系统核心配置项加密解决方案

目录

[系统核心配置项加密解决方案 1](#_Toc468889670)

[背景 1](#_Toc468889671)

[实现方案 1](#_Toc468889672)

[加解密工具 2](#_Toc468889673)

[windows平台下如何使用该加密工具？ 3](#_Toc468889674)

[1 配置项加密/解密 3](#_Toc468889675)

[2 获取密码字段摘要 3](#_Toc468889676)

[unix平台下如何使用该加密工具？ 3](#_Toc468889677)

# 背景

对系统安全稍微有点要求的系统，一般都会核心的配置项数据进行加密，尤其是在配置文件放在用户目录场景下。这类配置项一般包括但不限于数据库连接信息、核心资源服务器的连接信息(比如LDAP服务器)等。

# 实现方案

DES算法已经能够满足当前的安全需求，后续如果需要更高级别的加密算法，可以自行扩展，通过-Dsupport.crypt.coder={对称加密算法} 动态参数扩展。

用户账号密码字段的加密，当前采用MD5，主要考虑到各种前段技术对它支持比较全面，虽然该算法可以理论上被碰撞破解，但足以保障一般安全系统的运转。

配置文件如下：

support.auth.data.access.db.driver.classname=net.sourceforge.jtds.jdbc.Driver

support.auth.data.access.db.url=jdbc\:jtds\:sqlserver\://183.62.162.27\:21433/MNIS\_QM

**support.auth.data.access.db.username.crypt=oWeFY3ivVng**

**support.auth.data.access.db.password.crypt=oWeFY3ivVng**

需要加密的字段，使用.crypt后缀，配置文件加载会将读取的值进行解密。在spring版本较低的时候，一般通过扩展PropertyPlaceholderConfigurer来实现，但该组件已经被舍弃，为保持向前兼容，虽然还可以用，但是现在已经默认使用PropertySourcesPlaceholderConfigurer。（mnis、mnisqm还在使用旧式组件）。

工程中建议通过以下方式来加载配置文件。

<context:property-placeholder

properties-ref="cryptProperties"

local-override="true"

location="file:${user.home}/project-support-auth-config/support-auth-data-access.properties"

ignore-unresolvable="true"/>

<bean id="cryptProperties"

class="com.lachesis.support.common.util.bean.CryptPropertiesFactoryBean">

<property name="location"

value="file:${user.home}/project-support-auth-config/support-auth-data-access.properties" />

</bean>

这部分实现当前由support-common-util组件封装，其它工程需要引用时，只需要提供相关的配置路径。

<dependency>

<groupId>com.lachesis.support</groupId>

<artifactId>support-common-util</artifactId>

<version>1.0.0.0-SNAPSHOT</version>

</dependency>

# 加解密工具

考虑到有时需要重置配置项或用户的密码，提供了简单的命令行小工具，windows平台已验证过，unix下的脚本还没有经过测试。

## windows平台下如何使用该加密工具？

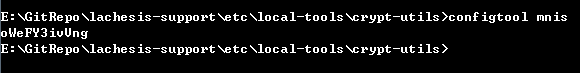
### 1 配置项加密/解密

a)加密

configtool[.CMD] 明文

[回车]

窗口输出该明文的密文

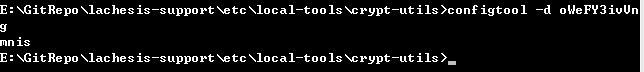


b)解密

configtool[.CMD] -d 密文

[回车]

窗口输出该密码的明文

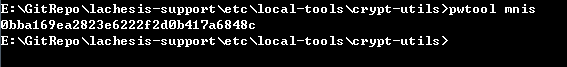


### 2 获取密码字段摘要

pwtool 明文密码值

[回车]

窗口输出该明文密码值的摘要，如



## unix平台下如何使用该加密工具？

类似windows平台，选择相应脚本