1. 加密目标

隐藏原始 dex 数据,保护程序。

- 2. 实现步骤
  - 1) 反编译原始 apk 文件(不反编译 dex 文件) ShakaApktool d –s xx.apk –o xx
  - 2) 加密原始 dex 文件 提取反编译后的原始 dex 文件,进行加密,保存到 assests 目录中。
  - 3) 添加 ProxyApplication 添加壳的入口点
  - 4) 添加 so 文件(可选) 添加解壳的 so 文件
  - 5) AndroidManifest.xml 修改

在 application 标签中添加 ProxyApplication 的入口点,如果程序本身就重载了 Application,则需要把原始的 Application 保存起来。

- 6) 回编译 apk 文件 ShakaApktool b xx -o xx.apk
- 3. 相关知识点
  - 1) 类装载器

public DexClassLoader (String dexPath, String optimizedDirectory, String libraryPath, ClassLoader parent)

可以用来动态加载 dex 文件 dexPath: dex 文件的存储路径

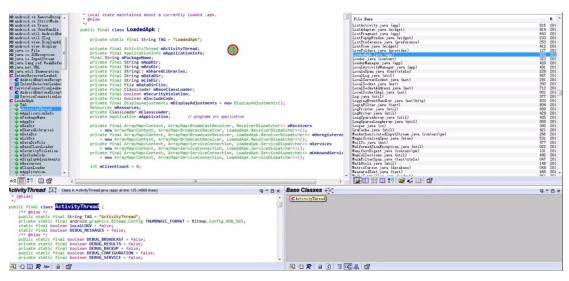
optimizedDirectory: 优化后的 odex 存储目录

libraryPath: 依赖的 so 文件目录(如需要加载的 so 目录)

Parent: 父 ClassLoader

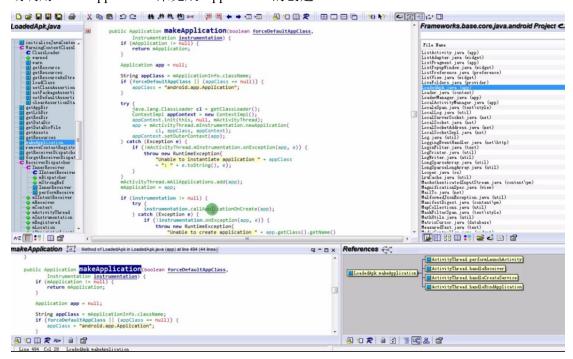
2) Android apk 加载与运行

frameworks/base/core/java/android/app/LoadedApk.java frameworks/base/core/java/android/app/Activity.java

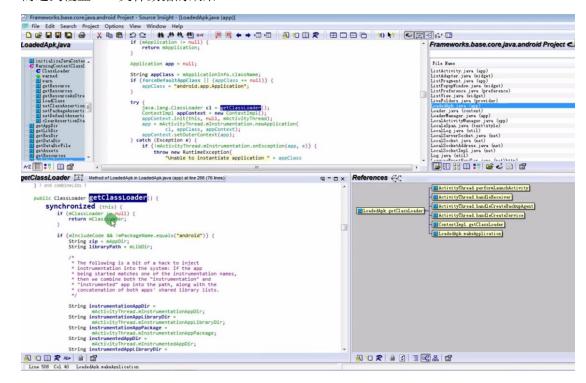


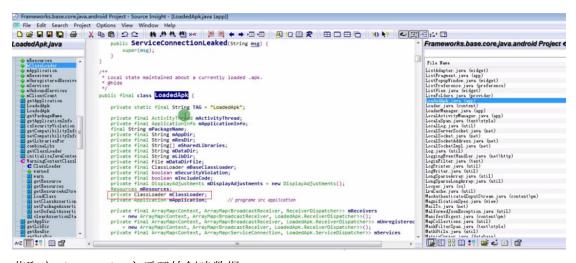
makeApplication: 创建 Application,对 Application 进行重新加载如果一个程序本来有 Application 数据,然而壳的代码把数据覆盖掉了,那么需要手

动调用 makeApplication 来完成 Application 的创建



getClassLoader: 对 classLoader 进行加载,当本身的 classLoader 调用该方法创建完之后,需要把创建后的 classLoader 值通过反射的方法把 mClassLoader 域覆盖掉,从而达到覆盖 dex 文件数据的效果。





获取完 classLoader 之后开始创建数据

加壳时需要对程序本身的 Application 方法通过反射方法来覆盖掉。

