基于JAAS和Struts的MIS安全性的实现方法研究

李 磊，杜平安，刘孝保，姜 伟

(电子科技大学机械电子工程学院 成都 610054)

【摘要】描述了基于Java认证与授权服务和Struts程序开发框架的企业MIS系统安全性的分析与实现过程；提出了一种三层结构的安全模型；采用将数据库管理系统和Web容器的角色权限体系相统一的方法，分别在浏览器层、Web容器层和数据库层采用协同的安全策略，实现了基于角色访问控制的MIS系统安全性。结果表明结合Java认证与授权服务与Struts框架较好地满足了MIS系统安全性的各种要求，不仅降低了系统运行期安全性维护的难度，而且提高了Web应用程序开发的效率。

关 键 词 Java认证与授权服务； MIS系统； 基于角色访问控制； 安全模型； Struts框架

中图分类号 TP309.2 文献标识码 A

Implementation of MIS Security Based on JAAS and Struts

LI Lei，DU Ping-an，LIU Xiao-bao，JIANG Wei

(School of Mechatronics Engineering, University of Electronic Science and Technology of China Chengdu 610054)

Abstract A 3-layer model based on Java Authentication & Authorization Service (JAAS) and Struts framework is presented to insure the security of Manufacturer Information System (MIS). In this model, the identical roles for both Database Management System (DBMS) and web container are defined. A coordinate security architecture is utilized to realize the Role-Based Access Control (RBAC) of MIS at web browser, web container and database, respectively. The results indicate that the implementation of this model meets various security requirements for MIS, and also makes it easier to administrate the system security during the runtime. In addition, the Struts framework enhances the development for web application effectively.

Key words JAAS; manufacturer information system; role-based access control; security model; Struts

随着制造业信息化的深入，如何保证管理信息系统运行的安全，对于企业至关重要。传统的基于B/S结构的应用程序安全性主要有两种实现方案：(1) 利用数据库自身的安全管理功能，根据建立数据库连接的用户身份限制对数据库资源的访问，但不能解决对Web资源的安全访问控制；(2) 依赖于Web容器所提供的安全性服务。目前流行的J2EE Web容器提供了基于角色访问控制模型(RBAC)的安全保护机制，通过判断用户角色来控制对容器中的JSP/Servlet等Web资源的安全访问，从而控制对服务器中资源的安全访问。但需要单独为用户和用户权限建立映射，通常用XML文件或者关系型数据库等来存储。在Web程序的业务逻辑层进行数据库访问时，为了提高性能，常采用统一的数据库连接或配置连接池，但很难实现权限的区分，不利于数据库资源的保护。

在某油田企业MIS系统的开发中，本文提出了一种三层安全性模型，统一了数据库的角色权限体系和Web容器的角色权限体系，实现了MIS系统多个层面的安全性。该模型以Java认证与授权服务(Java Authentication & Authorization Service)为核心，在容器层实现了系统安全性与程序业务逻辑的彻底分离，为系统运行提供了身份验证与资源授权功能；并结合Struts开发框架快速实现了基于MVC设计模式的Web应用程序结构，提高了开发效率。

1 JAAS安全框架

JAAS是SUN公司提出的在Java平台上实现以用户为中心的安全性框架[1-4]。JAAS框架包括认证(Authentication)与授权(Authorization)。认证取得当前运行代码用户的身份；授权则根据身份决定是否授予访问资源的权限。通过扩展JAAS通用框架，可实现自定义的认证与授权策略。

1.1 JAAS认证

JAAS采用了一种标准可插接认证框架方式(Pluggable Authorization Service)，允许Java应用程序

收稿日期：2005– 09 – 08
作者简介：李磊(1981 –)，男，硕士，主要从事CAD/CAE/CAM和制造业信息化等方面的研究.

保持与底层身份验证技术之间的独立，使应用程序开发者更大程度地将精力集中到实现程序的业务逻辑中，并可在应用程序中插入新的或者经过更新的身份验证技术，而无需修改应用程序本身。在Java平台下使用JAAS认证的实现过程如下[5]：

(1) 应用程序初始化登录上下文(LoginContext)对象实例；

(2) 登录上下文实例请求配置器加载为该应用程序配置的登录验证模块(LoginModule)实例；

(3) 应用程序通过登录上下文实例调用登录验证模块的Login方法，对该用户身份进行认证；

(4) 登录模块首先创建回调对象实例(CallbackHandler)，回调对象通过使用一个或多个CallBack对象与用户进行交互，获得登录模块所需的认证信息并进行认证。如果配置的所有登录模块登录验证成功后，通过调用Commit方法将特定的身份(Principal)和凭证(Credential)加入代表当前用户的Subject；如果在Login失败，调用abort方法来清除已执行的操作；

(5) 如果验证成功，应用程序通过登录上下文返回的结果获得用户的Subject，通过该Subject进行对敏感资源的访问。

1.2 JAAS授权

JAAS授权的实现过程如下：

(1) 配置基于身份的策略描述文件，其格式为：

grant <signer(s) field>, <codeBase URL>
<Principal field(s)> {

permission perm_class_name "target_name", "action"; …};

(2) 程序经过JAAS认证后取得代表用户身份的subject，调用Subject类的静态方法doAs或doAsPrivileged；

(3) doAs方法中将subject对象和执行敏感操作的类的实例作为参数传入，并将subject对象加入当前线程的访问控制上下文；doAsPrivileged则将subject加入传入的访问控制上下文对象中；

(4) 执行敏感操作的action实例被调用。action调用AccessController类的静态方法checkPermission，对比包含在访问控制上下文中的权限特征与策略文件中的权限配置，确定是否进入敏感操作区；

(5) 授权后应用程序执行action中的敏感操作。

2 基于MVC设计模式的Struts框架

Struts是基于Java Servlet和JavaServer Pages技术、用于开发Web应用程序的开放源码的程序框架，采用Model-View-Controller设计模式，减弱了业务逻辑接口和数据接口之间的耦合。用Struts框架开发Java Web程序时，多个视图能共享一个模型，模型响应用户请求并返回响应数据，视图负责格式化数据呈现给用户，业务逻辑和表示层分离，同一个模型可以被不同的视图重用，大大提高了代码的可重用性。

2.1 模型层

模型用于处理软件要涉及的业务范围，主要包括业务数据模型和业务逻辑模型。Struts中主要依靠Action、ActionForm、EJB或者JavaBean来处理业务逻辑。

2.2 视图层

视图在软件中通常是用户交互的界面，用于表示数据和提交用户对软件的请求。在Struts中视图通过一组JSP实现，这些JSP程序中不包含业务逻辑，也不包括模型的信息。模型的信息通过控制器传递，在Struts中ActionForm也被作为视图的一部分。

2.3 控制器层

在Struts中，ActionServlet发挥了一个控制器的作用，用于接受用户请求，并调用模型中的处理方法，然后选择相应的视图。ActionServlet是一个通用的控制组件，提供了处理所有发送到Struts的HTTP请求的入口，截取和分发这些请求到相应的动作类(这些动作类都是Action类的子类)。另外控制组件也负责用相应的请求参数填充Action Form，并传给动作类。动作类实现核心控制逻辑，它可以访问java bean 或调用EJB。所有这些控制逻辑利用Struts config.xml文件来配置。

2.4 Struts配置文件 struts-config.xml

该文件实现了控制器和视图的耦合。通过配置，控制器可以选择对应的视图，视图能通过对应的控制器查询到模型中的数据。在该配置文件中每个Action都对应一个<Action>标签，用来映射Action类的信息。

3 MIS系统安全模型的实现

该企业MIS系统的实际情况为: 同时在线用户少于200人；较少涉及海量数据的存取；明确的岗位责任分工和对业务数据的敏感性，对系统安全性提出较高要求。根据上述要求，本文为该MIS系统设计了三层结构安全性模型，如图1所示。系统选用BES6.5(Borland Enterprise Server)作为程序运行的Web容器，数据库使用Oracle9i。

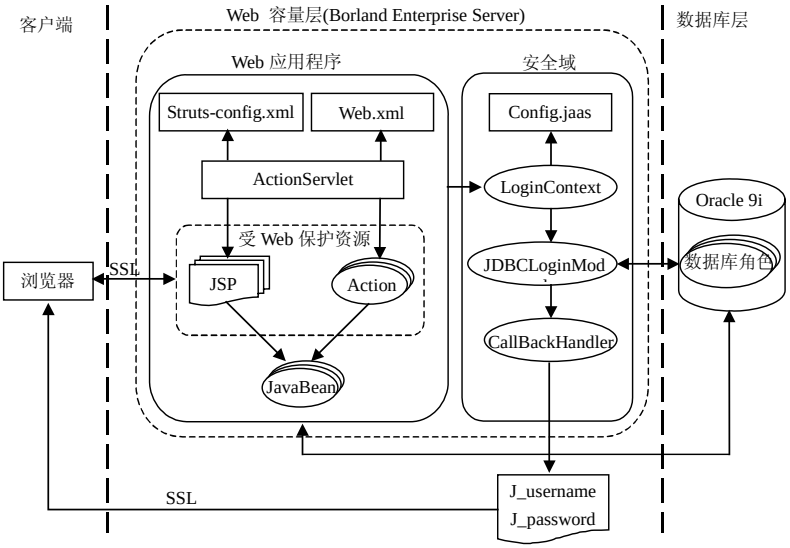

图1　MIS系统的三层安全模型

3.1 客户端安全

面向用户采用了基于角色权限的动态菜单，在用户界面层中限制用户对敏感资源的访问，实现过程如下：

(1) 按照系统的功能模块与企业内部各岗位职责的划分，在数据库端定义相应的角色。同时将各自所涉及的数据库对象，包括各种表、视图、存储过程和序列等分别授予对应的角色；

(2) 建立表示用户操作菜单与数据库角色的映射关系的数据库表。用户登录并通过身份验证以后，系统查询该用户所拥有的角色列表。并根据操作菜单与角色映射关系，动态显示该用户的操作菜单；

(3) 用户通过浏览器采用安全套接字协议(Server Socket Layer，SSL)与Web服务器通信。SSL使用基于证书的加密技术实现会话双方信息的安全传递，确保信息传递的保密性、完整性。

3.2 Web容器层安全

符合J2EE规范的Web容器提供声明性安全和编程性安全两种安全机制[6]。采用Web容器提供的声明性安全模型与指定的JAAS认证授权模块相结合，实现对Web资源的保护。实现过程如下：

1) 配置Web应用程序的发布描述符Web.xml，设置程序使用Web容器所提供的表单认证方式，以及受保护Web资源列表和能够访问这些资源的安全角色清单：

```
<login-config>
<auth-method>FORM</auth-method>
<form-login-config>
<form-login-page>/login.jsp</form-login-page>
<form-error-page>/error.jsp</form-error-page>
</form-login-config>
</login-config>
    …
 <security-constraint>
 <display-name>login</display-name>
 <web-resource-collection>
 <web-resource-name>login</web-resource-name>
 <description/>
 <url-pattern>/index.jsp</url-pattern>
 <http-method>POST</http-method>
    <http-method>GET</http-method>
    </web-resource-collection>
    <auth-constraint>
    <role-name>BASIC_ROLE</role-name>
    </auth-constraint>
 <user-data-constraint>
 <transport-guarantee>NONE</transport-guarantee>
 </user-data-constraint>
 </security-constraint>
    …
```

服务器端使用BES提供的JDBCLoginModule类作为JAAS登录验证模块。它使用标准JDBC数据库接口，根据数据库系统的用户和角色来验证用户。配置BES安全域和JAAS登录配置文件config.jaas，指

定所使用的登录验证模块[7-8]。

2) 当试图访问受保护的Web资源被时，Web容器就会激活为该资源配置的认证与授权机制，过程如下[9]：

(1) 如果对受限资源首次访问，则容器通过重定向到部署描述符中定义的注册页面，要求用户提供以用户名与密码作为身份凭证。基于表单验证的注册页面包含两个特殊文本域j_username和j_password，分别对应用户输入的用户名和密码。

(2) 根据Struts-config.xml配置的路径，请求被ActionServlet传递到一个Action对象中。该Action将用户名与密码转发到一个特殊的URL请求j_security_check，该请求负责调用验证方法对用户进行验证。

(3) 根据Web容器安全域的配置，JAAS安全管理器首先实例化一个LoginContext对象，它负责调用JDBCLoginModule认证模块的login方法，并通过CallbackHandler获取用户输入的用户名和密码；然后根据用户名和密码判断该用户是否为数据库的合法用户。如果用户身份合法，则查询Oracle数据库的系统视图[DBA_ROLE_PRIVS]，获取用户所拥有的数据库角色列表，并将其加入到用户的Subject中。Subject被维持在HTTP会话对象中，实现了用户的单点登录。

(4) 该认证模块将用户的数据库角色与Web资源访问的安全角色进行比较，容器负责判断得到认证的用户是否被授权访问发布描述符中的Web资源。如果通过验证，授权用户访问受保护的Web资源；如果身份验证失败，自动跳转到错误提示页面。

3.3 数据库层安全

采用每个用户独立的数据库连接，并将数据库用户和密码作为用户登录系统的用户名和密码。用户登录验证后，根据用户名和密码实例化一个独立的JDBC数据库连接对象，并与当前该用户对应的HTTP会话对象绑定[10]。当Struts模型层的JavaBean与数据库进行交互时，首先从会话对象中取得该数据库连接对象的引用，利用该连接对象访问数据库。尽管采用每个用户独立的连接比数据库连接池损失了一定的性能，但是不同用户拥有的数据库角色权限严格限制了用户对数据库对象的访问，并且用户对数据库的操作也能被数据库的审计日志中真实完整地记录下来。

4 结 束 语

通过项目的实施证明，本文设计的三层结构的安全模型实现了基于JAAS的安全框架和Struts Web程序框架，程序员在项目开发过程中无须关心权限。将权限管理和业务处理隔离不仅简化了程序开发，而且极大地提高了Web应用程序开发的效率，降低了MIS系统运行期安全性维护的难度。使用同一套角色权限体系在三个层面实现协同的安全策略，更有效地保护了系统和数据免受非法攻击。

参 考 文 献

[1] 胡海燕, 贺贵明. J2EE的安全机制及其应用研究[J]. 计算机应用, 2003, (S2): 153-154.
[2] 沈 耀, 陈昊鹏, 李新颜. EJB容器中基于JAAS的安全机制的实现[J]. 计算机应用与软件, 2004, 21(09): 61-65.
[3] 赵仲孟, 沈海斌. 王 瑞. J2EE应用服务器安全服务体系的分析与实现[J]. 计算机工程与应用, 2003, 39(21): 175-179.
[4] BODOFF S. J2EE1. 4标准教材[M]. 第2版. 北京: 电子工业出版社, 2005.
[5] 孙卫琴. 精通Struts: 基于MVC的Java Web设计与开发[M]. 北京: 电子工业出版社, 2004.
[6] PISTOIA M, NAGARATNAM N. Enterprise Java security: building secure J2EE applications[M]. 北京: 清华大学出版社, 2005.
[7] 刘孝保, 杜平安. J2EE模式下基于角色的访问控制的应用研究[J]. 计算机应用, 2006, 26(6): 1331-1333.
[8] 张志立, 张 鹏, 齐德昱. 基于J2EE的Web应用开发中安全问题的研究[J]. 武汉理工大学学报(交通科学与工程版), 2005, (2): 300-303.
[9] 罗 锐, 程文青. Java安全体系在Web程序中的研究和应用[J]. 计算机应用研究, 2006, 23(7): 114-115.
[10] 刘孝保, 杜平安. 基于角色的访问控制在多应用层CIMS中的应用[J]. 四川大学学报(工程科学版), 2007, 39(2): 140-144.

编 辑 黄 莘