Empirical Validation of Security Methods

Ph.D. Candidate: Katsiaryna Labunets
Advisor: Fabio Massacci

University of Trento, Italy
{surname}@disi.unitn.it

Abstract. Security requirements engineering is an important part of many software projects. Practitioners consider security requirements from the early stages of software development processes, but most of them do not use any formal method for security requirements engineering. According to a recent survey, only about 9% security practitioners implement formal process of elicitation and analysis of security requirements and risks.
However, a number of methods have been recently proposed in academia to support practitioners in collecting and analysing security requirements. Unfortunately, these methods are not widely adopted in practice because there is a lack of empirical evidence that they work. Only few papers in requirements engineering have a solid empirical evidence of efficiency of proposed solutions. So how can we know that security methods work in practice?
In this paper we propose to conduct a series of empirical studies to build a basis that *a*) will provide security practitioners with guidelines for selection of security requirements methods, and *b*) will help methods designer understand how to improve their methods.

1 Introduction

An increasing role of security in software development process is recognized by both industrial professionals [14] and academia members [10]. The security requirements and risk analysis plays a major role in delivery of secure software systems.

A variety of academic security methods like SREP [9] and CORAS [7], Secure Tropos [12] and SI* [6], LINDDUN [3] and misuse cases [15] have been proposed in the last years. However, these methods are not commonly used in industry. Only 9% of security practitioners implement formal process of elicitation and analysis of security requirements and risks [5]. The reason of this can be a lack of empirical study showing the effectiveness of these methods on real cases. In most of the papers in requirement engineering researchers propose a new methodology and shows that it works. This is acknowledged by a recent study of Condori-

Fernandez et al. [2] shows that only 13% of research works in Requirements Engineering relied on case studies[1].

So how can the practitioners decide which method is better for elicitation and analysis of security requirements and risks in their projects? This lack of empirical grounded knowledge on security methods effectiveness in real cases blocks a wide deployment of academic methods in industrial projects. Indeed, disregarding of validation activities is a drawback for both practitioners and methods designers. Practitioners do not know which methods to apply because designers of methods do not provide information about the effectiveness and usefulness of the methods in real cases. Methods designers do not know whether the methods are efficient in practice or not because there is no experience in practical application of the methods.

The main objective of our research is to investigate the effectiveness of security requirements and risk analysis methods and *the reasons* of their effectiveness through a series of empirical studies. The second objective is to build an empirical basis that *a*) can provide security practitioners with guidelines for selection of security requirements methods, and *b*) will help methods designers to understand how to improve their methods.

There is a number of empirical studies are dealing with requirements engineering. Morandini et al. [11] present qualitative study of requirements comprehension. They compare Tropos and Tropos4AS requirements methods. Opdahl et al. [13] carried out a pair of controlled experiments to compare two methods for security threats identification: misuse cases and attack trees. Asnar et al. [1] presents their experience in modeling and analysis of requirements in practice. They applied Secure Tropos method in air transport management system.

The expected contribution of our work to the field of Engineering Secure Software and Systems is practical guidelines for selecting security requirements and risk assessment methods.

The rest of the paper is organized as follows: Section 2 states the proposed research directions. The research methodology and research plan are presented in Section 3. The ongoing and future work are discussed in Section 4.

2 Research Objectives

There is a number of methods for elicitation and analysis of security requirements. Usually methods designers propose a method and claim that the method works. To show that the method works typically method designer apply the method to a real case. This kind of experimentation cannot be accepted as a solid evidence that the new method works in practice.

Without experimentation how do practitioners choose methods for elicitation of security requirements and risks? How do researchers understand ways to improve their method? These two questions lead to the third one: How to empirically validate the effectiveness of a new security method?

[1] Case study is an in-depth investigation of how and why a particular phenomenon happened in real-life conditions

In this work we propose to empirically validate the effectiveness of security requirements methods when they are applied by novices, i.e., users that have no prior knowledge of these methods. In particular, we wand to understand which security methods are effective and which are not, and what are the reasons behind.

The main outcome of our research is to build an empirical ground that *a*) will help security practitioners to select a security requirements method, and *b*) provides methods designers with ideas on how to improve their methods.

Thus, our research work aims to answer the following questions:

- **RQ1** *Do security methods work in practice when applied by novice users?*
- **RQ2** *Why do some methods work? Why others don't?*

We want to answer the above research questions by conducting a series of empirical studies using a mix-method methodology combining research approaches from qualitative research (e.g. grounded theory) and quantitative research (e.g. controlled experiments).

3 Research Methodology and Research Plan

The research can be organized into 3 main phases: *Design* (step 1), *Execution* (steps 2-3), and *Analysis* (step 4). The decomposition of research plan is shown in Figure 1.

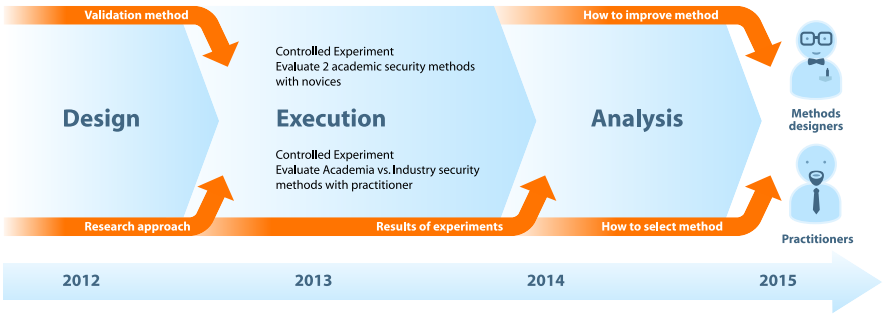

Fig. 1: Research plan.

3.1 Step 1: Selection of Research Methodology

The first step is to select a research methodology for our research. For this purpose we compare the research methodologies that are the most relevant to studies in the field of software engineering. We adopt the Easterbrook et al. [4]

taxonomy of research methods. This taxonomy considers five main classes of research methods: *a*) controlled experiments; *b*) case studies; *c*) survey research; *d*) ethnography; *e*) action research.

We propose to combine qualitative studies (grounded theory) with quantitative research (metrics measurements and statistics). We will apply quantitative methods to evaluate the effectiveness of academic methods for elicitation of security requirements (*RQ1*). Quantitative methods will allow us to collect wide knowledge base for following qualitative studies. For *RQ2* we plan to conduct qualitative studies to find out the underlying reasons of methods effectiveness (*RQ2*). The quantitative methods make it possible to advance hypotheses of security methods and check them experimentally.

We selected controlled experiment as a principal methodology for our research because it allows us to control necessary variables and collect reliable data for evaluation of the effectiveness of security requirements methods. However, we combine controlled experiments with quantitative (metrics measurements) and qualitative methods (focus group interview).

3.2 Step 2: Evaluate the Effectiveness of Academic Security Methods

At this step, in order to answer **RQ1**, we propose to evaluate the effectiveness of academic security requirements methods in case when they are applied by novices. A controlled experiment with master students will allow us to measure the effectiveness of security requirements methods in case when they are applied by users without prior knowledge of the methods.

3.3 Step 3: Academic security method vs. Industrial

The aim of this step it to understand which features make methods effective and working in practice. We propose to evaluate and compare academic versus industrial security methods in order to study which of the studied methods work better and why it happens. For this purpose we propose to conduct a controlled experiment with security practitioners and evaluate the effectiveness of one academic security requirements method (the best one by results of the previous experiment) versus the most used industrial method. The controlled experiment with practitioners and comparison of academic method with industrial one can give us an idea about those details and features that makes a method effective in real cases.

3.4 Step 4: Build Empirical Grounds

At this step we are going to aggregate and analyse the results of our experiments. Basing on these results we plan to develop empirical basis for selection of security methods. We believe this knowledge should help practitioners in understanding which of academic security methods are suitable for the purposes of their projects. At the same time the methods designers may find an idea about what makes their security methods applicable.

4 Ongoing and Future Work

This section presents the ongoing work and supplements the research plan with details of the future work.

The first experiment. Based on the results of eRISE challenges [8] conducted in 2011 and 2012 we selected 2 academic security methods, SREP and CORAS, to be evaluated and compared in our experiment. The last eRISE challenge showed that the SREP method was more appreciated by the participants than CORAS. Every participant that applied SREP method was able to finish all steps and identified a set of security requirements. In contrast, CORAS method showed worse results. The aim of the controlled experiment is to understand the reasons that make SREP better than CORAS. Quantitative research will help us to compare the effectiveness of the methods and find the reasons behind.

We are conducting controlled experiment with master students of Security Engineering course at University of Trento. The master students are divided into two groups. The first group of students uses SREP method while the second group applies CORAS method and vice-versa. During the experiment for each method-case combination we measure a number of metrics.

To evaluate the performance of the methods we take the following metrics:

- *Number of assets.* How many threats per asset does the participant identify for the case?
- *Number of threats.* How many threats does the participant identify for the case?
- *Number of security requirements.* How many security requirements does the participant identify for the case? [2]

To evaluate the perception of the users we take values similar to Opdahl et al. [13]:

- *Perceived usefulness.* How useful does the participant consider the method to be?
- *Perceived easy to use.* How easy to use does the participant consider the method to be?
- *Intention to use.* Does the participant intend to use the method again in the future?

In this experiment we propose to test more than 10 hypotheses that are dealing with correlation between performance and perception metrics, and the participants experience in application of security methods. Here are some examples of hypotheses to test:

[2] We use the notion of security requirements to denote both treatments in CORAS and security requirements in SREP because they both define a way to mitigate a threat.

- **H1** There will be a difference in the number of threats found with CORAS and with SREP within each case.
- **H2** The difference between the numbers of security requirements found with CORAS and with SREP will be correlated with the difference between the participants' preferences for CORAS and for SREP within each case
- **H3** The difference between the number of threats found with CORAS and with SREP will be correlated with participants' knowledge in security

We foreseen that the experiment will show *a*) how users without experience in application of security methods can comply with the method guidelines and *b*) how the first experience in application of one security method impacts on the work with other security methods (i.e., does the previous experience with security methods facilitate the application of new ones?)

eRISE 2013. In 2013 we will be involved in organization of eRISE 2013 challenge. This challenge aims to empirically evaluate security requirements and risk assessment methods. During eRISE we will study how participants apply the methods in practice, and which security methods are more effective and what features make them useful. The eRISE 2013 covers the objectives of our second experiment. We will conduct a controlled experiment with practitioners and compare the best *academic method* by the results of our first experiment and an *industrial method* that is the most used in practice. We expect to collect sufficient and reliable data to understand what the security methods need to become effective.

References

1. Y. Asnar, P. Giorgini, F. Massacci, A. Saidane, R. Bonato, V. Meduri, and V. Ricucci. Secure and dependable patterns in organizations: An empirical approach. In *Proc. of RE '07*, pages 287 –292, 2007.
2. N. Condori-Fernandez, M. Daneva, K. Sikkel, R. Wieringa, O. Dieste, and O. Pastor. A systematic mapping study on empirical evaluation of software requirements specifications techniques. In *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*, pages 502–505. IEEE Computer Society, 2009.
3. M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering*, 16(1):3–32, 2011.
4. S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian. Selecting empirical methods for software engineering research. In F. Shull, J. Singer, and D. Sjberg, editors, *Guide to Advanced Empirical Software Engineering*, pages 285–311. Springer London, 2008.
5. G. Elahi, E. Yu, T. Li, and L. Liu. Security requirements engineering in the wild: A survey of common practices. In *Computer Software and Applications Conference (COMPSAC), 2011 IEEE 35th Annual*, pages 314–319. IEEE, 2011.
6. P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone. Modeling security requirements through ownership, permission and delegation. In *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on*, pages 167–176. IEEE, 2005.

7. M. Lund, B. Solhaug, and K. Stølen. *Model-driven risk analysis: the CORAS approach*. Springer, 2010.
8. F. Massacci and F. Paci. How to select a security requirements method? A comparative study with students and practitioners. *Secure IT Systems*, pages 89–104, 2012.
9. D. Mellado, E. Fernández-Medina, and M. Piattini. Applying a security requirements engineering process. *Computer Security–ESORICS 2006*, pages 192–206, 2006.
10. D. Mellado and D. Rosado. An overview of current information systems security challenges and innovations J. UCS Special Issue. *Journal of Universal Computer Science*, 18(12):1598–1607, 2012.
11. M. Morandini, A. Marchetto, and A. Perini. Requirements comprehension: A controlled experiment on conceptual modeling methods. In *Empirical Requirements Engineering (EmpiRE), 2011 First International Workshop on*, pages 53–60. IEEE, 2011.
12. H. Mouratidis. Secure software systems engineering: the Secure Tropos approach. *Journal of Software*, 6(3):331–339, 2011.
13. A. L. Opdahl and G. Sindre. Experimental comparison of attack trees and misuse cases for security threat identification. *Inf. Softw. Technol.*, 51(5):916–932, 2009.
14. B. Schneier. The importance of security engineering. *Security Privacy, IEEE*, 10(5):88, Sept.-Oct. 2012.
15. G. Sindre and A. Opdahl. Eliciting security requirements with misuse cases. *Requirements Engineering*, 10(1):34–44, 2005.