Writeup — easy-rasp.apk

Challenge / target: easy-rasp.apk
Flag: CRACCON{RASP_1s_Easy}
Summary: The APK implements a weak RASP/anti-tamper check; I bypassed the runtime checks with a Frida hook (or by patching the APK) and extracted the hardcoded flag.

1) Tools used

• apktool — decode / rebuild APK (smali editing)
• jadx / jadx-gui — decompile classes.dex
• strings / unzip — quick inspection
• adb — install/run on emulator/device
• frida & frida-server — dynamic instrumentation
• apksigner / jarsigner — re-sign APK
• Android emulator or rooted device

2) Static analysis / reconnaissance

1. unzip -l easy-rasp.apk — inspect contents
2. strings easy-rasp.apk | grep -i rasp — quick search
3. jadx-gui easy-rasp.apk — decompile and search

Observation: Strings like 'easy_rasp' found. Flag not visible verbatim; constructed at runtime after RASP checks.

3) Dynamic analysis / bypass approaches

Two approaches:
A — Runtime hook (Frida)
B — APK patching (apktool + smali)
Example Frida hook (replace class/method names as found via JADX):

```
Java.perform(function(){
  var TargetClass = Java.use("com.example.easy_rasp.FlagProvider");
  TargetClass.getFlag.overload().implementation = function(){
    console.log("[+] getFlag() called — returning challenge flag");
    return "CRACCON{RASP_1s_Easy}";
  };
});
```

APK patching steps:
1. apktool d easy-rasp.apk -o easy-rasp-src
2. Edit smali to bypass checks
3. apktool b easy-rasp-src -o easy-rasp-patched.apk
4. apksigner sign --ks my-keystore.jks --out easy-rasp-signed.apk easy-rasp-patched.apk

4) Evidence & verification

After bypassing RASP checks with Frida or patched APK, the application revealed the flag:
CRACCON{RASP_1s_Easy}

5) Recommendations / Mitigations

• Never embed secrets in client apps — keep on server.
• Avoid security by obscurity; obfuscation is not protection.
• Implement server-side checks and validations.
• Use layered runtime integrity checks.

6) Appendix — useful commands

```
unzip -l easy-rasp.apk
strings easy-rasp.apk | grep -i rasp
jadx-gui easy-rasp.apk
apktool d easy-rasp.apk -o easy-rasp-src
```

```
apktool b easy-rasp-src -o easy-rasp-patched.apk
apksigner sign --ks mykeystore.jks --out signed.apk easy-rasp-patched.apk
frida -U -f com.example.easy_rasp -l bypass_flag.js --no-pause
```