Developing a Systematic Approach to Evaluate the Usability of Security APIs - Doctoral Consortium Submission

Chamila Wijayarathna
Supervisors - Dr. Nalin A. G. Arachchilage and Prof. Jill Slay
Australian Center for Cyber Security, School of Engineering and Information Technology
University of New South Wales - Canberra
c.diwelwattagamage@student.unsw.edu.au

1. Introduction

Modern software development is primarily driven by the use of Application Programming Interfaces (APIs). Rather than developing functionalities from scratch, programmers find and reuse APIs that provide the functionality they are looking to implement in their code (Wurster & van Oorschot, 2009) (Myers & Stylos, 2016). APIs allow programmers to embed various functionalities to applications they develop without requiring them to know the underlying implementation details of the functionality.

APIs that provide security related functionalities, such as encryption, decryption and hashing, are categorized as security APIs. Due to the complexity of security concepts, security functionalities are implemented by developers who are experts in computer and information security (Wurster & van Oorschot, 2009). Security APIs allow programmers who are not expert in security concepts to embed security functionalities into the applications they develop.

Even though APIs are important in software development process, often they are not very easy to learn and use in software development environment (Myers & Stylos, 2016). Less usability of APIs reduces efficiency of programmers where they have to spend significant time to learn the APIs. Furthermore, less usable APIs lead programmers to incorrectly use them, which causes unintended behaviors in resulting systems.

The situation is worse with less usable security APIs. When a programmer uses a security API incorrectly, that causes security vulnerabilities in the system s/he develops. In a study Fahl et al. (2012) carried out using 13500 popular free android apps, they found that 8% of the apps are vulnerable to attacks like man in the middle attack, due to improper use of the Secure Socket Layer (SSL)/Transport Layer Security (TLS) APIs (Fahl et al., 2012). The authors have identified that the cause for this is not only the carelessness of the programmers, but also the usability issues of the SSL/TLS APIs used by programmers for developing those apps. For example, some SSL APIs provide lower level of abstraction to programmers that makes it hard for them to understand, where they get tempt to use it in unintended manners.

On the other hand, programmers who use security APIs are not security experts in most cases. They are task oriented, which sometime negatively affects the security aspects of the application they develop with the use of security APIs(Wurster & van Oorschot, 2009). The software development style can affect security of the applications developed by programmers in many ways. Therefore, it is worth designing and developing security APIs with usability in mind so that non security experts can also utilize them within their applications.

If the usability of security APIs can be improved, those will be less prone to erroneous usage and therefore, will be less subjected to introduce security vulnerabilities to the applications (Myers & Stylos, 2016) (Wurster & van Oorschot, 2009). Currently there is no existing methodology to evaluate the usability of security APIs (Acar, Fahl, & Mazurek, 2016) (Myers & Stylos, 2016). Thus, my research will contribute to develop a systematic approach to evaluate the usability of security APIs. I am planning to identify solutions to following research questions in order to achieve this goal.

1. What are the usability aspects that need to be considered when evaluating the usability of security APIs?

2. What are the steps that need to be followed when evaluating the usability of security APIs?

2. Research Methodology

By doing a comprehensive literature survey, I identified five methodologies that are being used to evaluate the usability of general APIs, which are :

- Empirical evaluation
- Heuristic evaluation
- API walkthrough method
- API concept maps method
- Automated evaluation

By considering the strengths and weaknesses of these methodologies, I identified that empirical evaluation will be the most suitable methodology to start my experiments (Wijayarathna, Arachchilage, & Slay, n.d.). Furthermore, through the literature survey, I identified that using a generic cognitive dimensions questionnaire as proposed by Blackwell and Green (2000) is the most suitable methodology to collect feedback from participants who involve in the emperical evaluation process (Wijayarathna et al., n.d.). However, existing cognitive dimensions framework and the questionnaire for API ubsaility proposed by Clarke (2004) will not be sufficient to evaluate the usability of security APIs (Wijayarathna, Arachchilage, & Slay, 2017). Therefore, I proposed that this framework and questionnaire need to be improved to evaluate the usability of security APIs and I proposed an improved version of the cognitive dimensions framework and a questionnaire (Wijayarathna et al., 2017). Suggested improvements were derived by referring to literature on usability of security APIs (Green & Smith, 2016) (Gorski & Iacono, 2016).

First Experiment : I am in the process of conducting my first study to evaluate the proposed framework and the questionnaire. For this experiment, I selected four security APIs which cover different contexts and domains, and designed four programming tasks where participants will have to develop a program using one of these APIs . I recruited programmers to participate in this study where they did one of these tasks. While completing the task, they had to follow a think aloud study. Their computer screen and the think aloud output were recorded. Once they completed the task, they had to complete the cognitive dimensions questionnaire. Once I collect the data, I will analyze the screen recording and the think aloud results, and identify usability issues that each participant come up with. Then I will analyze the questionnaire answers and identify usability issues of the security API identified there separately. From the results, I am planning to identify answers to following questions.

- Is the cognitive dimensions framework proposed by Clarke (2004) sufficient to evaluate the usability of security APIs? If not, what are the aspects it does not cover?
- Is our proposed cognitive dimensions framework (Wijayarathna et al., 2017) sufficient to evaluate the usability of security APIs? If not, what are the alternations that need to be done for it?
- Are there any security API usability aspects that can not be evaluated using empirical usability evaluation? If yes, what are they?
- Is using a generic cognitive dimensions questionnaire (Blackwell & Green, 2000) effective in identifying usability issues in API usability evaluations?

I am submitting a work in progress paper based on this study for the PPIG workshop.

Second Experiment : I am planning to conduct the second study to identify methodologies that can be used to evaluate usability aspects that will be identified in the third question.

Third Experiment : From the results of first two experiments, I will propose a systematic approach to evaluate the usability of security APIs. Then I will conduct the third experiment to identify the applicability of proposed usability evaluation methodology to the agile software development process. This will be conducted by interviewing software quality assurance leads of software development firms that develop and deliver APIs and security APIs.

3. References

Acar, Y., Fahl, S., & Mazurek, M. L. (2016). You are not your developer, either: A research agenda for usable security and privacy research beyond end users. In *Ieee cybersecurity development conference (ieee secdev) 2016.*

Blackwell, A. F., & Green, T. R. (2000). A cognitive dimensions questionnaire optimised for users. In *Proceedings of the twelth annual meeting of the psychology of programming interest group* (pp. 137–152).

Clarke, S. (2004). Measuring api usability. *Doctor Dobbs Journal*, *29*(5), S1–S5.

Fahl, S., Harbach, M., Muders, T., Baumgärtner, L., Freisleben, B., & Smith, M. (2012). Why eve and mallory love android: An analysis of android ssl (in) security. In *Proceedings of the 2012 acm conference on computer and communications security* (pp. 50–61).

Gorski, P. L., & Iacono, L. L. (2016). Towards the usability evaluation of security apis. In *Proceedings of the tenth international symposium on human aspects of information security and assurance* (pp. 252–265).

Green, M., & Smith, M. (2016). Developers are not the enemy!: The need for usable security apis. *IEEE Security & Privacy*, *14*(5), 40–46.

Myers, B. A., & Stylos, J. (2016). Improving api usability. *Communications of the ACM*, *59*(6), 62–69.

Wijayarathna, C., Arachchilage, N. A. G., & Slay, J. (n.d.). An approach to evaluate the usability of security apis. *IEEE Security & Privacy (under review)*.

Wijayarathna, C., Arachchilage, N. A. G., & Slay, J. (2017). Cognitive dimensions to evaluate usability of security apis. In *19th international conference on human-computer interaction.*

Wurster, G., & van Oorschot, P. C. (2009). The developer is the enemy. In *Proceedings of the 2008 workshop on new security paradigms* (pp. 89–97).