Permission-Based Android Malware Detection

1.0 Introduction:

In order to mitigate all these mentioned Android malware problems (in the attached paper) There is a proposal of new model which is based on feature selection as a first phase and Classification based on built dataset as second phase.

The process followed to obtain data from the android application file. The general steps that have been followed for each application are:
1. Downloaded and collected malware and good ware applications from application market.
2. Decompressed downloaded applications to extract the content.
3. Extracted the permission request features (*android.permission.PERMISSION*) from each application.
4. Building a dataset in an ARFF or CSV file format with the extracted data

1.1 Getting Started

The list of .apk files that have been downloaded for the *android.permission* feature extraction in the **Apk Files** folder are.

Goodware/Benign Applications:

1. WhatsApp Messenger_v2.18.46_apkpure.com.apk
2. Facebook_v156.0.0.36.100_apkpure.com.apk

Malware Applications:

1. org.benews.apk
2. com.BioTechnology.iClientsService44370

1.2 Pre-requisites:

The list of softwares that have been used in the Module-01 as pre-requisites are,

1. PyCharm - For executing python codes.
2. Notepad++
3. Windows Command Prompt

The list of Folder and Sources in the zipped folder,

1. Apk files
2. Apk Source Files
3. Python Source Files
4. Screenshots - Testcases
5. Tools.
6. Extracted features - datasetandroidpermissions.zip
7. Readme.pdf
8. Permission-based-Android-Malware-Detection.pdf

2.0 Reverse Engineering:

2.1 Decompilation and Indexing:

`Step 1:`

Make a new folder and put .apk file in it (which you want to decode). Now rename the extension of this **`.apk`** file to **`.zip`** (eg.: rename from filename.apk to filename.apk.zip) and save it

`Step 2:`

Now extract this zip apk file in the same folder. Now download dex2jar from the Tools folder.

and extract it to the same folder. Now open command prompt and change directory to that folder. Then write **`dex2jar classes.dex`** and press enter. Now you get **`classes.dex.dex2jar`** file in the same folder.

`Step 3:`

Then download java decompiler from **Tools** folder

And now double click on jd-gui and click on open file. Then open **`classes.dex.dex2jar`** file from that folder. Now you get class files and save all these class files.

The extracted readable form of source files are curated in the **Apk Source Folder.**

2.2 Extracting android.permission features from the AndroidManifest.xml file and writing them into a get_permisions.txt file.

`Step 1:`

Execute the **`get_permissions.py`** file from the **Python Source Files** folder to extract all the android.permission features and concatenating the extracted features to “get_permissions.txt” files.
The output of extracting features have been captured in the **Screenshots - Testcases** folder in the following order,

1. **Whatsapp- get_permissions.jpeg -** Features of WhatsApp Messenger_v2.18.46_apkpure.com.apk
2. **Facebook- get_permissions.jpeg -** Features of Facebook_v156.0.0.36.100_apkpure.com.apk
3. **Benews- get_permissions.jpeg -** Features of org.benews.apk
4. **FacebookOTP- get_permissions.jpeg -** Features of com.BioTechnology.iClientsService44370

The Output of extracted features that have been written into get_permission.txt files are found accordingly.

1. **get_permissionsW.txt** from AndroidManifestW.xml
2. **get_permissionsF.txt from** AndroidManifestF.xml
3. **get_permissionsB.txt from** AndroidManifestB.xml
4. **get_permissionsO.txt from** AndroidManifestO.xml

The features extracted from over 350 apks have been curated into the dataset given in the Extracted feautres.zip folder, on which the classification will be applied in the second phase.