# 签名绕过

发现这个签名绕过，是在kingroot原版root后卸载可用，而重启后不可用。

## java绕过

搜索Landroid/content/pm/PackageInfo;->signatures:[Landroid/content/pm/Signature;定位。

定位到的方法为：

Lcom/kingroot/kinguser/ace;->dx(Ljava/lang/String;)Landroid/content/pm/Signature; = is call

Lcom/kingroot/kinguser/ace;->dv(Ljava/lang/String;)Landroid/content/pm/Signature; = is called

进一步查找：

Lcom/kingroot/kinguser/ace;->dx(Ljava/lang/String;)Landroid/content/pm/Signature; = is call----------------------------为获取手机已安装程序的

Lcom/kingroot/kinguser/ace;->du(Ljava/lang/String;)Ljava/lang/String; = is called

param = /data/app/com.kingroot.kinguser-1/base.apk-----------为本apk的签名信息获取

Lcom/kingroot/kinguser/ace;->dv(Ljava/lang/String;)Landroid/content/pm/Signature; = is called

Lcom/kingroot/kinguser/ace;->du(Ljava/lang/String;)Ljava/lang/String; : res = E89B158E4BCF988EBD09EB83F5378E87

因此，只需要将程序的返回值永远是真是的apk签名“191240FCB048127DB9110D1B30537FDE”信息就好了。

### other

在上面两个都搞定了以后，发现在kingroot(原始)root完毕，再安装我的包，还是出现下面的：



界面为：dialog\_repackage\_warning.xml

activity为：Lcom/kingroot/kinguser/activitys/RePackageWarningActivity;

Lcom/kingroot/kinguser/oq;->a(Lcom/kingroot/kinguser/xp;)V

->Lcom/kingroot/kinguser/atg;->w(Z)V======== Lcom/kingroot/kinguser/or;->w(Z)V

->Lcom/kingroot/kinguser/activitys/RePackageWarningActivity;

Lcom/kingroot/kinguser/oq;->a(Lcom/kingroot/kinguser/xp;)V

->Lcom/kingroot/kinguser/op;->c(Lcom/kingroot/kinguser/or;)Z

->Lcom/kingroot/kinguser/op;->b(Lcom/kingroot/kinguser/or;)Z

-> Lcom/kingroot/kinguser/op;->fX()Ljava/lang/String;

-> Lcom/kingroot/kinguser/ace;--------3种方式验证签名

其中一种能够方式ace.bk

Lcom/kingroot/kinguser/acu;->a(Lcom/kingroot/kinguser/acx;J)Lcom/kingroot/kinguser/acw;

-->Lcom/kingroot/kinguser/acw;-><init>(Ljava/lang/String;Ljava/lang/Integer;Ljava/lang/String;Ljava/lang/String;)V

-->iput-object p3, p0, Lcom/kingroot/kinguser/acw;->Hi:Ljava/lang/String;

在acu.a中

调用[Hh:export CLASSPATH=/data/data/com.kingroot.kinguser/applib/tm.dex && /system/bin/app\_process /system/bin com.kingroot.kinguser.TellMe 10124,Hl:export CLASSPATH=/data/data/com.kingroot.kinguser/applib/tm.dex && /system/bin/app\_process /system/bin com.kingroot.kinguser.TellMe 10124,mTimeout:120000,]

获取程序的签名md5

将命令的执行结果替换掉签名“191240FCB048127DB9110D1B30537FDE”就可以了。

## so绕过

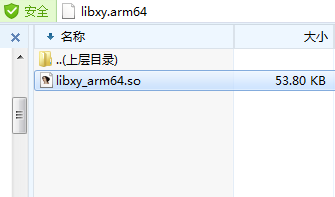
native 方法

load/loadLibrary

Lcom/kingroot/kinguser/zu;->F(Ljava/lang/String;Ljava/lang/String;)Z:load = /data/data/com.kingroot.kinguser/applib/libpu.so------------------------------------初步确定为反调试的so

Lcom/kingroot/kinguser/zu;->F(Ljava/lang/String;Ljava/lang/String;)Z:load = /data/data/com.kingroot.kinguser/applib/libhxy64.so -------------------------------初步确定为签名验证的so

进一步分析上面libhxy64.so,发现是libxy.arm64解压(为gzip格式)后得到的。



通过ida逆向，



因此将so修改后，重新打包为gz格式的压缩包，放到assets目录下。

## plugin java检查签名绕过

在Lcom/kingroot/kinguser/bqs;->ff(I)I中，加载dex文件：

DexClassLoaderHelper = init------------start--------

dexPath = /data/data/com.kingroot.kinguser/app\_workspace/app/com.kingroot.RushRoot-6122de20a94ae6d97defb274e7cccc84.apk

optimizedDirectory = /data/data/com.kingroot.kinguser/app\_workspace/dalvik-cache

libraryPath = /data/data/com.kingroot.kinguser/app\_workspace/data/com.kingroot.RushRoot/lib/armeabi

DexClassLoaderHelper = init------------end--------

因此，替换掉此Dex文件，并加入日志，发现在：Lcom/kingroot/RushRoot/ew;->b中，进行签名校验：

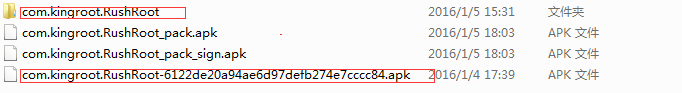
Lcom/kingroot/RushRoot/ew;->b : getPackageName = com.kingroot.kinguser

Lcom/kingroot/RushRoot/ew;->b : signature = E89B158E4BCF988EBD09EB83F5378E87

让其返回常量签名“191240FCB048127DB9110D1B30537FDE”

## plugin so检查签名绕过

com.kingroot.kinguser\app\_workspace\app，下有apk文件，反汇编为com.kingroot.RushRoot



在asserts目录下：

