Integration of Security Pattern Selection Practices with Pattern Storage

Jean-Michel Lehker
Department of Computer Science
The University of Texas at San Antonio
rpl599@my.utsa.edu

Rocky Slavin
Department of Computer Science
The University of Texas at San Antonio
koq441@my.utsa.edu

Jianwei Niu
Department of Computer Science
The University of Texas at San Antonio
niu@cs.utsa.edu

ABSTRACT
Security patterns represent reusable security practices that can be applied to a specific problem in order to generate a solution. The reuse of existing practices both decreases the time spent on solving a problem and improves the quality of the product by applying tried-and-tested solutions. In the wild, security patterns are documented in many forms from many sources. This increases the difficulty of locating the most appropriate pattern(s) for a situation. We address this issue by creating a security pattern repository using the structure of feature diagram hierarchies. Furthermore, we will conduct a study to better understand how software engineers search for patterns and apply the resulting information to our search engine.

General Terms
Management, Documentation, Experimentation, Security

Keywords
Security, Requirements, Patterns, Repository

1. INTRODUCTION

Based upon the idea of the *design pattern* [1], a *requirements pattern* is a structure that engineers can use to generate one or more requirements for a recurring situation [7]. Each requirements pattern describes a recurring problem as well as a general solution which can be used repeatedly, but not necessarily in the same way every time. The reuse of solutions provides a means for knowledge transfer as well as decreased time spent on generating a new solution. The use of such a mechanism is valuable for security practitioners in order to avoid the penetrate-and-patch [9] approach. *Security patterns* exist specifically to address security goals and risks for a system.

Security patterns of all types (e.g., requirements, design, architectural, procedural, etc) exist in a wide variety of disjoint sources [4]. The variety of sources combined with different documentation styles [2, 3, 6, 10, 11, 13] increases the challenge for engineers to select appropriate patterns [15]. From online databases to textbooks, one can scour a large number of sources before finding an appropriate pattern. Construction of a repository would directly address this issue. By collecting and organizing patterns in a centralized location, a security professional need not spend countless hours searching for a specific pattern (or set of patterns), but instead consult one organized source of patterns that would suggest related patterns as well as those which may be required in addition to selected patterns.

Our previous work on managing requirements patterns using *feature diagram hierarchies* [16] provides a framework for the organization of security requirements patterns. Using this framework, we will design and create a repository for the storage of security patterns. Expanding on the use of pattern hierarchies, we will conduct an empirical study on the trends in pattern selection to further the usability of the hierarchy. Information from this study will allow us to customize the repository's search feature to conform better to the habits of analysts.

2. PATTERN COLLECTION

Collection and storage of patterns is the first and most straight-forward step in construction of such a repository. In our previous work with pattern hierarchies, we collected a total of 188 security patterns from various textbooks and papers and added them to a preliminary database [8]. The approach for our collection began with the analysis of the existing state of the pattern landscape [5, 16] and branching our search through related works and references.

Our collection includes design, requirements, architectural, procedural, and implementation patterns. Due to the topic of our previous research and the framework we introduced for organizing security requirements patterns, our focus will be towards such patterns. We have classified 53 of the patterns in our database as requirements patterns. The inclusion of design, architectural, procedural, and implementation patterns will allow us to evaluate the possibility for relations between such types of patterns using our framework. Furthermore, the inclusion of security patterns of all types will benefit a broader user base.

3. UNDERSTANDING PATTERN SELECTION

We intend to create search engine which will assist analysts in using the repository. By analyzing the way security professionals from industry search for and select patterns for various scenarios, we believe we can collect valuable information regarding how analysts search for patterns. Such information could then be worked into the search engine to provide more useful results. We have begun designing a user study which will allow us to analyze security professionals and gather the data we need to specify inclusion and exclusion criteria.

3.1 Study Overview

This study will use a graphical user interface (implemented as a cross-platform web application) for our repository with which we will record information about search queries. The design of this study will include the presentation of scenarios describing common security problems to participants. As they use the interface, we will observe their process for searching and selecting patterns.

Our user study will investigate the following: context categorization, classification of problem domain, selected patterns, as well as any common keywords used by participants when searching for a pattern solution in our repository. Based on search queries and selected patterns, we will annotate subsets of patterns in the repository with these attributes. For example, if we observe participants classifying a scenario as "web application" and they subsequently select the "session" pattern, this pattern will then be annotated with "web application."

We plan to present the participants with enough scenarios to sufficiently annotate all the patterns in our subset. In order to not overwhelm our participants with a myriad of security patterns, and to simplify the study design, we have chosen to limit the subset to include only those patterns which we have categorized as "requirements" patterns. Due to the fact that an initial search function is necessary in conducting this study, we have annotated patterns in this subset with keywords to rank and filter search results. The keyword matching algorithm used will be very permissive and exclude few, if any, patterns from the search results. After a sufficient amount of preliminary data is collected, we will incorporate a stricter and more robust searching algorithm, which will use this data to further reduce search results. This may require multiple iterations in order to refine the search algorithm to an acceptable state.

3.2 Expectations

Once we have a complete set of annotated patterns, we expect to be able to implement a tool that should allow for repository users to narrow down relevant patterns more efficiently based on these annotations. Unlike the simple search function to be used in the study, this tool will provide more than just a search box where users can input keywords and phrases. Users will have the option to limit their search results to those patterns which contain specified annotations and/or attributes.

4. FUTURE WORK

The goal of this research is to provide usable and accessible patterns to software analysts. By understanding how such analysts search for and select their patterns, we can provide a means for the unification of the pattern landscape through a single pattern repository. The scope of research is currently limited to only those patterns which address security concerns and concepts. In future iterations of this repository, we hope to broaden the scope to include a wider variety of software and systems patterns.

Our work on feature diagram pattern hierarchies incorporates an Inquiry Cycle-based [12] approach for relating patterns. We plan to implement a questionnaire feature into the web interface of the repository so users would be able to utilize hierarchies for pattern selection by answering Inquiry Cycle questions.

5. ACKNOWLEDGMENTS

We extend our thanks to Travis Breaux and Hanan Hibshi for their contributions to our pattern research. This research was funded by Army Research Office (Award #W911NF-09-1-0273).

6. REFERENCES

[1] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel, A pattern language, Oxford University Press, 1977.

[2] D. Dietrich and J. M. Atlee, "A pattern for structuring the behavioral requirements of features of an embedded system," RePa'12, pp. 1-7, 2012.

[3] E. Gamma, R. Helm, R. Johnnson, and J. Vlissides, Design patterns: elements of reusable object-oriented software, Addison-Wesley, 1994.

[4] M. Hafiz, P. Adamczyk, R. E. Johnson, "Growing a pattern language (for security)," *Onward!'12* pp.129-158, 2012.

[5] T. Heyman, K. Yskout, R. Scandariato, and W. Joosen, "An analysis of the security patterns landscape," SESS '07. p. 3, 2007.

[6] M. Jackson, Problem frames; analyzing and structuring software develoment problems, Addison-Wesley, 2001.

[7] S. Konrad and B. H. Cheng, "Requirements patterns for embedded systems," RE'02, p. 127, 2002.

[8] J-M. Lehker. (2014). Security Pattern Repository [Online]. Available: http://sefm.cs.utsa.edu/repository/patterns/

[9] G. McGraw, "Testing for security during development: why we should scrap penetrate-and-patch," *IEEE T. Aero. Elec. Sys.* 13(4), 1998.

[10] J. Mylopoulos and J. Castro, "Tropos: a framework for requirements-driven software development," Information Systems Engineering: State of the Art and Research Themes, pp. 261-273, 2000.

[11] C. Palomares, C. Quer, X. Franch, C. Guerlain, and S. Renault, "A catalogue of non-technical requirement patterns," RePa'12, pp. 1-6, 2012.

[12] C. Potts, K. Takahashi, and A. I. Antón, "Inquiry-based requirements analysis," IEEE Softare, pp. 21-32, 1994

[13] M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann, and P. Sommerlad, Security patterns: integrating security and systems engineering, John Wiley & Sons, 2006.

[14] R. Slavin, J-M Lehker, J. Niu, and T. D. Breaux, "Managing security requirements patterns using feature diagram hierarchies," Tech. Rep. CS-TR-2014-002, Univ. Texas at San Antonio, 2014.

[15] M. Weiss and H. Mouratidis, "Selecting security patterns that fulfill security requirements," 16th ICSE'08, pp. 169-172, 2008.

[16] N. Yoshioka, H. Washizaki, and K. Maruyama, "A survey on security patterns," Progress in Informatics, No.5, pp. 35-47, 2008.