安卓逆向学习 之 KGB Messenger的writeup（1）

原创

CowboyBeboppp 于 2021-05-10 21:37:56 发布 214 收藏 2

分类专栏： 安卓逆向

本文链接：https://blog.csdn.net/weixin_38109420/article/details/116609430

KGB Messenger的writeup（1）

该题的 github地址为https://github.com/tlamb96/kgb_messenger,apk的下载地址为apk,

总结

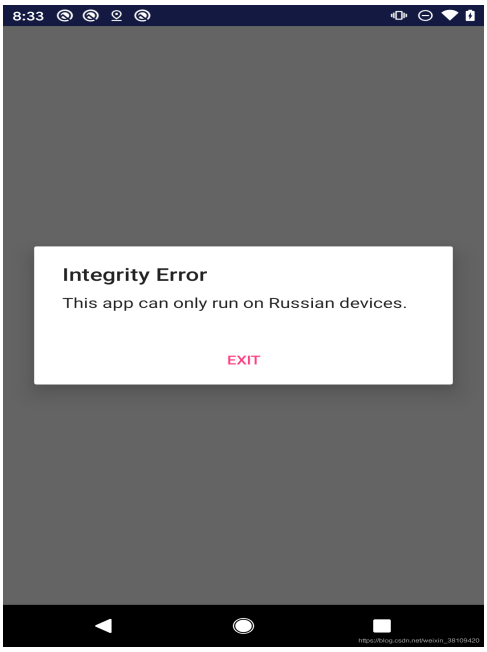
1. 清楚由apk到java源码的整个反编译过程：dex2jar+ jd-gui
2. **绕过**或者**去除**判断语句的思路
3. 作者使用**keytool**和**jarsigner**进行的签名，和平时使用的方法不同

challenges

就是说你是国际秘密情报局的逆向工程师。今天早上，你的团队负责人指派你检查一个有问题的APP。据传有个特工斯特林·阿切尔曾与一些克格勃间谍接触并使用了这个APP。你的工作是对这个APP进行逆向，以核实谣言。

Alerts(Medium)

第一个问题就是说当我们打开这个APP时他会弹出一个很恶心的窗口（苏卡不列！！），让我们调查一下。



点击exit直接就退出了。

获得FLAG

过程如下：

此弹窗在主界面中所以先找到intent-filter中的主activity，首先用APKtool解包：

```
C:\Users\DELL\Desktop\AndroidRe\tools\apktool>apktool d kgb-messenger.apk -o kgb
```

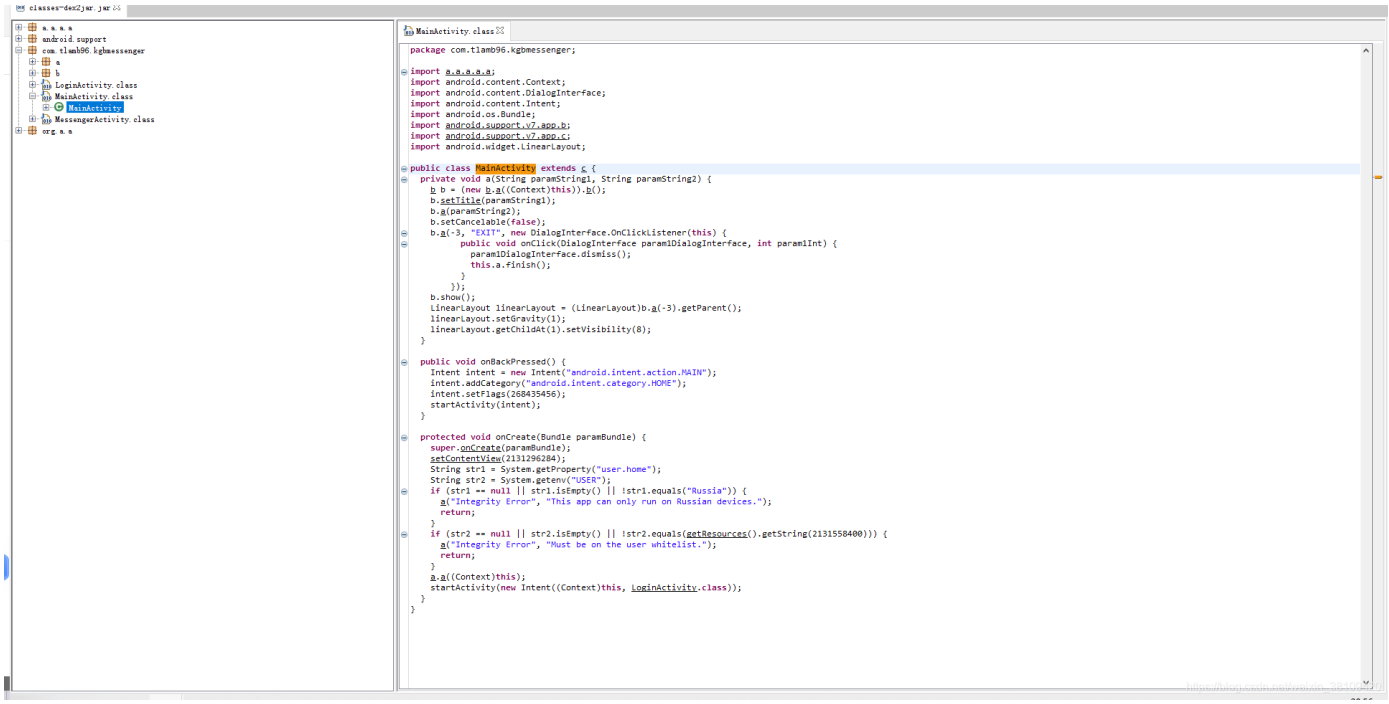
查看AndroidManifest.xml文件，发现com.tlamb96.kgbmessenger.MainActivity的属性如下：

```
1  <?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schema
2      <application android:allowBackup="true" android:icon="@mipmap/ic_kgb_launcher_icon" andr
3          <activity android:name="com.tlamb96.kgbmessenger.MainActivity">
4              <intent-filter>
5                  <action android:name="android.intent.action.MAIN"/>
6                  <category android:name="android.intent.category.LAUNCHER"/>
7              </intent-filter>
8          </activity>
9          <activity android:name="com.tlamb96.kgbmessenger.MessengerActivity"/>
10         <activity android:name="com.tlamb96.kgbmessenger.LoginActivity"/>
11         <meta-data android:name="android.support.VERSION" android:value="25.4.0"/>
12     </application>
13 </manifest>
```


所以将APK后缀改为zip后解压找到其中的dex文件用dex2jar反编译：

```
C:\Users\DELL\Desktop\AndroidRe\tools\apktool\kgb-messenger>d2j-dex2jar classes.dex
dex2jar classes.dex -> .\classes-dex2jar.jar
```

使用jd-gui打开查看java代码并找到上面的MainActivity，可以很清楚的看到里面的逻辑：



在onCreate中可以很明显的看到字符串的比较过程，若不满足条件就会弹窗，而且是会有两个弹窗。

第一个字符串若是Russia就可通过条件判断。

而第二个字符串与2131558400的字符串比较（这里是）

转化为16进制为

```
Type  help ,  copyright
>>> hex(2131558400)
'0x7f0d0000'
```

这里直接用Androidkiller简化后续过程（其他在terminal中查找字符串的方法可以看作者的workthrough video）

在Androidkiller中搜索此字符串，在pulic.xml中发现其name为User

```
<public type="string" name="User" id="0x7f0d0000" />
```

在string.xml文件中找到name='User'的字符串得到具体的值

```
<string name="User">RkxBR3s1N0VSTDFOR180UkNIM1J9Cg==</string>
```

使用base64解码可以获得**FLAG值**

```
FLAG{57ERL1NG_4RCH3R}
```


或者在linux中输入命令：

```
echo "RkxBR3s1N0VSTDFOR180UkNIM1J9Cg" | base64 -d
```

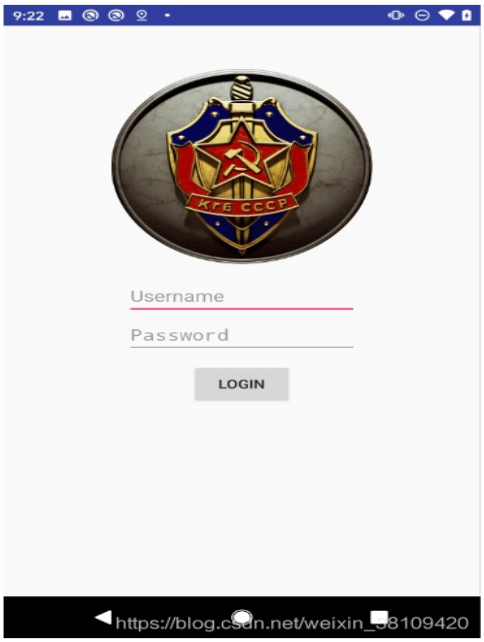
最终破解思路

在判断语句之前将s1和s2赋予这两个值就可以取消弹窗

```
invoke-super {p0, p1}, Landroid/support/v7/app/c;->onCreate(Landr
const v0, 0x7f09001c
invoke-virtual {p0, v0}, Lcom/tlamb96/kgbmessenger/MainActivity;-
const-string v0, "user.home"
invoke-static {v0}, Ljava/lang/System;->getProperty(Ljava/lang/St
move-result-object v0
const-string v1, "USER"
invoke-static {v1}, Ljava/lang/System;->getenv(Ljava/lang/String;
move-result-object v1
const-string v0, "Russia"
const-string v1, "RkxBR3s1N0VSTDFOR180UkNIM1J9Cg=="

if-eqz v0, :cond_0
invoke-virtual {v0}, Ljava/lang/String;->isEmpty()Z
```


使用android killer重新安装后打开app已经没有弹窗了至此alert破解完毕：



作者使用的方法是**删除中间的所有无关语句**从而达到效果。不过最后要注意末尾有一个goto:goto_0的语句要复制下来写到下面，是个return语句。

```
.method protected onCreate(Landroid/os/Bundle;)V
    .locals 3

    invoke-super {p0, p1}, Landroid/support/v7/app/c;->onCreate(Landroid/os/Bundle;)V

    const v0, 0x7f09001c

    invoke-virtual {p0, v0}, Lcom/tlamb96/kgbmessenger/MainActivity;->setContentView(I)V
```

中间的部分全部删除

```
    :cond_3
    invoke-static {p0}, La/a/a/a/a;->a(Landroid/content/Context;)V

    new-instance v0, Landroid/content/Intent;

    const-class v1, Lcom/tlamb96/kgbmessenger/LoginActivity;

    invoke-direct {v0, p0, v1}, Landroid/content/Intent;-><init>(Landroid/content/Context;Ljava/lang/Class;)V

    invoke-virtual {p0, v0}, Lcom/tlamb96/kgbmessenger/MainActivity;->startActivity(Landroid/content/Intent;)V

    goto :goto_0
    :goto_0
    return-void
.end method
```