Android Security Lab 2

Goal: patching the bytecode from app of Lab1

Report:

- 1 per group of 2 or 3 students
- no more than 2 or 3 pages
- to be provided at the latest on Monday January 20 2025 (i.e. before the exam) by email at polytech@securingapps.com

Bytecode patching allows to modify any part of the code, e.g. 2 lines in the middle of a function, and without a rooted device. Whereas hooking only allows to see input/outputs or to overload a full method.

- Intro: slides 16-17-18
- Download apktool.jar from https://bitbucket.org/iBotPeaches/apktool/downloads/ (or install with instructions at https://ibotpeaches.github.io/Apktool/install/)
- Make sure you have Java installed
- `java -jar apktool_2.10.0.jar d app1.apk`
- Look at smali code in `app1/smali`

- Open app1.apk with jadx
- Think about one line of code you would like to add in a specific location
- With Android studio, create a new “No activity” application called Dummy
- In this app, include only the line of code you would like to add in app1.apk
- Build it in as an apk with Build menu/ Build bundle(s)/Build APK(s)
- Click Locate to find app-debug.apk
- `java -jar apktool_2.10.0.jar d app-debug.apk`
- Look at smali code in `dummy/smali_*`
- Find where is your class and your corresponding line(s) of code

- Manually merge your smali class of app-debug in app1
  - Find in `app1` the smali file and location where to inject your code
  - Copy paste the relevant section of your smali code at this location
  - Adapt variable names if needed
- In `app1` folder, launch “`java -jar apktool_2.10.0.jar b .`” to build your patched apk
- Result is in `dist` subfolder
- Open `dist1/app1.apk` with jadx and verify if your code insertion looks OK
  - If not OK, try again editing smali/rebuild APK (and so on)

- When patched apk code finally looks OK in jadx, signature is required
- Generate a dummy signature key with

```
 keytool -genkey -v -keystore patch.keystore -alias patch
-keyalg RSA -keysize 2048 -validity 10000
```

- Sign patched apk with
  ```
  jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1
  -keystore patch.keystore app1.apk patch
  ```

- Now it's time to test in the emulator
- Find adb tool location: cf Lab1
- Run command: `./adb install -r app1.apk` (and check status message is successful)
- Launch app from the UI of the emulator
- Check that your extra code is executed and allows to steal the secret