Usable Security by Design: A Pattern Approach

Bilal Naqvi[1,2(✉)] and Jari Porras[1]

[1] LUT Software, LENS, LUT University, 53850 Lappeenranta, Finland
syed.naqvi@student.lut.fi
[2] Software Engineering, Mirpur University of Science and Technology, MUST, Mirpur, Pakistan

Abstract. Security and usability are often in conflict. There is a recognition that security cannot be achieved in real sense unless it incorporates the human factor (usability elements). Despite this recognition, the state of the art identifies many challenges and reasons for conflicts between security and usability. This paper discusses some of these challenges while proposing the use of design patterns to handle those challenges. While justifying the use of patterns as one of the effective ways of handling the problem (conflicts), the paper presents a proposal for participatory usable security design patterns workshop. The workshop provides a forum for discussing a variety of issues concerning the usability and security conflicts while documenting the instances of conflicts and suitable tradeoffs as design patterns for use by other designers and developers. A catalog of usable security design patterns can assist the system designers and developers by positively influencing their decision-making abilities when it comes to conflicts.

Keywords: Patterns · Security · Usability · Usable security

1 Introduction

Security and usability are essential quality characteristics in today's software systems. To address the quality demands, security and usability are considered in specialized teams where the focus of each team is specific, the security team focuses on making the system security as robust as possible against internal and external attacks, however, usability is a minor concern for them. Whereas the usability team focuses on improving usability issues arising with the use of the system while providing a positive user experience (UX). With this specific focus, the need for usable security is realized when the instances of conflicts between security and usability are identified. A classic example in this regard is the password for authentication. The security dimension suggests that the passwords should be sufficiently long, frequently changed, have different cases and special characters, etc. However, from the user's (usability) point of view, such passwords are hard to memorize. If the suggested security guidelines are implemented, they have an adverse impact on the usability of the system, and if not implemented the system security is at stake.

Recently, there has been a realization that security cannot be implemented effectively unless we pay attention to the usability aspects [1]. US National Institute of Standards and Technology (NIST) report NIST Special Publication 800-63B states "evaluating the

A. Moallem (Ed.): HCII 2020, LNCS 12210, pp. 609–618, 2020.
https://doi.org/10.1007/978-3-030-50309-3_41

usability of authentication is critical, as poor usability often results in coping mechanisms and unintended workaround that can ultimately degrade the effectiveness of security controls" [4]. Initially, usable security was considered as limited to the usability of security interfaces, however, with time aspects like, (1) correspondence between systems' internal procedures and user's thoughts, (2) incorporating user values into security design [2, 3], were identified as important aspect to be considered in development of simultaneously usable and secure systems. With correspondence between the system's internal procedures and human thoughts, it is meant that there should be compliance between user perceptions and the way security procedures are performed on the system. Such compliance could be achieved in two ways, (1) training the users, and, (2) designing the security systems while considering the human aspects, thereby decreasing the chances of human errors as the system works the same way as the user thinks it does.

Similarly, incorporating user values into security design can also contribute towards implementing security effectively. In the development of security systems, the goals are set by experts who are unaware that users might have different priorities and values concerning security [3]. Certain user value-based objectives associated with security include objectives such as minimize system interruptions, maximize information retrieval, maximize ease of use, enhance system-related communication, etc. [2]. Therefore, the elements of value-sensitive design (VSD) can improve users' engagement with security.

Despite the realization of aligning security and usability in the development of systems and services, the state of the art concerning usable security identifies many challenges. While considering all the challenges identified via literature review and conducting exploratory studies in the industry, this paper advocates the concept of '*usable security by design*'. The usable security by design concept is aimed at aligning security and usability right from the start of the system development lifecycle [5]. The concept is centered on the development of a catalog of usable security design patterns to assist the system designers and developers in dealing with the conflicts, thus delivering simultaneously secure and usable solutions. The fundamental question addressed in this paper is '*how do we develop a catalog of usable security patterns?'*. The paper presents a proposal for a participatory usable security design patterns workshop [6]. To conduct such a workshop, various templates to be used during the workshop are also presented.

The remainder of the paper is structured as follows. Section 2 presents the background. Section 3 presents the proposal for a participatory usable security design patterns workshop. Section 4 presents the related work and Sect. 5 concludes the paper.

2 Background

2.1 Challenges in the State of Art

The authors [8] state that "usable security assumes that when security functions are more usable, people are more likely to use them, leading to an improvement in overall security. Existing software design and engineering processes provide little guidance for leveraging this in the development of applications". Based on an analysis of existing literature and exploratory studies in the industry, the following are some of the challenges in aligning security and usability during the system development lifecycle.

- *Security and usability handled independently*: Security and usability are considered by different teams, where the focus of each team is specific i.e. the team working on security is focused on making the system secure; whereas the team focusing on usability and UX is focused on improving the human interaction with the system. There does not exist a mechanism where concerns from both teams can be integrated towards achieving the goal of simultaneously usable secure systems, therefore it is a tradeoff between security and usability.
- *Reliant on Skill of Developers*: Handling usable security in an organizational setting is reliant on the skill of developers [8]. Developers are either experts in security or usability. Despite this, there does not exist a mechanism (in practice) to assist developers in handling the issues where security and usability are in conflict.
- *Lack of emphasis during the early phases of development*: Security requirements are usually improperly specified, due to lack of emphasis on security during the early stages of development; the same holds for usable security [9]. The authors [10] argue that system security is usually considered in the production environment by employing protections like firewalls, IDS/IPS, AV servers, etc., which identifies the state of consideration on security during the system development phases, let alone its usability.
- *Existence of suitable technique for assessing adequacy*: Concerning the adequacy of security, techniques like vulnerability scan and penetration testing can be employed to check the robustness of security features, however, there is no such technique for evaluating the adequacy of usable security [16].
- *Constraint to a Constraint*: The requirement engineering community defines security as a constraint to the system's functional requirements [11]. The question is, if security is a constraint to the system's requirements, then usability of security could be a constraint to a constraint, which is one of the reasons that usable security requirements are neither specified nor addressed adequately.

The challenges discussed above often serve as contributing factors to the complexity of usable security problem. Furthermore, the standards concerning software quality in general and usability, security in particular, do not provide any guidance when these characteristics are in conflict. While considering all these aspects, this paper advocates the use of design patterns for handling security and usability conflicts.

2.2 Why Patterns?

A pattern expresses a relationship between three things, *context*, *problem,* and *solution*. Furthermore, the patterns have three dimensions: descriptive, normative, and communicative [6]. In its *descriptive* dimension, a pattern is an analytic form to describe problems, context and solutions. However, in the *normative* dimension, a pattern is a meta-design tool to identify key issues and propose a method for addressing them. It is a *communicative* tool to allow different communities to discuss and address issues [6].

Moreover, for multidisciplinary fields such as usable security, it is important to consider the concerns from both perspectives. Patterns can incorporate multiple concerns due to their descriptive nature and enable different communities to discuss design issues and solutions due to their communicative ability. Patterns' ability to evolve with time

Table 1. Challenges in the state of art with a description of how pattern addresses it

Challenges	Description	Involved patterns' dimension
Usability and security handled independently	Patterns allow concerns from both usability and security to incorporated before documenting a final solution	*Communicative*
Reliant on skill of developers	Information provided by the pattern including problem addressed, solution, context facilitates the developers in making reasonably accurate decisions in other similar contexts	*Descriptive*
Lack of emphasis during the early phases of development	Patterns can be incorporated right from the beginning of development life cycle and can be used by designers and developers as a meta-design tool for identification of key problems and solution for resolving them	*Descriptive/communicative*
Existence of suitable technique for accessing adequacy	Patterns ability to be improved with time helps in establishing adequacy of the solution presented by the pattern. Even when the patterns are disseminated, they are monitored and reviewed and proposal for amendments can be incorporated at any stage	*Descriptive/normative*
Constraint to a constraint	Security and usability are considered together thereby decreasing the chances of being considered as constraint to constraint or as after thoughts	*Normative*

also makes them suitable for problems like usable security. A pattern has different states, a *proto pattern* is a pattern which is newly documented after the first iteration, and it captures the basic elements of problem, context, and solution. However, after undergoing various refinement stages it is in *alpha-state,* ready to be released for use and testing by designers and developers.

Furthermore, patterns provide benefits like means of common vocabulary, shared documentation, improved communication among the different stakeholders during product development [5]. Patterns provide real solutions by explicitly mentioning the context and problem and summarizing the rationale for their effectiveness. Since the patterns

provide a generic "core" solution, its use can vary from one implementation to another [7]. All the challenges in the state of art identified earlier along with how the pattern approach helps in addressing them are presented in Table 1. The Table 1 also presents the dimension of the pattern involved in addressing a particular challenge.

As stated earlier, security and usability have evolved independently as different domains, therefore, expertise in both security and usability is hard to find in one person. Todays' industrial practices reflect that handling the security and usability conflicts is reliant on the skill of the developers [8]. The use of patterns provides a way of assisting developers at work by influencing their decision-making abilities when it comes to the conflicts between security and usability. Moreover, the patterns can be incorporated right from the start of the systems' development lifecycle, which helps in saving significant costs and effort associated with rework in contrast to the cases where security and usability are afterthoughts.

3 Proposal for Participatory Usable Security Design Patterns Workshop

Having discussed the problem and motivation for using design patterns, the question is how we can identify such patterns to be able to build a catalog of patterns for dissemination among common developers. One mechanism for creating such a catalog is a participatory usable security design patterns workshop. The activities during the workshop are to be performed in groups (3–5 participants each). Participants of the workshop are security and usability developers and designers. The key activities during the workshop are presented in Fig. 1, which include:

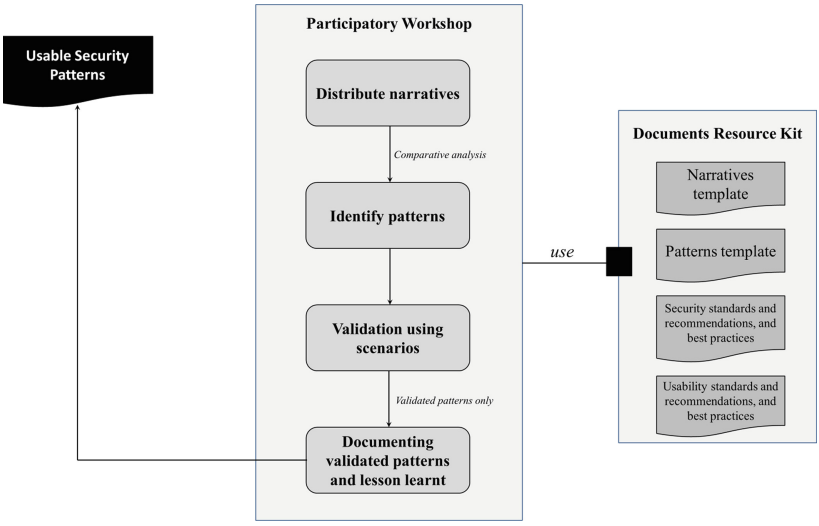
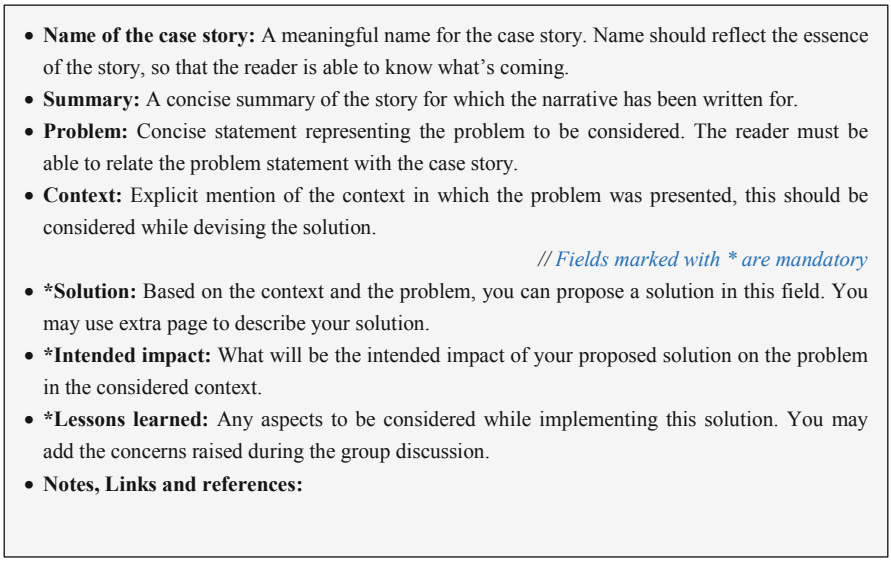


Fig. 1. Proposal for participatory usable security design patterns workshop

1. **Distribute narratives**: The narratives describing a usable security problem are distributed among the participants in groups. The narrative elicits a case story describing a usable security problem. The groups are tasked to design a solution of their own for the problem under consideration. The narrative template used during the first activity of the workshop is presented in Fig. 2.


- **Name of the case story:** A meaningful name for the case story. Name should reflect the essence of the story, so that the reader is able to know what's coming.
- **Summary:** A concise summary of the story for which the narrative has been written for.
- **Problem:** Concise statement representing the problem to be considered. The reader must be able to relate the problem statement with the case story.
- **Context:** Explicit mention of the context in which the problem was presented, this should be considered while devising the solution.

*// Fields marked with * are mandatory*

- ***Solution:** Based on the context and the problem, you can propose a solution in this field. You may use extra page to describe your solution.
- ***Intended impact:** What will be the intended impact of your proposed solution on the problem in the considered context.
- ***Lessons learned:** Any aspects to be considered while implementing this solution. You may add the concerns raised during the group discussion.
- **Notes, Links and references:**

Fig. 2. The narrative template to be used during the workshop

2. **Identify patterns**: The solutions from each group are subjected to comparative analysis in an attempt to identify instances of good design. The 'Rule of Three' also comes into play here. The rule of three requires at least three instances of similar implementations before a pattern could be identified and documented [6]. Once three instances of similar implementations for a problem are identified, the pattern is documented on a standard template.
3. **Validation using scenarios**: The participants are provided with a list of design patterns (already identified) and a problem scenario. The problem scenario being used during this stage involves a set of problems, and the task involves the selection of the patterns (from the list) that are applicable in the context being considered. The participants are tasked to document the description of a solution derived by applying a pattern in the considered context. If the right pattern is applied in the right context, it is validated; otherwise, it is subjected to a modification to ensure the use of the right patterns in the right scenarios.
4. **Documenting validated patterns and lessons learned**: In the end, the lessons learned and recommendations for future use of patterns are documented. The outcome of the activity is a catalog of validated usable security design patterns, which will be disseminated among the community of designers and developers to positively influence their decision-making abilities when it comes to conflicts.

An example of how a usable security pattern looks like is presented in Fig. 3. It is imperative to state that the pattern is documented on a standard template.

- **Title:** Toggle Password Visibility
- **Classification:** Authentication
- **Prologue:** To ensure secure authentication and users' privacy while preserving the usability element of feedback.
- **Problem statement**: Password for authentication is masked by default to protect against attacks such as shoulder surfing. This is done to preserve breach of privacy and authentication, but at the cost of 'feedback'. If the user makes an error while typing the long password s/he has to retype the entire password without just knowing and correcting the error.
- **Context of Use:** Whenever the password is masked to protect against shoulder surfing and other similar attacks.
- **Affected Sub Characteristics:** The sub characteristics of usability and security being affected/involved when this pattern is applied.
    - Usability: satisfaction, effectiveness in use, desirability
    - Security: privacy, confidentiality, authentication
- **Solution:** Provide the user with option to toggle password visibility by providing an icon or button. The button/icon should unmask the users' password. The password should remain unmasked until the button/icon is being clicked. The button/icon should be accessible with the mouse pointer.
- **Discussion:** This solution enhances the usability element of feedback while preserving users' privacy and security of the authentication process. The button/icon can be presented at the far end of password field or below it. This would help users in correcting the mistyped character in the password rather than retyping the entire password.
- **Type of service**: Desktop/ Web application requiring authentication with passwords.
- **Epilogue:** Increased user satisfaction, desirability of the service while providing the effectiveness in use.
- **Related Patterns:** To be added from the catalog

Fig. 3. Toggle password visibility pattern

The pattern presented in Fig. 3. addresses the conflict between authentication (security mechanism) and feedback (usability element) in cases where the user is confident that the password is not readable by the adversary. There are instances of this pattern on the authentication screens by major service providers, however, it is documented and intended for other designers and developers for consideration in newer versions of the system they develop. Moreover, other usable security patterns are available elsewhere [5, 7, 13, 15].

4 Related Work

The authors [7] presented a four staged framework for identification of conflicts and elicitation of suitable tradeoffs as patterns. In the first stage, the usable security problems are identified, which are modeled and quantified during the second stage. Standards and best practices on security and usability are accessed while developing suitable tradeoffs (solutions) to be documented as patterns. The documented patterns are applied to the software ecosystem.

Furthermore, the authors [15] presented a methodology for deriving usable security patterns during the requirements engineering stage of system development. The methodology is aimed at handling the conflict from the requirement engineering stage of system development. It does so by enumerating all security-related features. For all the enumerated features the security concerns are listed, and usability concerns arising from security features are identified. Once the concerns from both security and usability perspectives are known, the tradeoffs are explicitly elicited and then documented as patterns.

The authors [12], while listing 20 usable security patterns presented the results after analyzing applications such as Internet Explorer, Mozilla Firefox, and Microsoft Outlook. The authors state "patterns make sense and can be useful guide for software developers". However, the work was limited to listing the patterns and justifying their usage.

The authors [13] presented a list of patterns to align security and usability. They classified the patterns into two categories: data sanitization patterns and secure messaging patterns. Different patterns listed include, 'explicit user audit', 'complete delete', 'create keys when needed', among others.

The authors [14] proposed a set of user interface design patterns for designing information security feedback based on elements of user interface design. In addition, the authors created prototypes incorporating the user interface patterns in the security feedback to conduct a laboratory study. The results of the study showed that incorporating the elements of usability interface design patterns could help in making security feedbacks more meaningful and effective.

What distinguishes this work from others just discussed is that it provides a mechanism to involve a wider group of developers and designers during a workshop and identifying patterns based on their expertise. Though the work [7, 15] provides an avenue for identifying patterns, their scope and intended environment of application is during a project or in a team. However, the current proposal has been designed to hold good for participants from multiple projects and teams. We believe that the workshop proposal discussed in this paper can help attract a wider audience and identify usable security patterns.

5 Conclusion

There is a recognition that security and usability need to be handled together and integrated during the entire system development life cycle, rather than being considered as afterthoughts. With reference to the literature, we identified various challenges in the state of the art concerning usable security and proposed the use of design patterns as a way to handle the usable security challenge. While justifying the use of patterns in handling the usable security problem, we presented the proposal for a workshop for identifying and developing a catalog of usable security patterns. The catalog of patterns can help common security and usability designers and developers by influencing their decision-making abilities when it comes to security and usability conflicts in other but similar contexts.

Developing such catalogs requires a community-level effort and arranging various participatory workshops. We hope to gather the attention of the HCI community during the conference towards establishing a joint effort framework for arranging such workshops and collecting more usable security design patterns.

Moreover, the research advocates the shift in approach from 'user is the weakest link in security chain' to achieving, (1) correspondence between systems' internal procedures and human thoughts, and, (2) incorporating user values into security design. As an instance of the patterns' approach, a usable security pattern was also presented in the paper.

References

1. Garfinkel, S., Lipford, H.R.: Usable Security History, Themes and Challenges. Morgan and Claypool, San Rafael (2014)
2. Dhillon, G., Oliveira, T., Susarapu, S., Caldeira, M.: Deciding between information security and usability: developing value-based objectives. Comput. Hum. Behav. **61**, 656–666 (2016 2016
3. Dodier-Lazaro, S., Sasse, M.A., Abu-Salma, R., Becker, I.: From paternalistic to user-centered security: putting users first with value-sensitive design. In: CHI 2017 Workshop on Values in Computing, p. 7 (2017)
4. Grassi, P.A., Newton, E.M., Perlner, R.A., et al.: Digital identity guidelines: authentication and lifecycle management. Special Publication (NIST SP)-800-63B (2017). https://doi.org/10.6028/NIST.SP.800-63b
5. Naqvi, B., Porras, J., Oyedeji, S., Ullah, M.: Aligning security, usability, user experience: a pattern approach. In: IFIP Joint WG 13.2 & WG 13.5 International Workshop on Handling Security, Usability, User Experience and Reliability in User-Centered Development Processes held during International Conference on Human Computer Interaction (INTERACT) (2019)
6. Mor, Y., Winters, N., Warburton, S.: Participatory Patterns Workshops Resource Kit. Version 2.1 (2010). https://hal.archives-ouvertes.fr/hal-00593108/document
7. Naqvi, B., Seffah, A.: Interdependencies, conflicts and trade-offs between security and usability: why and how should we engineer them? In: 1st International Conference, HCI-CPT 2019 Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, pp. 314–324 (2019)
8. Caputo, D.D., Pfleeger, S.L., Sasse, M.A., Ammann, P., Offutt, J., Deng, L.: Barriers to usable security? Three organizational case studies. IEEE Secur. Priv. **14**(5), 22–32 (2016)
9. Riaz, M., Williams, L.: Security requirements patterns: understanding the science behind the art of pattern writing. In: Requirements Patterns (RePa 2012), pp. 29–34. IEEE (2012)
10. Parveen, N., Beg, R., Khan, M.H.: Integrating security and usability at requirement specification process. Int. J. Comput. Trends Technol. **10**(5), 236–240 (2014)
11. Xuan, X., Wang, Y., Li, S.: Privacy requirements patterns for mobile operating systems. In: IEEE 4th International Workshop on Requirements Patterns, pp. 39–42. IEEE (2014)
12. Ferreira, A., Rusu, C., Roncagliolo, S.: Usability and security patterns. In: Second International Conference on Advances in Computer-Human Interaction, pp. 301–305 (2009)
13. Cranor, L., Garfinkel, S.: Patterns for aligning security and usability. In: Symposium on Usable Privacy and Security (SOUPS), Poster (2005)
14. Munoz-Arega, J., et al.: A methodology for designing information security feedback based on user interact patterns. Adv. Eng. Softw. **40**(2009), 1231–1241 (2009)

15. Naqvi, B., Seffah, A.: A methodology for aligning usability and security in systems and services. In: 2018 Third International Conference on Information Systems Engineering, pp. 61–66 (2018)
16. Wang, Y., Rawal, B., Duan, Q., Zhang, P.: Usability and security go together: a case study on database. In: 2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM), pp. 49–54 (2017)