Assuring Security Properties of Device APIs by Automated Formal Analysis

Position Paper

Graham Steel

June 1, 2010

1 Background

Web applications running on mobile devices have potential access to a wealth of sensitive information and functionality. The applications that excite users need to access this information and functionality in a way that doesn't breach important security properties. We have been investigating the general problem of allowing untrusted code to access sensitive resources in a secure way for some time. Specifically, we have been investigating *security APIs*, application program interfaces which are designed not just to allow access to functionality but also to preserve a policy. The idea is that such an API should be designed so that no matter what functions in the interface are called, and no matter what the parameters, certain 'good properties' continue to hold.

Our approach is to use formal security analysis techniques. We build an abstract logical model of the interface, and treat the untrusted code as the adversary. We formalise the security goals of the interface as logical properties of the model. We then use automated tools such as model checkers and theorem provers to determine whether an adversary could breach these properties. Our results have been significant: we have discovered new PIN leakage attacks on the APIs of HSMs used in the cash machine network [6], new key extraction attacks on smartcards and USB security devices [2], and used our analysis techniques to propose improvements to the interfaces that eliminate the vulnerabilities [3, 4].

Since 2007, there has been an annual workshop on the analysis of security APIs (ASA). Recent editions have included proposals for APIs to ensure privacy in social networks and to ensure privacy in remote health monitoring

systems [1, 5]. In this paper, we sketch how our formal analysis techniques could be applied to privacy-preserving APIs, and propose a programme of research we expect to have near-term impact in the area.

2 Formal Analysis of Device APIs

Our plan is to apply our analysis techniques to device APIs. From the W3C device APIs committee we need examples of interface functionality and security goals, formally or informally specified, to begin our work. We expect that giving formal definitions for properties related to privacy will be one of the main challenges of our project, but we have some good starting points from our previous work on privacy in eVoting and automotive protocols, and from our information leakage analysis for PIN processing APIs.

A major challenge for automated formal analysis is mutable state. When an interface relies on state to make security critical decisions (such as whether the current session is still valid, the device is in a particular configuration etc.) the logical model becomes far more combinatorially complex. To counter this we need to find good abstractions and prove their soundness. On the positive side, our experience with cryptographic key management interfaces has shown that once good abstractions have been found they can be applied to many similar interfaces.

Our aim is to deliver device API specifications with verified security properties. Note that this would be a guarantee of a specific property in an abstract model of an interface and attacker. There would still be plenty of room for security flaws at the implementation level of the interface, or by an attacker using operations out of scope of the model, for example by attacks with corrupt the integrity of the interface. However, our previous work with APIs has shown that a large number of flaws prevalent in real devices do fall within scope of this model, and the problem of e.g. safely sandboxing untrusted code to be sure it can only call a fixed API can be seen as an important orthogonal problem to the analysis of the security properties of the API.

3 Summary

We propose to investigate the formal analysis of security and privacy preserving device APIs for web applications in conjunction with W3C. We would need engagement at the level of requirements, examples, and suitable languages for specifying device APIs and their security goals.

References

[1] Jonathan Anderson, Joseph Bonneau, and Frank Stajano. Security APIs for online applications. In *Third International Workshop on Analysis of Security APIs (ASA-3)*, Long Island, NY, USA, 2009.

[2] M. Bortolozzo, M. Centenaro, R. Focardi, and G. Steel. How Smart is your Smartcard? Attacking and Fixing PKCS#11 Devices. Under review, 2010.

[3] Matteo Centenaro, Riccardo Focardi, Flamina L. Luccio, and Graham Steel. Type-based analysis of PIN processing APIs. In Michael Backes and Peng Ning, editors, *Proceedings of the 14th European Symposium on Research in Computer Security (ESORICS'09)*, volume 5789 of *Lecture Notes in Computer Science*, pages 53–68, Saint Malo, France, September 2009. Springer.

[4] Véronique Cortier and Graham Steel. A generic security API for symmetric key management on cryptographic devices. In Michael Backes and Peng Ning, editors, *Proceedings of the 14th European Symposium on Research in Computer Security (ESORICS'09)*, volume 5789 of *Lecture Notes in Computer Science*, pages 605–620, Saint Malo, France, September 2009. Springer.

[5] Carl A. Gunter. Using APIs to assure conformance to privacy regulations. In *First International Workshop on Analysis of Security APIs*, Venice, Italy, 2007.

[6] G. Steel. Formal analysis of PIN block attacks. *Theoretical Computer Science*, 367(1-2):257–270, November 2006. Special Issue on Automated Reasoning for Security Protocol Analysis.