

专家和您一起谈谈java加壳的问题 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/214/2021_2022__E4_B8_93_E5_AE_B6_E5_92_8C_E6_c104_214235.htm 在C世界，对代码进行加密，主要通过加壳的方式。所谓加壳，就是先将程序代码加密，然后用特定的程序加载器，将代码解密后加载进内存，这样可以在防止代码的反编译，当然，有加壳工具，也有解壳工具，虽然不能100%防止crack，但仍然给代码增加一层有力的保护。然而在java世界，保护代码是件很困难的事情，因为class文件非常规范，极易反编译，且反编译后的代码清晰可读。常见的保护办法是使用混淆器，打乱class和function以及变量的名字，可以干扰反编译后的代码的可读性。虽然简单提高了代码的安全性，但还仅仅相当于未加壳的C程序。java可以加壳吗?以前我认为这是不可能的，因为动态加载代码这样的内存级别的操作，java无法做到，除非使用JNI(Java Native Interface)，调用自己编写的C代码，在C代码中实现动态加载java代码。但是，C如何加载java代码呢?这需要JVM相当的了解。所以当时的我认为这是不可能的。然而，最近接触的一些知识告诉我java也可以加壳!!

1.URLClassLoader。用URLClassLoader可以在java程序的运行期间，再将文件夹或者jar加入到classpath中，这个特性事实上就是动态加载。既然可以动态加载class的文件夹或者jar，为何不可以加载加密后的classes呢，将classes用自己的方式加密，在URLClassLoader调用时，使用自己的方式解密。不就是达到了加壳的目的了吗?不过解密的代码放在何处是个问题，就是说解壳器的代码暴露在外，还是很危险的。 2.javassist。这

实在是一个非常神奇的新技术，我是在学习Tapestry5时第一次遇到这个包的，他可以动态创建java字节码，甚至可以修改你已经写好的函数，比如你的getter和setter，只是简单的读取和赋值，你可以在Runtime用他修改你的getter和setter，让你的getter和setter每次调用时都可以触发某些代码。借助这项神奇的技术，一定可以实现更强大的动态加载，加壳也就有了可能。让我有些不爽的是，javassist又是小日本创建的项目。
<http://www.open-open.com/open54954.htm> Javassist是一个开源的分析、编辑和创建Java字节码的类库。是由东京技术学院的数学和计算机科学系的 Shigeru Chiba 所创建的。它已加入了开放源代码JBoss 应用服务器项目,通过使用Javassist对字节码操作作为JBoss实现动态AOP框架。 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