Monet: A User-oriented Behavior-based Malware Variants Detection System for Android

Abstract:

Android, the most popular mobile OS, has around 78 % of the mobile market share. Due to its popularity, it attracts many malware attacks. In fact, people have discovered around one million new malware samples per quarter, and it was reported that over 98 % of these new malware samples are in fact "derivatives" (or variants) from existing malware families. In this paper, we first show that runtime behaviors of malware's core functionalities are in fact similar within a malware family. Hence, we propose a framework to combine "runtime behavior" with "static structures" to detect malware variants. We present the design and implementation of MONET, which has a client and a backend server module. The client module is a lightweight, in device app for behavior monitoring and signature generation, and we realize this using two novel interception techniques. The backend server is responsible for large scale malware detection. We collect 3723 malware samples and top 500 benign apps to carry out extensive experiments of detecting malware variants and defending against malware transformation. Our experiments show that MONET can achieve around 99 % accuracy in detecting malware variants. Furthermore, it can defend against 10 different obfuscation and transformation techniques, while only incurs around 7 % performance overhead and about 3 % battery overhead. More importantly, MONET will automatically alert users with intrusion details so to prevent further malicious behaviors.

Introduction:

ANDROID is a mobile operating system from Google and it powered mobile devices dominate around 78.7 % of the smartphone OS market in the first quarter of 2016. Android applications (apps for short) can be downloaded not only from the Google's official market Google Play, but also from third-party markets. Although Google Play scans any uploaded apps to reduce malware, other markets/sites usually do not have sufficient malware screening, and they become main

hotbeds for spreading Android malware. As a result, Android attracts millions of malware. It is reported that 97 % of mobile malware is on the Android platform. Broadly speaking, there are two types of in-device malware detection systems. The first one is to perform static malware detection. This type of systems uses static information such as API calling information and control flow graphs to generate signatures for detection. For example, anti-virus, engines will scan files in apps after their installation. However, studies have shown that these types of anti-virus engines can be easily bypassed using transformation attacks (i.e., code obfuscation techniques like package name substitution and reflection technique). Furthermore, sophisticated signature generation and signature matching techniques based on control flow analysis incur considerable computation overhead, and consume energy on mobile devices which have limited battery resource, preventing them from being adopted as in-device detection systems