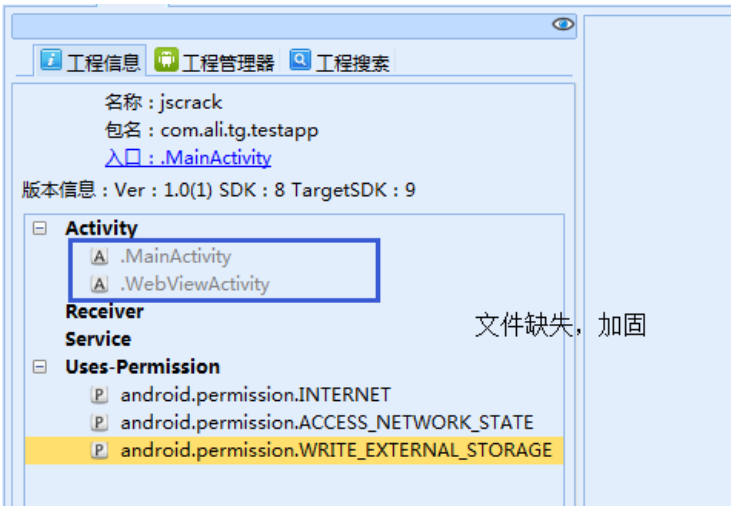
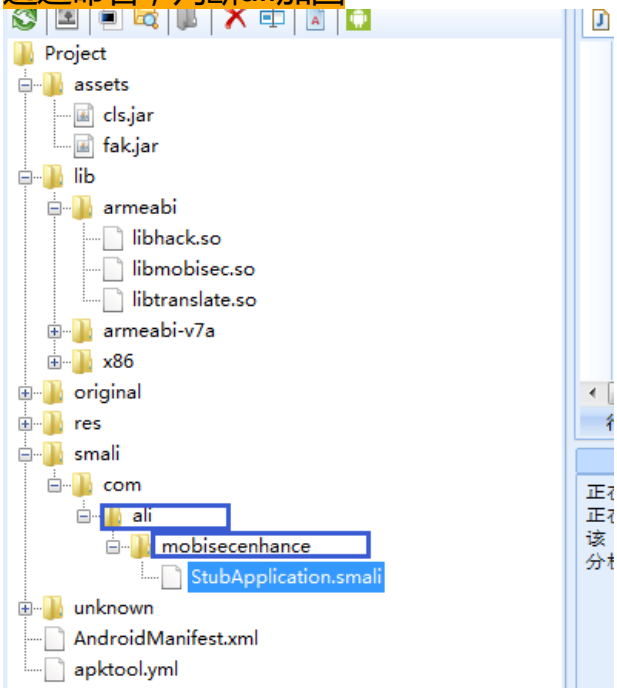
简单dex壳实例

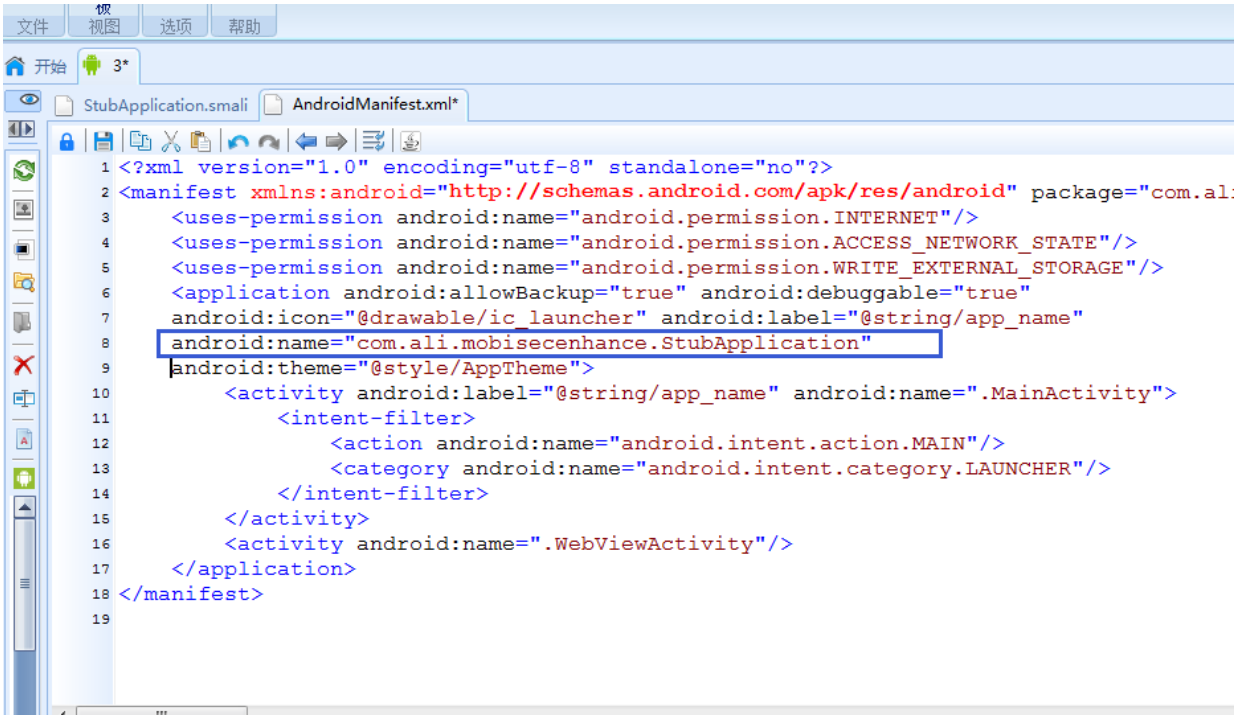
3.apk



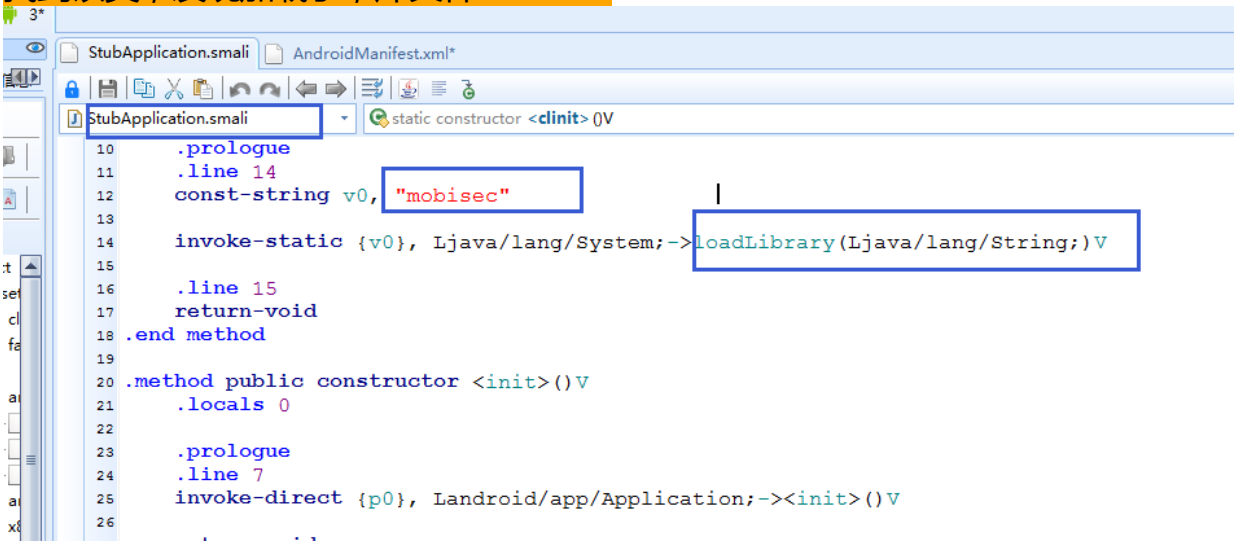
通过命名，判断ali加固



先于activity启动的类



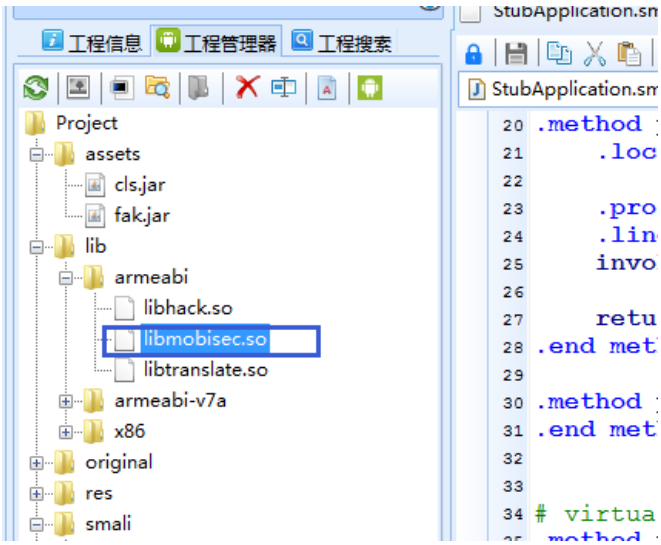
找到该类，发现加载了，库文件mobisec



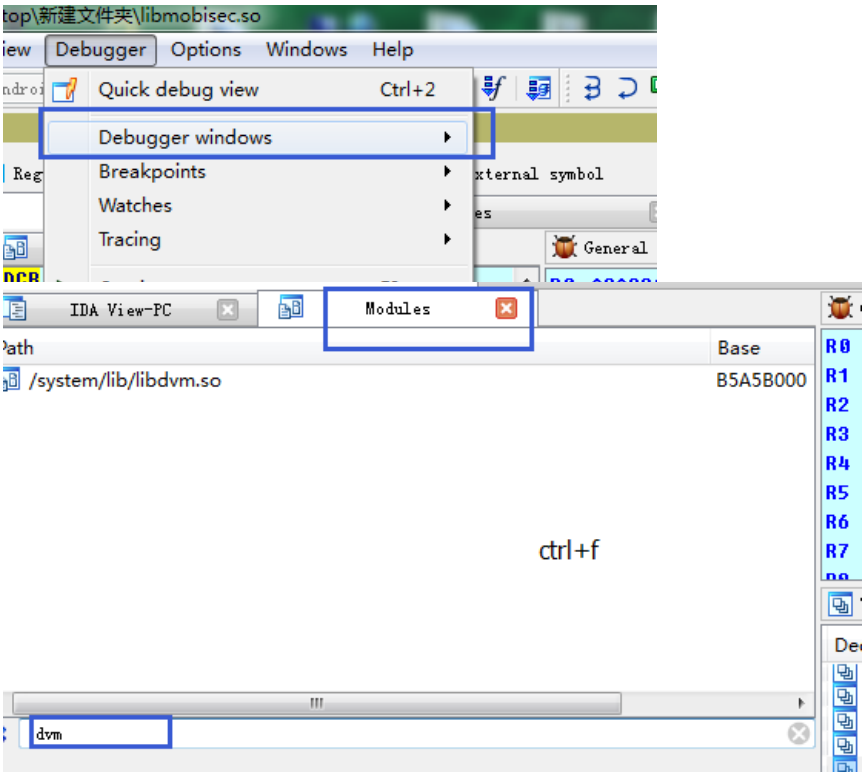
两个函数，一般用于加固

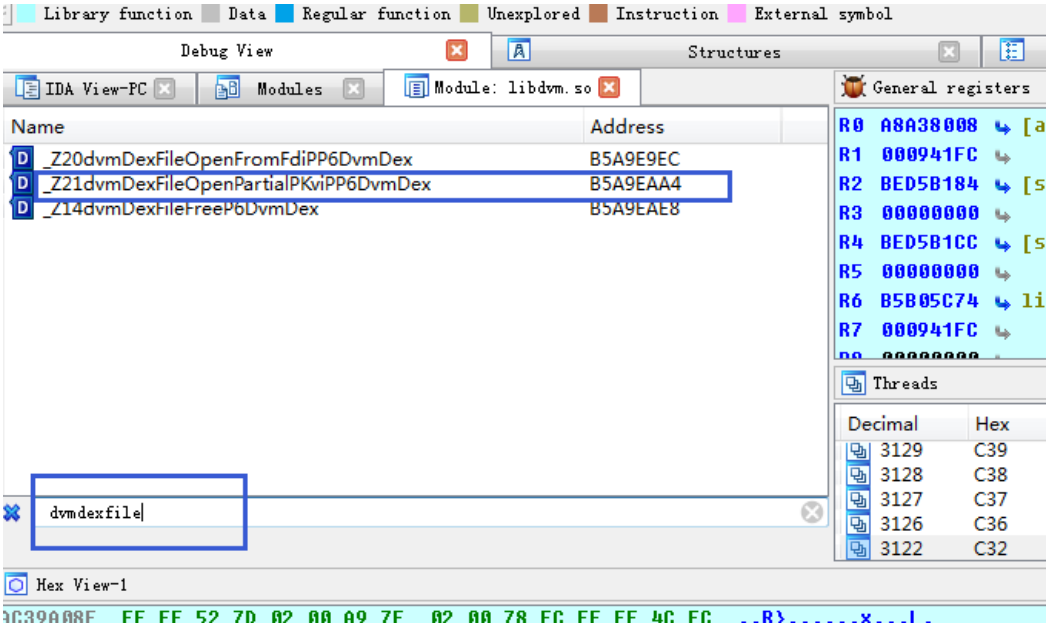
```
# virtual methods
.method protected native attachBaseContext(Landroid/content/Context;)V
.end method

.method public native onCreate()V
.end method
```

找到该so文件 ，IDA动态调试



打开模块窗口，Ctrl+F查找要下断的函数



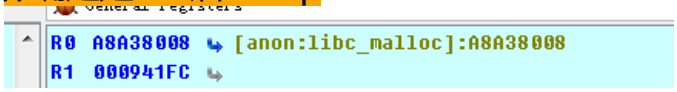


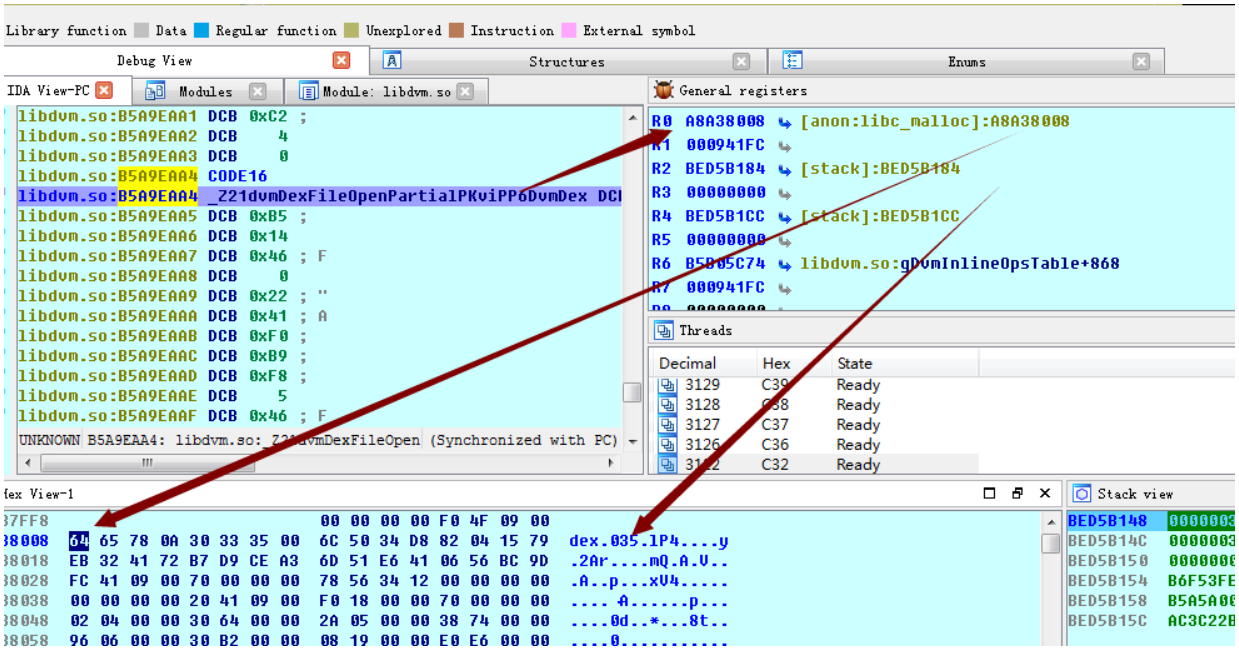
//安装要脱壳的软件
//命令行调试启动

```
D:\>adb shell
root@generic:/ # cd ./data/local/tmp
root@generic:/data/local/tmp # ./android_server
IDA Android 32-bit remote debug server(ST) v1.19. Hex-Rays (c) 2004-2015
Listening on port #23946...
=============================================================
[1] Accepting connection from 127.0.0.1...
```

```
D:\>adb forward tcp:23946 tcp:23946

D:\>adb shell am start -D -n com.ali.tg.testapp/.MainActivity
Starting: Intent { cmp=com.ali.tg.testapp/.MainActivity }

D:\>jdb -connect com.sun.jdi.SocketAttach:hostname=localhost,port=8700
设置未捕获的java.lang.Throwable
设置延迟的未捕获的java.lang.Throwable
正在初始化jdb...
>
```

//断下来后查看下寄存器值，r0,r1,在内存中
//确定是dex后dump





static main(void)

{

auto fp, begin, end, dexbyte;

fp = fopen("d:\\dump1.dex", "wb");

begin = r0;

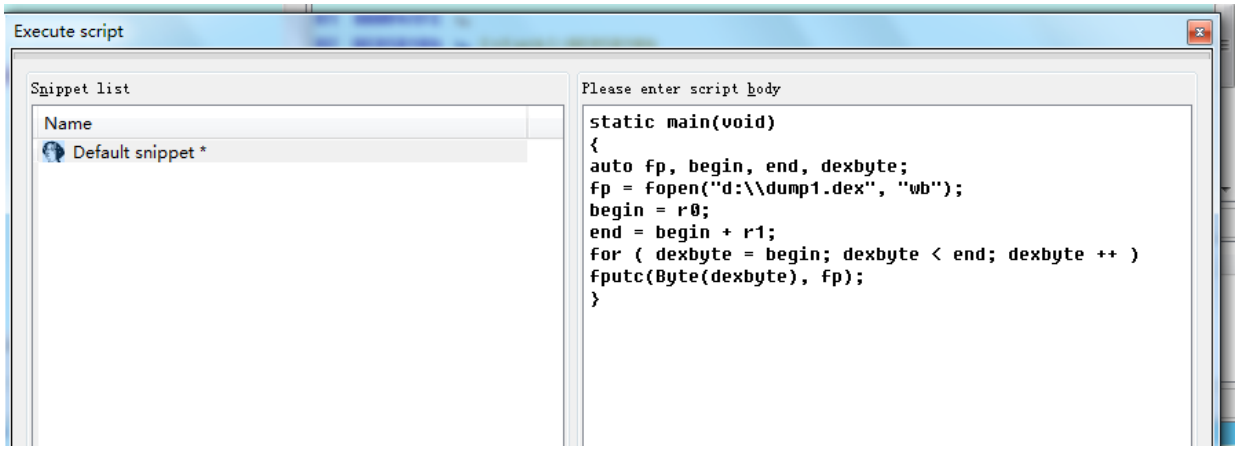
end = begin + r1;

for (dexbyte = begin; dexbyte < end; dexbyte ++)

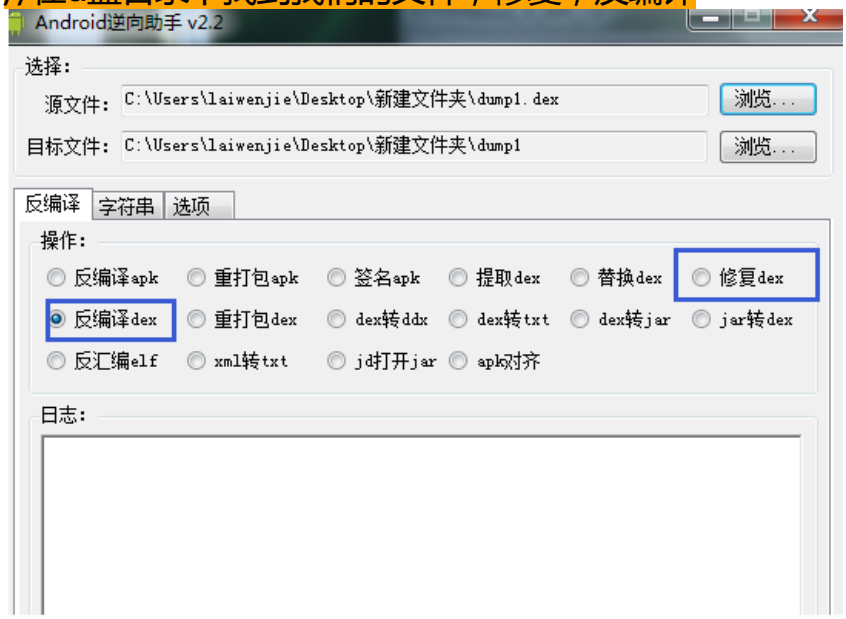
fputc(Byte(dexbyte), fp);

}

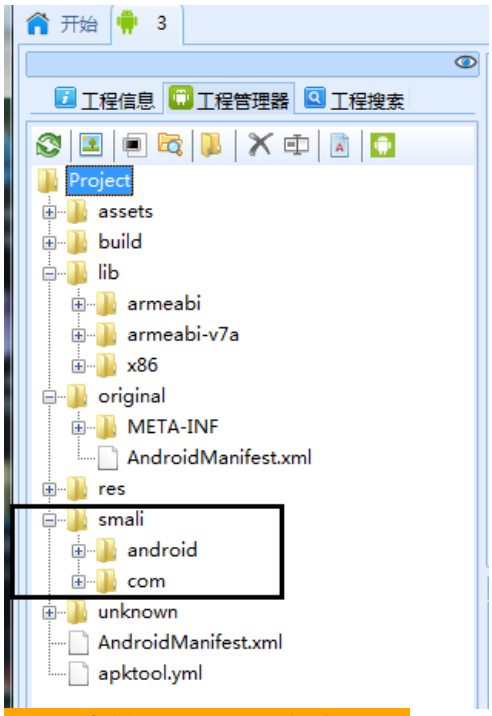
//运行脚本：菜单file->SCripcomman



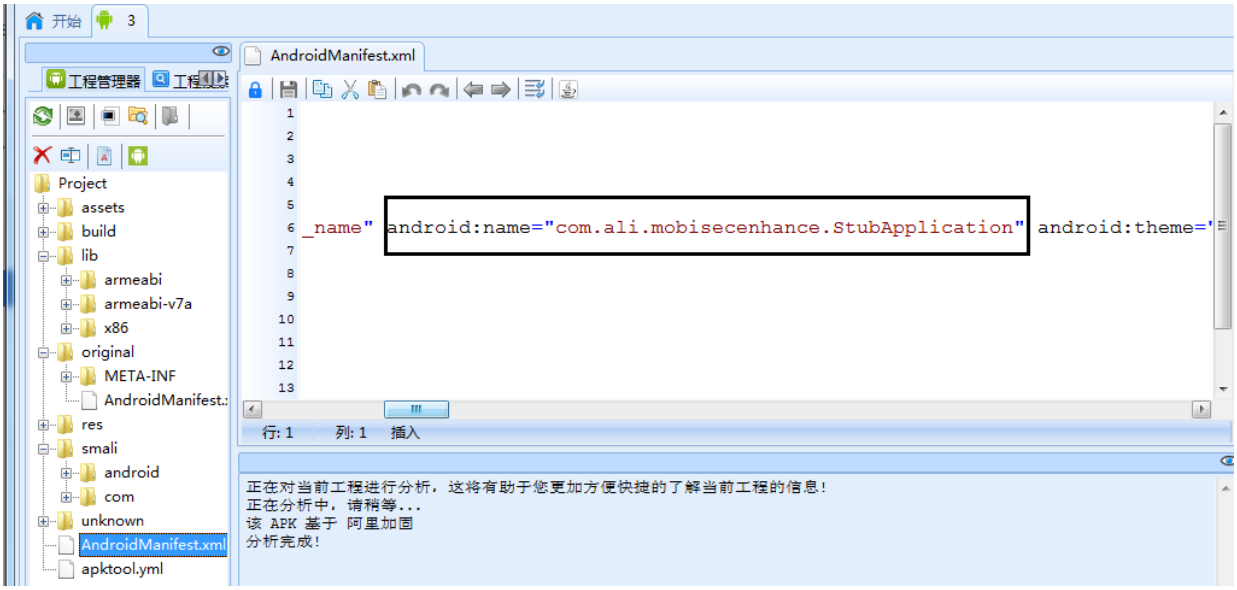
//在d盘目录下找到我们的文件，修复，反编译



在原有的smali目录下，右键，路径。将修复反编译的dex文件拷贝在打开的路径下



删除清单文件的加密启动类


开始
3
AndroidManifest.xml
工程管理器
Project
assets
build
lib
armeabi
armeabi-v7a
x86
original
META-INF
AndroidManifest.
res
smali
android
com
unknown
AndroidManifest.xml
apktool.yml
_name" android:name="com.ali.mobisecenhance.StubApplication" android:theme='
行: 1
列: 1
插入
正在对当前工程进行分析，这将有助于您更加方便快捷的了解当前工程的信息!
正在分析中，请稍等...
该 APK 基于 阿里加固
分析完成!

最后保存编译就可以了