Preventive Measures for BadNews Malware

Adrien Guerard
Danny Park

October 2013

1 Abstract

Android security has become a growing concern in the industry as the number of devices in circulation grows. Old school versions of Android malware might have included a small, stupid app that would request particular permissions and then send the data it was able to collect over the network to an evil server. A new form of malware was recently discovered by security experts called BadNews. Instead of having the app itself be malicious, BadNews was a malicious ad network, thus not detectable by Google prior to allowing distribution on the Google Play store (e.g. app marketplace). We are proposing a tool for helping to identify these sorts of malicious ad networks, as well as allowing users to see the information that is being gathered about them and sent to servers by these advertising services. We would also like to suggest why it is generally a bad idea to grant ad networks the same permissions as apps, in order to better protect users from unknowingly sharing their private information.

2 Motivation

Mobile devices are essential items in everyday life. Daily interactions with applications on our phone are assumingly safe and not malicious; however, this is not the case, and as smartphone usage continues to grow, so will the number and prevalence of malicious software. There are occasions when, unbeknownst to the user, attackers retrieve sensitive data from a users phone (phone number, IMEI, etc) and send them to a remote server for assumingly malicious use. Attackers can also push malicious content to devices that might prompt users to engage in unwanted activities. We believe that it is vital to preserve the integrity of devices in order to protect users and their sensitive information.

Most Android analysis tools are static, i.e. analyze the code as is to check for any unsafe execution patterns. However, with libraries that allow arbitrary HTML5 to be downloaded/new instructions to be sent over the network to the device, the potential for an app to be a vehicle for malware is harder to identify. One often needs to observe an app for an extended period of time before being able to observe its malicious capabilities. Ads are also an important source of revenue for developers, and so it is important that users are comfortable both with the app they are downloading, but also with any third-party libraries that that content creator has included (ads are most often displayed using an API provided to the developer by the ad network).

3 Background

The BadNews phony ad network is a relatively new form of attack on mobile devices, and so has yet to be researched heavily. BadNews works because it is included within an infected app (usually a small game or other dumb app), which asks for specific permissions that allows access to private or sensitive data. BadNews then functions like a normal ad network, except it promotes malicious software that is not carried on the Google Play store. It can also spoof fake updates to certain apps, by sending a notification message along with a .apk for the device to install. Users that install the updates are actually installing some Android malware or spyware.

Certain systems like TaintDroid **?** have sought to make more apparent the potential use of sensitive information by tainting the data and creating rules for propagating tainted markers. We are not nearly as concerned with apps having access to certain private data as we are with fake ad networks transmitting that data.

4 Our Idea

Our basic idea is to understand how ad networks adversely interact with the Android OS (i.e. understand how BadNews works from an implementation level), and then develop some sort of monitoring tool that would help a security researcher determine if a particular application is periodically sending private or sensitive information to an unknown, third-party server. The tool could either be an Android application, or some other software that can monitor the phones network traffic, and organize it by whatever application or package is making the request. Our tool would categorize certain requests as safe or malicious based on the frequency, contents, and URL of the request. Although unable to work on iOS, our model would supposedly be able to extend beyond Android and be applicable to other smartphone ecosystems for analyzing the security and safety of their ad networks.

We realize that this is an ambitious endeavour, and in the event that we fall short of our goal, we hope to at least gain a detailed understanding of BadNews implementation.

5 Milestones

Milestone 1: Analyze the BadNews codebase (infected apps are available online). Understand exactly how ad networks are implemented in Android, and what security exploits BadNews takes advantage of.

We would also start using tools like Fiddler to be able to determine how we would monitor and filter network traffic (HTTP/HTTPS) to serve our needs. This would involve playing around with different ways of organizing and classifying the traffic. We would be experimenting with both legitimate and malicious apps.

Milestone 2: By M2 we would have decided on our general approach for monitoring the network traffic. We would begin coming up with algorithms or general rules for determining whether a particular pattern of network access by an application is malicious or not, and how best to generate a profile of a particular application or device running multiple applications.

We would also begin tweaking BadNews for testing, so that we could push the limits of what is acceptable by the Google Play store, while still compromising users private or sensitive information.

Milestone 3: By M3 we would have finished our own version of BadNews as well as how we categorize and collect the network traffic. Development in M3 would focus testing our framework on lots of apps in order to see if we can identify potentially malicious network traffic automatically (i.e. by analyzing the history and contents of HTTP/HTTPS requests). We would continue to tweak our algorithms and classification heuristics to reflect new information.