

如何有效防止Java程序源码被人偷窥？[2] PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/244/2021_2022__E5_A6_82_E4_BD_95_E6_9C_89_E6_c104_244675.htm ClassLoader有几个重要的方法。创建定制的ClassLoader时，我们只需覆盖其中的一个，即loadClass，提供获取原始类文件数据的代码。这个方法有两个参数：类的名字，以及一个表示JVM是否要求解析类名字的标记(即是否同时装入有依赖关系的类)。如果这个标记是true，我们只需在返回JVM之前调用resolveClass。

【Listing 2：ClassLoader.loadClass()的一个简单实现】以下是引用片段：

```
public Class loadClass( String name, boolean resolve )
throws ClassNotFoundException { try { // 我们要创建的Class对象
Class clazz = null. // 必需的步骤1：如果类已经在系统缓冲之中
, // 我们不必再次装入它 clazz = findLoadedClass( name ). if
(clazz != null) return clazz. // 下面是定制部分 byte classData[] = /*
通过某种方法获取字节码数据 */. if (classData != null) { // 成功
读取字节码数据，现在把它转换成一个Class对象 clazz =
defineClass( name, classData, 0, classData.length ). } // 必需的步
骤2：如果上面没有成功， // 我们尝试用默认的ClassLoader装
入它 if (clazz == null) clazz = findSystemClass( name ). // 必需的步
骤3：如有必要，则装入相关的类 if (resolve &amp; clazz != null)
resolveClass( clazz ). // 把类返回给调用者 return clazz. } catch(
IOException ie ) { throw new ClassNotFoundException(
ie.toString() ). } catch( GeneralSecurityException gse ) { throw new
ClassNotFoundException( gse.toString() ). } }
```

 Listing 2显示了一个简单的loadClass实现。代码中的大部分对所有ClassLoader对象

来说都一样，但有一小部分(已通过注释标记)是特有的。在处理过程中，ClassLoader对象要用到其他几个辅助方法：

- findLoadedClass：用来进行检查，以便确认被请求的类当前还不存在。loadClass方法应该首先调用它。
- defineClass：获得原始类文件字节码数据之后，调用defineClass把它转换成一个Class对象。任何loadClass实现都必须调用这个方法。
- findSystemClass：提供默认ClassLoader的支持。如果用来寻找类的定制方法不能找到指定的类(或者有意地不用定制方法)，则可以调用该方法尝试默认的装入方式。这是很有用的，特别是从普通的JAR文件装入标准Java类时。
- resolveClass：当JVM想要装入的不仅包括指定的类，而且还包括该类引用的所有其他类时，它会把loadClass的resolve参数设置成true。这时，我们必须在返回刚刚装入的Class对象给调用者之前调用resolveClass。

三、加密、解密 Java加密扩展即Java Cryptography Extension，简称JCE。它是Sun的加密服务软件，包含了加密和密匙生成功能。JCE是JCA(Java Cryptography Architecture)的一种扩展。JCE没有规定具体的加密算法，但提供了一个框架，加密算法的具体实现可以作为服务提供者加入。除了JCE框架之外，JCE软件包还包含了SunJCE服务提供者，其中包括许多有用的加密算法，比如DES(Data Encryption Standard)和Blowfish。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com