

1 File Contents

Apart from this file, the contents of this repositories are:

Extended Paper.pdf Contains the extended contents for the paper

files/simpleapp.apk A simple test app which contains no method calling and exception handling in the program. This app is compiled using the option of no-optimize.

files/apkreader.apk A prototype of the proof of concept. This app can read other app's APK and then check it against the certificate. The checked app can not contain method calling and exception handling. The app must also compiled with the flag no-optimize.

files/APKReader/ The source code for the prototype. Two main contents of this app are the component to parse another file's apk, and the type check it.

files/dx.ocaml/ The source code for the OCaml files to provide a certificate for an app given its classes source. This component will first parse the JVM class, do a naive JVM type inference, then translate the certificate.

files/SimpleApp/ The source for the test app.

2 Instructions for Constructing Certificate

At this stage, since we only have the translation from JVM classes to DEX, we need to get the classes from a half compiled Android app. Using the Android Studio we can do this by first building the non-optimized app, then get the classes from the intermediates directory. After we get the classes we feed these classes into the OCaml program to get the certificate which contains the typing for the Android app. Finally we rename the certificate into "Certificate.cert" and insert it into the "assets/" directory, and compile the application.

3 Instructions for Compilation

3.1 APK Reader

We developed this APK Reader using the Android Studio tool, which you can get from <http://developer.android.com/develop/index.html>. We have tested this app in our own phone: Samsung Galaxy 4, model number of GT-I9506, and Android version 5.0.1.

3.2 Test application

We developed this test using the Android Studio tool, which you can get from <http://developer.android.com/develop/index.html>. A note about this test application is that we have to put the option of no-optimize for the DX compiler. This is a little bit complicated with the Android studio which uses gradle version 1.5. One way to circumvent this problem is to build the app using lower version of gradle (we used 1.3.1). To do this we can modify the graddle setting for the project (contained in “build.gradle” at the root of the project directory), changing the value of classpath from ‘com.android.tools.build:gradle:1.5.0’ to ‘com.android.tools.build:gradle:1.3.1’. Then we modify the graddle setting for the module (contained in “app/build.gradle”) . We inject the following code into the gradle file:

```
afterEvaluate {  
    tasks.matching { it.name.startsWith('dex') }.each { dx →  
        if (dx.additionalParameters == null) { dx.additionalParameters = [] }  
        dx.additionalParameters += '-no-optimize' }  
}
```

I have tested this app in my own phone: Samsung Galaxy 4, model number of GT-I9506, and Android version 5.0.1.

3.3 Certificate Translation

There is a makefile for the OCaml program already, so we just need to type “make” in in the root directory. The output will be a binary called “dx_ocaml”.

4 Instructions for Type Checking

After opening the application, just provide the package name for the application that you want to type check into the input text, and then click on the button. The application will then parse the DEX file (which takes a really long time) and the certificate, then provide the final judgement whether the application is type check or not.