反编译：

apktool d xxx.apk

重新打包

apktool b 代码目录

打包签名

jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore 签名文件名 -storepass 签名密码 待签名的APK文件名 签名的别名

Sign Your App Manually

You do not need Android Studio to sign your app. You can sign your app from the command line using standard tools from the Android SDK and the JDK. To sign an app in release mode from the command line:

1. Generate a private key using [keytool](http://docs.oracle.com/javase/6/docs/technotes/tools/solaris/keytool.html). For example:
2. $ keytool -genkey -v -keystore my-release-key.keystore
3. -alias alias\_name -keyalg RSA -keysize 2048 -validity 10000

This example prompts you for passwords for the keystore and key, and to provide the Distinguished Name fields for your key. It then generates the keystore as a file called my-release-key.keystore. The keystore contains a single key, valid for 10000 days. The alias is a name that you will use later when signing your app.

1. Compile your app in release mode to obtain an unsigned APK.
2. Sign your app with your private key using [jarsigner](http://docs.oracle.com/javase/6/docs/technotes/tools/windows/jarsigner.html):
3. $ jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1
4. -keystore my-release-key.keystore my\_application.apk alias\_name

This example prompts you for passwords for the keystore and key. It then modifies the APK in-place to sign it. Note that you can sign an APK multiple times with different keys.

1. Verify that your APK is signed. For example:
2. $ jarsigner -verify -verbose -certs my\_application.apk
3. Align the final APK package using [zipalign](https://developer.android.com/tools/help/zipalign.html).
4. $ zipalign -v 4 your\_project\_name-unaligned.apk your\_project\_name.apk

zipalign ensures that all uncompressed data starts with a particular byte alignment relative to the start of the file, which reduces the amount of RAM consumed by an app.

Apktool|ShakaApktool 简体中文汉化版|APK反编译工具  
ShakaApktool源码：<https://github.com/rover12421/ShakaApktool> 作者：rover12421  
apktool源码：<https://github.com/iBotPeaches/Apktool>  作者：iBotPeaches  
smali/baksmali源码：<https://github.com/JesusFreke/smali> 作者：JesusFreke  
  
简体中文汉化：越狱（pwelyn）欢迎关注我的微博：<http://weibo.com/206021119>  
**本人水平比较小白，如有不对的地方请指出。强烈推荐使用ShakaApktool**  
  
ShakaApktool完整更新日志：<https://github.com/rover12421/ShakaApktool/commits/master>  
apktool完整更新日志：<http://ibotpeaches.github.io/Apktool/changes/>  
smali/baksmali更新日志：<https://github.com/JesusFreke/smali/wiki>

ShakaApktool功能：  
-ShakaApktool前身算的上是RsApktool  
-支持简体中文，繁体中文，英文 三种语言  
-支持非Android标准目录打包  
-支持png,9.png异常图片回编译  
-支持smali/baksmali功能独立运行  
-增加res资源没有被arsc引用,没有生成id,就会丢失问题  
-增加String Style 中连续`;`引发解析错误修正  
-添加选项[df|default-framework]参数，使用默认的框架资源文件  
-添加选项 [mc|more-recognizable-characters]显示更多的可识别字符.比如中文,不在以\uxxxx编码显示,而是直接显示中文.让smali更容易读  
-添加选项[fui|fuck\_unkown\_id],对未知ID,强制处理  
-添加选项[ir|ignore\_res\_decode\_error],忽略资源decode异常  
-添加选项[n9|no-9png] 参数，不解析.9格式的资源  
-添加选项[fnd|fuck-not-defined-res] 参数  
支持标准资源名的`Public symbol drawable/? declared here is not defined.`异常打包  
不支持带点资源名的打包  
-添加选项[xn|xml\_attribute\_name\_correct] 参数  
已经测试mobileqq,qq浏览器可以正常使用  
xml 属性名实际是通过id来查找的,但是baxml中保留了一份显示的属性名,QQ浏览器这里实际是错误属性字段.  
-资源名，XML属性名无字母数字的限制，可以是大写字母，可以是中文等等  
-腾讯加固Xml修复  
xml修复正常,但是直接安装会失败[INSTALL\_PARSE\_FAILED\_MANIFEST\_MALFORMED]  
[INSTALL\_PARSE\_FAILED\_MANIFEST\_MALFORMED]原因是<meta-data android:name="@anim/push\_top\_out2" android:value="meta-data"/> 这段中的android:name有问题.删除这行即可安装.  
-更多请查看RsApktool说明：<http://bbs.chinaunix.net/thread-4096302-1-1.html>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*apktool & ShakaApktool安装说明\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
1.Windows:  
下载Windows文件夹apktool.bat  
下载apktool\_xxxxx.jar或ShakaApktool\_xxx.jar重命名为apktool.jar  
复制(apktool.jar & apktool.bat)到Windows目录（通常是C:\Windows\System32）  
也可以把这这几个文件放到任意一个文件夹，然后添加这个文件夹路径到系统的环境变量  
然后通过cmd命令窗口运行apktool  
  
2.Linux:  
下载Linux文件夹apktool  
下载apktool\_xxxxx.jar或ShakaApktool\_xxx.jar重命名为apktool.jar  
复制(apktool.jar & apktool.bat)到/usr/local/bin（需要root权限，可以使用在终端使用命令：sudo cp apktool /usr/local/bin）  
也可以把这这几个文件放到任意一个文件夹，然后添加这个文件夹路径到系统的环境变量  
然后给文件添加可执行权限（chmod +x）  
然后通过终端命令窗口运行apktool  
  
3.MAC OS X:  
下载OS X文件夹apktool  
下载apktool\_xxxxx.jar或ShakaApktool\_xxx.jar重命名为apktool.jar  
复制(apktool.jar & apktool.bat)到/usr/local/bin（需要root权限，可以使用在终端使用命令：sudo cp apktool /usr/local/bin）  
也可以把这这几个文件放到任意一个文件夹，然后添加这个文件夹路径到系统的环境变量  
然后给文件添加可执行权限（chmod +x）  
然后通过终端命令窗口运行apktool  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*apktool & ShakaApktool 使用方法\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
-advance,--advanced        查看更多信息.  
-lng,--language <Locale>   显示语言, e.g. zh-CN, zh-TW  
//ShakaApktool特有功能，如果需要英文输入直接修改apktool.bat 内 -Duser.language=en 中文即zh  
-version,--version         查看版本信息  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*安装框架指令\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
if|install-framework  
//框架文件一般在system/framework/\*.apk 每个系统不一样框架也不一样，有的是1个有的2个有的或更多  
$ apktool if framework-res.apk  
I: 框架安装到: $HOME/apktool/framework/1.apk  
$ apktool if com.htc.resources.apk  
I: 框架安装到: $HOME/apktool/framework/2.apk  
-p,--frame-path <dir>   保存框架文件到指定目录  
$ apktool if framework-res.apk -p foo/bar  
I: 框架安装到: foo/bar/1.apk  
$ apktool if framework-res.apk -t baz -p foo/bar  
I: 框架安装到: foo/bar/1-baz.apk  
//foo/bra 为命令所在的目录并非$HOME/apktool/  
-t,--tag <tag>          保存框架文件为指定名称  
$ apktool if com.htc.resources.apk -t htc  
I: 框架安装到: $HOME/apktool/framework/2-htc.apk  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*反编译指令\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
d[ecode] [options] <file\_apk>  
//d 反编译参数，一般命令主要是apktool d file.apk 以下参数为d 后面可选参数  
   --api <API>                       将按照API级别生成文件信息, 例如.14是ICS.  
-b,--no-debug-info                   不输出debug信息 (.local, .param, .line, etc.)  
-d,--debug                           反编译调试模式. 查看更多信息.  
    --debug-line-prefix <prefix>      反编译调试模式下, 给Smali添加行前缀 默认是 "a=0;// ".  
-k,--keep-broken-res                 当出现错误或者一些resources被放弃时使用，例如."Invalid config flags detected. Dropping resources", 即使在有错误的情况下，你还是想要强行反编译. 你之后必须手动修复相关错误才能进行编译.  
-m,--match-original                  保持尽可能的接近原始文件.防止重新生成.  
-df,--default-framework              使用默然的框架资源文件  
//ShakaApktool独有功能，主要作用删除$HOME/apktool/framework/1.apk然后释放新的1.apk(主要解决apktool内android-framework.jar升级后本地1.apk版本太低导致反编译失败问题)  
-f,--force                           强制删除目标文件夹  
//这个应该用的比较多，在使用apktool d file.apk时候提示你目标文件夹已存在，加-f就会直接删除目标文件夹  
-fui,--fuck\_unkown\_id                反编译遇到未知资源id继续执行  
//ShakaApktool独有功能，主要是反编译遇到一些未知的id直接跳过  
-ir,--ignore\_res\_decode\_error        忽略资源反编译的错误  
//ShakaApktool独有功能，主要是忽略资源反编译的错误  
-mc,--more-recognizable-characters   显示更多的可识别字符  
//ShakaApktool独有功能，显示更多的可识别字符，比如中文不在以\uxxxx编码显示，而是直接显示中文，让smali更容易读  
-n9,--no-9png                        不解析.9格式的资源  
//ShakaApktool独有功能，在编译时直接不解析.9格式的文件  
-o,--output <dir>                    输出文件夹名字. 默认是 apk.out  
//输出文件夹名，例如：apktool d -f file.apk -o out  
-p,--frame-path <dir>                使用指定目录下的框架文件  
//使用指定目录下的框架文件，例如：apktool d -f -p foo/bar file.apk -o out  
//这里的-p foo/bar按照安装框架那一步来执行  
-r,--no-res                          不反编译resources.arsc  
//不反编译resources.arsc，只反编译classes.dex  
-s,--no-src                          不反编译classes.dex  
//不反编译classes.dex，只反编译resources.arsc  
-t,--frame-tag <tag>                 使用指定名称的框架文件  
//使用指定名称的框架文件，例如：apktool d -f -t htc file.apk -o out  
//这里的-t htc按照安装框架那一步来执行  
-xn,--xml\_attribute\_name\_correct     xml attribute name correct. May be has problem, not recommended.  
//ShakaApktool独有功能，xml 属性名实际是通过id来查找的,但是baxml中保留了一份显示的属性名。xml 属性名实际是通过id来查找的,但是baxml中保留了一份显示的属性名，已经测试mobileqq,qq浏览器可以正常使用  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*回编译指令\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
b[uild] [options] <app\_path>  
-a,--aapt <loc>         从指定路径加载aapt  
//从指定路径加载aapt，例如：apktool b -a $HOME/sdk/build-tools/23.0.1/aapt out -o new.apk  
-c,--copy-original      复制原始AndroidManifest.xml和META-INF文件.可以查看项目更多信息  
-d,--debug              调试模式编译. 检查项目的更多信息  
-f,--force-all          跳过已编译检查,强制编译所有文件  
//覆盖已经存在的文件，强制编译resources.arsc 和 classes.dex  
-o,--output <dir>       输出apk路径. 默认是 dist/name.apk  
//输出apk路径，默认在dist/xxx.apk 例如：apktool b out -o new.apk(new.apk路径在命令执行的目录)  
-p,--frame-path <dir>   使用指定目录下的框架文件  
//使用指定目录下的框架文件，例如：apktool b -p foo/bar out  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*其他说明\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
-q 和 -v 命令  
//-q 编译时不输出任何信息，直接静默模式编译，例如：apktool -q d或apktool -q b  
//-v 编译时输出更多详细信息，例如：apktool -v d或apktool -v b  
  
也可以直接不借助apktool[apktool.bat] 直接使用  
java -jar apktool.jar[ShakaApktool.jar]  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*smali/baksmali\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
ShakaApktool.jar目前版本已经完全支持smali/baksmali功能  
  
ShakaApktool s[mali] [options] [--] [<smali-file>|folder]  
ShakaApktool bs|baksmali [options] <dex-file>