Static Analysis For Android Security: Building the Map of Android Inter-Application Communication
project title
University of Luxemburg
project place

Project description:

Android is the most widespread smartphone operating system in the world accounting 70% market share. More than 600 000 Android applications available on dozens of application markets can be installed by end users. On the official market of Google (AndroidMarket), more than 10 000 new applications are available every month. For the end user, downloading an application on his smartphone is similar to choosing an apple on an apple tree: he only sees the surface and has no evidence that there is no worm in it. Unfortunately, there are many worms of different kinds waiting to invade smartphones: malware leaking data, applications eating all the battery, adware calling premium-rate numbers, etc. In an open world as the Android world is, the end user receives and install an application file and can then only pray that the application is not harmful in any sense. Hopefully, it's not inevitable! Recent research works try to propose different kind of security analyses on Android applications.

Nevertheless, analyzing one Android application in isolation is not sufficient. Indeed, even if a permission-based architecture (as the one of Android) ensures that an application A can only access the resources for which A has the permission, the specificities of Android make communication between applications (and the components constituting an application) possible through elements called Intents. Consequently, on Android, several applications can collude to leak sensitive information. For instance, an application can get the user's location and send it to another application which then leaks the sensitive information to an untrusted third party.

The first expected outcome of AndroMap is a detailed map of Android application components and the links that exist between them. The map can be used to detect problems such as privacy leaks and click fraud, or to determine interesting properties such as long or otherwise interesting communication paths, or paths with loops, hinting at potential for a Morris-worm or denial-of- service attacks. A direct application is to use the map to warn the user when she is about to install an app that would yield suspicious links in the user's "device map".

The second expected outcome of AndroMap is to go beyond simply warning the user by providing a tool which allows the correction of the identified security flaws. The main research difficulty is to modify the code of an android application without altering the nominal functionalities of the application.