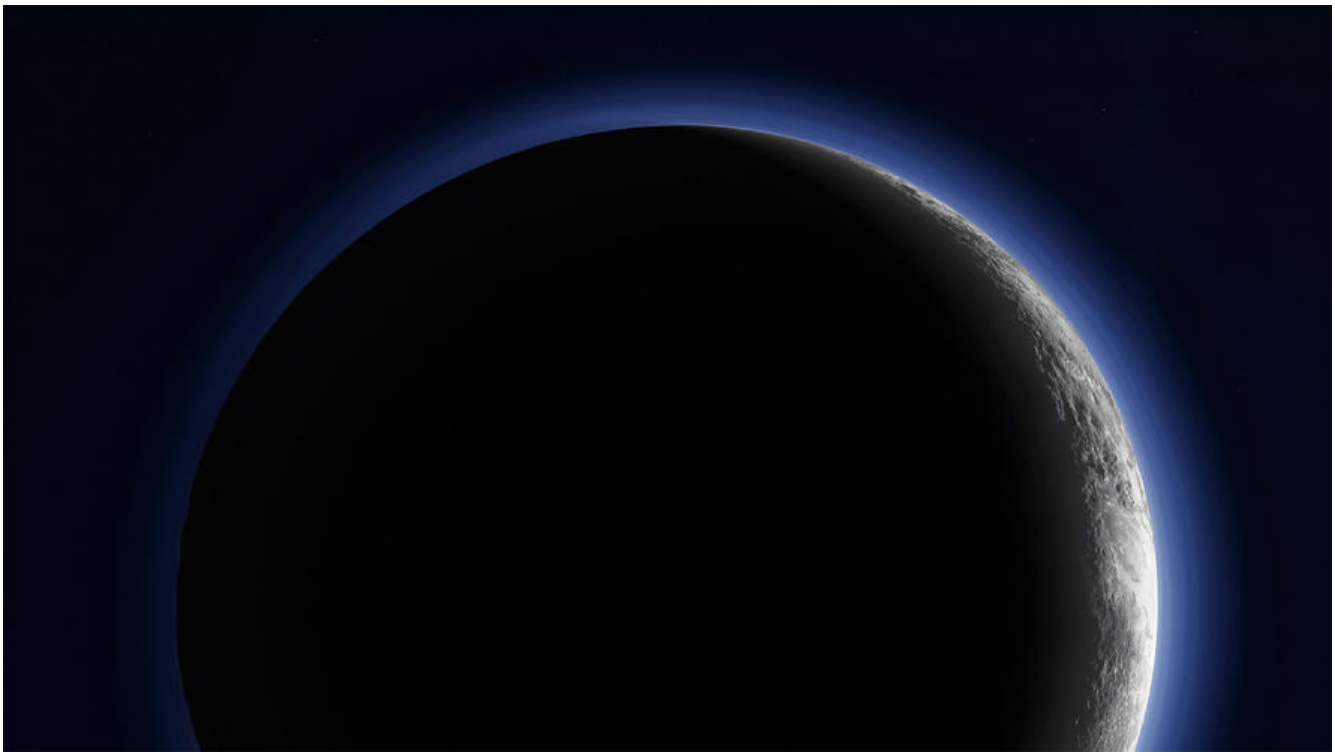
链滴

yaml 敏感数据加密

作者：sirwsl

原文链接：https://ld246.com/article/1632927960332

来源网站：链滴



故事

最近在出差，一直需求调研，下班后什么也不做，闲的有点蛋疼，每天浑浑噩噩，实在手痒痒了，想前段时间被朋友问到一个如何对yaml文件里的账号密码以及一些关键密钥进行加密。今天就来简单记一下。

jasypt进行yaml加密

场景重现

- 个人

我们在开发过程中，特别是个人项目，有时候在git的时候，一不小心把yaml文件里的一些配置的数库连接密码、以及一些密钥提交了，这时候就会被其他的加以利用，对方不友好，那你就完犊子了。

- 公司

一个公司代码安全性很重要，关键信息泄露更是严重，在开发人员的频繁流动中，很容易导致敏感信的泄露，然后被对手拿到了相关数据库、中间件、以及一些开发密钥，哦豁，去数据库、中间件中一捣鼓，perfect，这公司离凉凉不远了。

Jasypt是啥

简单点来说就是一个简单的加密工具，可以对你的关键信息进行加密，防止关键信息泄露，遭受对方接连接或者攻击。

如何使用

1、引入依赖

```
<dependency>
        <groupId>com.github.ulisesbocchio</groupId>
        <artifactId>jasypt-spring-boot-starter</artifactId>
        <version>3.0.3</version>
    </dependency>
```

2、配置yaml（可省略）

```
jasypt:
  encryptor:
    property:
      prefix: ENC(
      suffix: )
```

PS：就是简单配置一下前缀后缀，为了解析加密字符串用的

3、如何使用

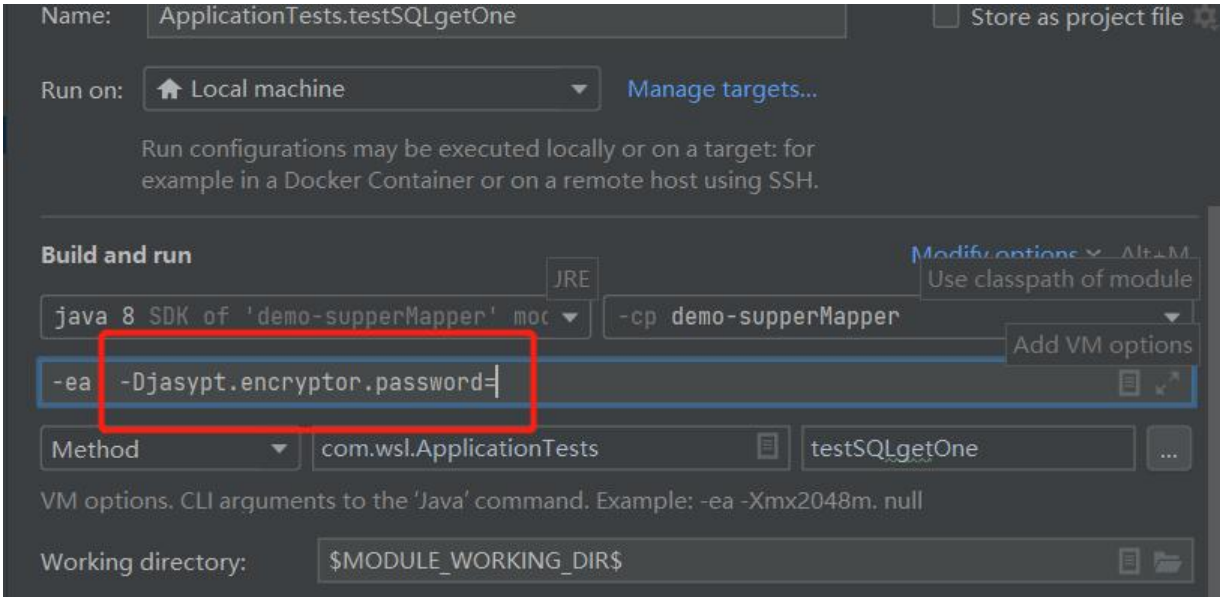
使用方式主要分为两种：配置yaml文件或者运行项目加入启动参数。

1）配置yaml（及其不推荐）。推荐在项目启动时加入启动参数：

```
jasypt:
encryptor:
property:
prefix: ENC(
suffix: )
password: # 你的密钥
```

推荐方式

VM options:-Djasypt.encryptor.password=xxx



原文链接：yaml 敏感数据加密

4、获取加密的数据

```
public class Jasypt {

    public static void main(String[] args) {
        BasicTextEncryptor textEncryptor = new BasicTextEncryptor();
        // 加密密钥
        textEncryptor.setPassword("sirwsl");
        // 要加密的数据（如数据库的用户名或密码）
        String username = textEncryptor.encrypt("root");
        String password = textEncryptor.encrypt("123");
        System.out.println("加密：username:" + username);
        System.out.println("加密：password:" + password);
    }
}
```

或者你可以配置jasypt的password后直接用DI进行

```
import org.jasypt.encryption.StringEncryptor;
import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;

import javax.annotation.Resource;

@SpringBootTest
class JasyptTest {

    @Resource
    private StringEncryptor stringEncryptor;
    @Test
    public void test() {
        //加密
        String username = stringEncryptor.encrypt("root");
        System.out.println("加密username: " + username);

        String decUsername = stringEncryptor.decrypt(username);
        System.out.println("解密username: " + decUsername);

        //加密
        String password = stringEncryptor.encrypt("123456");
        System.out.println("password: " + password);
        String decPassword = stringEncryptor.decrypt(password);
        System.out.println("解密password: " + decPassword);
    }

}
```

5、将获取到的加密数据配置到yaml中

```
# 应用名称
spring:
  application:
```

```
    name: demo-encryption
# 数据库驱动：
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://localhost:3306/demo?serverTimezone=UTC&useUnicode=true&characterncoding=UTF-8&zeroDateTimeBehavior=convertToNull&allowMultiQueries=true
    password: ENC(Y8CJa4AfPV+/snhdJ6ADg0wWuNQIJ1v2UQuyejJm7lPE76jdbr2I82rMvLRX2s9)
    username: ENC(cN5buxefuYaZBJ8/XCXWA3GAJdkPz0hBogIwG9uGjI8DH1v2oKLm1TQYD8ahX9A)

jasypt:
  encryptor:
    property:
      prefix: ENC(
      suffix: )
    #password: xxx
```

扩展

host实现地址隐藏

通常，在公司开发时，常通过配置HOST文件实现隐藏数据库或者其他地址，在开发时候也可以很好使用代理。win中host地址

C:\Windows\System32\drivers\etc

MyBatis-plus实现数据库加密

从MP的官网(https://baomidou.com/)上我们可以看到有说明：在版本3.3.2开始实现了数据安全。

配置

配置yaml

```
// 加密配置 mpw: 开头紧接加密内容 （ 非数据库配置专用 YML 中其它配置也是可以使用的 ）
spring:
  datasource:
    url: mpw:qRhvCwF4GOqjessEB3G+a5okP+uXXr96wcucn2Pev6Bf1oEMZ1gVpPPhdDmjQqo

    password: mpw:Hzy5iliJbwDHhjLs1L0j6w==
    username: mpw:Xb+EgsyuYRXw7U7sBJjBpA==
```

获取加密数据

```
// 生成 16 位随机 AES 密钥
String randomKey = AES.generateRandomKey();

// 随机密钥加密
String result = AES.encrypt(data, randomKey);
```

如何使用

配置启动参数

```
// Jar 启动参数（idea 设置 Program arguments，服务器可以设置为启动环境变量）
--mpw.key=d1104d7c3b616f0b
```

相关链接

MyBatis-Plus 数据安全保护：https://baomidou.com/guide/safety.html