IPHV8I2021X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .II
ISSN Online: 2319 – 9253
Print: 2319 – 9245

EFFIFCIENT DESIGN AND EVALUTION OF SECURITY ISSUES IN OSN'S APPLICATIONS

[#1]K. SHRUTHI, M.Tech Student,
[#2]D.THIRUPATHI, Assistant Professor,
Department Of CSE
SAHAJA INSTITUTE OF TECHNOLOGY & SCIENCES FOR WOMEN, KARIMNAGAR, TS, INDIA.

ABSTRACT: There is ongoing interest in utilizing user experiences associated with security and privacy to better inform system design and development. However, there are few studies demonstrating how, together, security and usability design techniques can help in the design of secure systems; such studies provide practical examples and lessons learned that practitioners and researchers can use to inform best practice, and underpin future research. This paper describes a three-year study where security and usability techniques were used in a research and development project to develop webinos — a secure, cross platform software environment for web applications. Because they value innovation over both security and usability, research and development projects are a particularly difficult context of study. We describe the difficulties faced in applying these security and usability techniques, the approaches taken to overcome them, and lessons that can be learned by others trying to build usability and security into software systems.

Keywords— *Security; Usability; Context of Use; Personas.*

I.INTRODUCTION

The security community has long recognized the need for incorporating a human dimension into its work. Surprisingly, however, while understanding how to design for user experience has been recognized as an important area for study, there has been comparatively little work examining the practical activities associated with building usable and secure systems. This observation was highlighted by Birge [2], who noted that there has been considerable research in Human-Computer Interaction and Security (HCISec) in recent years, but the bulk of this work has pertained to studying the usability of security controls and conceptual investigations about terms such as 'trust' and 'privacy', rather than activities associated with designing systems. In particular, much of this research focuses on the needs of end-users, rather than those of designers — few studies have attempted to tackle the question of how the latter should approach both usability and security as design concerns. At a NIST-hosted event on aligning software and usable security with engineering in 2011 [15], several recommendations for progressing the state-of-the-art were proposed, with one being that more case studies demonstrating the efficacy of best practice should be published. By showing how techniques were (or were not) applied successfully in contemporary design situations, practitioners could be better informed when selecting approaches for their own projects, while researchers could better understand the problems faced and develop research agendas for dealing with them. An interesting source of case studies comes from collaborative research and development (R&D) projects with teams drawn from both industry and academia. The importance of security has been recognized in recent years, partly due to a number of high-profile incidents reported in the media. Exploring the security and privacy implications of planned innovations is now a common pre-requisite for the funding of R&D projects. Unfortunately, designing a project for security and usability is easier said than done. In many conventional projects, security and usability are not considered primary goals, making them likely candidates for sacrifice in the rush to meet project deadlines. Unlike most conventional projects, however, R&D projects usually tackle risky, ill-defined problems with the aim of achieving the greatest possible impact. But impact can mean different things to different people: industrial partners measure success by developing technology which promises to be commercially successful, whereas academic partners might use the technology to explore research questions that surface during development. This apparent conflict can mean that, when facing impending deadlines, different partners have different priorities and areas of interest. Because this can further undermine quality concerns such as security, usability, or performance, creative solutions to design problems need to be explored to account for potential conflict. This paper describes the challenges faced designing security and usability into webinos: a software environment for running web applications securely across different device platforms. We begin in Section II by reviewing the issues faced in projects where the primary aims are to innovate, as well as the experiences of the HCISec community in contending with both security and usability during such projects. In Section III, we then present the webinos project and its objectives, the design approach taken to develop the webinos platform, and the part played by security and usability design techniques. We describe the difficulties

faced in applying these techniques in Section IV, together with the approaches taken to tackle them. Finally, in Section V, we reflect on lessons for those looking to align security and usability design in projects of this nature.

II. RELATED WORK

A. Innovation design in EU research projects

Designing for innovation might appear to be synonymous with general information system design, but several differences have been identified between the roles played by system architects and entrepreneurs [9] in conventional and innovation design respectively. The most prominent of these differences is that architects are system-centered and their architectures are realisations of system goals, whereas entrepreneurs are opportunity-centered, such that their architectures fit into an innovation strategy. This distinction is important: while architectural design is concerned with shaping an architecture to fit a socio-technical environment, innovation is equally concerned with shaping the socio-technical environment around the architecture. Because of this difference in perspectives, designing for innovation is often hampered by what has been described as the innovation design dilemma [14]: structured processes (such as those typically associated with secure system design) generate few ideas, while more unstructured processes generate more diversity – but at the cost of conflict that may hamper the implementation of innovation. The priority of innovation over system design qualities is evident when considering the role usability has played in several European security and privacy projects. The Prime Life project [22] demonstrated how privacy technologies can enable people to control their on-line personal information based on their legal rights. Based on their experience developing a number of Privacy Enhancing Technology (PET) exemplars, the Prime Life team identified several useful heuristics and idioms when designing and evaluating user interfaces for PETs. Usability researchers played an important role in evaluating Prime Life, but there is less evidence of their influence in the design of the exemplars themselves. Representations of users were created during the project to guide design decisions. However, in an example of privacy in social software [3], one of these representations is used only to illustrate access control policies in web forums, rather than describing how it influenced design decisions. While it is possible that these representations were used to inform architectural and application design, this is not reported.

B. Security and Usability Design

While the need for usable security had long been recognized, the work of Zurko and Simon on user-centered security — security models, mechanisms, systems and software having usability as a primary motivation or goal — was one of the first acknowledgements of the part designers can play in achieving this [37]. As well as reinforcing the need for security models that are sensitive to the mental models of different types of users, they propose combining security design techniques with established design techniques from HCI. Work by Sasse et al. [25] has illustrated how such usability design approaches can be applied to the design of security mechanisms. Based on the empirical findings from several studies about password usage, they found that framing the design of security controls from technology, user, goal, task, and context perspectives can lead to useful insights. Considering security as an enabling task that needs to be completed before a main task begins explains why users choose to circumvent controls getting in the way of their work. Although the work of Sasse et al. and Zurko and Simon continues to inspire HCISec research, much of this community's work focuses on studying the usability of security controls and interfaces rather than activities associated with designing interactive secure systems. Birge [2], in outlining five broad categories of research in HCISec, observed that the only one that mentions design is the 'Usability and Design Studies' category: exploring how traditional usability methods can be used to make decisions about user interfaces with security or privacy implications. In light of the lack of published work illustrating how usable and secure systems might be designed, a workshop was held in 2011 to explore whether it was possible to blend security, usability and software engineering lifecycles [15], with a number of informal anecdotes about promoting usability design during secure software development being discussed. Case studies exploring such insights are valuable, but still in short supply.

C. Lessons learned from e-Science and Neuro Grid

Some problems faced by R&D projects have already been encountered by the e-Science community, which is concerned with building IT infrastructures that facilitate global scientific collaboration. e-Science projects need to address two challenges relating to usable security design. First, global collaboration entails building coalitions and working partnerships between stakeholders who are distributed and culturally diverse. Cultural differences can lead the same artifacts in the same apparent context of use being perceived and used in different ways based on the norms and values of different scientific communities [8]. Distributed stakeholders also mean that carrying out formative evaluation activities to glean and reason about these differences can be expensive and time-consuming. Second, security is not usually treated as a priority when attempting to cast light on scientific uncertainties. Prioritizing core functionality does not mean that security is ignored in e-Science, but, as is suggested in [17], there is a tendency to treat design for security in an ad-hoc manner. Given the different perceptions stakeholders might hold

IPHV8I2021X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .II
ISSN Online: 2319 – 9253
Print: 2319 – 9245

about assets in an e-Science project, security design decisions might be ill suited to the assets needing protection. For example, some stakeholders may not believe highly aggregated data sources to be a valuable target for an attacker; however, with the right search criteria, they could be de-anonymised to the detriment of another stakeholder. The NeuroGrid project is a useful exemplar of the implications of usability and security in the design of e-Science projects. This project is also the subject of one of the few published studies describing the application of security and usability practices in e-Science projects [30]. NeuroGrid was a three-year project funded by the UK Medical Research Council to develop a Grid-based collaborative research environment; this was designed to enhance collaboration both within and between clinical research communities [18]. The sensitivity and distributed nature of the clinical data drove the need to find secure and effective ways of accessing and managing it. The project was ultimately successful, but several problems stymied the adoption of NeuroGrid in its targeted research communities; these included failing to specifically address usable security problems. Although usability researchers had elicited requirements from a wide range of project stakeholders, many security decisions with an implication on user interaction had been made before their engagement. As a result, the usability researchers felt they had little control over any design or implementation decisions affecting the security architecture; this meant there was little direct engagement with the team responsible for building and maintaining NeuroGrid's core security mechanisms. Because of this, when scientists expressed problems using the Public Key Infrastructure during formative evaluation activities, application developers were left with the responsibility of addressing them. While simple interface changes helped explain terms to non-specialists, as did writing documentation to explain security controls, developers were also left with the responsibility of implementing components to manage users' digital certificates on their behalf; this effectively replaced digital certificates with passwords as an authentication mechanism. Although this appeared to improve user perception of authentication usability, the design and implementation of these security components was not subject to the same security evaluation criteria as the rest of the security architecture, and, as such, the vulnerabilities associated with it were largely unknown.

III. APPROACH

In this section, we describe how security and usability design activities were incorporated into the development process of webinos. We begin by summarizing the webinos project and the motivations behind it, and then explain the development process, before describing the security and usability design activities in more detail.

A. About webinos

The webinos project was funded by the EU with a project team drawn from 24 organizations across Europe, including universities, mobile network providers, handset and automotive manufacturers, mobile software houses, and market analysts. The project ran from September 2010 to August 2013. The primary objective was to develop a federated software platform for running web applications consistently and securely across mobile, PC, home media, and in-car systems. The webinos platform provides a software runtime environment that allows the discovery of devices and services based on technical and contextual information. The platform also offers a set of APIs providing access to cross-user, cross-service, and cross-device functionality. More information about the webinos architecture can be found at [35]. Because webinos applications can access physical and informational resources across different devices, and this information may describe a user's personal habits and preferences, managing access is a complex security and usability design problem. Questions are also raised about the needs of application developers who must request permissions for their applications to access user data and provide facilities for users to manage their security and privacy. This is exacerbated by the lack of prior art and experience in cross-platform and multi-device personal networking, especially given the different physical and social contexts of use. The large number of collaborating partners, the innovative nature of the project, and the significant security concerns all influenced the need to integrate security and usability design techniques in the development process.

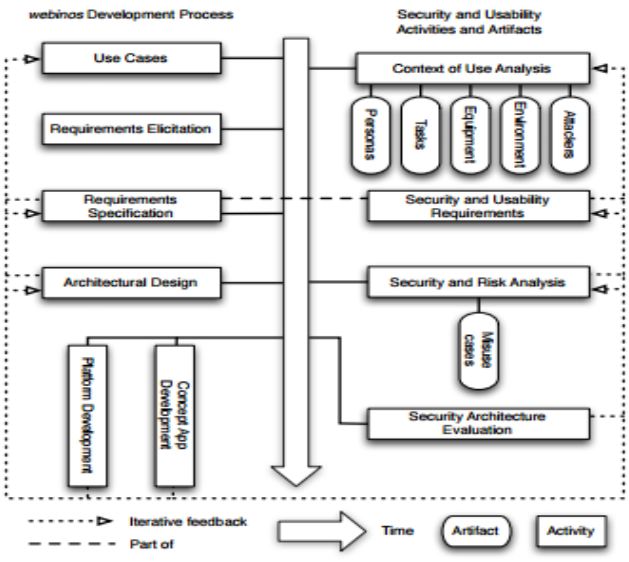

Fig. 1. webinos development process

The Design and Development process: A first version of the webinos platform was publicly released in March 2012, following an 18-month development effort that is

IPHV8I2021X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .II
ISSN Online: 2319 – 9253
Print: 2319 – 9245

summarized in Figure 1. A use-case driven approach was used to elicit and specify software requirements for the initial version of webinos. During the first six months of development, over 80 use cases were drafted and ranked by importance and novelty by the project team. As the project scope became clearer, the number of use cases were reduced to 33. These final use cases, and the process used to create them is described in [33]. The use cases, along with an industry and software ecosystem analysis, informed an up-front requirements elicitation and specification activity for the webinos platform. In the subsequent six months, these requirements informed the development and architectural design of webinos, its APIs, and the security and privacy frameworks. In the last six months, two concurrent activities took place: platform development, and development of a collection of concept apps. Both the platform and concept apps were implemented via an agile model based on the Scrum method [27], with sprints of approximately two weeks. Scrum teams were created for representative webinos platforms, with the core architectural components and features developed being derived from the architectural design documentation. Insights from the implementation led to updates to the use cases and requirements, which were delivered at the same time as the platform. In parallel, based on the API specifications, several concept apps were developed to demonstrate webinos' capabilities and provide feedback to platform developers.

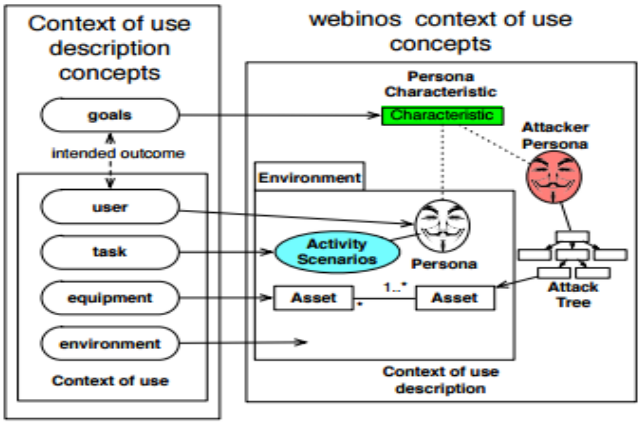

Fig. 2. Context of use description concepts

2) The Usable Security design team: Security and usability design activities were integrated into the design and development process. These activities took place during the project's first two years, and were carried out by a team of 10 practitioners and academics drawn from five different organisations; this team included some of the authors of this paper. Because the team was responsible for both security and usability, the team would avoid the engagement problems found on NeuroGrid. The distributed nature of the team meant that on-going progress was discussed in fortnightly telephone conferences during the first year. Although all team members had an information security background and an appreciation of the importance of usability in the design process, only three had any practical experience of applying usability design techniques to a real project. For this reason, detailed guidelines and examples of how to use the techniques were documented on a project wiki, and time was set aside at each telephone conference for answering questions about them. In the second year, telephone conferences and meetings were less frequent, and multi-day meetings were held during the project meetings that took place throughout the year. In the remaining 18 months of the project, the webinos platform was continually refined, with its source code being released to the community as multiple open-source projects [34].

B. Context of Use Analysis

To begin driving the project's security and usability design activities, the team created an adapted context of use description. This description, which is specified in [32], was used to support the specification and architectural design of webinos by capturing and modeling the security expectations of webinos stakeholders. Context of use descriptions describe the characteristics of a system's intended users, the tasks they are expected to perform, and the environments a system is expected to operate in [16]. Although context of use descriptions were devised to support formative and summative usability evaluation activities, the team wanted to use this description to motivate and justify security and usability design decisions; this would avoid issues found in previous European security and privacy projects where usability artifacts were disconnected from subsequent design decisions. Because of this, the context of use description developed for webinos was aligned with usability and secure system design techniques and concepts in a number of different ways, as summarized in Figure 2.

IV. SECURITY-USABILITY ANALYSIS OF SECURE SYSTEMS

To analyze the security and usability of a system based on the threat model, we use the concept of usage scenarios (or simply scenarios) and threat (negative) scenarios [33]. In our context, we define usage scenarios as actions that are desirable to stakeholders of a secure system and threat scenarios as actions that are not desirable and hence the system should not allow them to happen. HCISec, on one hand, is concerned with making usage scenarios accessible to the user with low mental and physical workload. For example, in an email application, usage scenarios could be composing an email, locating a contact, sending email, or creating new contact. On the other hand, HCISec is concerned with threat scenarios, undesired actions, that may cause non-malicious users to break the security of a system. The focus is on nonmalicious users who may break the system due to factors discussed in the security-usability threat model. While threat scenarios are usually associated

with malicious attackers, we associate them with legitimate users whose goal is no malicious. For example, in a secure email application, as much as we are concerned about whether users can encrypt emails, we are also concerned with whether they may accidentally encrypt a particular email with a key belonging to an unintended recipient. While usage scenarios and threat scenarios are traditionally used during requirements gathering and design [34], we apply them to security-usability analysis during system development life cycle as well as after product release. Figure 2 summarizes the steps in the security-usability analysis process.

A. Identify usage scenarios

In HCI, usage scenarios are identified before a usability evaluation. The scenarios are specific tasks that a typical user of a system would endeavor to accomplish. Usage scenarios are presented to participants (e.g. in usability testing) or evaluated by experts and performance measures are recorded. An evaluation of scenarios provides performance data on all the usability factors in the threat model. Evaluated scenarios are deemed usable if they meet pre-agreed criteria.

B. Identify threat scenarios

As earlier pointed, HCISec is also concerned about legitimate users making mistakes that break security of a system. We propose that events that may result in such behaviour (threat scenarios) are modeled and evaluated. The goal is to measure how easy legitimate users may unknowingly break a system. In device pairing, for example, a threat scenario could be users not paying attention to comparing strings which may result in indicating that the strings are matching when in fact they are not. Since the security of device pairing relies on users ensuring that the strings displayed are matching before they accept the pairing, lack of attentiveness may results in one or both devices pairing with an unintended device. To model this threat scenario and determine how likely users may break the system, no matching strings should be presented to participants and evaluated.

C. Assess difficult-of-use of usage scenarios

Usability of usage scenarios is important. We want users to perform them with minimal physical and mental effort. It is, therefore, crucial that we identify and minimize or eliminate elements that introduce difficult-of-use into a system. An assessment of difficult-of-use of usage scenarios can be in the form of usability experiments, cognitive walkthroughs, interviews, etc. Each usage scenario should be evaluated against usability factors in the threat model, that is: effectiveness, satisfaction, accuracy, efficiency, memorability, and knowledge/skill of users. It is important to note, however, that these factors are system specific. For example, while memorability is crucial in many authentication systems, it is not in most PKI or security tools. A security-usability evaluator must identify usability factors that affect a particular system. • Assess system de-motivators – with usage scenarios, we are interested in identifying system properties that may de-motivate users from using the system in a desired and prescribed manner. Identifying system demotivators is the first step to addressing them. Data collected while assessing difficult-of-use of usage scenarios provides a starting point to identifying system de-motivators. For example, the amount of time to accomplish a particular task may deter users from following prescribed procedure in using a system. Identifying system de-motivators focuses on elements of a system that deter or make it difficult for users to use a system effectively. • Identify external de-motivators – users may also be de-motivated from performing usage scenarios by factors that are external to the system. This is because systems, together with their users, do not operate in a vacuum but rather in concert with other systems. For example, external de-motivators may include environmental variables such as light intensity and noise, social variables such as pressure from people around, and personal variables such as age, gender, culture, and education. Users can also be de-motivated if they have access to a competing system that they perceive to be more usable. The competing system may be insecure— sending unencrypted email, for example—but may be seen as effective and efficient by users. The aim is to ensure that we identify as many external de-motivators as possible and eliminate or minimize their effect on system usability.

D. Assess easy-of-use of threat scenarios

Users follow the path of least resistance. Threat scenarios are the direct opposite of usage scenarios and we are interested in understanding how easily users can access them. If threat scenarios are much more difficult to accomplish compared to usage scenarios, legitimate users are unlikely to perform the former. Despite having good intentions, users may start performing threat scenarios if usage scenarios are harder to carry out. For example, an evaluation of an authentication system may consider how difficult it is for users to memorise secrets (usage scenario)—which may force users to write them down (threat scenario)—while an evaluation of a P2P system may consider how easy it is for users to share files they do not intend to share. • Identify system motivators – to understand why users may perform threat scenarios, system motivators must be identified. System motivators are elements of the system that may help users to perform threat scenarios.

If a threat scenario is more usable than a usage scenario, this can be seen as a system motivator for users to perform threat scenarios. Data collected when assessing the difficulty of use of threat scenarios and ease of use of usage scenarios will identify most system motivators. We can compare usability of usage scenarios and threat scenarios using

completion times, completion rates, etc, for example. • Identify external motivators – factors external to the system may motivate users to perform threat scenarios. For example, imperfect lighting conditions may make it harder for users to compare strings in device pairing and provide sufficient reason for users not to compare strings at all increasing the chance of performing threat scenarios. Similarly to usage scenario's de-motivators, the aim is to minimise external motivators for threat scenarios. We earlier pointed out how social context is an external motivator: users may share passwords or security certificates among themselves, or may share passwords with outsiders whom they perceive as trying to help.

E. Make recommendations

The final stage is making recommendations based on the preceding steps. Recommendations will be in the form of areas that need improving to make usage scenarios easily accessible to legitimate users and also areas that need to be hardened for threat scenarios. In addition to users and the system, the analysis process focusses on external factors. A system may be usable or secure in itself but may not when in actual use because external factors outweigh internal ones. For example, an employee who is aware of password security and of the need to avoid sharing passwords may be forced to share it with a colleague in stressful situations such as being late for work and needing to send an urgent report. It is unrealistic to expect to achieve maximum usability and security in all secure systems. In most systems, there will be a trade-off between security and usability. The goal is to minimize as much as possible the possibility of threat scenarios and maximize the accessibility of usage scenarios. For example, allowing users to write passwords down may be acceptable if the threat from attackers using dictionary based password cracking tools is particularly severe. It is also unrealistic to expect a complete reduction of all internal and external motivators, for threat scenarios, or demotivators, for usage scenarios. In either case, we want to minimize these factors to an acceptable level. An acceptable level varies from case to case and needs to be assessed based on the system and its context. In order to determine what usage and threat scenarios to be attended to first, both de-motivators and motivators can be prioritized using a risk-level matrix [35]. Using the matrix, each motivator or de-motivator can be ranked according to its likelihood and impact on the system.

V. CONCLUSION AND FUTURE WORK

We have proposed a security-usability threat model for conducting security-usability analyses. We have employed usage scenarios and threat scenarios to understand and identify both system and external elements that are threats to a system's usability, security, or both. Usage scenarios are used to identify areas that may hinder the usability of a system, whereas threat scenarios are used to identify areas that may help non-malicious users to break the security of the system. When a system's threat scenarios are more usable compared to the usage scenarios, users are more likely to perform the former. External factors, too, may cause users to perform actions that they may not normally perform. This is the initial effort in building a security-threat model for HCISec security-usability analysis. Future work will involve adding detailed metrics that can be used to calculate the likelihood of users performing a threat scenario over a usage scenario. Further work is also necessary to extend the threat model to malicious users.

REFERENCES

[1] K.-P. Yee, "Aligning Security and Usability," IEEE Security & Privacy, vol. 2, no. 5, pp. 48–55, 2004.

[2] I. Flechais, "Designing secure and usable system," Ph.D. dissertation, University of London, 2005.

[3] D. Balfanz, G. Durfee, D. Smetters, and R. Grinter, "In search of usable security: five lessons from the field," IEEE Security & Privacy, vol. 2, no. 5, pp. 19–24, 2004.

[4] R. Anderson, "Why cryptosystems fail," CCS '93: Proceedings of the 1st ACM conference on Computer and communications security, pp. 215–227, 1993.

[5] A. Whitten and J. Tygar, "Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0," in Proceedings of the 8th USENIX Security Symposium, August 1999, Washington, 1999, pp. 169–183.

[6] A. Adams and M. A. Sasse, "Users are not the enemy," Commun. ACM, vol. 42, no. 12, pp. 40–46, 1999.

[7] A. Brostoff, "Improving password system effectiveness," Ph.D. dissertation, University of London, 2004. [Online]. Available: http://hornbeam.cs.ucl.ac.uk/hcs/ publications/Brostoff Improving%20Password%20System% 20Effectiveness Thesis%20final.pdf

[8] S. Brostoff and M. A. Sasse, "Are Passfaces More Usable Than Passwords? A Field Trial Investigation," in Proceedings of HCI 2000, 2000. [Online]. Available: http://www.cs.ucl.ac.uk/staff/S.Brostoff/ index\ files/brostoff\ sasse\ hci2000.pdf

[9] C. Kuo, S. Romanosky, and L. F. Cranor, "Human selection of mnemonic phrase-based passwords," in SOUPS '06: Proceedings of the second symposium on Usable privacy and security. New York, NY, USA: ACM, 2006, pp. 67–78.

[10] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon, "Authentication Using Graphical Passwords: Effects of Tolerance and Image Choice," in SOUPS '05: Proceedings of the 2005 symposium on Usable privacy and security. New York, NY, USA: ACM, 2005, pp. 1–12.