servlet Specification version 2.3 [EB/OL]. http://www.jcp.org/aboutJava/communityprocess/final/jsr053/.

[9] 许春根,江于,严悍. 基于角色访问控制的动态建模[J]. 计算机工程, 2001, 27(1): 116-119.

[10] 叶锡君,许勇,等. 基于角色的访问控制在 Web 中的实现技术[J]. 计算机工程, 2002, 28(1): 167-169.

作者简介:

陈阳(1978-),男,四川人,硕士,主要研究方向为分布式计算与网络安全;佘堃(1969-),男,四川人,硕士生导师,主要研究方向为信息安全;周明天(1939-),男,广西人,教授,博士生导师,主要研究方向为分布式计算与信息安全。

(上接第 113 页)

参考文献:

[1] un WD, *et al*. An Artificial Immune System Architecture and Its Applications[J]. IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences, 2003, E86A(7): 1858-1868.

[2] Harmer PK, *et al*. An Artificial Immune System Architecture for Computer Security Applications [J]. IEEE Trabsactuibs on Evolutionary Computation, 2002, 6(3): 252-280.

[3] 张彦超,阙喜戎,王文东. 一种基于免疫原理的网络入侵检测模型[J]. 计算机工程与应用, 2002, (10): 159-161.

[4] Anil Somayaji, Stephanie Forrest. Automated Responsed Using System-Call Delays [C]. Proceedings of the 9th USENIX Security Symposium, Denver, Colorado, USA, August 2000. 14-17.

[5] 肖人彬,王磊. 人工免疫系统:原理、模型、分析及展望[J]. 计算机学报, 2002, 25(12): 1281-1293.

[6] 王风先,刘振鹏,李继民,等. 一种仿生物免疫的计算机安全系统模型[J]. 小型微型计算机系统, 2003, 24(4): 698-701.

[7] Anil Buntwal Somayaji. Operating System Stability and Security through Process Homeostasis[D]. Dissertation for the Degree of Doctor of Philosophy Computer Science. The University of New Mexico, Albuquerque, New Mexico, 2002.

作者简介:

孙照焱(1975-),男,湖北十堰人,博士研究生,主要研究方向为快速并行数据存储处理、附网存储设备及信息安全;董永贵(1965-),男,河北滦县人,副研究员,博士学位,主要研究方向为非接触式位移传感器、石英晶体化学传感器、振动信号分析与处理、信息存储与传输;贾惠波(1945-),男,河北辛集人,副院长,教授,主要研究方向为精密仪器、光存储技术;冯冠平(1946-),男,江苏人,院长,教授,硕士,主要研究方向为传感器与智能仪器。