On Relevance of String Analysis in Android

Aravind Machiry

January 14, 2016

1 Project Overview

I want to explore the idea of performing string analysis on Android application. I plan to divide my project into following 3 sub-goals.

1.1 Finding Composition Vulnerabilities

Many android applications are composed by native component. Here, application calls into native code to perform some task. These applications could provide user controlled strings to native code, It is well known observation that handling strings in native code is tricky and could lead to vulnerabilities. We try to find these type of vulnerabilities.

Specifically, First, we find all invokes that call into native code. Second, check if arguments to these functions could be strings controllable by user (This is where we use string analysis). These provide potential vulnerable points. Finally, (If time permits), we check native code to see if these strings are used unsafely.

1.2 Usage of reflection in Android Apps

Most of the android vulnerability detection papers claim reflection is major problem. I intend to verify this claim by analyzing large corpus of applications (both benign and malicious).

Specifically, following are the questions for which I intend to answer:

- Do apps (benign/malware) use reflection? if yes, How often they use? What percentage of invokes are resolved thru reflection?
- How reflection is distributed against various(45) category of applications?
- Can these reflection be resolved using existing string analysis tools? If yes, how effective are they?

1.3 Intercomponent Communication mapping

Every Android application is composed of several components. Each component has a name (string), and they interact with each other using these names. Inter-application communication can also be achieved by using these component names (prefixed by application name). It is useful to have the Intercomponent invocation graph of an app, this graph could be used to understand the app better.

Specifically, I use existing tools to construct the above said graph of a given application.

2 Plan

I will try using JSA on Dalvik bytecode. JSA has front-end and back-end (which works on normalized representation). I will try to implement front-end for Dalvik and see its effectiveness.