Towards Identification of Patterns Aligning Security and Usability

Bilal Naqvi(✉), Jari Porras, Shola Oyedeji, and Mehar Ullah

Software Engineering, LENS, LUT University, 53850 Lappeenranta, Finland
syed.naqvi@student.lut.fi

Abstract. Academic research and existing implementations of various systems and services identify instances of conflict between security and usability. Engineering the right trade-offs between security and usability is often not an easy task. Engineering of such trade-offs is mainly reliant on developers' skills, who are either experts in security or usability. This research aims to assist the developers in engineering the right trade-offs by proposing the use of patterns. Patterns provide benefits like means of common vocabulary, shared documentation, reuse, among others. The use of patterns can assist security and usability developers by influencing their decision-making abilities when dealing with conflicts in other but similar context of use. For the identification of such patterns, the paper presents a three-stage methodology. To instantiate the methodology, a case study was conducted whose results are also presented in the paper.

Keywords: Security · Usability · Usable security · Patterns

1 Introduction

Security and usability are considered as conflicting goals [1]. The trade-offs between the two are discussed at different forums not limited to cyber-security and Human Computer Interaction (HCI). Typical examples of the security and usability conflict include (1) complex password guidelines having an impact on memorability, (2) implementation of password masking to protect against 'shoulder surfing attacks' but at the cost of feedback (usability element), among others.

Traditionally security and usability have evolved independently and as different domains, therefore, expertise in both security and usability is hard to find in one person [2]. Despite this, the developers are ones who face most of the criticism when the security solutions are unusable, or when usability features pose a threat to systems' security. The domain considering the integration of principles of security and aspects of usability is known as *usable security*.

The early efforts in the field of usable security date back to 1998 when different properties of usability problems relevant to the development of security systems were identified [3]. Despite this recognition, state of the art concerning usable security still has some catching up to do. Practices and trends followed in the large organizations reveal a

© IFIP International Federation for Information Processing 2020
Published by Springer Nature Switzerland AG 2020
J. Abdelnour Nocera et al. (Eds.): INTERACT 2019, LNCS 11930, pp. 121–132, 2020.
https://doi.org/10.1007/978-3-030-46540-7_12

lack of motivation in considering usable security as a quality dimension [4]. One possible reason for this state is the cost associated with usable security [19]. The implementation of security due to the constantly evolving threat environment and usability due to rapid technological advancements has been so demanding that it leaves less time and costs to manage the trade-offs between the two. Among the other reasons for the current state of the art, it is imperative to discuss the following.

- *Different perceptions concerning security and usability*: The community has a different opinion concerning the existence of trade-offs between security and usability. Most of the research argues the existence of trade-offs between security and usability [5, 6]. However, in parallel with the research establishing the existence of the trade-offs, there is some research classifying security and usability trade-offs as mere myths [7, 8]. When the opinion on the existence of the problem is divided, then it is difficult to effectively contribute towards solving it.
- *Varying types of users:* In the community of users of the same device or application, opinions and requirements concerning security and privacy differ. Therefore, it is difficult to cater to the requirements of such a diverse category of users, which further complicates the task of finding common ground between security and usability and delivering a usable secure system.
- *Studying the conflicts by different communities in silos*: Various communities and interest groups have been studying usable security in silos, independently from each other. Some of these include, *(1)* SOUPS (Symposium on Usable Security and Privacy), small community studying trends, avenues and advancements in usable security. Much of the content is tactical, rather than being strategic, *(2)* The cybersecurity community dealing with the wider scope of security services; usability is a minor concern for this community, *(3)* The software engineering community where security and usability are considered as quality characteristics. Some of the standards provide contradictory perceptions and models for the same software quality characteristics, e.g. definition of usability in ISO 9126 and ISO 9241-11, *(4)* The HCI community, where the researchers try to explain from a cognitive perspective how users make poor security decisions leading to system compromises. There is no medium for collaboration that enables views from different communities and perspectives to be incorporated.
- *Ineffective joint working groups*: Because of independent activities, there is a lack of joint efforts concerning usable security. However, there exist multiple working groups specifically on usable security, but combining their findings to come up with a strategic vision for usable security, remains a challenge.
- *Lack of strategic approach*: Much of the work related to usable security suffers from a cosmetic approach that is the solutions are limited to specific problems, rather than contributing towards the management of the conflicts in general [2]. For example, there was a perception that CAPTCHA (*Completely Automated Public Turing Test to Tell Computers and Humans Apart*) poses *readability* problems for the users, therefore, new CAPTCHAS were developed that allow the user to select relevant images in response to the challenge. The question that remains valid for the community to address is, '*do we need CAPTCHAS?*'. The prime purpose of CAPTCHA is to protect against denial of service (DoS) attacks, which is the responsibility of the service provider, and then why the user should bear the burden to deal with the CAPTCHA

> especially when they cause deviation from the users' primary task. Likewise, the majority of the work on usable security has been on the operational and tactical level and therefore, has a cosmetic effect on the usable security problem. However, what is required in this regard are the long term and strategic solutions, for example, a requirement-engineering framework for aligning security and usability during the phases of the system development lifecycle (SDLC).

Moreover, one aspect on which there is a consensus among different groups working on usable security is to focus on learning and assisting the developers in handling the security and usability conflicts. This forms the primary research question addressed in this paper, which is '*how to assist security and usability developers in handling the conflicts and identifying suitable trade-offs while enabling learning in a specific context of use*?'. This research advocates the concept of '*usable security by design*', which is aimed at assisting the developers in handling the conflicts and identifying suitable trade-offs by using design patterns. Each design pattern solves a recurring design problem in a particular context of use. Using the patterns' approach can be advantageous not only for the developers but for the organizations as well. Software development organizations can also contribute to the catalog of patterns, based on previous experiences from the projects. Furthermore, using the patterns while ensuring effective management of the trade-offs does not affect the timely completion and costs associated with the project.

There are some existing usable security design patterns, but there is a need to collect those patterns, add them to a catalog and disseminate the catalog among the developers and designers. Furthermore, it is imperative to identify more patterns to be added to the catalog. For identifying more usable security patterns, the proposal for a three-stage methodology is presented in this paper. The remainder of the paper is organized as follows. Section 2 presents the background and literature review. Section 3 presents the proposed methodology for the identification of usable security patterns from existing implementations. Section 4 presents a case study to instantiate the proposed methodology. Section 5 presents the discussion and avenues for future investigation identified after the workshop, and Sect. 6 concludes the paper.

2 Background and Literature Review

In line with the research question addressed in this paper, the literature review was conducted considering the following objectives.

1. To rationalize the use of patterns as a way of assisting developers in handling interdisciplinary conflicts e.g. security and usability conflicts.
2. To identify existing usable security patterns (if any) and methodologies for identification for such patterns.

The authors [9] state, "insufficient communication with users produces a lack of user-centered design in security mechanisms". Both usability and security professionals recognize the importance of incorporating their concerns throughout the design cycle and acknowledge the need for an iterative rather than a linear design process. The use of patterns allows the concerns from both security and usability viewpoints to be incorporated right from the start of system development lifecycle. Patterns' ability to be improved over time and incorporate multiple viewpoints make them suitable for interdisciplinary fields like usable security [1]. Handling the security and usability concerns earlier in the development lifecycle helps in saving significant costs and delays associated with re-work.

An architect Christopher Alexander in the book 'A Pattern Language' originally introduced the concept of patterns [10]. Deriving inspiration from this, the same concept was implemented in computer science particularly in software engineering to assist the designers of the system, while providing guidelines and high-level principles. A similar concept was introduced in HCI to assist the development of user interface design (e.g. [11, 12]).

Each pattern expresses a relation between three things, *context*, *problem,* and *solution*. Patterns provide real solutions, not abstract principles, 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.

Furthermore, the patterns have three dimensions: descriptive, normative, and communicative [17]. From the perspective of usable security, the communicative dimensions of the patterns enable different communities to discuss design issues and solutions. Patterns also prove effective in the domains, which lack an existing body of knowledge; in such cases, the patterns assist in identifying effective practices as they emerge and capture them as objects for discussion, scrutiny, and modification [17].

In line with the second objective of the literature review, it was identified that the authors [13] while listing 20 usable security patterns also presented the results after analysis of commonly used software browsers like Internet Explorer, Mozilla Firefox and email clients like Microsoft Outlook. It was revealed that the identified patterns had a 61.67% application in the analyzed software implementations. 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 [14] 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 [15] proposed a set of user interface design patterns for designing information security feedback based on elements of user interface design. Furthermore, 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.

The authors [1] presented a methodology for deriving usable security patterns during the requirements engineering stage of system development. The methodology relies on handling the conflicts during the early stages of system development and documenting the suitable trade-offs in the form of design patterns for reuse. What distinguishes the methodology presented in this paper from the work [1] is that the methodology discussed in this paper focuses on identifying and documenting instances of good implementations by experienced developers in the form of design patterns. This is more of a bottom-up approach involving the identification of the patterns from existing implementations. However, the work [1] focuses on the creation of new patterns based on system requirements where possible trade-offs are identified and managed. The managed trade-offs are documented as patterns for implementation in the specific project and reuse by other developers.

3 Methodology for the Identification of Usable Security Patterns

In this section, the proposed three-stage methodology for the identification of usable security patterns is presented. As stated earlier, the methodology is based on identifying new patterns from existing implementations, which are setting good practices in the industry (*see* Fig. 1). This methodology provides uniform means to identify new patterns, and an opportunity for various stakeholders to contribute towards identification of the patterns and building the usable security patterns catalog. Particularly, from the industrial perspective, it can enable documenting new patterns from the implementations by experienced developers, thereby facilitating the learning and training of new developers.

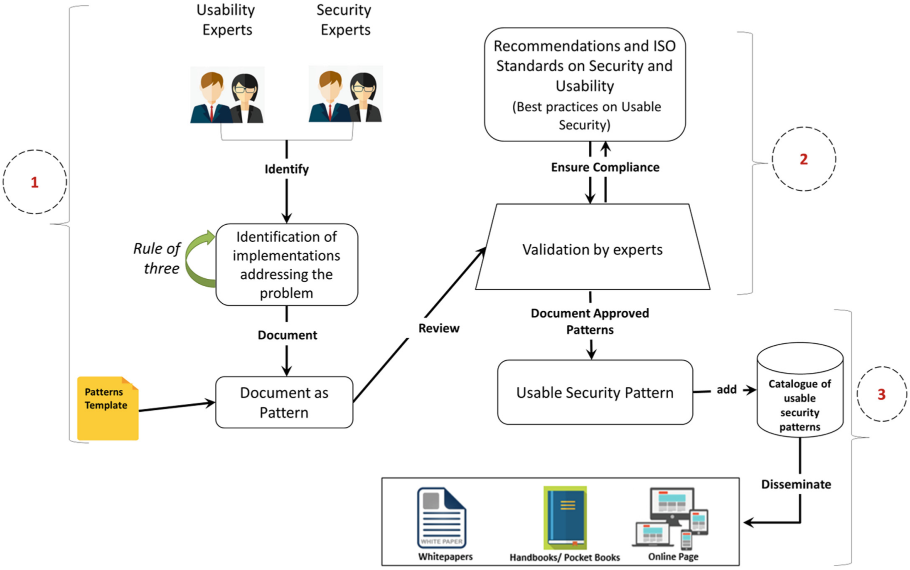


Fig. 1. The proposed methodology for identification of usable security patterns

- **Stage-1**: The first stage involves the selection of a common usable security problem. For the selection of a usable security problem, the experts can utilize one of the instruments such as surveys involving end-users, cognitive walkthroughs, heuristic evaluations, to mention a few. The next step is to identify existing implementations addressing the problem. Since the implementations can have different ways of approaching the problem, therefore, to document the best implementation as a pattern it is imperative to fulfill the '*Rule of Three*'. The rule of three requires at least three instances of similar implementations before a pattern could be identified and documented [17]. Once three instances of similar implementations for a particular problem are identified, the pattern is documented on a standard template. The details of usable security patterns' template are presented elsewhere [16]. Furthermore, selection of the best implementation is mainly based on the expertise of the professionals who are identifying it, however, to formalize the process it can also include evaluating the implementation with respect to a pre-defined set of heuristics. Defining such a set of heuristics for the evaluation would be considered as a part of the future work.

- **Stage 2**: The second stage involves a review of the newly documented pattern by one or more experts in the field. This stage involves activities like selection of expert(s) and gathering the reviews. Based on reviews the pattern is either accepted, which means it is ready to be finalized (*Stage 3*), or require modification, which means it goes back for modification to the experts who identified it during Stage 1, and in other cases, it may be rejected, which means it is discarded. The review by experts besides validation of the pattern has two advantages, (1) ensuring compliance with the underlying standards and best practices concerning security and usability, and (2) ensuring that the solution proposed in the pattern manages the trade-off effectively. The expert(s) review concerning each pattern is recorded on a checklist (see Table 1).

Table 1. Usable security pattern review checklist

Usable security pattern review checklist										
Description: For the pattern under consideration fill in the columns below. Accessing ISO standards on security and usability is highly recommended to ensure compliance										
Name of the pattern	Relevant to usable security		Effectively manages the trade-off			Compliance with the standards an best practices			Decision	Additional recommendations
/*Unique name of the pattern */	**Y**	**N**	**Y**	**N**	**Y/N**	**Y**	**N**	**Y/N**	□ Accept □ Modify □ Reject	Include recommendations for improvement of pattern, proposal for modification, compliance to the standard, reasons for rejection, etc.

- **Stage 3**: This stage comprises the following activities subject to the decision by the expert(s):
  - *Accept:* The accepted patterns are added to the catalog. The patterns in the catalog can be disseminated among the community of developers and designers. The ways

of disseminating the patterns include online pages, pocketbook for developers, and whitepapers.

- *Modify:* The documented pattern is referred back to the security and usability experts who identified it. The proposal for modification is considered and after necessary amendments, the pattern is subjected to review for the second time.
- *Reject:* The rejected patterns are discarded; however, the recommendations are considered for compliance in the other identified patterns with similar as well as the other context of use.

4 Instantiating the Methodology: A Case Study

To instantiate the methodology and identify a usable security pattern from existing implementations, a case study was conducted. The participants in the case study were the members of the software engineering laboratory at LUT University. Participation in the case study was voluntary. The objective behind the case study was to identify instances of good implementations by experienced developers, which set best practices in the field concerning the problem described below.

Case Description:
Mobile devices, particularly smartphones and tablets have become an inseparable companion for human users, as they have a wide range of features not just limited to communication. With such increased usage, we have seen an increase in cases of loss/theft of mobile devices, which ultimately leads to data breaches.

Consider a scenario when someone's smartphone is lost. Even if the lost smartphone it was locked, the victim would still be worried about ways in which an adversary could bypass the authentication mechanism and get access to the device. Access to the device could mean a breach of privacy and identity (if payment options were linked to the lost device). The authors [18] report a user study revealing that 50% of the respondents did not feel protected in case of loss/theft of their smartphone. Based on the scenario, the following problem statement was formulated.

Problem Statement:
In case of loss/theft of the users' device, the data on the device increases the impact of loss in the form of breach of privacy. The user needs to have trust and protection feelings to be able to use the device for personal/work purposes.

Stages of Case Study:

- **Stage 1**: This first stage involved the selection of the usable security problem. The results of a survey [18] led to the selection of the problem. While identifying the implementations addressing this problem, a solution 'remote data deletion' was identified. The next step involved the application of the 'rule of three'. Once three similar implementations addressing the problem were identified, the pattern (presented in Fig. 2) was documented on the standardized template. The solution offered by the pattern for the problem stated above is to "Offer the user with remote deletion functionality hosted by the mobile vendor or mobile service provider via a usable secure interface".

A secure service available online will work in this regard. It should offer the remote deletion by invoking the restore factory settings procedure, which would erase all the information from the device in case of loss/theft. This procedure not only ensures the security of data but also incorporates the human aspect of security, achieving human satisfaction and trust (elements of the global usability).

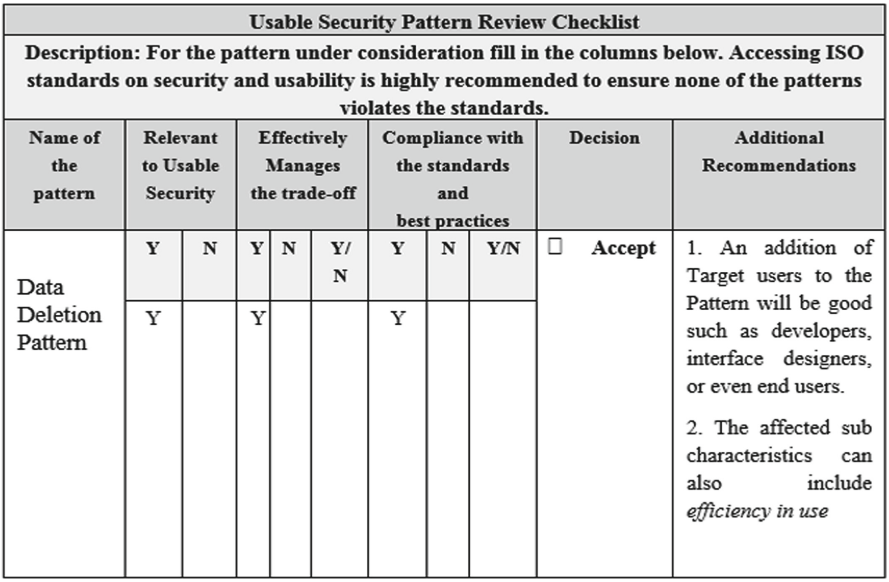
- **Title:** Data Deletion Pattern
- **Classification:** Data Protection, Device protection
- **Prologue:** To reduce the impact of loss in case of loss/theft of a device carrying sensitive personal/business information.
- **Problem statement**: In case of loss/theft of the users' device, the data on the device increases the impact of loss in the form of breach of privacy. The user needs to have trust and protection feelings to be able to use the device for personal/work purposes.
- **Context of Use:** Whenever there is loss/theft of device carrying user's data, which can lead to a breach of data.
- **Affected Sub Characteristics:** The sub- characteristics of usability and security being affected/involved when this pattern is applied.
    - Usability: satisfaction, trust, *efficiency in use*
    - Security: privacy, confidentiality, integrity
- **Solution:** Offer the User with remote deletion functionality hosted by the mobile vendor or mobile service provider via a usable secure interface.
- **Discussion:** Even if the lost smartphone was locked, the human user can still be bothered by breach of their privacy and the device's security. However, when the data has been removed from the device, the impact of loss can be minimized to an exclusively monetary loss.
- **Type of service**: Mobile devices or similar used in the same context.
- ***Target Users**: developers, designers*
- **Epilogue:** Improved data protection and reduced impact of loss.
- **Related Patterns:** Can be added later from the catalog.

Fig. 2. Data deletion pattern

Implementations of this pattern are available in the form of a "remote data deletion" functionality made available by smartphone manufacturers like Samsung and Apple for their users. Now the question arises who will use this pattern when this feature is already implemented? One scenario for the application of this pattern is in the case of other mobile devices including PDAs for inventory records, GPS, etc.

It is imperative to state that as per classification, the pattern (presented in Fig. 2) is a data/device protection usable security pattern. Other classifications of usable security patterns include usable authentication, usable security interface, among others. For example, a usable security interface pattern is presented in [16].

- **Stage 2**: This stage involved the validation of the pattern by the experts. It is pertinent to state that the pattern presented in Fig. 2 is a validated version of the pattern after reviewing by the experts. The items in *italic* were added based on experts' recommendations. The pattern review checklist from one of the experts is presented in Fig. 3.



Usable Security Pattern Review Checklist

Description: For the pattern under consideration fill in the columns below. Accessing ISO standards on security and usability is highly recommended to ensure none of the patterns violates the standards.

Name of the pattern	Relevant to Usable Security		Effectively Manages the trade-off			Compliance with the standards and best practices			Decision	Additional Recommendations
	Y	N	Y	N	Y/N	Y	N	Y/N	□ Accept	1. An addition of Target users to the Pattern will be good such as developers, interface designers, or even end users. 2. The affected sub characteristics can also include *efficiency in use*
Data Deletion Pattern	Y		Y			Y				

Fig. 3. Data deletion pattern review checklist

- **Stage 3:** Involved addition of this pattern to the catalog we are maintaining for dissemination and reuse by other developers.

5 Discussion

The presentation of the methodology during the workshop generated a discussion from which we identified the following avenues for future consideration.

- *Evaluation of instruments for identification of the usable security problem*: There is a need to evaluate different instruments that can help the security and usability experts in identifying the usable security problems with efficacy. For example, some of these instruments include:

  - *Surveys*, involving end-users' feedback and qualitative assessment of the problems faced by users while using a security system or service.
  - *Heuristic Evaluations*, which are conducted by experts to identify usable security problems due to violations of usability heuristics and security policies.
  - *Cognitive Walkthrough*, the security and usability experts inspect the user interfaces of security systems and services by going through a set of tasks and evaluating its understandability, ease of use and learning from the perspective of the targeted population.
  - *Contextual Inquiry,* that consists of observing services and systems in use within the context of participants' daily activities and asking for explanations as interesting events arise (security problems, usability problems, comments from users)
  - *Semi-structured interviews,* online or on-site with the users of security systems and services. The interviews would be focused on specific usability problems arising from security implementations.
  - *Use of tools*, within a lab, the users can be recruited to use security systems and services in a controlled environment. The human system interactions can be recorded using specialized tools like *Morae* or Observer *XT*.

- *Adding quantitative aspects to the methodology*: One dimension that needs further investigation is the addition of a quantitative method in the selection of the best implementations while documenting patterns. As stated in Sect. 3, the methodology considers only the qualitative aspects (expertise of professionals) in the selection of best implementations, therefore, considering the quantitative aspects will support the security and usability experts in selecting the best implementations for identifying and documenting new patterns. A quantitative methodology would also require a set of metrics to assist the identification of best implementations, for example, NUC (number of user complaints) is one such metric that can help in determining the best implementations from the user perspective. The lesser is the NUC, the better is the implementation from the user point of view. However, there is a need to identify a set of these metrics and incorporate their values by assigning weights to come up with a final valuation of the implementations quantitively. This valuation can be used by experts in the selection of the best implementations addressing the usable security problem under consideration.
- *Assessing the across system properties perspective*: Bouzekri *et al.*, presented their work on "Characterizing Sets of Systems: Across-Systems Properties and their Representation" during IFIP WG 13.2 & WG 13.5 Workshop at INTERACT 2019, an interesting aspect to consider from the perspective of our work is the effect of within systems and across system properties on the identified patterns. Considering the across system properties perspective, an important question to address is, do we need different patterns addressing the same usable security problem but requiring different solutions due to the nature of the context in which these systems are deployed?
- *Formalizing the process of selection of experts for review*: To have a set of experts for validation of the identified patterns, the work presented by Larusdottir and Kyas during the workshop identifies a mechanism that can be incorporated for selecting the right set of people for performing a validation job. The authors presented their work

related to the selection of an agile team for a developing development task. However, learning from their approach can be useful in formalizing the process of selection of experts.

6 Conclusion

Inter-dependencies and trade-offs between security and usability need to be approached strategically. The three-stage methodology presented in this paper is an attempt in this regard. Efforts need to be put in to develop a framework within the scope of the system development life cycle (SDLC) for eliciting the conflicts between security and usability while identifying suitable trade-offs between the two. The use of patterns can also be influential in documenting the outcomes of employing such frameworks. Patterns can assist also assist in improved communication between various segments working on the project more precisely the security and usability teams.

Additionally, the use of patterns does not only assist the developers within the organizational setting but also free-lancers in assessing the usability of their security options and vice versa. Furthermore, one pattern only solves one problem in a particular context of usage; therefore, an entire catalog of usable security patterns is required just like the user interface patterns catalog. The development of such a catalog is a time-consuming process and requires community-level efforts, therefore, we intend to present our proposal of using patterns and the methodology for identifying patterns to participants of the Human-Centered Software Engineering and HCI community for their feedback and participation in the development of the usable security patterns catalog.

Acknowledgment. The first author wishes to thank Professor Ahmed Seffah for his feedback during the initial phases of this research.

References

1. Naqvi, B., Seffah, A.: A methodology for aligning usability and security in systems and services. In: 2018 3rd International Conference on Information Systems Engineering, pp. 61–66 (2018)
2. Garfinkel, S., Lipford, H.R.: Usable Security History, Themes, and Challenges. Morgan and Claypool, New York (2014)
3. Whitten, A., Tygar, J.D.: Usability of security: a case study. School of Computing Science, Carnegie Mellon University. Rep. Technical Report CMU-CS-98-155 (1998)
4. Caputo, D.D., et al.: Barriers to usable security? Three organizational case studies. IEEE Secur. Priv. **14**(5), 22–32 (2016)
5. Garg, H., Choudhury, T., Kumar, P., Sabitha, S.: Comparison between significance of usability and security in HCI. In: 2017 3rd International Conference on Computational Intelligence Communication Technology (CICT), pp. 1–4 (2017)
6. Kulyk, O., Neumann, S., Budurushi, J., Volkamer, M.: Nothing comes for free: how much usability can you sacrifice for security? IEEE Secur. Priv. **15**(3), 24–29 (2017)
7. Sasse, M.A., Smith, M., Herley, C., Lipford, H., Vaniea, K.: Debunking security-usability tradeoff myths. IEEE Secur. Priv. **14**(5), 33–39 (2016)

8. Cranor, L.F., Buchler, N.: Better together: usability and security go hand in hand. IEEE Secur. Priv. **12**(6), 89–93 (2014)
9. Cranor, L., Garfinkel, S.: Security and Usability. O'Reilly Media, Inc., Sebastopol (2005)
10. Alexander, C., Ishikawa, S., Silverstein, M.: A Pattern Language. Oxford University Press, Oxford (1977)
11. Tidwell, J.: Designing Interfaces. O'Reilly Media, Inc., Sebastopol (2005)
12. Welie, M.V.: Patterns in interaction design (2008). https://www.welie.com/patterns/
13. Ferreira, A., Rusu, C., Roncagliolo, S.: Usability and security patterns. In: 2nd International Conference on Advances in Computer-Human Interaction, pp. 301–305 (2009)
14. Garfinkel, S., Miller, R.C.: Patterns for aligning security and usability. In: Symposium on Usable Privacy and Security (SOUPS) (2005). https://cups.cs.cmu.edu/soups/2005/2005posters/13-garfinkel.pdf
15. 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)
16. Naqvi, B., Seffah, A.: Interdependencies, conflicts and trade-offs between security and usability: why and how should we engineer them? In: Moallem, A. (ed.) HCII 2019. LNCS, vol. 11594, pp. 314–324. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22351-9_21
17. Mor, Y., Winters, N., Warburton, S.: Participatory patterns workshops resource kit. Version 2.1 (2010). https://hal.archives-ouvertes.fr/hal-00593108/document
18. Sophos: Security threat report (2010). https://www.sophos.com/sophos/docs/eng/papers/sophos-security-threat-report-jan-2010-wpna.pdf
19. Kirlappos, I., Sasse, M.A.: What usable security really means: trusting and engaging users. In: Tryfonas, T., Askoxylakis, I. (eds.) HAS 2014. LNCS, vol. 8533, pp. 69–78. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07620-1_7