

华为云·云享读书会



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

- 1 为什么需要密码加密
- 2 加密方案进化史
- 3 PasswordEncoder
- 4 加密方案自动升级



为什么需要密码加密



CSDN密码泄露事件

2011年12月21日,有人在网络上公开了一个包含600万个CSDN用户资料的数据库,数据全部为明文储存,包含用户名、密码以及注册邮箱。事件发生后CSDN在微博、官方网站等渠道发出了声明,解释说此数据库系2009年备份所用,因不明原因泄漏,已经向警方报案,后又在官网发出了公开道歉信。在接下来的十多天里,金山、网易、京东、当当、新浪等多家公司被卷入到这次事件中。整个事件中最触目惊心的莫过于CSDN把用户密码明文存储,由于很多用户是多个网站共用一个密码,因此一个网站密码泄漏就会造成很大的安全隐患。由于有了这么多前车之鉴,我们现在做系统时,密码都要加密处理。



加密方案进化史



密码加密"打怪升级"之路

- Hash 算法
- 密码加盐
- 自适应单向函数



PasswordEncoder



常见实现类

- BCryptPasswordEncoder
- Argon2PasswordEncoder
- Pbkdf2PasswordEncoder
- SCryptPasswordEncoder



PasswordEncoder



DelegatingPasswordEncoder

DelegatingPasswordEncoder是一个代理类,而并非一种全新的密码加密方案。 DelegatingPasswordEncoder主要用来代理不同的密码加密方案。为什么采用DelegatingPasswordEncoder 而不是某一个具体加密方式作为默认的密码加密方案呢?主要考虑了如下三方面的因素:

- (1)兼容性:使用DelegatingPasswordEncoder可以帮助许多使用旧密码加密方式的系统顺利迁移到Spring Security中,它允许在同一个系统中同时存在多种不同的密码加密方案。
- (2) 便捷性:密码存储的最佳方案不可能一直不变,如果使用DelegatingPasswordEncoder作为默认的密码加密方案,当需要修改加密方案时,只需要修改很小一部分代码就可以实现。
- (3) 稳定性:作为一个框架, Spring Security不能经常进行重大更改,而使用Delegating Password Encoder可以方便地对密码进行升级(自动从一个加密方案升级到另外一个加密方案)。

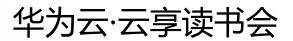


加密方案自动升级



"老破旧"系统整合利器

如果开发者使用了DelegatingPasswordEncoder,只要数据库中存储的密码加密方案不是 DelegatingPasswordEncoder中默认的BCryptPasswordEncoder,在登录成功之后,都会自动升级为 BCryptPasswordEncoder加密。在同一种密码加密方案中,也有可能存在升级的情况。







谢谢