Feature Enhancement of Amandroid, an Open Source Static Analyzer for the Security Vetting of Android Applications.

Author:Shiva Bhusal, Supervisor: Dr. Sankardas Roy

Department of Computer Science

The Android Ecosystem

- Android is the most popular mobile operating system in the world.
- Android has caught up to windows in terms of the internet use. (Businessinsider, 2017)
- The Android Ecosystem consists of three major players: the users, the publishers and the app Store agency.
- Android Ecosystem is not void of malicious apps.

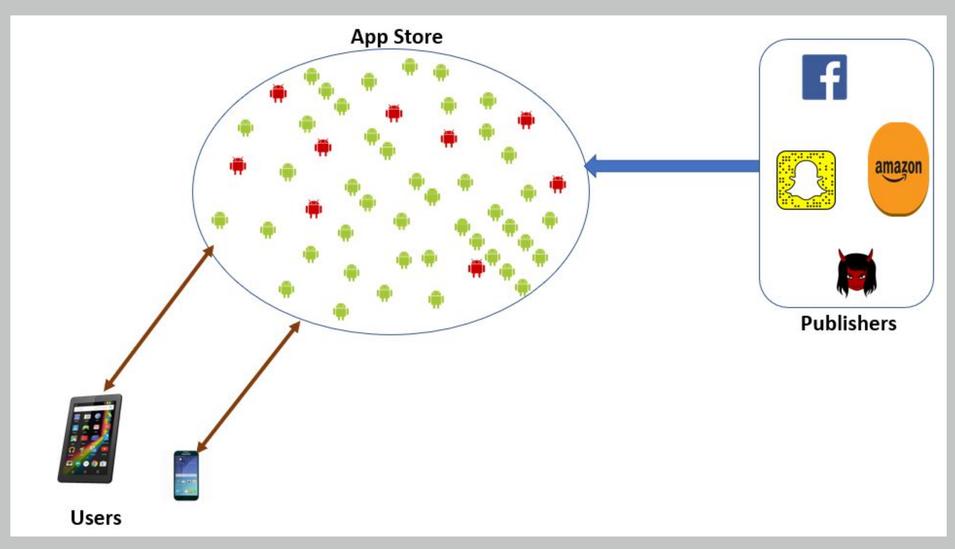

Figure 1:Android Ecosystem

The Art Of Static Analysis

- Static Analysis is the way of analyzing an application without actually running it.
- Static analysis is useful in Android Ecosystem to detect malicious apps before users install in their machine or even before they are uploaded to the App store.
- Static analysis is not perfect.It is not void of false positives and false negatives.

Amandroid: A Tool Of The Trade

- Amandroid is an open source static analyzer used for the security vetting of Android Apps.
- The idea behind Amandroid is to perform data flow, control flow and the data dependence analysis for each of the components of an Android App.
- This helps detect possibilities of malicious activities such as:
  - Sensitive data leakage.
  - Data injection.
  - Misuse of sophisticated APIs.

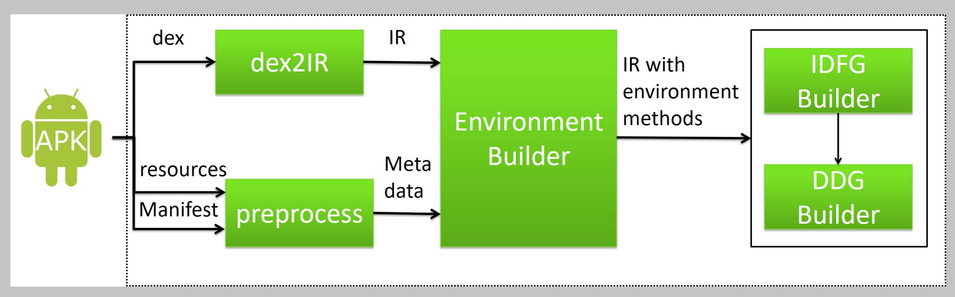

Figure 2:Amandroid block diagram(Wei Et. Al)

- Amandroid can be downloaded in the form of a jar file, and can be used to detect malicious activities in any APKs.

Problem Statement

- Attackers are implementing newer techniques to create malicious apps.
- Every static analysis tools have to be updated and enhanced to detect the newer forms of malicious activities.
- Currently, Amandroid doesn't have plugins to detect potential malicious activities such as:
  - Blackmailing users through the misuse of LockScreen.
  - Dynamic loading of Hexcode from network or the asset folder of the Android app. Etc.
- The motto of this work is how the features of Amandroid can be enhanced so that it detects newer forms of potential malicious activities.

Motivating example I : LockScreen

- **Malicious Symptoms:**
  - Once the app is installed, it covers the entire screen of the Android device.
  - The victim is not able to close the malicious App or access any other apps already installed in their device.
  - The app may consist of some blackmailing information.
- **The lockScreen detector plugin:**
  - Checks for the presence of signature that can be malicious.
  - Checks for a particular value of the parameter in the call statement which confirms the presence of LockScreen.

Motivating Example II: Dynamic loading

- **Malicious Symptoms:**
  - Loads Dalvik Bytecode from asset folder or network.
  - The hidden Dalvik Bytecode has malicious intent.
- **Why attackers use dynamic loading ?**
  - Because it's difficult for an static analysis application to analyze the source code dynamically loaded from another source.
- **The Dynamic loading detector plugin:**
  - Checks for the presence of signature that can be malicious.
  - Checks for the use of DexClassLoader class.

Testing and Verification

- Both the LockScreen and DynamicLoading detector plugins were tested using **ScalaTests.**
- Both the plugins passed 5 different test cases.

Experimental Results

-
-

Future Works

- Writing more plugins to detect newer tricks used by attackers.
- Using Machine Learning techniques to detect malicious and benign apps.
  - Result of each plugin becomes one feature for the classifier.

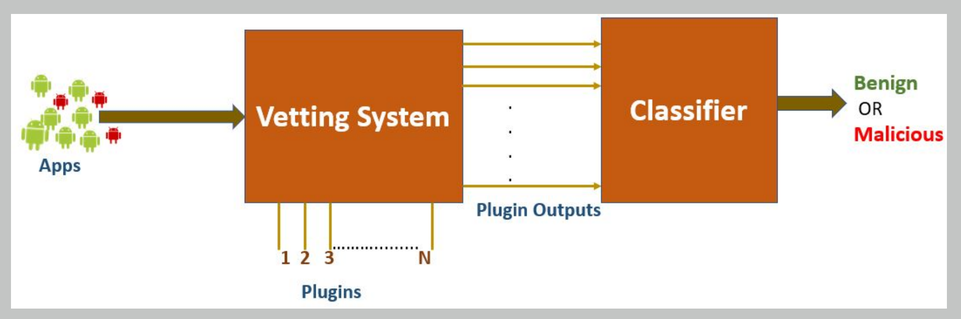

Figure 3:Machine Learning based Classifier

References

1. Amandroid: ACM CCS 2014, http://pag.arguslab.org/argus-saf

Acknowledgments

- Fengguo Wei, Dewan Chaulagain, Steven Arzt.