·安全技术· 文章编号：1000-3428(2018)11-0135-05 文献标志码：A 中图分类号：TP311

Shiro 框架在 Web 系统安全性上的改进与应用

易文康，程 骅，程耕国

（武汉科技大学 信息科学与工程学院，武汉 430081）

摘 要：针对 Web 系统中普通用户越权访问未授权的系统资源以及未授权用户非法访问系统资源带来的信息安全问题，分析越权访问的发生原理，利用基于角色的访问控制技术和 Shiro 框架的授权机制，同时结合文本设计权限访问控制算法，实现对系统各模块不同权限灵活、安全的管理。分析结果表明，该算法能有效阻止越权访问，提高系统安全性。

关键词：授权；越权访问控制；信息安全；角色；权限

中文引用格式：易文康，程 骅，程耕国. Shiro 框架在 Web 系统安全性上的改进与应用[J]. 计算机工程，2018，44(11)：135-139.

英文引用格式：YI Wenkang，CHENG Hua，CHENG Gengguo. Improvement and application of Shiro framework in Web system security[J]. Computer Engineering，2018，44(11)：135-139.

Improvement and Application of Shiro Framework in Web System Security

YI Wenkang，CHENG Hua，CHENG Gengguo

（Institute of Information Science and Engineering，Wuhan University of Science and Technology，Wuhan 430081，China）

【Abstract】For information security issues that the normal users unauthorized access to unauthorized system resources and unauthorized users access to system resources illegally in Web system，this paper analyzes the principle of the Broken Access Control(BAC)，uses the access control technology based on the role and authorization mechanism of the Shiro framework，and proposes an authority access control algorithm，which realizes flexible and safe management to different modules in the system. Analysis result shows that this algorithm can effectively prevent the BAC and improve security of the system.

【Key words】authorization；Broken Access Control(BAC)；information safety；role；authority

DOI：10.19678/j.issn.1000-3428.0048417

0 概述

随着互联网技术的飞速发展，学校、企业和政府机关等机构所建立的信息管理系统，其管理的对象日益复杂，用户所能访问的数据资源日趋庞大，导致用户的权限管理和日常维护变得越来越繁琐。为在实现资源信息共享的同时，避免出现授权用户越权访问未授权的系统资源以及未授权用户非法访问系统资源等一系列 Web[1] 系统中的信息安全问题[2]，需要进一步深入研究基于角色的访问控制技术[3-4]和 Shiro 的授权机制，确保用户对系统数据资源的访问都是经过授权的，并防止非法用户的访问。这对保证系统中数据资源的安全性、保密性和完整性是非常必要的。本文主要研究如何运用 Shiro 安全框架[5]来阻止 Web 系统中的越权访问控制(Broken Access Control，BAC)。

越权访问[6]即跨越权限访问，是 Web 应用中一种常见的安全缺陷，其产生原因是没有对访问用户提交的数据参数进行权限检查或者缺少跨域访问的限制。越权访问可以直接绕过基础的网络安全服务防御，它通过前端对数据或者参数进行请求构造、遍历，在取得完整的 Web 应用程序或者数据库权限前就可以得到相关用户的个人信息。若没有对越权访问采取有效的解决或防御措施，将极有可能泄露客户个人、企业和政府等机构的重要信息，存在很大的网络信息安全隐患，造成不可估量的经济损失。

针对Web系统访问控制中越权访问的安全

基金项目：国家自然科学基金(61304129)；湖北省教育厅科学技术研究项目(q20121107)；武汉科技大学基金(2012x2009)。

作者简介：易文康(1991—)，男，硕士研究生，主研方向为 Web 系统安全；程 骅，副教授；程耕国，教授。

收稿日期：2017-08-21 **修回日期：**2017-11-15 **E-mail：**924523558@qq.com

隐患,研究者提出了各种解决方案,其中 Shiro 框架逐渐受到人们的重视。本文通过使用 Shiro 框架的授权方式和一种权限访问控制算法来解决此安全问题。

1 基于 Shiro 的 Web 数据库访问

1.1 Shiro 框架简介

Shiro 是 Apache 系列的一个 Java 开源安全开发框架[7],其本身具有良好的健壮性和易用性。Shiro 提供了认证、授权、会话管理、加密等功能,可以为学校、企业和政府机关等机构提供信息安全解决方案。同时,Shiro 本身集成了许多保护 Java 应用的特性,如支持 Web 应用[8]、线程和并发、缓存机制和测试工具等。与其他安全框架相比,Shiro 的安全认证和授权方式比较简洁且容易操作,编写的代码量显著减少。本文主要研究如何将 Shiro 的授权功能应用在 Web 网络安全问题中,解决越权访问的问题。

1.2 数据库访问过程

当用户登录系统并进行数据操作访问 Web 数据库[9]时,可能由于本身系统设计上存在的安全漏洞[10]或者用户权限分配和管理上存在的缺陷,导致授权用户越权访问未授权的数据库资源或者未授权用户非法访问数据库资源。为了阻止这些安全隐患的发生,可在 Web 应用中集成 Shiro 安全框架。

首先在 Web 工程里的 web. xml 配置文件中定义 Shiro Servlet 过滤器实现 Web 应用和 Shiro 框架的集成。该过滤器会过滤用户发送来的所有请求,根据实际需求执行特定的逻辑判断,当请求满足一定要求时才允许通过,从而保证用户在进行 Web 数据库访问时,会先对用户的访问权限进行判断,若拥有相应的权限,则执行对应的业务逻辑操作,访问数据库并得到数据。数据库访问流程如图 1 所示。

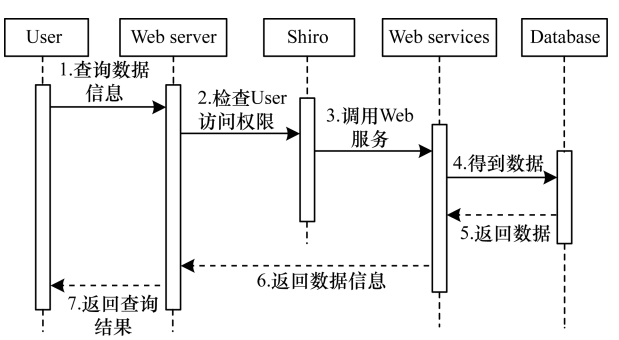

图 1 基于 Shiro 的数据库访问流程

2 基于 Shiro 框架的访问控制原理

2.1 Shiro 访问控制的实现

授权,其实质就是访问控制[11-12],控制系统用户能够访问 Web 系统中的哪些资源,能执行哪些操作(如访问页面、编辑数据等),这些一般是通过系统管理员分配给用户的角色和权限来决定的。Shiro 支持权限的概念,能友好地与 Web 系统集成[13],还能够通过运用通配符灵活地对权限进行匹配和检查。目前,Shiro 提供以下 3 种方式的授权:

1)编程式授权。通过编写 if/else 等一系列 Java 的授权代码块来实现当前用户的授权操作,例如:

```
Subject currentUser = SecurityUtils. getSubject( );
if ( currentUser. hasRole( " read" ) ) {
//有角色" read" 权限
} else {
//没有角色" read" 权限}
```

2)JSP/GSP 标签式授权。在 JSP/GSP 页面通过相应的标签来实现授权,例如:

```
< shiro:hasRole name = " read" >
< ! — 拥有角色" read" 权限 — >
< /shiro:hasRole >
```

3)注解式授权。在执行的 Java 方法上放置相应的注解来实现系统用户的授权操作(前提是开发的系统需要支持面向切面的编程[14]),例如:

```
@ RequiresPermissions( " user:read" )
//拥有角色" user" 中的" read" 权限
void someMethod( );
```

以前访问数据资源时,需要先对访问的 url 进行权限配置,在 spring-shiro. xml 中编写大量的 Java 代码,与最开始的 Hibernate 框架相同,需要编写大量的 hbm 配置文件,显得十分冗余,不利于后期维护,因而迫使开发者们逐渐选择注解形式来进行程序的开发和授权等操作。本文采用基于 Shiro 框架的 JSP/GSP 标签式授权和注解式授权来实现权限操作,配置非常简洁方便,显著减少了授权部分的代码量,同时还方便后期的管理和维护。

2.2 Shiro 权限注解

以如下 Shiro 权限注解为例进行分析:

```
@ RequiresPermissions( " user:find" , " user:add" , … )
void someMethod( );
```

注解@ RequiresPermissions 要求当前登录用户必须同时具备 user:find 和 user:add 权限时,才能继续执行注解下面的方法 someMethod( ),否则系统将会抛出异常 AuthorizationException。

在 Web 系统中,该注解作用于 Java 方法上,表示对访问这些方法的用户进行访问拦截,当判断出用户拥有该权限时才能继续访问,使得在用户访问后台数据资源时,先判定其是否拥有访问此方法的权限,对阻止越权访问有一定的作用。但在设计 Web 系统时,可能在开发授权这块考虑不周或者部分业务逻辑代码遗漏权限判断功能,导致用户在未授权的情况下越权访问这部分数据,甚至非法访客通过一些黑客技术来非法访问系统数据库中的数据资源,泄露了客户的个人隐私和一些敏感重要的商业信息,造成无法挽回的损失。

为避免越权访问情况的发生,本文在 Shiro 提供的权限注解@ RequiresPermissions 基础上,提出一个作用于 Java 类上的改进的权限注解。

2.3　权限注解作用范围的扩展

2.3.1　实现方法

根据 Shiro 提供的作用于 Java 方法的权限注解@ RequiresPermissions,本文提出一个改进的作用于 Java 类的权限注解@ ClassPermissions,实现步骤为:

1)编写一个 ClassPermissions. java,在其中定义基于类的权限注解@ ClassPermissions:

```
public @ interface ClassPermissions {
String[ ] value( );}
```

2)在 Spring[15]配置文件 spring-shiro. xml 中添加:

```
<bean class = "com. xxx.
ClassAuthorizationAttributeSourceAdvisor" >
<property name = "securityManager" ref = "securityManager"/ >
</bean >
```

3)继承 AuthorizingAnnotationHandler 类,将构造函数中 new 的对象替换成自己的 AOP 实现。关键代码如下:

```
public ClassAuthorizingAnnotationHandler( ) {
super( ClassPermissions. class) ;}
@ Override
public void assertAuthorized ( Annotation an ) throws
AuthorizationException {
ClassPermissions per = ( ClassPermissions) an;
String[ ] permissions = per. value( ) ;
getSubject( ). checkPermissions( permissions) ;
return;} }
```

4)把 spring 所提供的 AopAllianceAnnotations-AuthorizingMethodInterceptor 重写一个自己的,修改权限验证拦截器栈的设置,修改权限验证的拦截器。关键代码如下:

```
interceptors. add( new ClassAnnotationMethod
Interceptor( resolver) ) ;
//自定义
interceptors. add( new ClassAuthorizingAnnotation
MethodInterceptor( ) ) ;
```

5)实现所需的权限验证拦截器,根据 spring 所提供的类 AuthorizingAnnotationMethodInterceptorl 重写一个自己的类。关键代码如下:

```
publicClassAuthorizingAnnotationMethod-
Interceptor ( ) {
super( new ClassAuthorizingAnnotationHandler( ) ) ;}
public ClassAuthorizingAnnotationMethod-
Interceptor( AnnotationResolver resolver) {
super( new ClassAuthorizingAnnotationHandler( ) ,
resolver) ;}
```

6)实现所需的权限处理器,继承 spring 提供的类 AuthorizationAttributeSourceAdvisor,实现作用于类的权限注解,获取类上的注解以及类下所有方法的注解。关键代码如下:

```
private static final Class <? extends Annotation > [ ]
annotationClass = new Class[ ] {//注解权限
ClassPermissions. class,
RequiresPermissions. class,……} ;
//匹配带有注解的方法
@ Override
public boolean matches( Method m,Classc) {
boolean flag = super. matches( m,c) ;
//若方法上没有权限注解,则获取类上权限注解
if( ! flag && isAuthzAnnotationPresent( c) && isWebAnnota
tionPresent( m) ) {
flag = true;}
return flag;}
//查看方法上是否有权限注解
private boolean isAuthzAnnotationPresent( Method m) {
for( Class <? extends Annotation > ann;
annotationClass {
Annotation a = AnnotationUtils. findAnnotation( m,ann) ;
if ( a ! = null) { return true;}
} return false;}
```

2.3.2　类与方法之间的安全作用机制

对集成了 Shiro 框架的 Web 系统各个功能模块的 url 访问都需要角色权限。当用户访问时,后台会根据前端的 url 地址定位到所对应方法上的权限注解。spring 会先扫描 Shiro 注解类的 matches 方法(上述第 6 个步骤),并通过返回 true/false 的方式来判断某个方法是否带有 Shiro 权限注解。若返回 true,则再判断当前访问操作是否满足方法上的注解权限;若返回 false,表示方法上没有权限注解,则会去获取所属类上的权限注解@ ClassPermissions,再判断当前访问操作的权限。

3 授权策略的设计与实现

3.1 权限访问控制算法

结合上节中实现的作用于 Java 类上的权限注解@ ClassPermissions,本文提出一个基于 Shiro 标签式授权和注解式授权的权限访问控制算法。

算法 基于 Shiro 标签和注解的访问控制算法

输入 访问者的请求 request = < user,messege > , user = {username,password,roleId}

输出 访问者得到的视图

步骤 1 根据请求 request 获得 user 对应的角色信息,并通过角色 ID (useId)获取 Shiro 中保存的对应的权限信息。

步骤 2 Shiro 过滤器拦截所有的请求,对每个请求进行权限判断。

步骤 3 若是前端请求,判断相应方法上的 JSP/GSP 权限标签。若拥有该权限,则跳转至步骤 7,否则跳转至步骤 8。

步骤 4 若是后台请求,判断相关 Java 方法上是否有权限注解。若有,则跳转至步骤 5,否则跳转至步骤 6。

步骤 5 若拥有权限注解@ RequiresPermissions 中标识的权限,则跳转至步骤 7,否则跳转至步骤 8。

步骤 6 获取并判断所属 Java 类上的权限注解@ ClassPermissions,若拥有注解中标识的权限,则跳转至步骤 7,否则跳转至步骤 8。

步骤 7 允许访问,获取数据库中的匹配信息,然后跳转至步骤 9。

步骤 8 禁止访问,抛出异常。

步骤 9 输出经过权限过滤后的数据信息。

3.2 授权流程

用户登录时发出请求 request = < user,messege > ,其中,user 包含账号(username)、密码(password)和角色 ID(roleId)的信息,messege 包含操作数据库的信息。Shiro 过滤器会拦截所有的请求并做判断:若是访问前端 JSP/GSP 页面,则使用标签进行权限控制;若是访问后台,则使用注解进行权限控制。

基于 Shiro 的授权流程如下:首先系统会调用 Subject. isPermitted(" 权限串") 方法,然后委托给 securityManager 进行处理,通过 securityManager 内部的 Authorizer(默认是实现 ModularRealmAuthorizer)来进行真正的授权处理。ModularRealmAuthorizer 会调用 Realm 的授权方法 doGetAuthorizationInfo,从数据库查询权限数据,返回 ModularRealmAuthorizer,ModularRealmAuthorizer 会调用 PermissionResolver 进行权限串比对。若从 Realm 中获取的当前 Subject 的角色信息(即权限串)与传入的 isPermitted(" 权限串") 相匹配,则返回 ture,有访问权限;否则返回 false,没有访问权限。

对于后台的访问请求,系统会先定位到具体的 Java方法上,并查找方法上的权限注解,例如@ RequiresPermissions(" user:find") 。若方法上存在权限注解,则按照上述 Shiro 的授权流程来进行授权处理;若方法上没有权限注解,则会去获取所在类上的权限注解@ ClassPermissions(" user:find") 再进行授权处理,从而实现对后台访问请求的权限控制。

综上所述,基于权限访问控制的用户授权流程如图 2 所示。

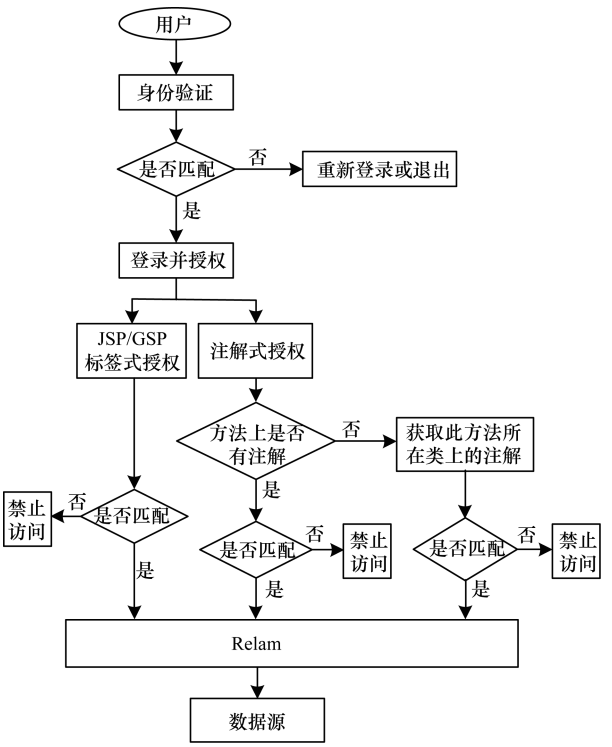

图 2 基于权限访问控制的用户授权流程

3.3 实现效果

3.3.1 安全性

一个集成了 Shiro 的 Web 系统,基本每个功能模块的 Controller 层都有同样的 add、list 等方法,这些方法不处理业务逻辑,只是把请求从 Controller 层转到 Service 层处理完后,将结果再转给相应的视图。此时只需在 Controller 层编写抽象类,并实现这些方法,再使用@ RequestMapping 标记此类。其他需要实现增删改查功能的 Controller 都继承此抽象

类,每个类只需编写自己的视图地址即可。此时,需要注解@ RequiresPermissions 能实现对类的注解。但是,根据文献[5]和 Apache Shiro 官网可知,该注解只能作用在方法上,导致其无法在 Controller 层的抽象类上实现权限验证功能。

本文提出的权限注解@ ClassPermissions 可针对某个类进行注解,解决了上述技术难题。其相应的权限访问控制算法与 Apache Shiro 官网提供的@ RequiresPermissions的权限验证过程相比,相当于增加了 3.1 节中步骤 4 和步骤 6 的判断,避免了开发过程中某些方法上缺少权限验证的问题,提高了系统的安全性。同时,不再需要对 Controller 层的所有方法进行注解,节省了大量的开发时间,使得授权更加灵活,便于管理和维护。

3.3.2　有效性

本文采用某公司的角色信息管理系统进行效果测试。该系统类似于文献[13]中的系统,其采用了 B/S 体系结构,以 Spring + SpringMVC + Hibernate 开源框架设计开发,整合了 Shiro 框架,使用 MySQL 数据库。

创建一个没有任何操作权限的普通用户,测试 100 个 url:http://localhost:8080/spring/rest/…①…/…②…,其中:①表示系统的功能模块名;②表示每个模块中增删改查等方法的方法名,观察并记录网页的输出结果;然后在各个功能模块的 Controller 层上实现权限注解@ ClassPermissions,再观察输出结果。实验结果对比如表 1 所示。由对比数据可知,实现类上的注解后可有效地阻止越权访问。

表 1　访问结果对比

实现方式	测试的 url 个数	允许访问的 url 个数	禁止访问的 url 个数
实现方法上的注解	100	12	88
实现类上的注解	100	0	100

3.3.3　普适性

只要开发的 Web 应用系统支持面向切面的编程,即可很容易地实现作用于 Java 类的权限注解@ ClassPermissions,从而实现方法和类上的权限控制。因此,本文算法在实际开发应用中具有很好的普遍适用性。

4　结束语

针对 Web 系统中的越权访问问题,本文提出一种基于 Shiro 安全框架的权限访问控制算法,通过协同使用作用于 Java 方法和类上的权限注解,实现整个 Web 系统后台的授权。与之前仅采用 Shiro 本身提供的权限注解来实现系统授权相比,该算法可有效阻止越权访问的发生,提高系统的安全性,同时降低开发者工作的复杂度,使操作更加灵活方便。下一步工作将继续运用 Shiro 和其他安全框架来解决 Web 系统中存在的各种安全问题。

参考文献

[1] 雷　敏,刘晓明,张　鸿,等. 面向 Web 信息系统安全威胁和风险评估分析[J]. 北京邮电大学学报,2016,39(增刊):87-93.

[2] HUANG H C,ZHANG Z K,CHENG H W,et al. Web application security: threats, countermeasures, and pitfalls[J]. Computer,2017,50(6):81-85.

[3] ZHANG Y S,WU M F,WU L,et al. Attribute-based access control security model in service-oriented computing[J]. Lecture Notes in Electrical Engineering,2014,163:1473-1479.

[4] 庞希愚,王　成,仝春玲. 基于角色-功能的 Web 应用系统访问控制方法[J]. 计算机工程,2014,40(5):144-148.

[5] 徐孝成. 基于 Shiro 的 Web 应用安全框架的设计与实现[J]. 电脑知识与技术,2015(16):93-95.

[6] 杨　静,季新生,刘彩霞. 基于角色访问控制的 HSS 数据库越权访问防护[J]. 电子技术应用,2011,37(4):145-148.

[7] 高秀慧,高建华. 基于 J2EE 框架的 Web 应用可靠性研究[J]. 计算机工程与设计,2013,34(4):1270-1275.

[8] LI X,XUE Y. A survey on server-side approaches to securing Web applications[J]. ACM Computing Surveys,2014,46(4):1-29.

[9] XU Z. Performance optimization of Web database application program based on JDBC[M]//ANGRISANI L,ARTEAGA M,PANIGRAHI B K,et al. Lecture Notes in Electrical Engineering. Berlin,Germany:Springer,2014.

[10] 王　丹,赵文兵,丁治明. Web 应用常见注入式安全漏洞检测关键技术综述[J]. 北京工业大学学报,2016,42(12):1822-1832.

[11] 贺正求,张叶琳,许俊奎,等. Web 服务访问控制策略研究[J]. 计算机应用,2015,35(8):2184-2188.

[12] 李怀明,王慧佳,符　林. 基于组织的 Web 服务访问控制模型[J]. 计算机工程,2014,40(11):65-70.

[13] 宋成明. 基于 Shiro 的某高校科研信息管理系统的设计与实现[J]. 智能计算机与应用,2017,7(4):62-63.

[14] SCHAEFER C,HO C,HARROP R. Introducing Spring AOP[M]. [S.l.]:Apress,2014:147-197.

[15] 薛　峰,梁　锋,徐书勋,等. 基于 Spring MVC 框架的 Web 研究与应用[J]. 合肥工业大学学报(自然科学版),2012,35(3):337-340.

编辑　金胡考