# **Intent Attack Surface CTF Write-up**

**Author**: Nephat  
**CTF Series**: Intent Attack Surface

## **Tools Used**

* **ADB (Android Debug Bridge)**: For launching and interacting with Android activities.
* **APKTool**: For decompiling the APK and analyzing the structure.
* **Jadx**: For viewing the decompiled source code to understand activity behavior.On kali linux you can use *sudo apt install jadx*

**Challenge:Flag1**  
**Challenge Name**: Basic Exported Activity  
**Objective**: Identify and interact with an exported activity io.hextree.attacksurface/.activities.Flag1Activity in the Android application to retrieve the flag.

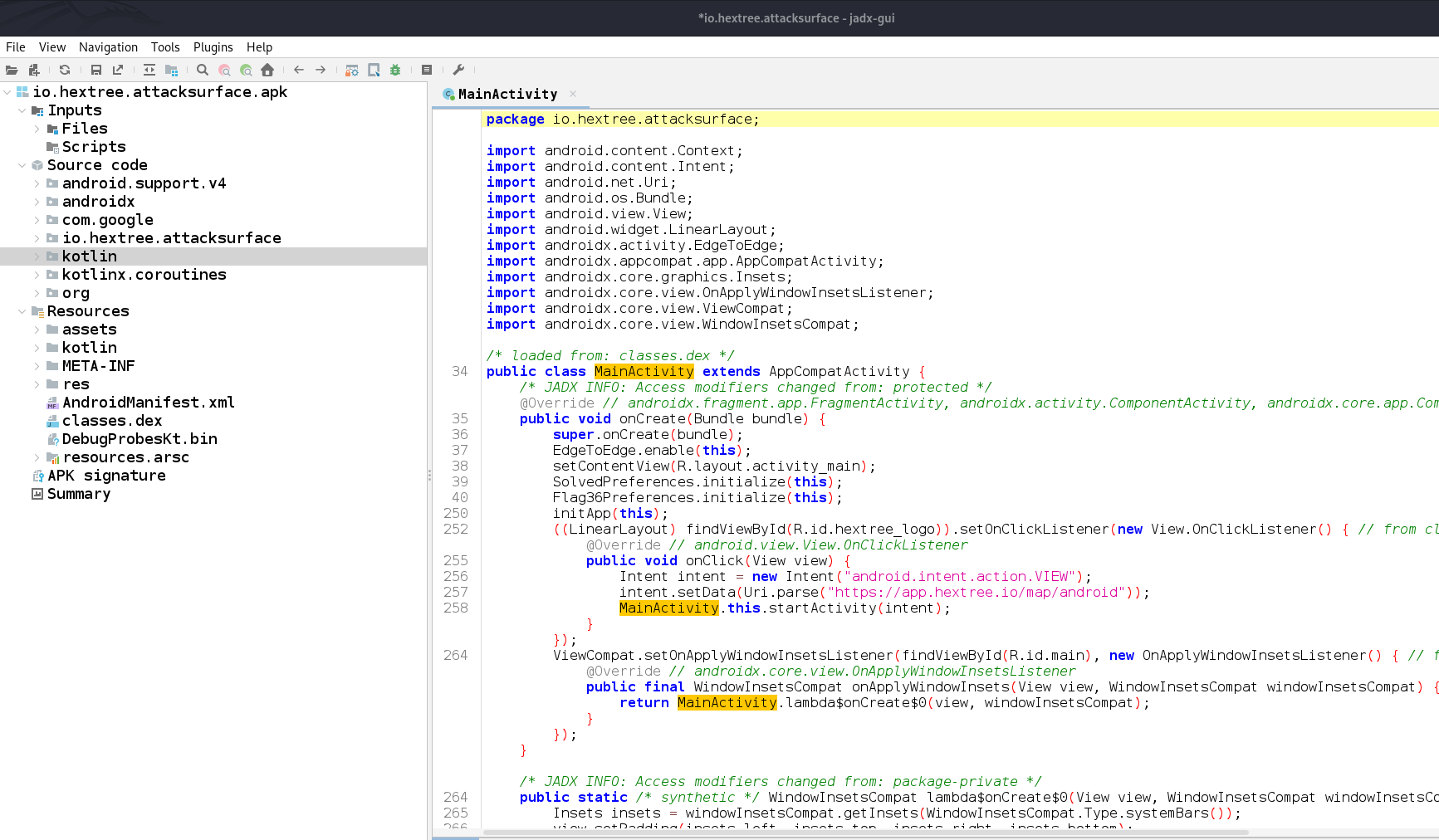
## **Introduction**

The "Basic Exported Activity" challenge is part of the Intent Attack Surface CTF series, focusing on understanding and exploiting exported activities within Android applications. Exported activities can sometimes be unintentionally exposed, potentially creating security vulnerabilities. In this challenge, I located and accessed an exported activity to retrieve the flag.

## **Solution**

### **Decompiling the APK**

I began by decompiling the APK using **jadx** to analyze its structure:



### **Inspecting AndroidManifest.xml**

* I examined the AndroidManifest.xml file to identify exported activities. I looked for activity tags with the attribute android:exported="true", which indicates that they can be accessed by other applications or via ADB commands.

The relevant activity I found was:  
xml  
<activity

android:name="io.hextree.attacksurface.activities.Flag1Activity"

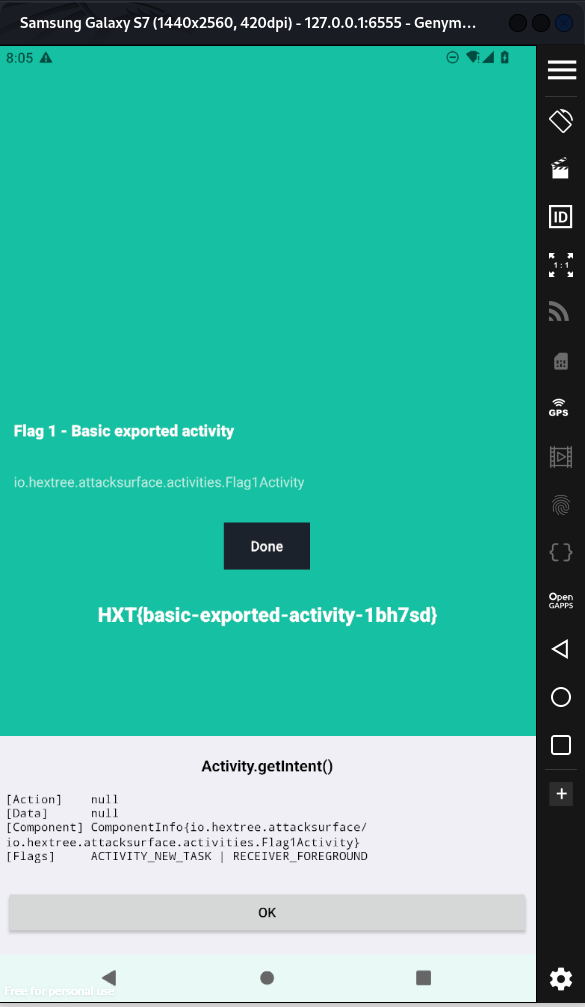
android:exported="true"/>

### **Launching the Exported Activity with ADB**

I used ADB to start the exported activity directly, bypassing the main entry point of the application. The command used:

adb shell am start -n io.hextree.attacksurface/.activities.Flag1Activity

* This launched the Flag1Activity, which displayed a screen with a flag.



## **Conclusion**

The "Basic Exported Activity" challenge demonstrated the potential security implications of exported activities in Android applications. By decompiling the APK, identifying the exported activity, and accessing it directly, I was able to retrieve the flag. This exercise underscores the importance of reviewing AndroidManifest.xml configurations to ensure sensitive activities are not inadvertently exported.