Improving Mobile Application Security via Bridging User Expectations and Application Behaviors

Wei Yang[1] Xusheng Xiao[2] Rahul Pandita[2] William Enck[2] Tao Xie[1]
[1]Dept. of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA
[2]Dept. of Computer Science, North Carolina State University, Raleigh, NC, USA
[1]{weiyang3, taoxie}@illinois.edu, [2]{xxiao2,rpandit,whenck}@ncsu.edu

ABSTRACT

To keep malware out of mobile application markets, existing techniques analyze the security aspects of application behaviors and summarize patterns of these security aspects to determine what applications do. However, user expectations (reflected via user perception in combination with user judgment) are often not incorporated into such analysis to determine whether application behaviors are within user expectations. This poster presents our recent work on bridging the semantic gap between user perceptions of the application behaviors and the actual application behaviors.

Categories and Subject Descriptors

D.2.4 [**Software Engineering**]: Software/Program Verification—*Validation*; K.4.4 [**COMPUTERS AND SOCIETY**]: Electronic Commerce—*Security*

General Terms

Human Factors, Security, Verification

Keywords

Privacy Control, Natural Language Processing

1. INTRODUCTION

The popularity of mobile phone has been centered around smartphone applications. Mobile applications have played an important role in numerous application areas such as email, social networking, entertainment, and e-commerce [2]. The increasing popularity of mobile applications spurs the occurrences of security attacks and privacy leakage on the applications. To address these issues, existing research proposes various approaches for application analysis. These proposed approaches [3, 5, 7, 13, 14] use program analysis techniques to extract security aspects of an application's behaviors such as information flows, and attempt to summarize patterns of malicious behaviors from the patterns of these security aspects. These analyses identify useful information for security analysts to identify suspicious applications, but the information provided by the analyses is not sufficient to differentiate suspicious application behaviors from benign application behaviors. More specifically, these analyses fail to determine whether an application's behaviors are within user expectations.

Classifying an application as malicious, privacy infringing, or benign is non-trivial. Because previously described formal analysis tool [1, 3–5, 7, 10, 13, 14] (directed towards detecting malicious applications) do not make a distinction between user-expected application behaviors and the unexpected application behaviors, these previous analyses may potentially report all security/privacy-sensitive operations as malicious. To address such issue, we have applied natural language processing (NLP) to aid the risk assessment of mobile applications. Our WHYPER [8] approach takes an application's description from the application market as input (before installing an application, users may often read the description of the application to understand the features provided by the application). The goal of WHYPER is to compare an application's description to permissions requested by the application to automatically identify mismatches for the users or application reviewers to inspect.

Inferring user expectations from an application's description is just a starting point. Various textual information in application-development process can also be leveraged (e.g., our work [11] that automatically extracts access control rules out of requirements documents and our work [9] 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 application behaviors, we have developed an approach of user-aware privacy control [12]. This approach allows users to perform inspection of the outgoing information at runtime to decide whether the functionality offered by a particular application 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 application behaviors.

HotSoS '14 Raleigh, NC USA

2. AUTOMATING RISK ASSESSMENT

We have developed WHYPER [8], an approach to automatically identify sentences that describe the uses of security permissions in an application's description. Specifically, in this work, we leveraged MLP techniques by using domain-specific models inferred from API documents to distinguish such sentences from the other sentences. These domain-specific models describe various actions performed on the resources protected by permissions, representing common uses of permissions. Our evaluation results on about 600 application descriptions show great promise in using NLP techniques to bridge the semantic gap of user expectations to aid the risk assessment of mobile applications.

Potential Use Cases. WHYPER is an enabling technology for a number of use cases. For users who install applications from application markets, WHYPER could enhance user experience for installing applications by highlighting the sentences that correspond to a specific permission. For market providers who desire to force developers to disclose functionality to users, WHYPER could ensure that permission requests have justifications in the description. For security analysts, WHYPER could help triage markets [1] for dangerous and privacy-infringing applications. Finally, for security researchers, WHYPER could be used in concert with existing crowd-sourcing techniques [6] designed to assess user expectations of application functionality.

3. USER-AWARE PRIVACY CONTROL

To enhance user perceptions of application behaviors and improve the privacy control mechanism of mobile platforms, we developed an approach of user-aware privacy control [12]. The approach (1) notifies users of potential information leak via presenting information flows that show what private data type flows to what output channels, and (2) allows users to perform inspection of the outgoing information at runtime. However, some information may flow to output channels where users cannot perform runtime inspection such as a network socket (referred to as escaping flows), and may tamper with the information before the information is presented to users for inspection (referred to as tampering flows). To differentiate such information flows from other information flows where users can inspect untampered information, our approach provides tamper analysis that tracks whether information is tampered before the information flows to output channels, and identifies escaping flows and tampering flows for users to inspect.

Our approach makes users aware of an application's behaviors that may compromise the users' security and privacy, explaining how the application may use the users' private information. Such approach is a first step towards bridging the semantic gap between what the user expects an application to do and what it actually does. This approach focuses on identifying and analyzing information flows from an application, and enhances the user perception of the application's behaviors to determine whether the application's behaviors in using the permission are expected based on the functionality of the application.

4. REFERENCES

[1] S. Chakradeo, B. Reaves, P. Traynor, and W. Enck. MAST: Triage for market-scale mobile malware analysis. In *Proc. Sixth ACM Conference on Security and Privacy in Wireless and Mobile Network*, pages 13–24, 2013.

[2] CNN Money. Mobile apps overtake PC Internet usage in U.S., Feburary 2014.

[3] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. Sheth. TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones. In *Proc. 9th USENIX Conference on Operating Systems Design and Implementation*, pages 1–6, 2010.

[4] W. Enck, M. Ongtang, and P. McDaniel. On lightweight mobile phone application certification. In *Proc. 16th ACM Conference on Computer and Communications security*, pages 235–245, 2009.

[5] M. Grace, Y. Zhou, Z. Wang, and X. Jiang. Systematic detection of capability leaks in stock Android smartphones. In *Proc. 19th Annual Network and Distributed System Security Symposium*, 2012.

[6] J. Lin, S. Amini, J. I. Hong, N. Sadeh, J. Lindqvist, and J. Zhang. Expectation and purpose: Understanding users' mental models of mobile app privacy through crowdsourcing. In *Proc. ACM Conference on Ubiquitous Computing*, pages 501–510, 2012.

[7] L. Lu, Z. Li, Z. Wu, W. Lee, and G. Jiang. CHEX: Statically vetting Android apps for component hijacking vulnerabilities. In *Proc. 2012 ACM Conference on Computer and Communications Security*, pages 229–240, 2012.

[8] 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.

[9] 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.

[10] H. Peng, C. Gates, B. Sarma, N. Li, Y. Qi, R. Potharaju, C. Nita-Rotaru, and I. Molloy. Using probabilistic generative models for ranking risks of Android apps. In *Proc. 2012 ACM Conference on Computer and communications security*, pages 241–252, 2012.

[11] 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.

[12] 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.

[13] Y. Zhou and X. Jiang. Dissecting Android malware: Characterization and evolution. In *Proc. IEEE Symposium on Security and Privacy*, pages 95–109, 2012.

[14] Y. Zhou, Z. Wang, W. Zhou, and X. Jiang. Hey, you, get off of my market: Detecting malicious apps in official and alternative Android markets. In *Proc. 19th Annual Network and Distributed System Security Symposium*, pages 5–8, 2012.