(Hey, You, Get Off of My Market: Detecting Malicious Apps in Official and Alternative Android Markets)

Due on May 11, 2015

Instructor: Evangelos Markatos
Smartphones are becoming increasingly popular, with over one hundred millions being sold in the first quarter of 2011, and over two hundred thousands applications in Google market. With such popularity, and since Android is an open project by its nature, malware authors are attracted as well. In 2010-2011 new android malware families emerged, such as Geimimi, ADRD, Pjapps, DroidDream, BaseBridge, Zsone and DroidDreamLight, that can compromise your device, steal your personal information and even send sms to premium numbers.

It is important to understand the overall health of Android Markets in terms of malware applications. The authors approach is to crawl different markets, and perform a detection analysis in the applications downloaded. The opt for 3 design goals.

(a) Accuracy: In order to achieve a low false/positive rate.

(b) Efficiency: In order to process apps in a reasonable amount of time and thus a deep scanning technique such as an antivirus is not an option.

(c) Scalability: In order to handle even more applications.

They proposed DroidRanger, a malware detection system. DroidRanger crawled two hundred thousand apks and uses two methods for analysis. The first method detects known malware with a permission based footprinting. The second method can detect unknown malware based on heuristics filtering. Overall they found one hundred and seventy one applications with 2 of them being a zero day malware.

**Method1: Detecting known malware.** Detecting known malware is a two step process. Step one, uses a permission based filtering that matches apps permissions with permissions requested by known malware. For this step a set of 10 known malware families was used. If an application needs those malware-friendly permissions then it is kept for further examination. This is indented to exclude unrelated apps and reduce the total amount of apk's that need to be processed. Step two matches apps to malware footprints in multiple dimensions. By gathering information from the manifest file, checking the semantics in the byte code and by observing the structural layout of the code. An example of a footprint may check for a receiver that listens to the SMS service and calls abort broadcast to hide messages from the user. It can also check if an app sends or receives messages from premium numbers. With this footprint DroidRanger found 9 instances of Zsone malware, in the official Android market.

**Method2: Detecting Unknown malware.** Method two again uses a two step process. Step one, uses a heuristic based filtering. The first heuristic, selects apps with dynamic loading of java code, such as loading classes from external sources. This is used mostly in advertisements, to fetch code from remote servers. On the other hand, malware tries to execute code from external files, such as apk or jar files. The second heuristic, selects apps with native code loading. Usually android apps are written in java. Android also supports, for performance and compatibility issues, the use of native code. Native code by default is placed in a special directory. By examining known malware applications they observed that applications with malicious activities hide their native code in other directories than the default. Using method 2 to identify malware applications, they discovered zero-day Plankton spyware and zero-day DroidKungFu malware. Step two, involves dynamic execution of the application uncovered in step one. For example, during a call to SmsManager.sendTextMessage, the analysis can get the destination phone number and content. Also, they collected different system calls from known Android root exploits and compared them with the system calls being made during the dynamic execution of the apps.

As for the evaluation, DroidRanger took 4 and a half hours to process all the apps. It detected 2 zero day malware using heuristics and 40 different samples using behavioral footprints. Overall, it detected 211 malicious apps and they highlight the need for better policing, both in official and alternatives markets.