Improving Mobile Application Security via Bridging User Expectations and Application Behaviors

Wei Yang
Dept. of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA
weiyang3@illinois.edu

1. INTRODUCTION

To keep malware out of mobile application(app) markets, existing techniques analyze the security aspects of app behaviors and summarize patterns of these security aspects to determine what apps do. However, malware and benign apps could present the same behaviors(e.g., sending SMS). The difference is that the behaviors of malware are unexpected while the behaviors of benign apps are expected by users. User expectations (reflected via user perception in combination with user judgment) should be incorporated into security analysis to determine whether app behaviors are within user expectations. This presentation presents our recent work on bridging the semantic gap between user perceptions of the app behaviors and the actual app behaviors.

Our first step is to incorporate user-perceivable information(e.g., app descriptions) into the traditional security analysis. We have applied natural language processing (NLP) to aid the risk assessment of mobile apps. Our WHYPER [1] approach takes an app's description from the app market as input (before installing an app, users may often read the description of the app to understand the features provided by the app). The goal of WHYPER is to compare an app's description to permissions requested by the app to automatically identify mismatches for the users or app reviewers to inspect.

Inferring user expectations from an app's description is just a starting point. Various textual information in app-development process can also be leveraged (e.g., our work [3] that automatically extracts access control rules out of requirements documents and our work [2] that automatically extracts method specifications out of API documents). The analysis results can be combined with different levels and complexities of program analysis to better enhance general user perceptions. Ultimately, we see NLP playing a big role as users are increasingly responsible for managing the security of many devices.

To further assist user perceptions of app behaviors, we have developed an approach of user-aware privacy control [4]. This approach allows users to perform inspection of the outgoing information at runtime to decide whether the functionality offered by a particular app is worth the cost of giving up sensitive information. We also identify the information flows whose output channels are not user-perceptible (referred to as escaping flows) and the information flows with the information tampered before the information is presented to users (referred to as tampering flows) for users to inspect. Our empirical study shows that users are more comfortable in using our approach when users are informed about these inconsistencies between user perceptions and app behaviors.

Our recent work Appcontext [5] takes a step further to automatically differentiate malicious and benign mobile app behavior though contexts of the behavior (i.e., the events and conditions that cause the security-sensitive behaviors to occur). Appcontext takes as input a specification of sensitive methods, computes activation events(i.e., external events that trigger the security-sensitive method calls) based on backward traversal in an extended call graph that includes edges for inter-component communication (via intent messages), and computes context factors(i.e., environmental attributes that decide whether the security-sensitive method calls will be invoked or not) by analyzing dependences of conditional statements (on the paths from activation to the sensitive method call) on different environmental attributes. Our empirical results indicate that the idea that the context of a security-sensitive behavior can help separate malicious app behaviors from benign behaviors is useful and can lead to an effective technique for identifying malicious uses of security-sensitive methods.

2. REFERENCES

[1] R. Pandita, X. Xiao, W. Yang, W. Enck, and T. Xie. WHYPER: Towards automating risk assessment of mobile applications. In *Proc. 22nd USENIX Security Symposium*, pages 527–542, 2013.

[2] R. Pandita, X. Xiao, H. Zhong, T. Xie, S. Oney, and A. Paradkar. Inferring method specifications from natural language API descriptions. In *Proc. 34th International Conference on Software Engineering*, pages 815–825, June 2012.

[3] X. Xiao, A. Paradkar, S. Thummalapenta, and T. Xie. Automated extraction of security policies from natural-language software documents. In *Proc. ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, pages 12:1–12:11, 2012.

[4] X. Xiao, N. Tillmann, M. Fahndrich, J. De Halleux, and M. Moskal. User-aware privacy control via extended static-information-flow analysis. In *Proc. of the 27th IEEE/ACM International Conference on Automated Software Engineering*, pages 80–89, 2012.

[5] W. Yang, X. Xiao, B. Andow, S. Li, T. Xie, and W. Enck. Appcontext: Differentiating malicious and benign mobile app behavior under contexts. In *Proc. 37th International Conference on Software Engineering*, May 2015.