

CTF MAKER	Kyle B3nac
CTF NAME	Injured_Android
CTF PLATFORM	ANDROID

### الأدوات :

- APKTOOL
- DEX2JAR
- JD-GUI
- ARTK

### المعلومات :

- عكس تطبيق الاندرويد
- تحليل [androidmanifest.xml](#)
- استغلال ثغره [PendingIntent](#)
- انشاء تطبيق لاستغلال الثغره

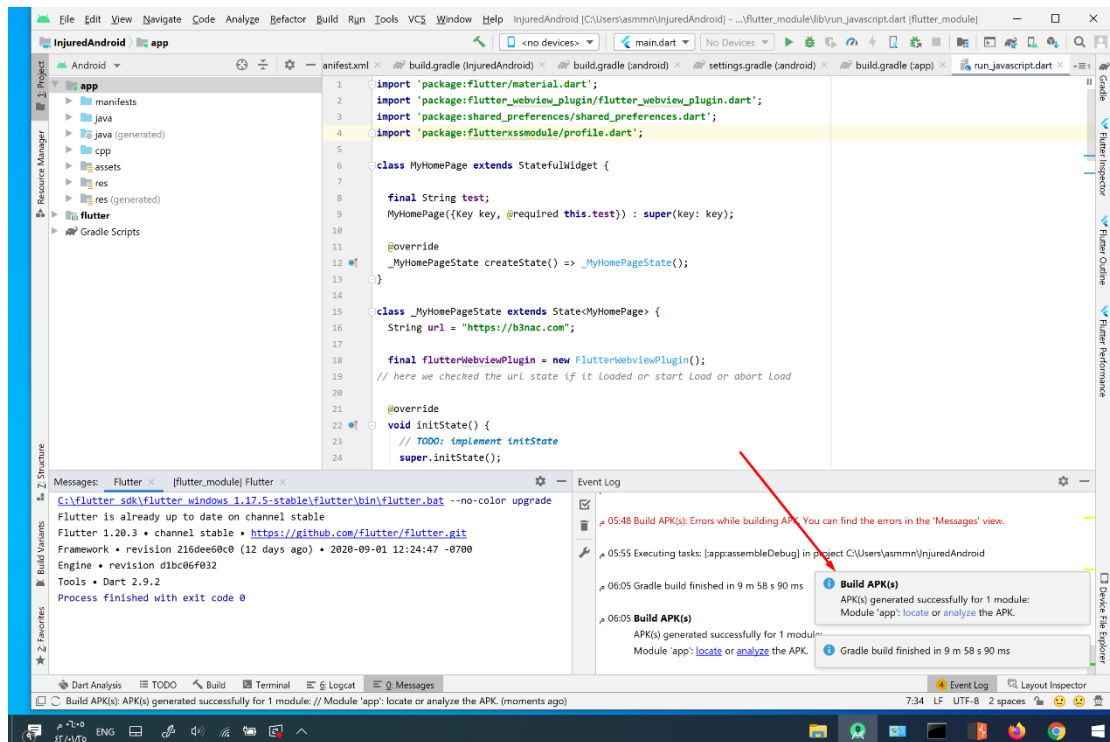
## تنويه :

ال CTF فيه اكثر من فلاق واكثر من ثغره ولكن المتطرق له هو ال **PendingIntent** وتكلمت عنها بالمختصر في قناة ال mobile-pentest

## ● البداية :

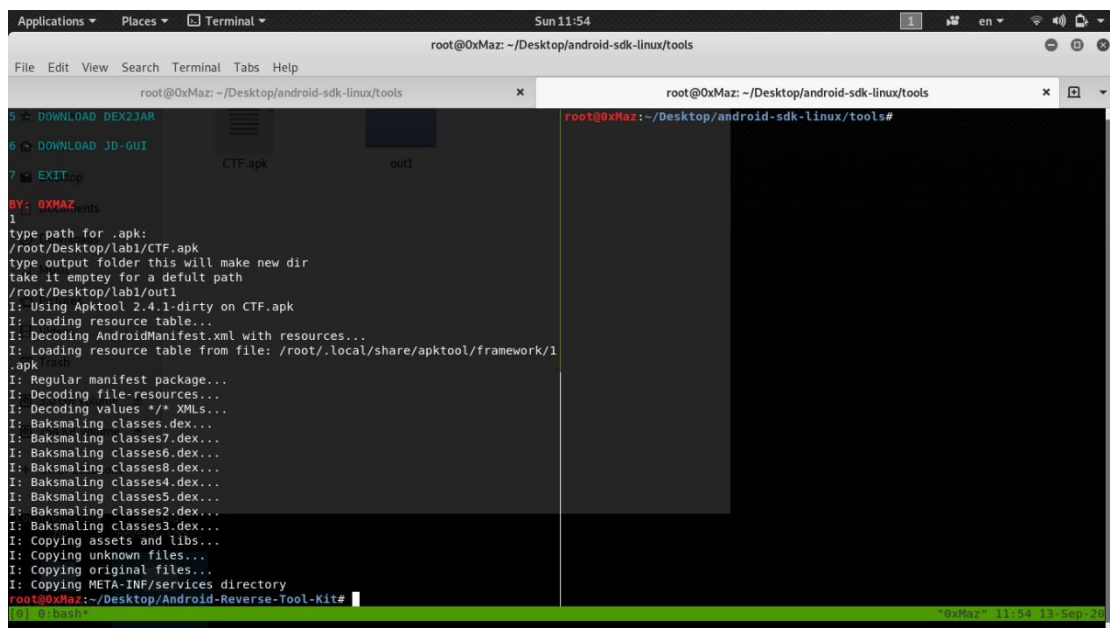
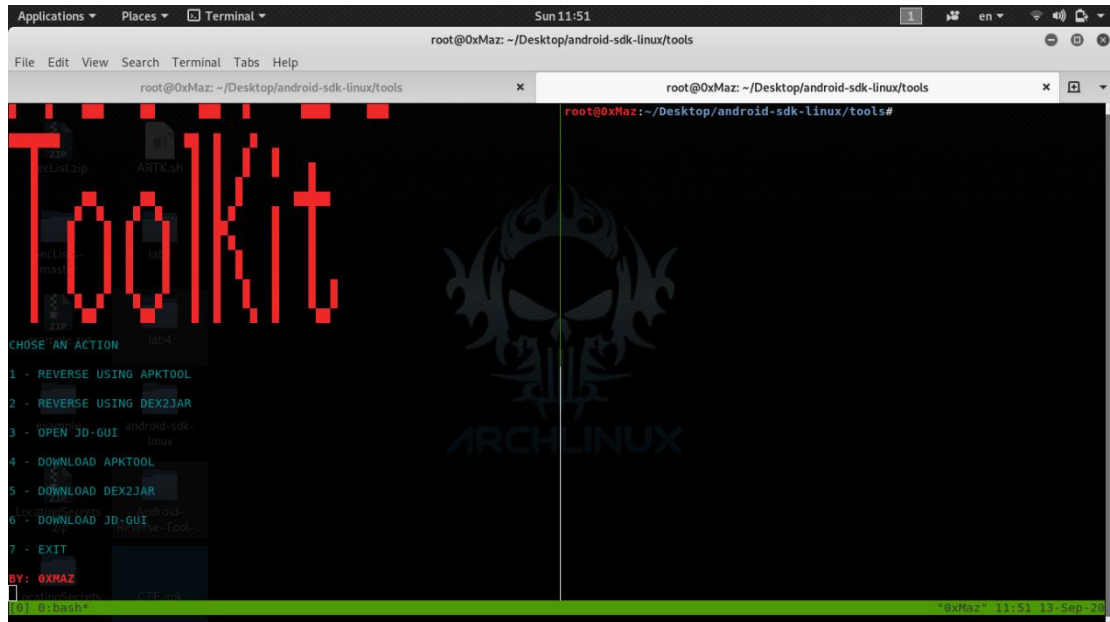
فالبداية مطور ال CTF حط السورس كود الخاص بالتطبيق ولازم انك تحل المشاكل وبعد كذا تسوي له كومبايل

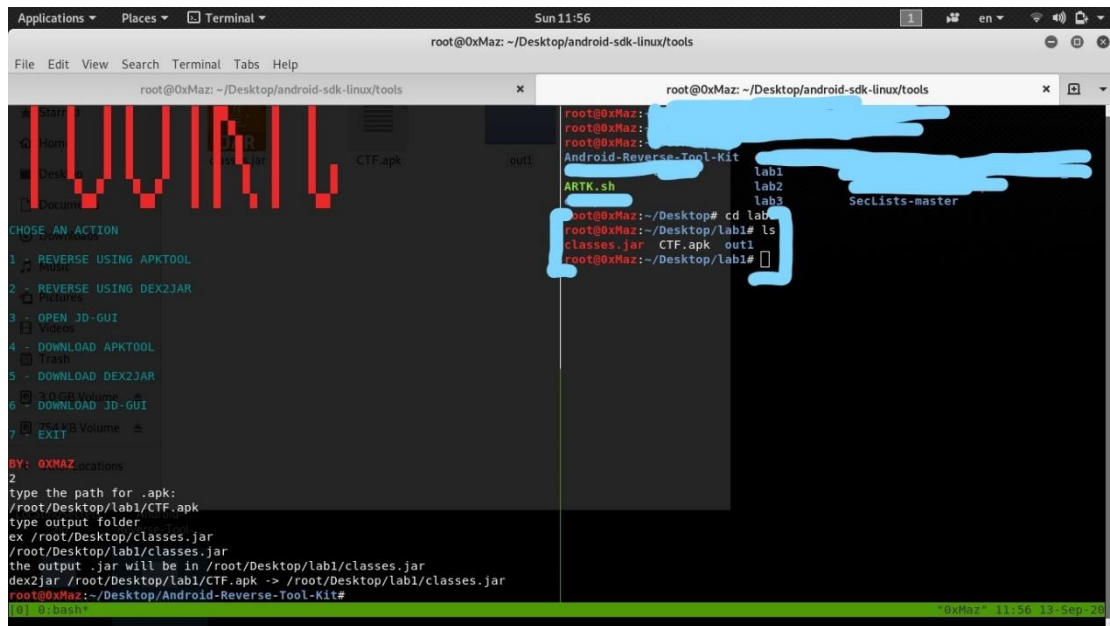
Build from source Build steps in progress. The flutter module makes this slightly more complicated



● عكس التطبيق :

## اول خطوه سویت ریفرس للتطبيق :





زي مو واضح بالصور استخرجنا الملفات .

## ● تحليل الكود :

بعد كذا ننتقل للخطوة الأهم وهي تحليل الكود لاستخراج الثغرات منه

```
root@0xMaz:~/Desktop/lab1# ls
classes.jar  CTF.apk  out1
root@0xMaz:~/Desktop/lab1# cd out1
root@0xMaz:~/Desktop/lab1/out1# ls
AndroidManifest.xml  lib          smali        smali_classes5  unknown
apktool.yml          META-INF    smali_classes2  smali_classes6
assets              original    smali_classes3  smali_classes7
kotlin              res         smali_classes4  smali_classes8
root@0xMaz:~/Desktop/lab1/out1#
```

## راح نبداً بملف ال androidmanifest.xml

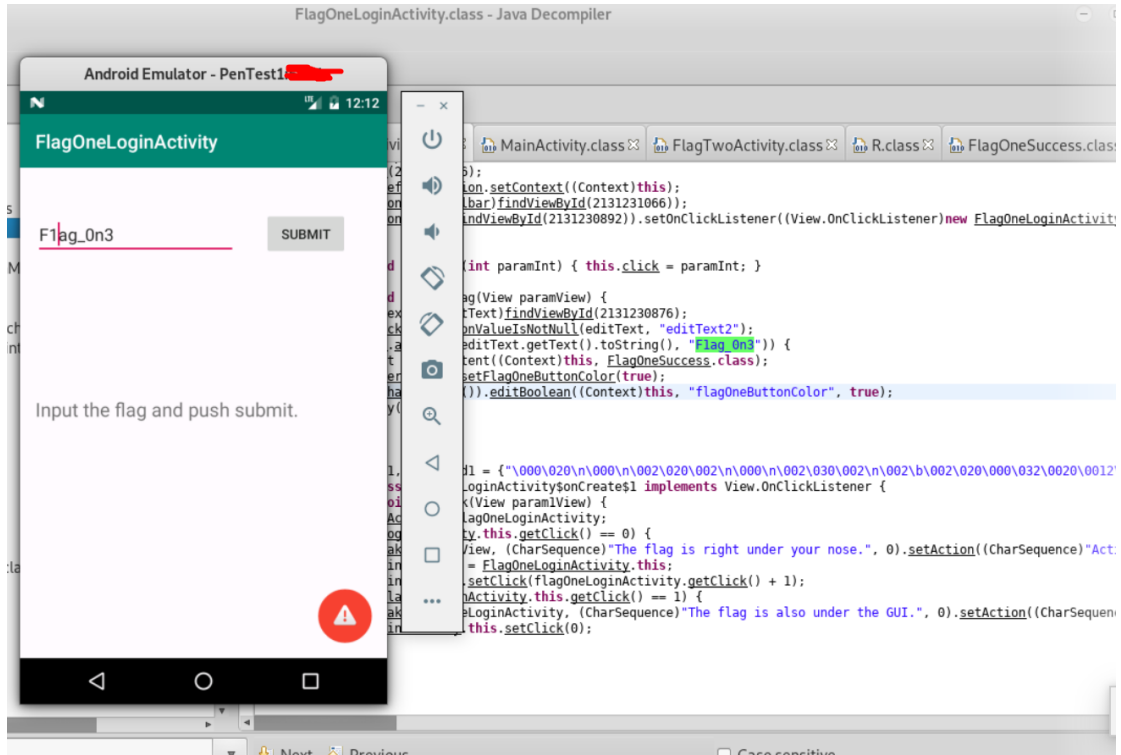
```
GNU nano 5.2      AndroidManifest.xml
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:and>
  <uses-permission android:name="android.permission.ACCESS_NETWORK_STAT>
  <uses-permission android:name="android.permission.INTERNET"/>
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STOR>
  <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
  <uses-permission android:name="android.permission.READ_EXTERNAL_STORA>
  <application android:appComponentFactory="androidx.core.app.CoreCompo>
    <activity android:label="@string/title_activity_assembly" android>
    <activity android:configChanges="density|fontScale|keyboard|keybo>
    <activity android:label="@string/title_activity_rce" android:name>
      <intent-filter android:label="filter_view_flag11">
        <action android:name="android.intent.action.VIEW"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <category android:name="android.intent.category.BROWSABLE>
        <data android:host="rce" android:scheme="flag13"/>
      </intent-filter>
    </activity>
    <activity android:name="b3nac.injuredandroid.SettingsActivity"/>
    <activity android:exported="true" android:label="@string/title_ad>
    <activity android:exported="true" android:name="b3nac.injuredandr>
    <activity android:label="@string/title_activity_flag_twelve_prote>
    <activity android:label="@string/title_activity_deep_link" android>
      <intent-filter android:label="filter_view_flag11">
        <action android:name="android.intent.action.VIEW"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <category android:name="android.intent.category.BROWSABLE>
        <data android:scheme="flag11"/>
      </intent-filter>
      <intent-filter android:label="filter_view_flag11">
        <action android:name="android.intent.action.VIEW"/>
      </intent-filter>
  </application>
</manifest>
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute  
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

"0xMaz" 11:59 13-Sep-20

بدايه حصلنا معلومه جميله وهو فلتر لل intent

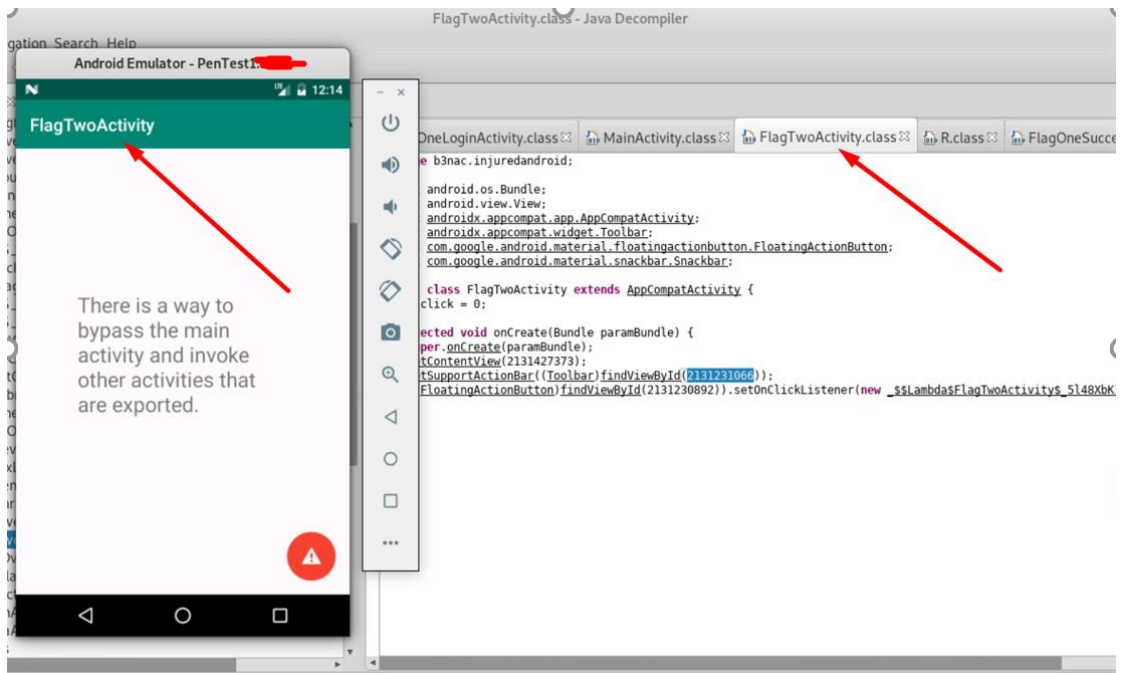
بعد كذا نستمر في التحليل وننتقل للكود



ضمن البحث حصلنا الفلاق للتحدي الأول (١)

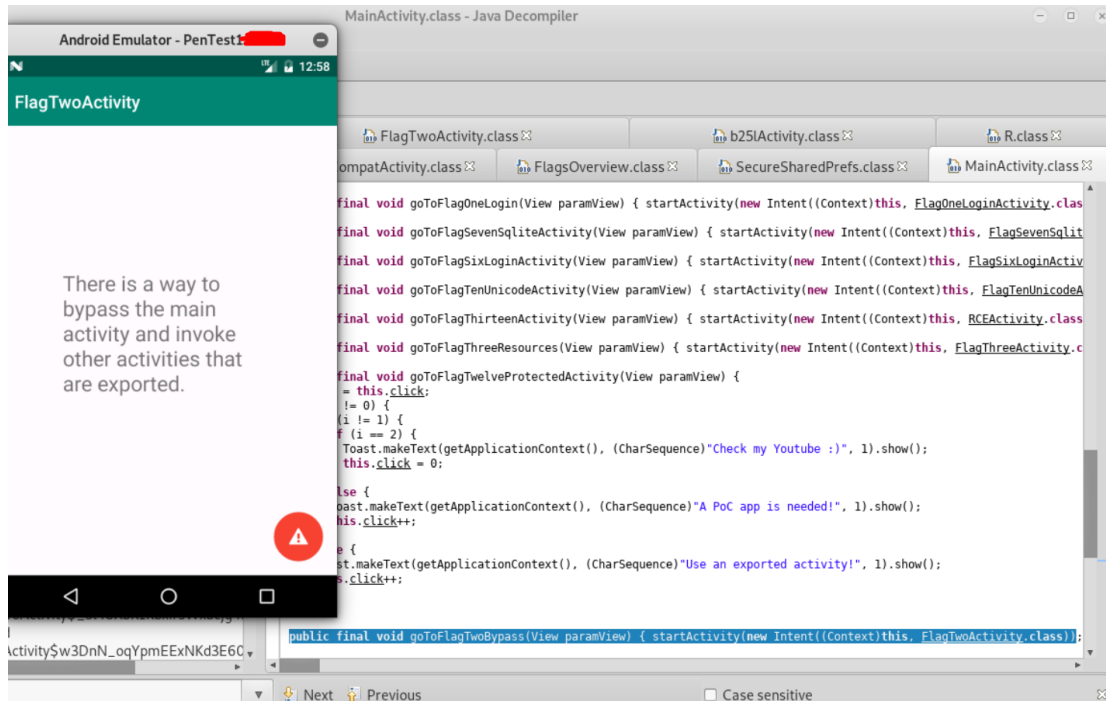
ولكن مو هذا اللي يهمنى

نستمر في التحليل وننعمق اكثر

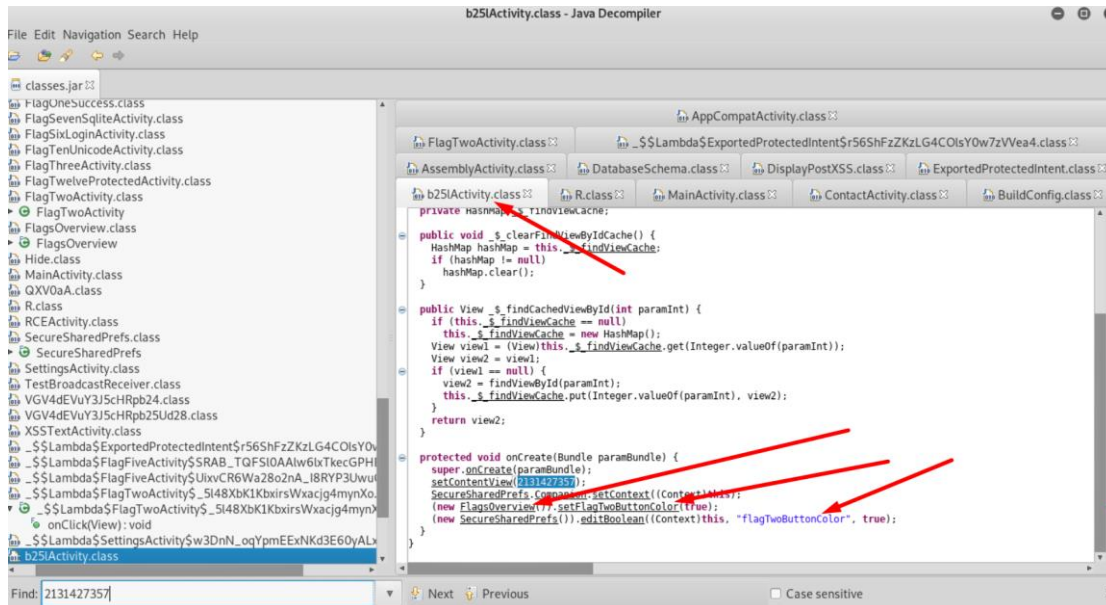




حصلنا الكلاس الخاص بالفلاق الثاني (وهو اللي يهمنى لانه متضمن للثغره)

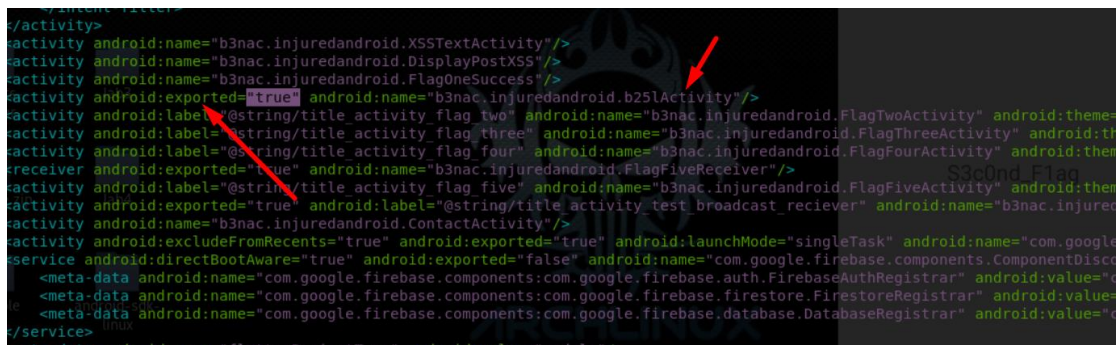


حصلنا كلاس وكانه راح يفيدنا ، نستمر برضه بالتحليل



برضه كلاس ثاني واسمه غريب ومرتب بالكللاس اوفر فيو ومنه مرتبط بفلاق 2 كلاس

## نبحث عنه في androidmanifest.xml



```
<activity
android:name="b3nac.injuredandroid.XSSTextActivity"/>
<activity android:name="b3nac.injuredandroid.DisplayPostXSS"/>
<activity android:name="b3nac.injuredandroid.FlagOneSuccess"/>
<activity android:exported="true" android:name="b3nac.injuredandroid.b25lActivity"/>
<activity android:label="@string/title_activity_flag_two" android:name="b3nac.injuredandroid.FlagTwoActivity" android:theme=
<activity android:label="@string/title_activity_flag_three" android:name="b3nac.injuredandroid.FlagThreeActivity" android:th
<activity android:label="@string/title_activity_flag_four" android:name="b3nac.injuredandroid.FlagFourActivity" android:them
<receiver android:exported="true" android:name="b3nac.injuredandroid.FlagFiveReceiver"/>
<activity android:label="@string/title_activity_flag_five" android:name="b3nac.injuredandroid.FlagFiveActivity" android:them
<activity android:exported="true" android:label="@string/title_activity_test_broadcast_reciever" android:name="b3nac.injured
<activity android:name="b3nac.injuredandroid.ContactActivity"/>
<activity android:excludeFromRecents="true" android:exported="true" android:launchMode="singleTask" android:name="com.google
<service android:directBootAware="true" android:exported="false" android:name="com.google.firebase.components.ComponentDisc
    <meta-data android:name="com.google.firebase.components:com.google.firebase.auth.FirebaseAuthRegistrar" android:value="c
    <meta-data android:name="com.google.firebase.components:com.google.firebase.firestore.FirestoreRegistrar" android:value=
    <meta-data android:name="com.google.firebase.components:com.google.firebase.database.DatabaseRegistrar" android:value="d
</service>
```

نلاحظ اننا نقدر نسوي له exported

خلونا نشرحها ع السريع :

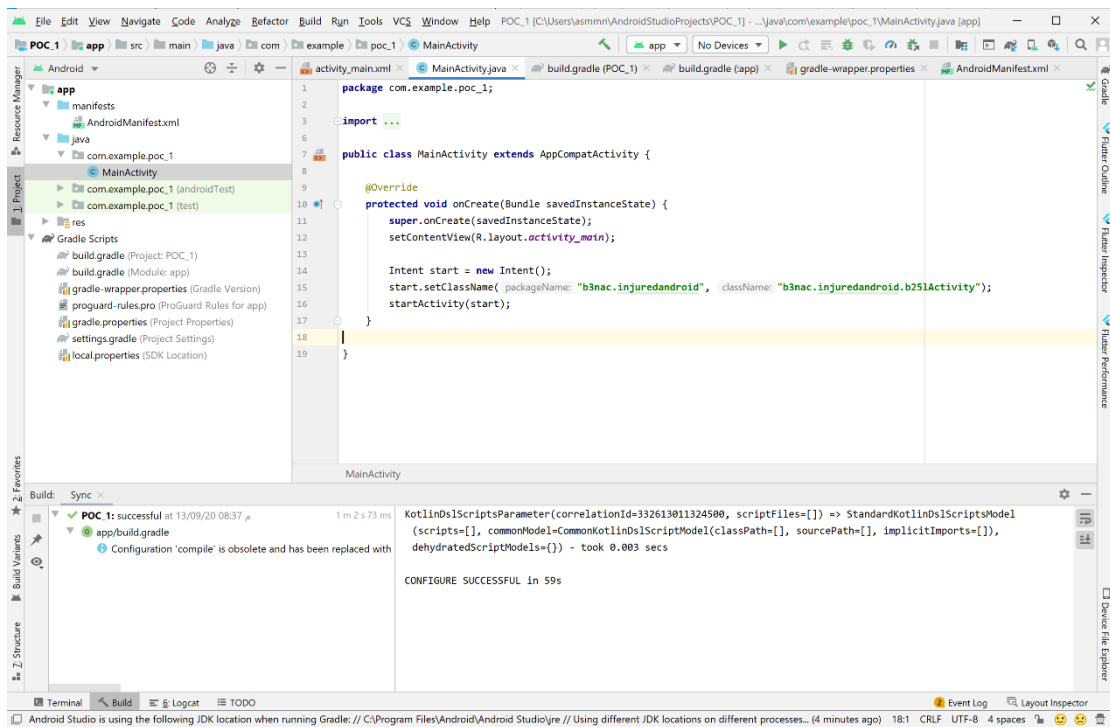
يسمونها ال content providers ومهمتها هي مشاركة ال structured data بين التطبيقات وعن طريقها تقدر تسمح للتطبيقات بالتواصل مع التطبيق الخاص بك عن طريق ال androidmanifest.xml زي ماحنا شايفين ان الكلاس .....b25 نقدر نسوي له اكسبورت لان موجود صلاحية

وموجود عندنا في AndroidManifest ثلاث انشطه او كلاسات نقدر نسوي لها exported وهو MainActivity وهو نقدر نسوي له exported افتراضيا لان فيه عامل تصفيه نيه intent filter وبرزه عندنا TestBroadcastReciever و b25lActivity وهي exported عن طريق الصلاحيات الموجوده في AndroidManifest زي مو موضح في الصورة



## • الاختراق :

نجي الحين للخطوة المهمة وهي استغلال ال pendingintent vulnerability راح اسوي كود يوضح ويفصل طريقه عمل الثغره بطريقه برمجيه



شرح مبسط للكود :

سويت intent اسمها start بعدها استدعيت البكج الخاص بالتطبيق وبعده الكلاس المصاب بالثغره، وبعد بكل بساطه سويت تشغيل لل intent اللي سويتها

فيديو لطريقه عمل الثغره وراح يوضح لكم اكثر :

[فتح الفيديو فالمتصفح](#)