# 四叶草移动网络安全学习日志

原创

WayneYHN 于 2020-07-25 11:11:31 发布 286 收藏

文章标签： android java



本文链接：https://blog.csdn.net/qq_44365878/article/details/107575821



## 四叶草学习日志

## DAY_1

**Android Studio之前已经安装**

**adb使用时出现**

```
connect error for write: no devices/emulators founds
```

解决：手机打开usb调试

**CrackeMe_clover01解题思路：**

再进行反编译后发现他会在最开始进行一个用户字段的判断，如下

```
{
  if (paramString1 != null) {
    try
    {
      if (paramString1.length() == 0) {
        return false;
      }
      if ((paramString2 != null) && (paramString2.length() == 16))
      {
        Object localObject = MessageDigest.getInstance("MD5");
```

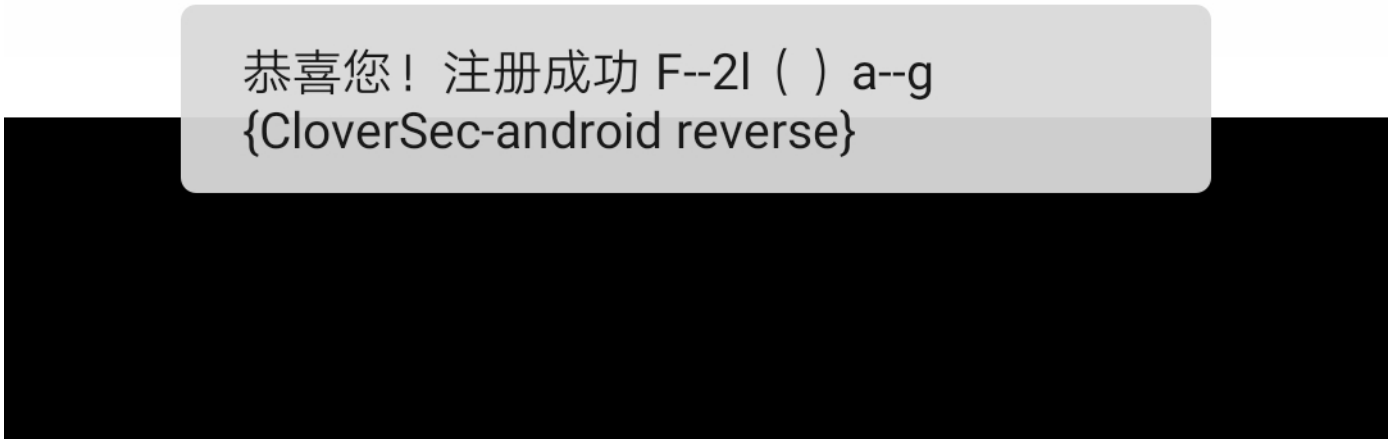
如果为空返回false，我让他在这直接返回true

根据这个思路，我找到了下面的返回值在源码中的表示

```
    .prologue
    const/4 v7, 0x0

    .line 54
    if-eqz p1, :cond_0

    :try_start_0
    invoke-virtual {p1}, Ljava/lang/String;->length()I

    move-result v8

    if-nez v8, :cond_1
```

```
.line 76
:cond_0
:goto_0
return v7
```

将v7的值由0x0改为0x1然后编译安装后，得到flag



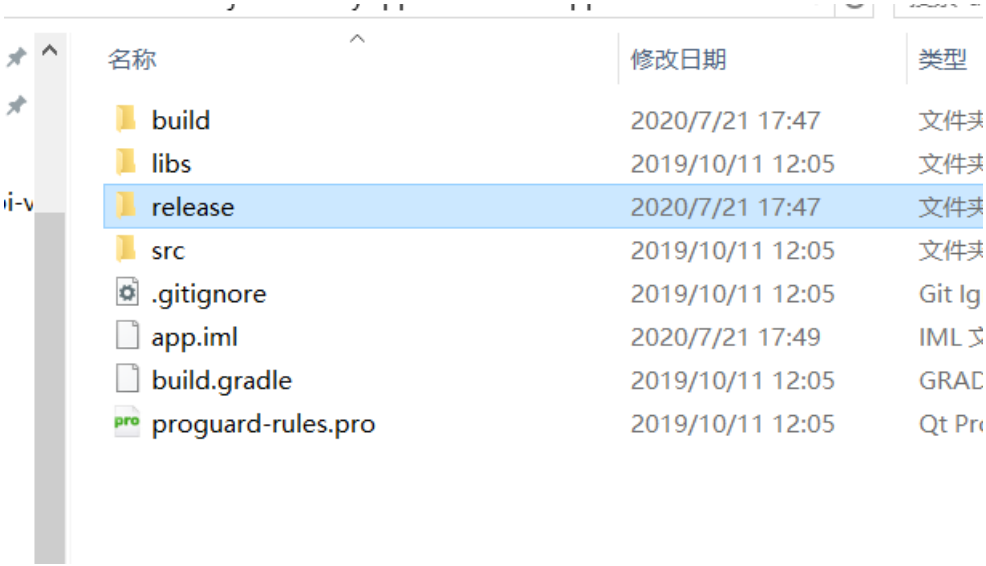

## DAY_2

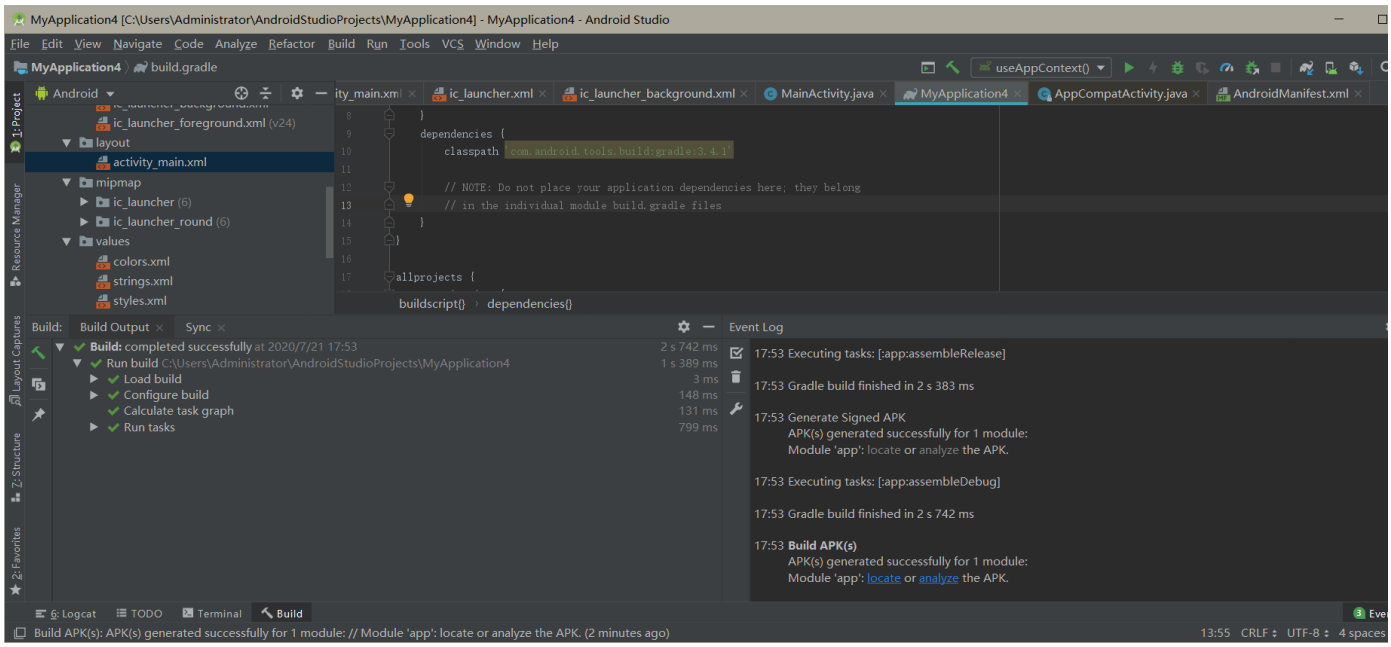
自己创建的app

BUTTON

杨昊南

并根据网上教程生成release版本https://www.jianshu.com/p/ba385af853c4并且安装成功




CTF100解题

```
177     .locals 6
178     .param p1, "savedInstanceState"    # Landroid/os/Bundle;
179
180     .prologue
181     const/4 v5, 0x1
182
183     .line 22
184     invoke-super {p0, p1}, Landroid/support/v7/app/AppCompatActivity;->onCreate(Landroid/os/Bundle;)V
185
```

将0x0改为0x1

```
super.onCreate(paramBundle);
setContentView(2130968601);
((Button)findViewById(2131492950)).setClickable(false);
this.has_gone_int = 0;
```

在编译时遇到如下问题

```
>W: fakeLogOpen(/dev/log_stats) failed
>W: libpng error: Not a PNG file
>W: ERROR: Failure processing PNG image G:\吾爱破解工具包2.0,解压密码 www.52pojie.cn\吾爱破解工具包\Tools\AndroidTools\AndroidKiller_v1.3.1\projects\CFF_100\Project\res
\mipmap-xxhdpi-v4\ic_launcher.png
```

直接删除图片后成功编译（也可以用010code查看前面几位16进制数来判断实际文件类型然后进行修改）

“爬到了看FLAG”可以点了，但是一点会闪退，将安装包发给同学运行发现可以，现在不知道是啥问题

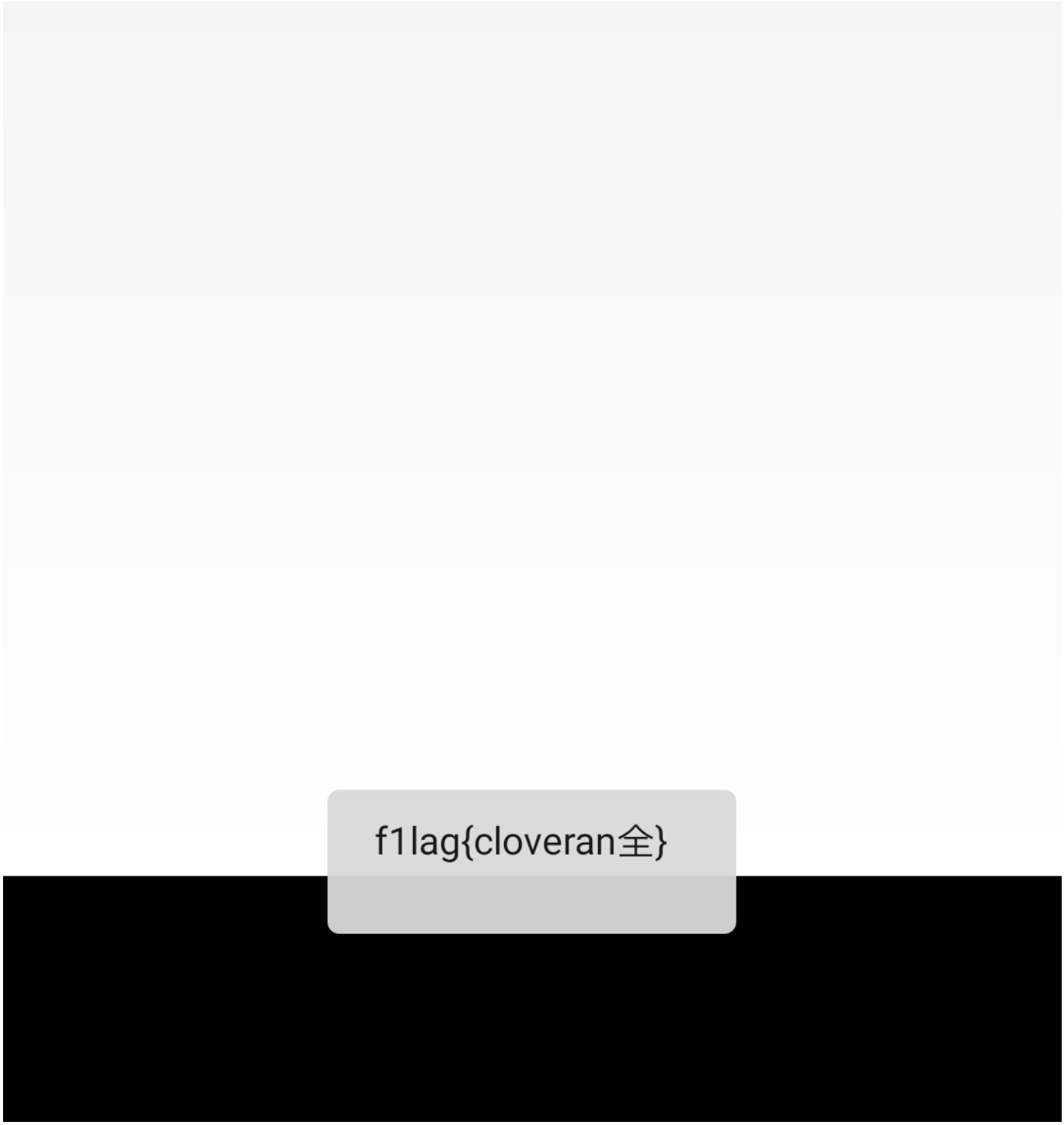

CrakeMe_clover03:
通过反编译查看java代码找到注册码验证得函数

```
public boolean isRegistered()
{
  boolean booll = false;
  int j = 0;
  if ((this.sn == null) || (this.sn.length() < 8)) {
    return false;
  }
  int k = this.sn.length();
  if (k == 8) {
```

在smali源码中发现调用这个函数后，如果他的返回值为零的话就不会输出下面的flag，

```
    invoke-direct {v0, v2, v3}, Lcom/droider/crackme0502/MainActivity$SNChecker;-><init>(Lcom/droider/crackm

    .line 45
    .local v0, "checker":Lcom/droider/crackme0502/MainActivity$SNChecker;
    invoke-virtual {v0}, Lcom/droider/crackme0502/MainActivity$SNChecker;->isRegistered()Z

    move-result v2

    if-eqz v2, :cond_0

    const-string v1, "fllag{cloveran\u5168}\r\n"

    .line 46
```

于是修改if-eqz v2 为 if-nez v2，编译运行后成功




## DAY_3

早上我们通过学习了解了Android逆向工程的流程，学习了如何利用apktool工具进行该过程

首先我们对apk文件进行反编译

```
java -jar .\apktool_2.4.0.jar d .\crackme02.apk
```

```
PS G:\NewJourney\pratice\day_3test> java -jar .\apktool_2.4.0.jar d .\crackme02.apk
I: Using Apktool 2.4.0 on crackme02.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: C:\Users\Administrator\AppData\Local\apktool\framework\1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
```

```
I: Copying unknown files...
I: Copying original files...
```

然后又练习了回编译的

```
java -jar .\apktool_2.4.0.jar b .\crackme02\
```

```
PS G:\NewJourney\pratice\day_3test> java -jar .\apktool_2.4.0.jar b .\crackme02\
I: Using Apktool 2.4.0
I: Checking whether sources has changed...
I: Checking whether resources has changed...
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk...
```

回编译生成的apk在\crakme02\dis\目录下，但在通过adb进行安装的时候出现如下错误，没有签名

```
* daemon not running; starting now at tcp:5037
* daemon started successfully
Performing Streamed Install
adb: failed to install .\crackme02\dist\crackme02.apk: Failure [INSTALL_PARSE_FAILED_NO_CERTIFICATE
S: Failed to collect certificates from /data/app/vmdl209832183.tmp/base.apk: Attempt to get length
of null array]
```

使用signapk对其进行一个签名之后成功安装

```
PS G:\NewJourney\pratice\day_3test\signapk> ls


    目录: G:\NewJourney\pratice\day_3test\signapk


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         2020/6/11     17:13         155433 1.apk
-a----         2020/6/11     17:20             67 signapk.bat
-a----         2011/5/13     15:34           7369 signapk.jar
-a----         2012/3/18     15:00           1217 testkey.pk8
-a----         2012/3/18     15:00           1675 testkey.x509.pem


PS G:\NewJourney\pratice\day_3test\signapk> java -jar .\signapk.jar .\testkey.x509.pem .\testkey.pk8 ..\crackme02\dist\c
rackme02.apk My_Crakeme02.apk
PS G:\NewJourney\pratice\day_3test\signapk> ls


    目录: G:\NewJourney\pratice\day_3test\signapk


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         2020/6/11     17:13         155433 1.apk
-a----         2020/7/22     14:36         159356 My_Crakeme02.apk
-a----         2020/6/11     17:20             67 signapk.bat
-a----         2011/5/13     15:34           7369 signapk.jar
-a----         2012/3/18     15:00           1217 testkey.pk8
-a----         2012/3/18     15:00           1675 testkey.x509.pem


PS G:\NewJourney\pratice\day_3test\signapk> adb install .\My_Crakeme02.apk
Performing Streamed Install
Success
PS G:\NewJourney\pratice\day_3test\signapk>
```

## CrackMe_clover02解题

使用apktool对其进行反汇编，并通过JEB查看Java代码

```
int v0 = this.a();
AlertDialog$Builder v1 = new AlertDialog$Builder(((Context)this));
v1.setTitle("想要flag吗？充个money吧");
if(v0 < 30) {
    v1.setMessage("现在你的积分值为：" + v0 + "！你的cloversec的积分也太少了吧！努力赚取积分，就能买得起flag了。然
    v1.setNegativeButton("俺去banzhuan赚积分咯！", new a(this));
}
else {
    v1.setMessage("要充钱才能有积分哦。确定充钱？");
    v1.setPositiveButton("充钱", new b(this));
    v1.setNegativeButton("不玩了 exit", new c(this));
}

v1.show();
```

发现onclick函数中出现判断，尝试在smali中进行修改

```
invoke-virtual {v1, v2},
Landroid/app/AlertDialog$Builder;->setTitle(Ljava/lang/CharSequence;)Landroid/app/AlertDialog$Builder
;

const/16 v2, 0x1e

if-ge v0, v2, :cond_0

new-instance v2, Ljava/lang/StringBuilder;

const-string v3, "\u73b0\u5728\u4f60\u7684\u79ef\u5206\u503c\u4e3a\uff1a"
```

if-lt

修改后，通过apktool和signapk工具进行安装后发现成功

# 向至尊vip致敬

至尊VIP

by SecClover

优秀，flag{Y3liZXJzY2hvb2w9c2VjY2xvdmVyPW5ldHdvcms=}！


对flag中的内容进行base64解密：cyberschool=secclover=network

下午我们对smali的语法和进行了学习了解了,同时也练习了baksmali.jar 与 smali.jar的使用

使用baksmali.jar 对CrackMe_clover02中的dex 进行反编译

```
java -jar .\baksmali.jar -o [输出文件夹]  dex文件
```

```
PS G:\NewJourney\pratice\JarPackageTools> java -jar .\baksmali.jar -o MyCrackMe_clover02 .\classes.dex
PS G:\NewJourney\pratice\JarPackageTools> ls


    目录: G:\NewJourney\pratice\JarPackageTools


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2020/7/22     15:50                CrackMe_clover02
d-----        2020/7/22     15:55                MyCrackMe_clover02
-a----        2020/7/21     12:45       16314178 apktool_2.4.0.jar
-a----        2012/9/17     22:12         524051 baksmali.jar
-a----        2020/7/22     10:36         286512 classes-dex2jar.jar
-a----        2020/4/21     13:25         148552 classes.dex
-a----        2020/4/21     13:25         499432 CrackMe_clover02.apk
-a----        2012/6/18     17:06         130691 ddx.jar
-a----        2012/9/17     22:12         693227 smali.jar


PS G:\NewJourney\pratice\JarPackageTools>
```

使用smali.jar回编译 dex

```
java -jar .\smali.jar -o 目标dex文件 [smali文件夹]
```

```
PS G:\NewJourney\pratice\JarPackageTools> java -jar .\smali.jar -o  newClass.dex .\MyCrackMe_clover02\
PS G:\NewJourney\pratice\JarPackageTools> ls


    目录: G:\NewJourney\pratice\JarPackageTools


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2020/7/22     15:50                CrackMe_clover02
d-----        2020/7/22     15:55                MyCrackMe_clover02
-a----        2020/7/21     12:45       16314178 apktool_2.4.0.jar
-a----        2012/9/17     22:12         524051 baksmali.jar
-a----        2020/7/22     10:36         286512 classes-dex2jar.jar
-a----        2020/4/21     13:25         148552 classes.dex
-a----        2020/4/21     13:25         499432 CrackMe_clover02.apk
-a----        2012/6/18     17:06         130691 ddx.jar
-a----        2020/7/22     15:57         148532 newClass.dex
-a----        2012/9/17     22:12         693227 smali.jar


PS G:\NewJourney\pratice\JarPackageTools>
```

选取 CrackMe_clover02中的MainActivity中的smali 选取一个函数 对其samli执行流程进行分析

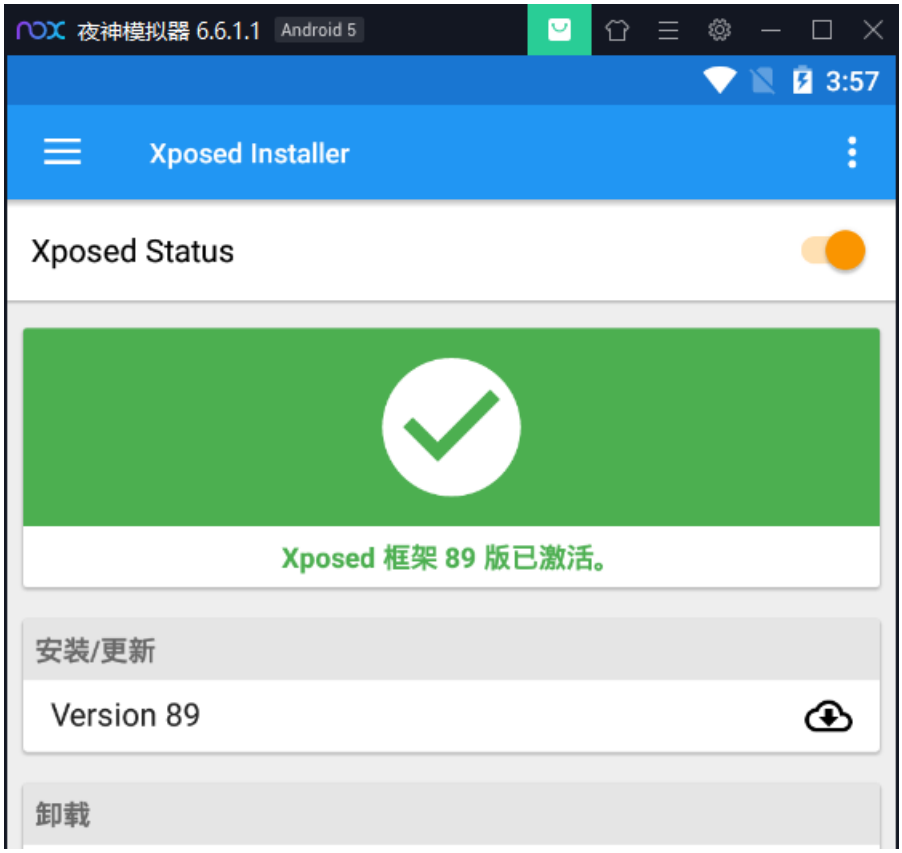
```
.method private a()I
    .locals 3

    const/4 v2, 0x0

    const-string v0, "data"

    invoke-virtual {p0, v0, v2}, Lcom/example/crackme/MainActivity;->getSharedPreferences(Ljava/lang/String;I)Landroid/content/SharedPreferences;

    move-result-object v0

    const-string v1, "points"

    invoke-interface {v0, v1, v2}, Landroid/content/SharedPreferences;->getInt(Ljava/lang/String;I)I


    move-result v0

    return v0
.end method
```

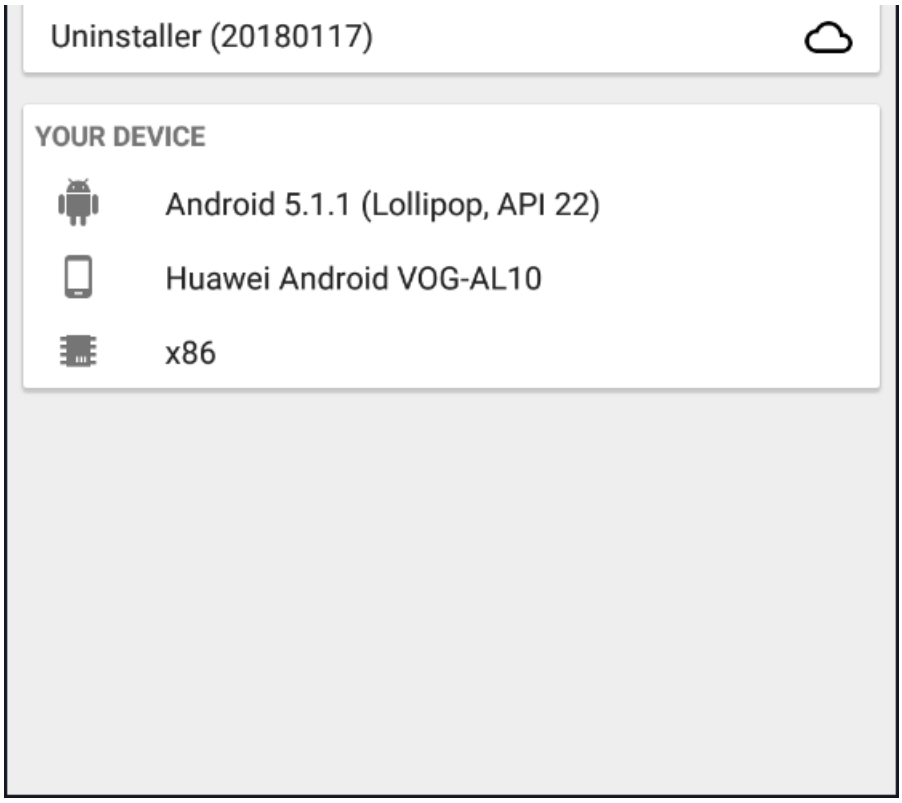
这个函数a, 它的返回值是一个整形，有3个局部变量：v0, v1, v2，先是调用方法getSharedPreferences(v0, v2), 然后将返回值存放在v0中，再调用接口方法getInt(),将返回值存放在v0中返回

## DAY_4

## 一、Xposed模块的安装， APKshelling的调试

根据老师给的步骤安装Xposed

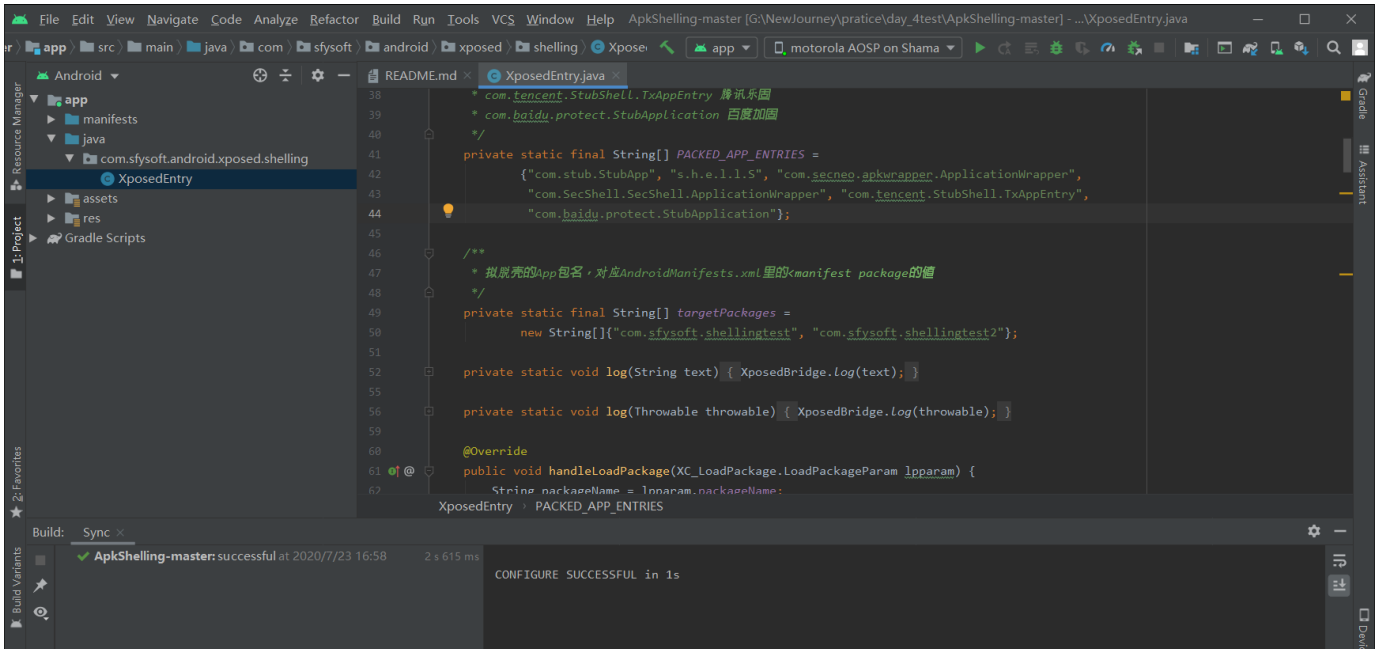




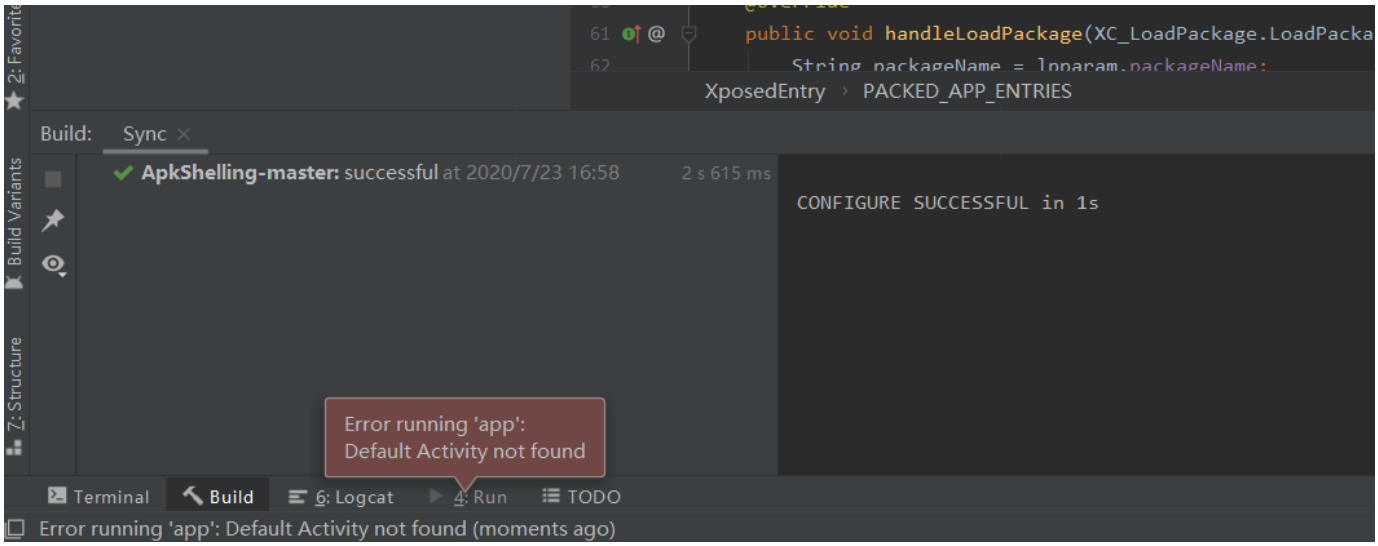

在夜神模拟器安装目录下，使用nox-adb命令连接Android Studio

```
Microsoft Windows [版本 10.0.18362.778]
(c) 2019 Microsoft Corporation。保留所有权利。

D:\Program Files\Nox\bin>nox_adb.exe connect 127.0.0.1:62001
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
already connected to 127.0.0.1:62001

D:\Program Files\Nox\bin>
```

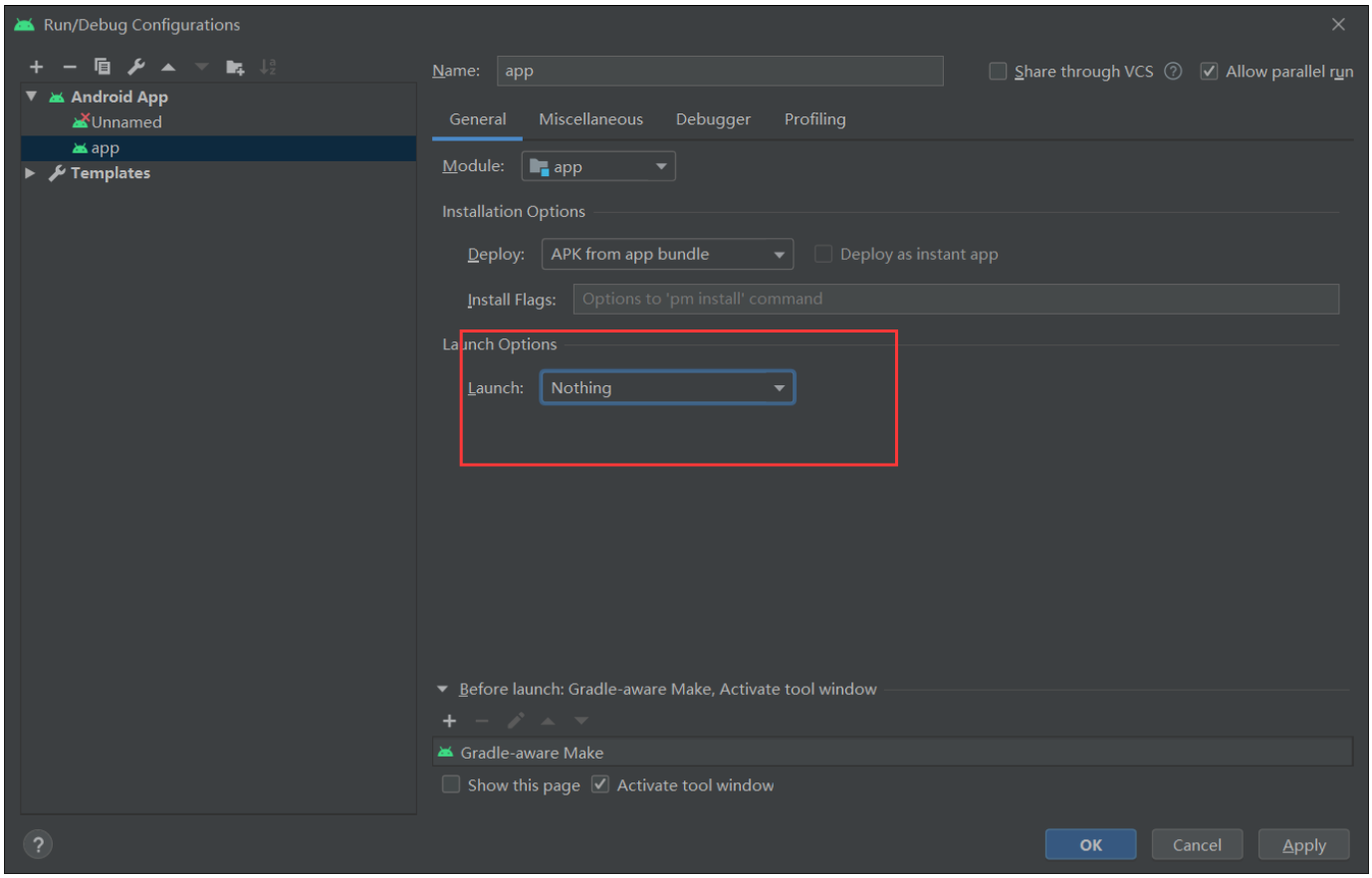
使用Android Studio打开ApkSheeling-master文件夹,

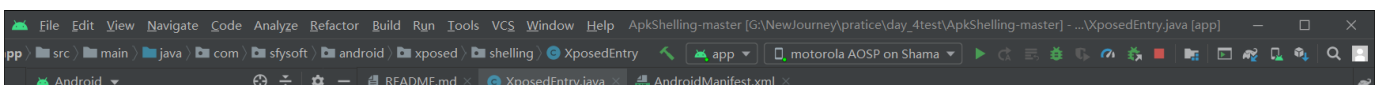


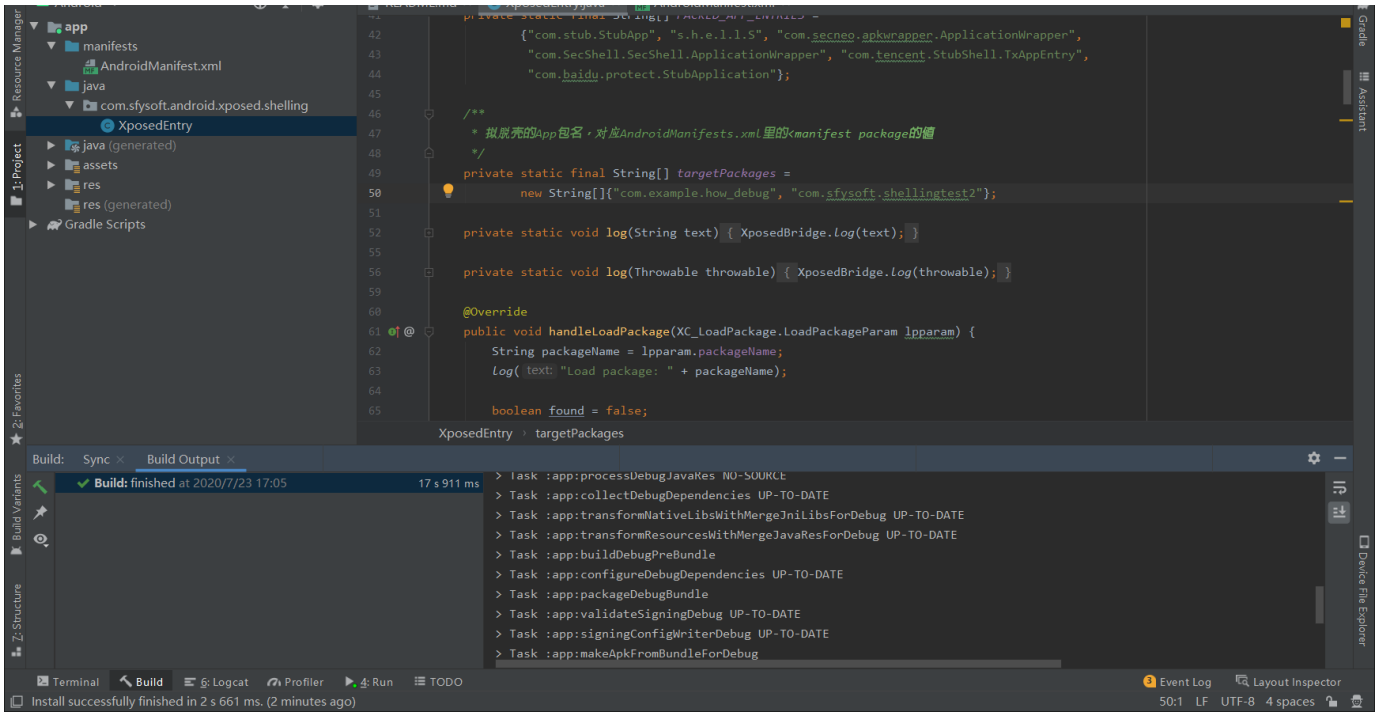

点运行时出现如下错误




将Launch中的Defaut Activity改为Noting后再次运行成功



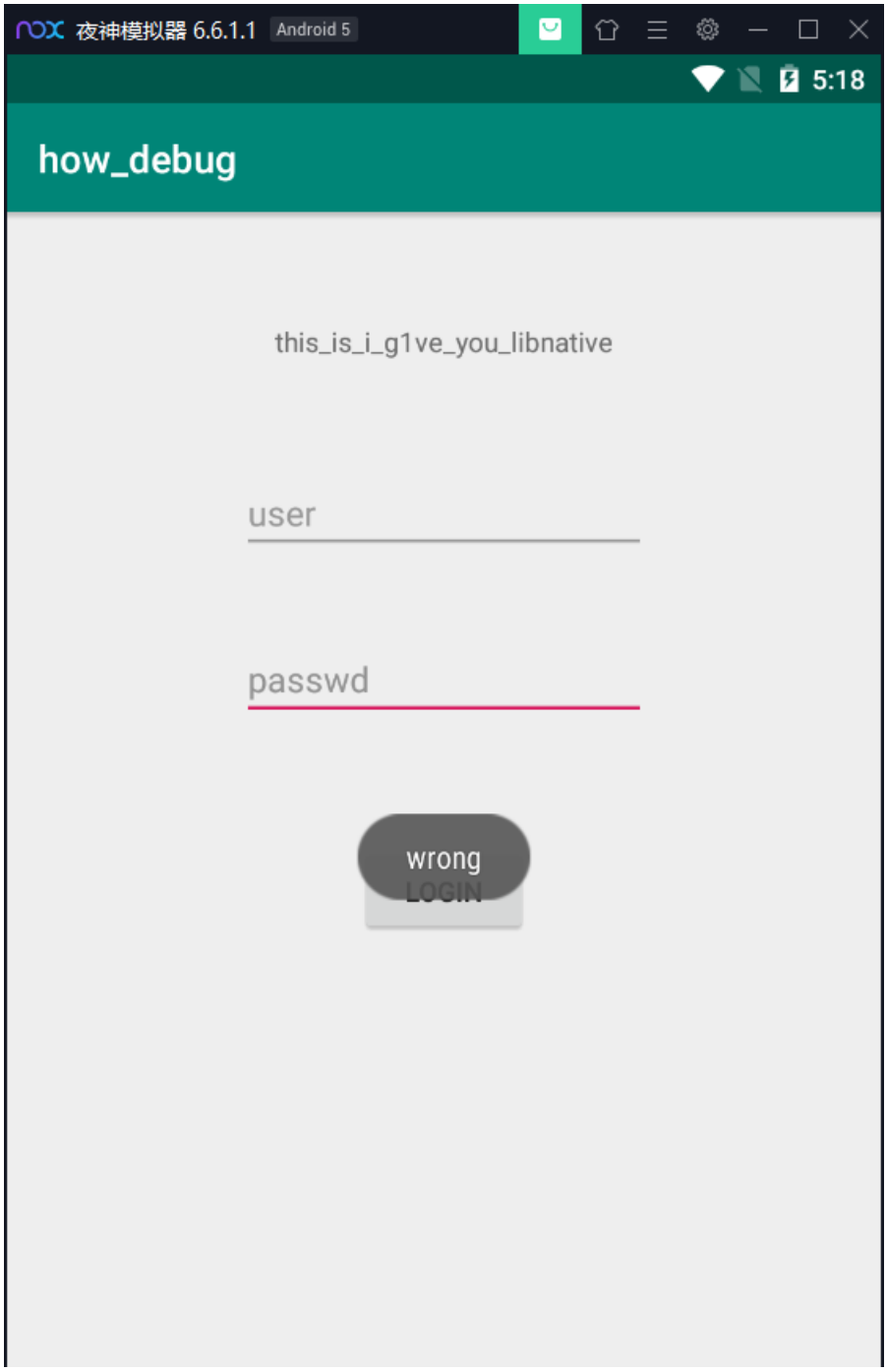



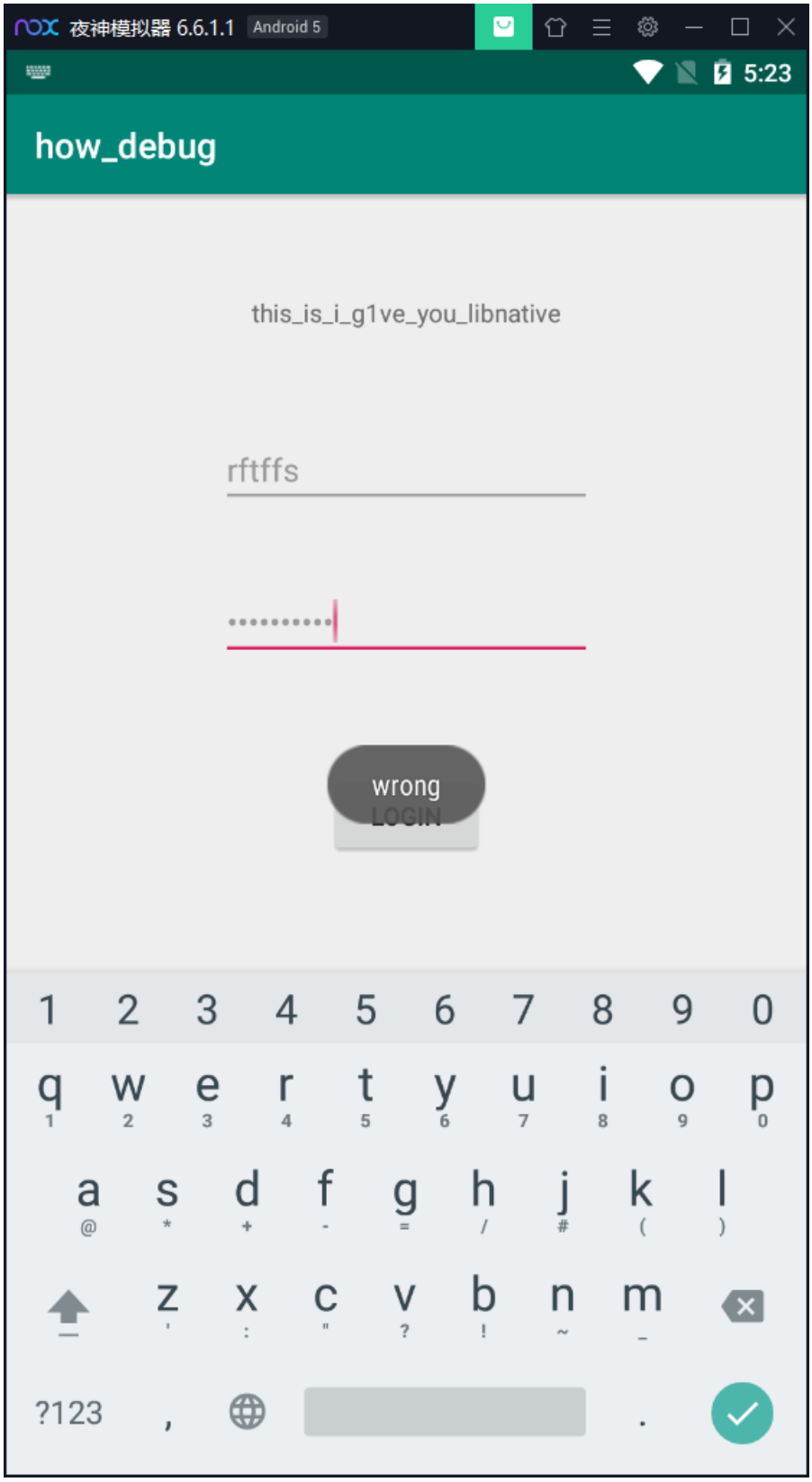



在模拟器中重新打开Xposed，在左上角的模块中选中脱壳,并重启设备

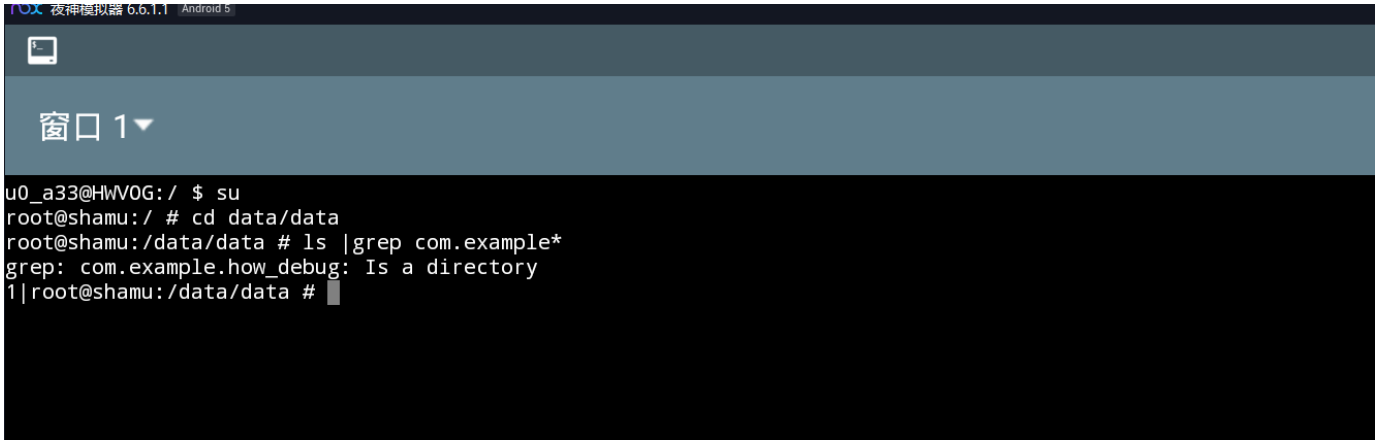
## 二、Bang获取flag

在模拟器进行安装signed.apk

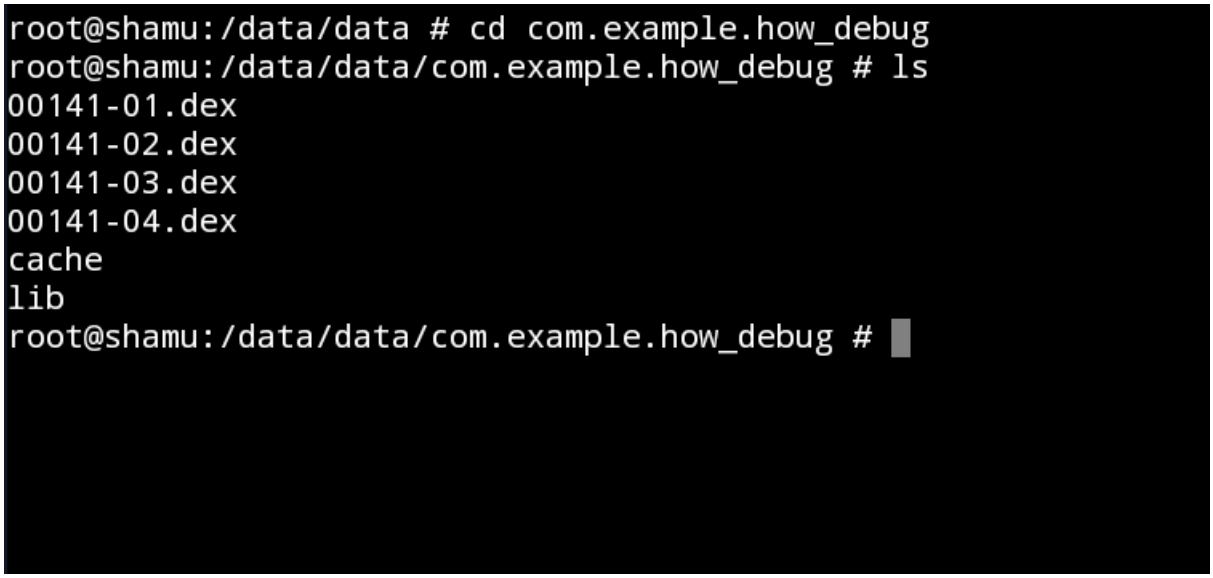




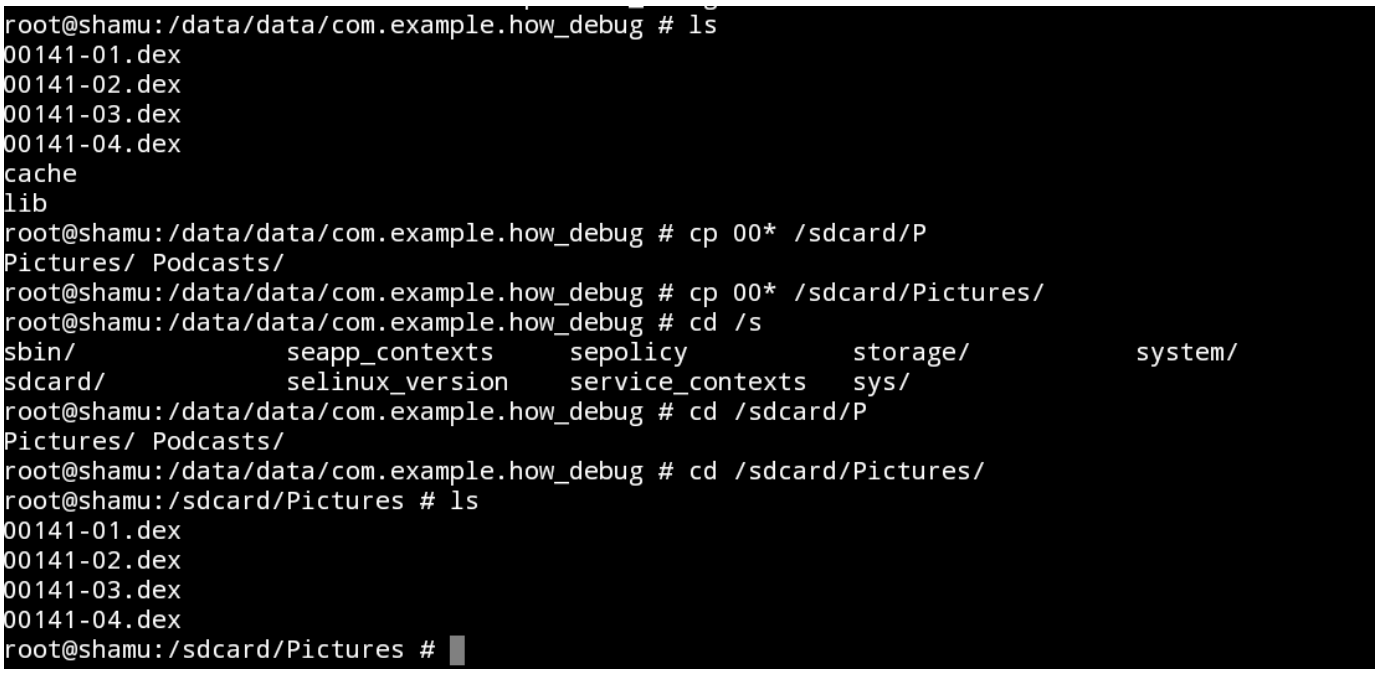

打开终端模拟器，获得root后，切换目录到/data/data，并通过grep查询到com.example.how_debug



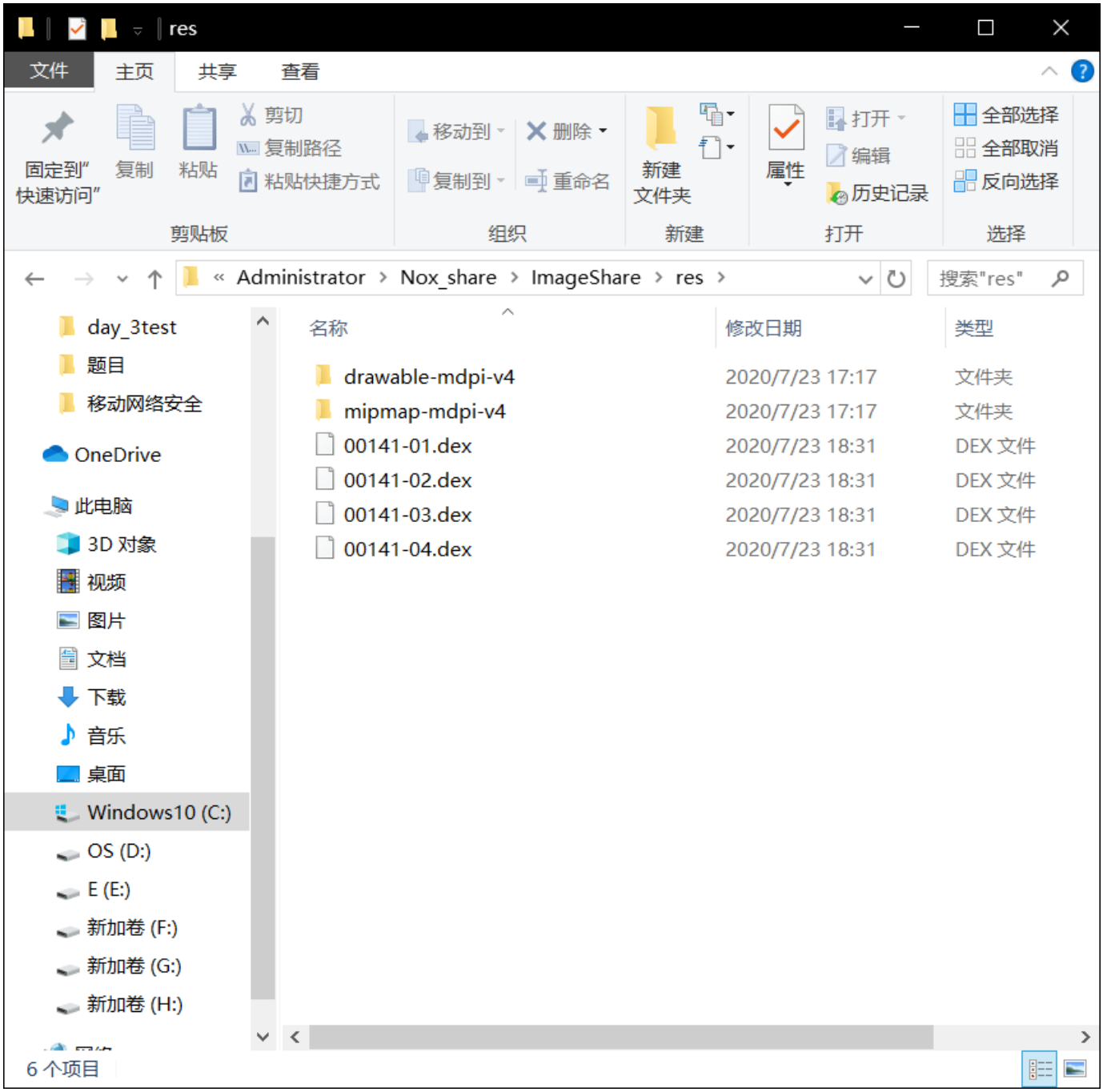

通过ls查看/data/data/com.example.how_debug



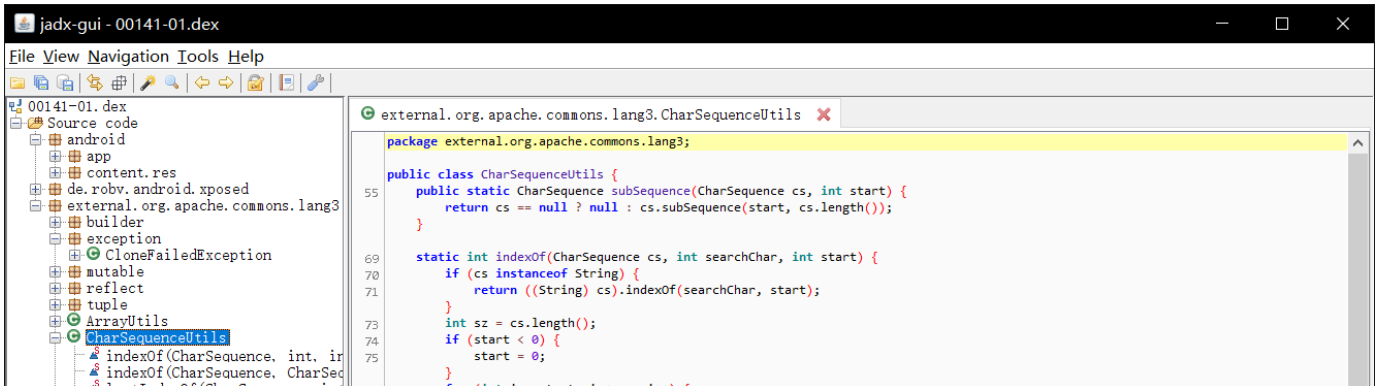

通过cp命令将*.dex文件复制到与windows的共享文件夹下

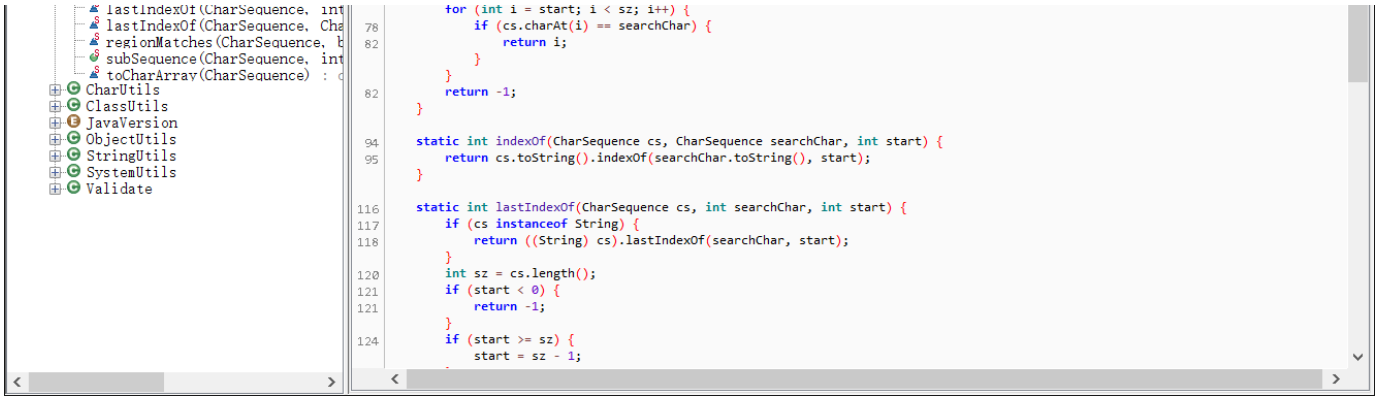


在windows共享目录下找到四个文件

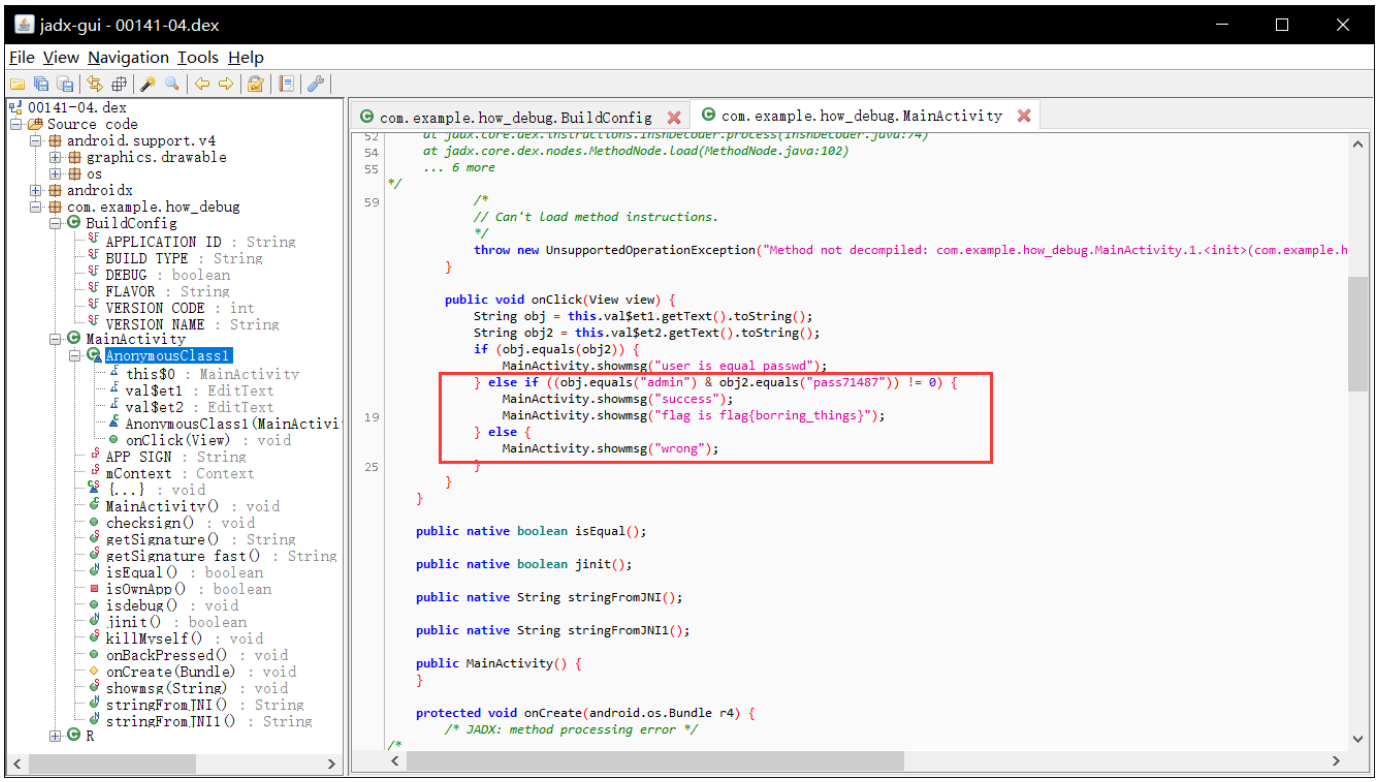



分别用jadx-gui打开进行查看






分别查看四个文档后得到flag: borring_things




**这周的学习已经临近尾声，一周的学习让我学到了不少之前没有接触过的东西，经过老师的讲解和自己的一些练习对这一系列的东西有了大致的了解，很高兴能在这里收获这么多新的东西。**

片转存中…(img-GyD0oqMH-1595646588392)]

分别查看四个文档后得到flag: borring_things

[外链图片转存中…(img-rVT0O4lz-1595646588394)]

**这周的学习已经临近尾声，一周的学习让我学到了不少之前没有接触过的东西，经过老师的讲解和自己的一些练习对这一系列的东西有了大致的了解，很高兴能在这里收获这么多新的东西。**