深入浅出 Spring Security

一场系统安全之旅

江南一点雨

华为云MVP、华为云云享专家

目录

Spring Security 简介

缘起

Spring Security最早叫Acegi Security，这个名称并不是说它和Spring就没有关系，它依然是为Spring框架提供安全支持的。Acegi Security基于Spring，可以帮助我们为项目建立丰富的角色与权限管理系统。Acegi Security虽然好用，但是最为人诟病的则是它臃肿烦琐的配置，这一问题最终也遗传给了Spring Security。

Acegi Security最终被并入Spring Security项目中，并于2008年4月发布了改名后的第一个版本Spring Security 2.0.0。

Spring Security 简介

Spring Security Vs Shiro

和Shiro相比，Spring Security重量级并且配置烦琐。其实，自从Spring Boot推出后，就彻底颠覆了传统了JavaEE开发，自动化配置让许多事情变得非常容易。在一个Spring Boot项目中，我们甚至只需要引入一个依赖，不需要任何额外配置，项目的所有接口就会被自动保护起来了。在Spring Cloud中，很多涉及安全管理的问题，也是一个Spring Security依赖两行配置就能搞定，在和Spring家族的产品一起使用时，Spring Security的优势就非常明显了。

因此，在微服务时代，我们不需要纠结要不要学习Spring Security，我们要考虑的是如何快速掌握Spring Security，并且能够使用Spring Security实现我们微服务的安全管理。

Spring Security 简介

为什么选择 Spring Security

不同于其他领域，在Java企业级开发中，安全管理方面的框架非常少，一般来说，主要有三种方案：

- Shiro
- Spring Security
- 开发者自己实现

Shiro 本身是一个老牌的安全管理框架，有着众多的优点，例如轻量、简单、易于集成、可以在JavaSE环境中使用等。不过在微服务面前，它无法充分展示自己的优势。

也有开发者选择自己实现安全管理，不过一个系统的安全，不仅仅是登录和权限控制这么简单，我们还要考虑各种各样可能存在的网络攻击以及防御策略，从这个角度来说，只有大公司才有足够的人力物力去支持这件事情。

Spring Security作为Spring家族的一员，在和Spring家族的其他成员进行整合时，具有其他框架无可比拟的优势，同时对OAuth2有着良好的支持，再加上Spring Cloud对Spring Security的不断加持，让Spring Security成为微服务项目的首选安全管理方案。

Spring Security 核心功能

对于一个安全管理框架而言，无论是Shiro还是Spring Security，最核心的功能，无非就是如下两方面：

- 认证
- 授权

通俗点说，认证就是身份验证（你是谁？），授权就是访问控制（你可以做什么？）。

Spring Security 核心功能

认证

Spring Security支持多种不同的认证方式，主要有如下几种：

表单认证。
OAuth2.0认证。
SAML2.0认证。
CAS认证。
RememberMe自动认证。
JAAS认证。
OpenID去中心化认证。
Pre-Authentication Scenarios认证。
X509认证。
HTTP Basic认证。
HTTP Digest认证。

Spring Security 核心功能

认证

作为一个开放的平台，我们还可以通过引入第三方依赖来支持更多的认证方式，同时，如果这些认证方式无法满足我们的需求，我们也可以自定义认证逻辑，特别是当我们和一些“老破旧”的系统进行集成时，自定义认证逻辑就显得非常重要了。

Spring Security 核心功能

授权

无论采用了哪种认证方式，都不影响在Spring Security中使用授权功能。Spring Security支持基于URL的请求授权、支持方法访问授权、支持SpEL访问控制、支持域对象安全（ACL），同时也支持动态权限配置、支持RBAC权限模型等，总之，我们常见的权限管理需求，Spring Security基本上都是支持的。

Spring Security 核心功能

其他

在认证和授权这两个核心功能之外，Spring Security还提供了很多安全管理的“周边功能”，这也是一个非常重要的特色，例如：

- 密码加密
- RememberMe
- 会话固定攻击防御
- CSRF 防御
- Http 防火墙
- ...

Spring Security 整体架构

认证和授权

在Spring Security的架构设计中，认证（Authentication）和授权（Authorization）是分开的，无论使用什么样的认证方式，都不会影响授权，这是两个独立的存在，这种独立带来的好处之一，就是Spring Security可以非常方便地整合一些外部的认证方案。

在Spring Security中，用户的认证信息主要由Authentication的实现类来保存，当用户使用用户名/密码登录或使用Remember-me登录时，都会对应一个不同的Authentication实例。

Spring Security中的认证工作主要是由AuthenticationManager接口来负责，在该接口中通过authenticate方法来做认证。

AuthenticationManager最主要的实现类是ProviderManager，ProviderManager管理了众多的AuthenticationProvider实例。

在一次完整的认证流程中，可能会同时存在多个AuthenticationProvider，多个AuthenticationProvider统一由ProviderManager来管理。同时，ProviderManager具有一个可选的parent，如果所有的AuthenticationProvider都认证失败，那么就会调用parent进行认证。

Spring Security 整体架构

认证和授权

在Spring Security的授权体系中，有两个关键接口：

AccessDecisionManager
AccessDecisionVoter

AccessDecisionVoter是一个投票器，投票器会检查用户是否具备应有的角色，进而投出赞成、反对或者弃权票；AccessDecisionManager则是一个决策器，来决定此次访问是否被允许。

Spring Security 整体架构

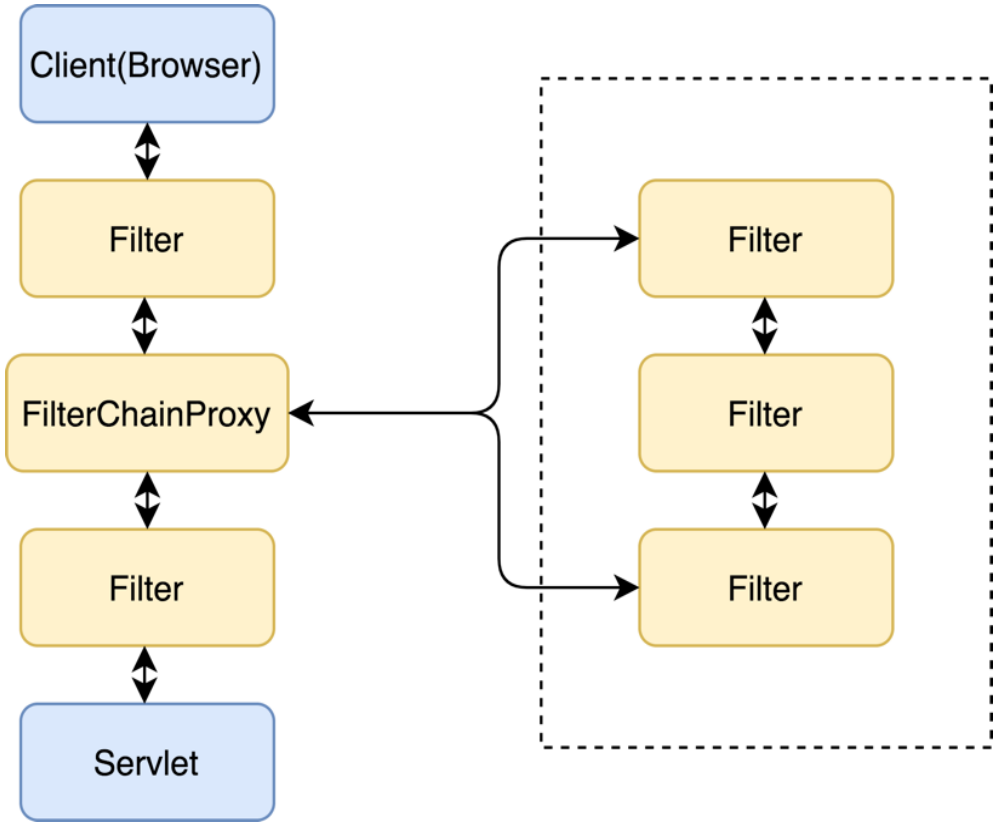
Web 安全

在Spring Security中，认证、授权等功能都是基于过滤器来完成的。开发者所见到的Spring Security提供的功能，都是通过这些过滤器来实现的，这些过滤器按照既定的优先级排列，最终形成一个过滤器链。开发者也可以自定义过滤器，并通过@Order注解去调整自定义过滤器在过滤器链中的位置。
需要注意的是，默认过滤器并不是直接放在Web项目的原生过滤器链中，而是通过一个FilterChainProxy来统一管理。

Spring Security 整体架构

Web 安全

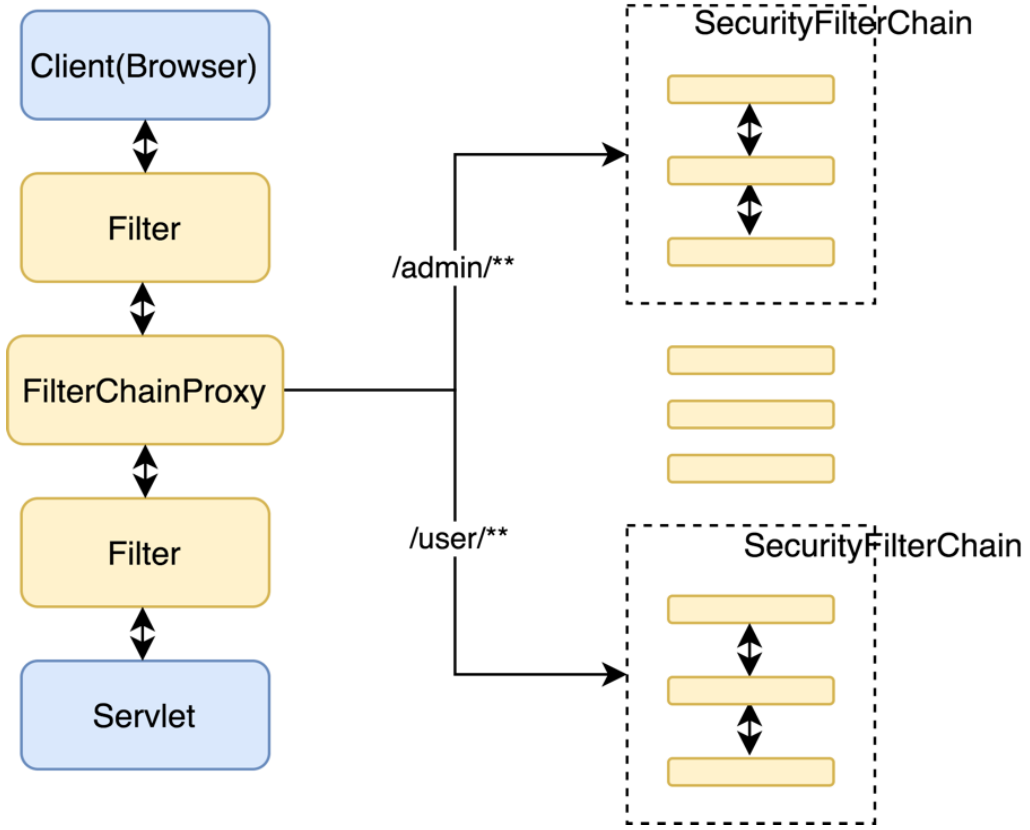
Spring Security中的过滤器链通过FilterChainProxy嵌入到Web项目的原生过滤器链中。



Spring Security 整体架构

Web 安全

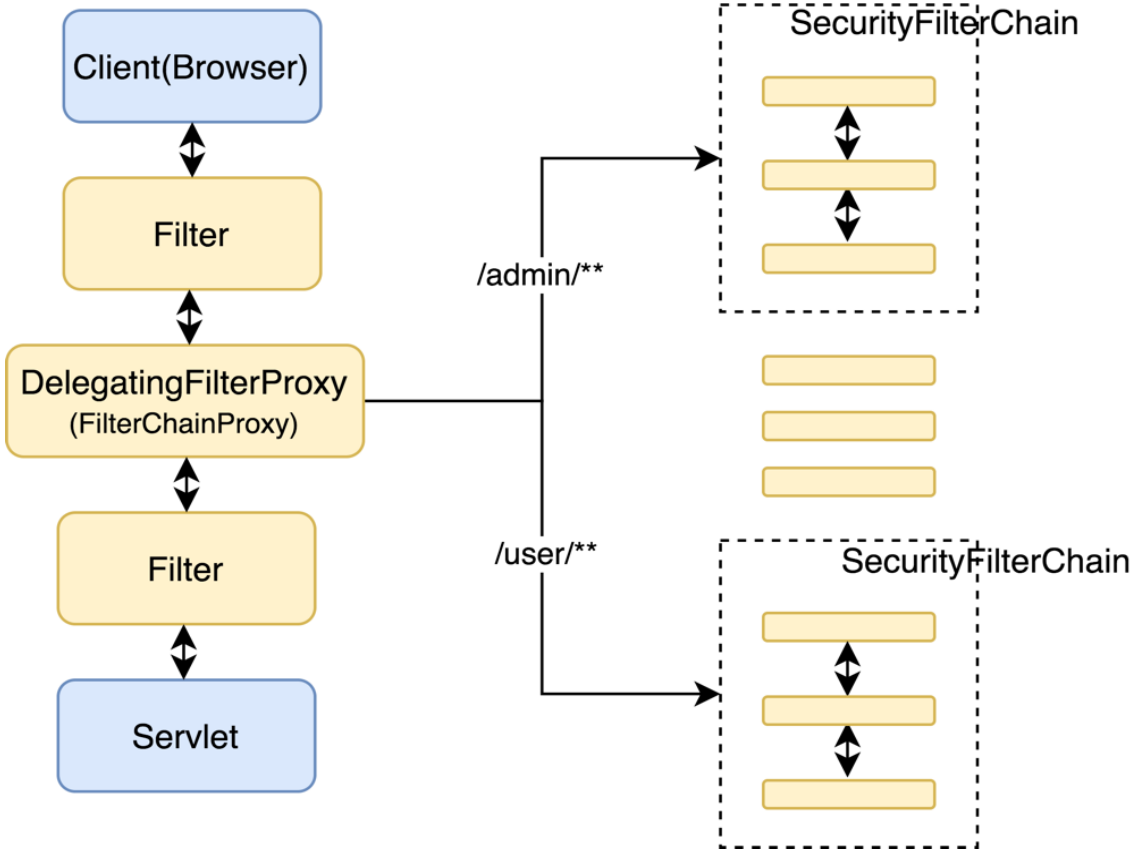
在Spring Security中，这样的过滤器链不仅仅只有一个，可能会有多个。当存在多个过滤器链时，多个过滤器链之间要指定优先级，当请求到达后，会从FilterChainProxy进行分发，先和哪个过滤器链匹配上，就用哪个过滤器链进行处理。



Spring Security 整体架构

Web 安全

FilterChainProxy作为一个顶层管理者，将统一管理Security Filter。FilterChainProxy本身将通过Spring框架提供的DelegatingFilterProxy整合到原生过滤器链中。



谢 谢