AsDroid: Detecting Stealthy Behaviors in Android Applications by User Interface and Program Behavior Contradiction Critique Paper for CS5371 Soft Test for Mobile & Emb Sys

Jeremy Solmonson
School of Security Engineering
University of Colorado at Colorado Springs
Colorado Springs, CO 80922
Email: jsolmons@uccs.edu

***Abstract*—This is paper is a critique of ”AsDroid: Detecting Stealthy Behaviors in Android Applications by User Interface and Program Behavior Contradiction.” In accordance with homework 3 requirements the critique will be based on: suggestion for acceptance, summary of paper, evaluation of paper, positive points, negatives points, and potential future work.**

I. Suggestion for Acceptance

I would strongly recommend to accept this paper. The primary idea useful in the security realm and can be build upon for future applications.

II. Summary of Paper

Finding hidden malware within Android application remains challenging because anyone, including hackers, can publish an application that contains unintended features. The most common features are high priced phone calls, premium SMS text messages, and http request connections. These features are executing without the operators permission or knowledge. By associating the function calls behavior with the overall program intent, stealthy malware can be identified through weak associations (i.e. having a calculator make an http connection). Asdroid was developed to identify these weak associations (intent vs. behavior) and can assist in finding stealthy malware.

III. Evaluation

The authors contributions to finding hidden (stealthy) malware is extremely useful to the security community. Most malware is identified through signatures which are after the fact detections. The authors propose a different method by analyzing the functions intent and comparing the functions behavior to that intent. If a mismatch exists, the program is flagged as suspicious. With more work, this could be a third method for malware identification beyond signature and heuristic bases detection.

The datalog was useful to identify the logic behind intent propagation. Without the rules explicitly stated, the examples would have been almost impossible to follow. Perhaps creating some easier to follow examples would better illustrate the datalog purpose. Further, by showing the datalog rules, future work can be built on within other areas of research.

The AsDroid application had a 79.5% success rate on identifying the stealthy behavior within 182 applications. Considering this is an introductory tool to correlate behavior and intent, the results are noteworthy.

IV. Positive Points

+ Novel idea to match program intent vs. program behavior to find malicious programs

+ The datalog logic was useful to build for future work

+ AsDroid seems to perform as intended to meet the authors works

+ Good use of graphs

V. Negative Points

- Some examples were rather long and difficult to follow

VI. Potential Future Work

While no future work was directly given within the paper, expansion of this feature could continue. First, more intents could be integrated into the datalog and toolset to encompase a larger number of stealthy behaviors. This would broaden the application beyond SMS, calls, and http requests. Perhaps including pictures, contracts, data files and other information that is more sensitive to the user.

Additionally, anti-virus and intrusion detection system vendors would be interested in this technology. The primary types of malicious detection is signature based (after the fact) and heuristic based (comparing against previous behavior), with signature based being he primary detection method within current industry use. This paper could broaden the idea of malicious to an intent vs. behavior correlation.