Harvesting Runtime Data in Obfuscated Android Applications

Eric Bodden

April 1, 2015

It is generally challenging to tell apart malware from benign applications: obfuscation and string encryption, used by malware as well as goodware, often render static analyses ineffective. In addition, malware frequently tricks dynamic analyses by detecting the execution environment emulated by the analysis tool and then refraining from malicious behavior.

In this work we thus present Harvester, a novel approach that combines a variation of program slicing with dynamic execution and show that it can be highly effective in the triage of current mobile malware families. For this malware, Harvester allows a fully automatic extraction of runtime values at any position of the app's bytecode. Using forced execution, Harvester reliably extracts target phone numbers and contents of SMS messages, decryption keys or concrete URLs that are called inside an Android application, even if the application is highly obfuscated, and even if the application uses anti-analysis techniques (emulator detection or delayed execution / "time bombs"), dynamic code loading and native method calls for string decryption. Harvester not only aids human malware analysts, but as we show also acts as an automatic deobfuscation tool that reverts the introduction of encrypted strings and reflective method calls.

Harvester is currently being integrated into a commercial product but we will release a research version as open source. Experiments on 13,502 current malware samples show that Harvester can extract many sensitive values from applications, usually in under one minute, and this fully automatically and without requiring the simulation of UI actions. Our results further show that Harvester's deobfuscation can increase the recall of existing static and dynamic analyses, for instance FlowDroid and TaintDroid.