Security and Usability: Analysis and Evaluation

Ronald Kainda and Ivan Flechais and A.W. Roscoe
Oxford University Computing Laboratory
{*ronald.kainda, ivan.flechais, bill.roscoe*}*@comlab.ox.ac.uk*

***Abstract*—The differences between the fields of Human-Computer Interaction and Security (HCISec) and Human-Computer Interaction (HCI) have not been investigated very closely. Many HCI methods and procedures have been adopted by HCISec researchers, however the extent to which these apply to the field of HCISec is arguable given the fine balance between improving the ease of use of a secure system and potentially weakening its security. That is to say that the techniques prevalent in HCI are aimed at improving users' effectiveness, efficiency or satisfaction, but they do not take into account the potential threats and vulnerabilities that they can introduce. To address this problem, we propose a security and usability threat model detailing the different factors that are pertinent to the security and usability of secure systems, together with a process for assessing these.**

***Keywords*-Security; Usability; HCISec; Threat model; Evaluation**

I. Introduction

Security and Usability seem to be at odds. It is a view of some that improving one affects the other in a negative way. Yee [1] attributes this conflict to system implementers treating security or usability as an add-on to a finished product. The second reason why security and usability seem to be opposed is the conflict of interest that exists between the system owner and its users. For example, in the music industry, some implementations of Digital Rights Management (DRM) have caused a lot of concern from genuine customers who would like to play their media on different devices and yet are prevented from doing so. However, because security is aimed at making undesirable actions more difficult while usability aims at making desirable ones easier for the user [2], it may also be true that improving one also improves the other. A usable system will minimise unintentional errors, while a secure system will aim at ensuring that undesirable actions in a system are prevented or mitigated. Human-Computer Interaction and Security (HCISec) arose because of the need that was identified by Human-Computer Interaction (HCI) experts to improve the usability of secure systems. This need has motivated the research community to re-examine the design and implementation of secure systems. However, despite efforts put into this research area, there are still many examples of secure systems being designed without enough consideration of usability.

Research in HCI started as early as 1975 [3] focusing on improving the usability of software through a systematic approach to design. However, despite its long existence, Balfanz et al [3], as late as 2004, pointed out that very little work was focusing on the usability of secure systems. As a result, secure systems are poorly designed leading to cases where users find alternative interactions with the system or avoid it completely [4]. Flechais [2] pointed out that HCISec is focusing nearly exclusively on improving the user interface of secure systems; while he recognises the importance of the user interface in making a secure system usable, he also gives examples of scenarios where a user interface alone is not sufficient. For example, [5] found that despite an improved graphical user interface in PGP users had significant difficulties in executing email encryption tasks.

With a growing recognition for the need to design systems that are both secure and usable, HCISec is increasingly becoming active. Usability studies have been conducted on authentication systems [6]–[10], email encryption [5], [11], security tools [12], [13], and secure device pairing [14]–[16]. These studies, however, follow standard HCI methodologies and procedures; methodologies and procedures designed for evaluating the usability of software systems in general, from which recommendations are made to improve ease-of-use.

Usability evaluations of secure software systems require procedures that deviate from standard HCI techniques. Whitten [17] highlights the differences between secure software and other software and why usability evaluation of secure software is difficult. In addition to encompassing main elements of education software (such as learnability), safeware (no undo for dangerous errors), and general consumer software (all kinds of user, goals set by users, no training), she highlights properties that make security difficult. These include the secondary goal property, hidden failure property, barn door property, abstraction property, and weakest link property. A usability evaluation of secure software should not focus on usability to the exclusion of security: in certain cases it is necessary, for the purposes of security, to include behaviour that is complex. Conversely it is possible to weaken the security of a system by simplifying or automating certain elements, which usually improve usability. Usability and security have a closely tied relationship, it is important to consider both factors when evaluating a system.

In this paper, we propose a security-usability threat model detailing the different factors that are pertinent to the security and usability of secure systems, together with a process for assessing these. The paper is organised as follows:

secure system evaluation is reviewed in Section II, followed by a description of our security-usability threat model in Section III. We present the process of analysing security and usability of a system in Section IV before concluding and discussing future work in Section V.

II. Background

There have been a number of attempts to define usability: the International Standard Organisation (ISO) [18] defines it as *the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*. This definition focusses on users' goals (effectiveness), the speed with which goals are achieved (efficiency), and users' satisfaction with the system within a specified context. The definition implies that usability is contextual. A system deemed usable in one context may not be in another. Other definitions of usability include other elements such as learnability [19] and memorability [20].

To consolidate all definitions of usability, it can be stated that usability consists of effectiveness, efficiency, satisfaction, learnability, and memorability. A usability evaluation of a system, therefore, focusses on one or more of these elements of usability.

Definitions of security, however, revolve around attackers. These are typically regarded as agents with a malicious intent rather than legitimate users of a system. The problem with focussing on malicious attackers is that it ignores the fact that non-malicious users may also compromise the system. Flechais [2] points out why this may be so: the user may not perceive the interaction to be detrimental to the system or has a greater incentive to engage in an insecure interaction.

In developing the security-usability threat model, we looked at current HCISec usability studies and identified the main factors that were the target of measurement for both usability and security. While reviewing, we noted that the studies fell into one of six categories:

- *Authentication*–authentication mechanisms have been, and continue to be, extensively studied. Studied mechanisms range from traditional authentication mechanisms (such as text passwords) to new proposals such as grid entry passwords or image based authentication. Studies on authentication mainly focus on measuring elements such as memorability or cognition, and efficiency (the speed with which one can successfully authenticate).
- *Encryption*–these studies focus on secure email. Unlike studies on authentication mechanisms that focus on memorability/cognition, studies on secure email focus on users' understanding of the mechanisms to send email securely. Knowledge of the mechanisms is a crucial factor for correct execution of email encryption as was found in [5].
- *PKI (Public Key Infrastructure)*–the main problem studied here is that of identity; whether a user can correctly identify a website to be secure or otherwise. Studies have been conducted on browser indicators that are supposed to provide a user with information about the security and identity of a particular website. Indicators include the traditional padlock symbol at the bottom right corner of a browser, colouring the address bar, and symbols and logos on a web page. Like encryption mechanisms, the major focus in these studies is on users' knowledge of particular indicators as well as factors such as vigilance (always looking out for indicators) and attention.
- *Device pairing*–studies on device pairing focus on efficiency, effectiveness (failure/success of pairing), and security failures. Security failures cannot be classified as part of effectiveness as they do not result in failure of the pairing, but result in a user pairing their device with an unintended one. In this case, the user has accomplished the pairing, but with the wrong device.
- *Security tools*–these are systems that help users manage their security. They include firewalls, password managers, and privacy managing tools. Studies of these systems focus on the users' knowledge in using the tool and whether it is used correctly or otherwise. Usability is mainly measured by whether users are able to achieve what they want or not, and whether what the user thinks the system has achieved is what the system has actually achieved.
- *Secure systems*–systems that do not fall into the above categories fall into secure systems. These systems are aimed at achieving user goals (that are not related to security) but have an element of security. For example, studies of peer-to-peer software have found that while users are able to achieve the goal of sharing files, many users unknowingly end up sharing files they would not want to share.

Reviewing studies in each of the above categories highlighted not only the main usability factors but also the security factors that were evaluated. There is, however, a blurred line dividing usability and security factors—some of the usability factors cause users to behave insecurely, and some of the security factors obviously impair performance.

Usability studies of secure systems focus entirely on elements of usability. All studies that we reviewed had a section on *usability analysis* that discussed various elements of usability. A complete evaluation must consider factors that may affect security as well. Evaluating one and not the other introduces similar problems as those discussed by [2]—a user perceives an action as harmless when it is not; a user has greater incentive to engage in a dangerous interaction; or a user is incapable of desirable interaction. It is, therefore, crucial and timely that a security-usability threat model that helps analyse both security and usability of a system is developed.

III. Security-usability threat model

HCISec is centred around the user. The user needs a system that is both secure and usable. Legitimate non-malicious users should not compromise or be duped into compromising a system's security. HCISec, therefore, requires methods and procedures that lessen user burden and protect the system from the very user. It requires a security threat model that encompasses elements of usability as a difficult-to-use system may force users to resort to insecure behaviour such as circumventing security processes—making protected assets insecure.

The HCISec security threat model must be different from standard security threat models. Standard security threat models focus primarily on malicious attackers who may or may not be legitimate users. HCISec's primary focus is on legitimate users' mistakes that may compromise the system. The security-usability threat model we present in Figure 1, therefore, centres around a legitimate user who has no intention of breaking the system.

The security-usability threat model depicts the critical factors that need investigation during the evaluation of usability and security. It identifies factors that are related to either usability or security and also factors that are related to both. Both security and usability factors relate to the legitimate user who has no malicious intent to harm the system. We discuss each of these factors below.

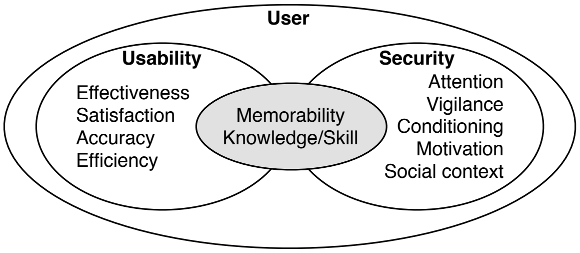

Figure 1. Security-usability threat model

A. Usability

- *Effectiveness*–a system is only useful if its users are able to achieve intended goals. An ineffective system is likely to be abandoned. Effectiveness is measured by whether users are able to complete a particular task or not. This approach is appropriate for most studies where a task consists of a single step that can be achieved through a single path. However, complex and multi-step tasks may require a more fine grained definition of success or failure which may include levels such as partial failure/success.
- *Satisfaction*–while objective analysis of usability analysis of systems is common, users' subjective assessment is crucial to a systems success. For example, a system may be usable (by usability standards) but users may label it unhygienic [21]. In other words, a system is bound to fail even when it is usable if it is not acceptable to users. User satisfaction can be assessed through interviews and rating scales.
- *Accuracy*–the accuracy factor was identified in authentication and device pairing studies. In many cases, authentication systems require users to enter passwords with 100% accuracy while certain mechanisms in device pairing require 100% accuracy when entering or comparing short strings [22]. Accuracy demands on users are impacted by other demands such as recall of required information, environmental, or personal factors.
- *Efficiency*–while users may use a system to achieve a specific goal, achievement in itself is not sufficient. The goal must be achieved within an acceptable amount of time and effort. What is the acceptable amount of time or effort in one system or context may not be in another. To this regard, a system is rated as efficient in relation to other similar systems or established benchmarks. Efficiency is captured by measuring the time to complete a task or the number of clicks/buttons pressed to achieve required goals.
- *Memorability*–many authentication systems require users to memorise secrets that they should recall whenever they want to be authenticated by a system. The number of secrets one is required to keep increases with the number of different authentication systems that individual interacts with. This results in memorability problems where users experience difficulties authenticating themselves to various systems and often ends in requests to reset those secrets [23].
- *Knowledge/skill*–usability definitions often use learnability to refer to how easy it is to learn to use a system. This is based on the assumption that users will learn or actually attempt to learn and understand the system. This assumption is flawed particularly in personal secure systems. What we found in the studies reviewed is that despite using a system, users only care about those parts that they think are important to specific operations they need—while in many cases security tasks are not seen to be important. For example, a study of a P2P system found that users did not know that the system shared folders on their local drives that were viewable on the internet [24]. More so, users of banking websites cannot distinguish between a padlock at the bottom of a browser window or one displayed as an image on a web page [25]. Previous studies have also found that training users in using secure systems is ineffective [26]. These problems exist because the

tasks—sharing folders, checking presence of padlock and learning about good security practices—are not users' goals in many cases.

The above factors have a direct effect on the usability of a system. A usability evaluation must determine which factors apply to a specific application and context.

B. Security

- *Attention*–users can easily be distracted, causing them to shift their attention from a task at hand. Security tasks must not demand undivided attention from users as this is likely to cause frustrations, and possibly security failures, in scenarios where the user is distracted. For example, device pairing methods such as comparing sound require users to be attentive while the sound is played; any distraction requires restarting from the beginning but may result in the user proceeding without cautious comparison. Moreover, users have a view that secure systems are disruptive—they often disrupt one's attention in order to attend to security prompts, for example. It is now, however, common knowledge that both disruptive (such as certificate prompts) and passive (e.g. browser padlock) approaches are usually ignored by users.
- *Vigilance*–secure systems tend to expect users to be alert and proactive in assessing the security state of a system. This has been problematic as studies have shown that even experts (people who understand the working of a secure system) are not always alert. For example, [27] found that experts on web site security indicators did not even look in places where those indicators were, hence falling for simulated phishing attacks which they would have avoided had they looked and noticed the absence or presence of indicators. Tasks that pose this security risk tend to be those that require users to divert attention from a primary task in order to attend to a security task. Such tasks should be analysed and integrated into users' work flow or eliminated if possible.
- *Motivation*–users have different levels of motivation to perform security tasks in different circumstances. For example, participants in a study in [22] indicated that they would prefer typing passkeys longer than 6 digits for financial transactions exceeding a certain monetary value. In this case, participants saw the risk to be more direct to them (losing money) than in a case where risk is perceived to be low or directed at someone else.
- *Memorability*–authentication systems often require users to memorise secrets that are difficult for someone else to guess or even attack by brute force. As the number of secrets one has to memorise increase, it can become more difficult to recall a particular secret when confronted with a system asking for one—particularly if the system is not used frequently. As a precaution to avoid forgeting and reseting, users write down these secrets. This in itself impacts the security of the system because written down secrets can be found by others who may then use them for malicious purposes.
- *Knowledge/skill*–users' knowledge or skill level plays a major role in the security of a system. Many users enter sensitive information on unprotected websites because they lack the knowledge or skill to distinguish between a secure and an insecure website [25]. Users also share sensitive information using P2P software unknowingly because they lack knowledge about the operation of P2P software. An evaluation of a secure system should ask questions such who are the users? what do they know about the system? what should they know?
- *Social context*–humans are social beings. They help each other and share various things. While sharing is generally a good thing, it is bad for security if users share their security secrets. For example, [28] found that users working on a particular project shared one digital certificate rather than each having their own as intended by system designers. Another common example is found in [29] where users shared passwords for various social reasons. Users also share secrets because someone is offering to help them if they disclose the secret. This has been exploited in many situations [30] and that is why it has been named *social engineering*. An assessment of a secure system should analyse how social context affects security.
- *Conditioning*–repetitive security tasks for which users can predict an outcome can become a threat to the security of a system. A common example are pop-up boxes that ask users whether a particular certificate should be trusted or not. A few encounters with such pop-up boxes make one realise that clicking a particular button will make the pop-up disappear and allows one to continue with a task. A security-usability analysis of a system should assess whether security tasks have the potential for condition users.

C. Measurable metrics

For a successful evaluation, both security and usability elements must be measurable. Measurements are crucial for comparative analysis and basic quantification of specific usability or security criteria. Table I summarise the measurable metrics for each of the elements in the threat model.

1) Usability metrics: Each usability factor can be measured and quantified using one or more metrics that are part of that factor. We can measure effectiveness by task success rates. Satisfaction is a subjective measure. It can be measured using rating scale questionnaires such as ASQ [31] or interviews. Accuracy can be measured as success rate on tasks that requires a certain degree of accuracy. For example, the number of users who successfully log on to a system using current text password methods provides a

Usability		Security	
Factor	**Measurable metrics**	**Factor**	**Measurable metrics**
Effectiveness	task success	Attention	Attention - failures
Satisfaction	Satisfaction	Vigilance	Vigilance - failures
Accuracy	Success rates	Conditioning	Conditioning - failures
Efficiency	Completion times, number of clicks/ buttons pressed	Motivation	Perceived, benefits, susceptibility, barriers, severity
Memorability	Recall	Memorability	Recall
Knowledge/ skill	Task success, errors, mental smodels	Knowledge /skill	Task success, mental models
		Social context	Social behaviour

Table I
MEASURABLE METRICS

measure of accuracy. A system's efficiency is measured as the amount of effort users expend to accomplish a task. This can be captured as the amount of time it takes to complete a task and number of clicks or buttons pressed. While memorability can be measured as the number of users who successfully recall a previously memorised password, many usability studies are conducted in laboratory environments where the length of time or usage pattern may be unrealistic. If a system studied is already deployed but a longitudinal study is not possible, users can be asked to report on their experiences in using the system. Users' understanding of a system can be shown through task completion rates, users' mental models, as well errors committed. A mental model can be measured by comparing a user's perceived security state of a system with actual state. Tasks may be completed successfully but with errors. Thus, it is essential that errors that do not lead to task failures are measured too.

2) Security metrics: Security factors must also be measurable. We can measure attention by monitoring and determining whether or not a security failure is due to lack of attention to specific piece of information. For example, eye tracking has been employed in studies of website security indicators to capture whether users take time to look at indicators or otherwise. We can also capture vigilance by monitoring whether users are consistence in paying attention to security tasks. This information can also be captured through self report questionnaires. In addition, we can capture conditioning by analysing users' mistakes and errors and determining whether previous events had an effect on the occurrence of those errors. Motivation cannot be directly measured but research shows that motivation to engage in a security task is driven by perceived susceptibility to attacks, benefits of and barrier to engaging in a security behaviour, and severity of a security failure [32]. Measuring these factors gives an indication of users' motivation to use a execute security tasks effectively. We can capture memorability by counting successful recalls and asking users whether they have memorability problems or not while knowledge/skill can be captured in form of task success, mental models, and errors. To capture effects of social context on the security of a system requires a qualitative approach; studying users behaviours in relation to those around them and elicitation of information on how they interact in relation to a studied system.

IV. Security-usability analysis of secure systems

To analyse 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 non-malicious 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 non-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 summarises 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 endeavour 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, non-matching strings should be presented to participants and evaluated.

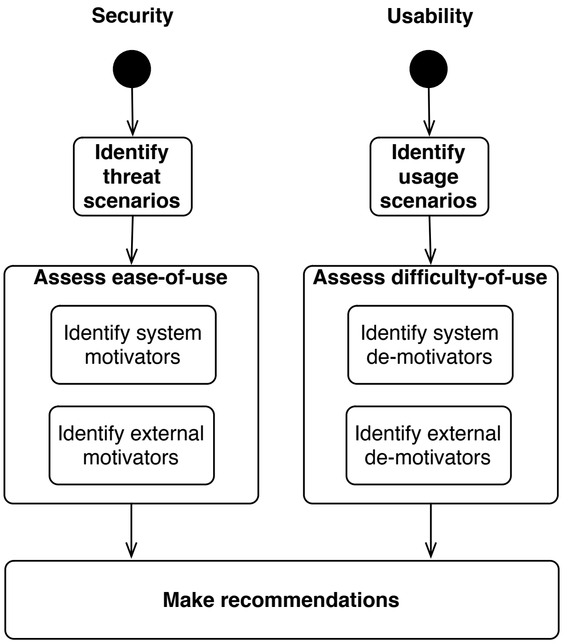
Security
Usability
Identify threat scenarios
Identify usage scenarios
Assess ease-of-use
Identify system motivators
Identify external motivators
Assess difficulty-of-use
Identify system de-motivators
Identify external de-motivators
Make recommendations

Figure 2. Process for security-usability analyses

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 minimise 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 de-motivators 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 focusses 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 minimise 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 minimise as much as possible the possibility of threat scenarios and maximise 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 de-motivators, for usage scenarios. In either case, we want to minimise 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 prioritised 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.

[11] S. L. Garfinkel and R. C. Miller, “Johnny 2: A User Test of Key Continuity Management with S/MIME and Outlook Express,” in *SOUPS ’05: Proceedings of the 2005 symposium on Usable privacy and security*. New York, NY, USA: ACM, 2005, pp. 13–24.

[12] L. Cranor and S. Garfinkel, *Security and Usability*. O’Reilly Media, Inc., 2005.

[13] U. Jendricke, U. Jendricke, and D. Gerd tom Markotten, “Usability meets security - the identity-manager as your personal security assistant for the internet,” in *Proc. 16th Annual Conference Computer Security Applications ACSAC ’00*, D. Gerd tom Markotten, Ed., 2000, pp. 344–353.

[14] R. Kainda, I. Flechais, and A. Roscoe, “Usability and Security of Out-Of-Band Channels in Secure Device Pairing Protocols,” in *SOUPS ’09: Proceedings of the 5th symposium on Usable privacy and security*, 2009.

[15] A. Kobsa, R. Sonawalla, G. Tsudik, E. Uzun, and Y. Wang, “Serial Hook-ups: A Comparative Usability Study of Secure Device Pairing Methods,” in *SOUPS ’09: Proceedings of the 5th symposium on Usable privacy and security*, 2009.

[16] E. Uzun, K. Karvonen, and N. Asokan, “Usability Analysis of Secure Pairing Methods,” in *Financial Cryptography and Data Security*, 2007, pp. 307–324. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-77366-5_29

[17] A. Whitten, “Making Security Usable,” Ph.D. dissertation, Carnegie Mellon University, 2004.

[18] T. I. S. Organisation, “Ergonomic Requirements for Office Work with Visual Display Terminals, ISO 9241-11,” 1998.

[19] B. Shackel, “Usability—context, framework, definition, design and evaluation,” pp. 21–37, 1991.

[20] J. Nielsen, *Usability Engineering*. Boston; London : Academic Press, 1993.

[21] M. A. Sasse, “Red-Eye Blink, Bendy Shuffle, and the Yuck Factor: A User Experience of Biometric Airport Systems,” *IEEE Security and Privacy*, vol. 5, no. 3, pp. 78–81, 2007.

[22] R. Kainda, “Human factors in hcbk protocols,” Master’s thesis, Oxford University, 2007.

[23] S. Brostoff and M. A. Sasse, ““Ten strikes and you’re out”: Increasing the number of login attempts can improve password usability,” in *Proceedings of CHI 2003 Workshop on HCI and Security Systems*. John Wiley, 2003.

[24] N. S. Good and A. Krekelberg, “Usability and Privacy: A Study of Kazaa P2P File-sharing,” in *CHI ’03: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM, 2003, pp. 137–144.

[25] S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer, “The Emperor’s New Security Indicators,” in *SP ’07: Proceedings of the 2007 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 51–65.

[26] M. A. Sasse, “Computer Security: Anatomy of a Usability Disaster, and a Plan for Recovery,” in *Proceedings of CHI2003 Workshop on Human-Computer Interaction and Security Systems*, 2003. [Online]. Available: http://www.andrewpatrick.ca/CHI2003/HCISEC/hcisec-workshop-sasse.pdf

[27] R. Dhamija, J. D. Tygar, and M. Hearst, “Why Phishing Works,” in *CHI ’06: Proceedings of the SIGCHI conference on Human Factors in computing systems*. New York, NY, USA: ACM, 2006, pp. 581–590.

[28] B. Beckles, V. Welch, and J. Basney, “Mechanisms for increasing the usability of grid security,” *International Journal of Human-Computer Studies*, vol. 63, no. 1-2, pp. 74 – 101, 2005, hCI research in privacy and security. [Online]. Available: http://www.sciencedirect.com/science/article/B6WGR-4G94J0R-3/2/5091b0c39ca9b9e17034b62db0004969

[29] M. A. Sasse, S. Brostoff, and D. Weirich, “Transforming the ’weakest link’ — a human/computer interaction approach to usable and effective security,” *BT Technology Journal*, vol. 19, no. 3, pp. 122–131, 2001.

[30] K. D. Mitnick and W. L. Simon, *The Art of Deception: Controlling the Human Element of Security*. New York, NY, USA: John Wiley & Sons, Inc., 2003.

[31] J. R. Lewis, “Psychometric evaluation of an after-scenario questionnaire for computer usability studies: the ASQ,” *SIGCHI Bull.*, vol. 23, no. 1, pp. 78–81, 1991.

[32] B.-Y. Ng, A. Kankanhalli, and Y. C. Xu, “Studying Users’ Computer Security Behavior: A Health Belief Perspective,” *Decision Support Systems*, vol. 46, no. 4, pp. 815 – 825, 2009, iT Decisions in Organizations. [Online]. Available: http://www.sciencedirect.com/science/article/B6V8S-4TYYMNV-3/2/842f546339406093d3e1fb6c28e5ef71

[33] I. Alexander and M. Neil, *Scenarios, Stories and Use Cases*. John Wiley, 2004.

[34] R. Kazman, G. Abowd, L. Bass, and P. Clements, “Scenario-Based Analysis of Software Architecture,” *IEEE Softw.*, vol. 13, no. 6, pp. 47–55, 1996.

[35] G. Stoneburner, A. Goguen, A. Feringa, N. I. of Standards, and T. (U.S.), *Risk Management Guide for Information Technology Systems [Electronic Resource] : Recommendations of the National Institute of Standards and Technology / Gary Stoneburner, Alice Goguen, and Alexis Feringa* . U.S. Dept. of Commerce, National Institute of Standards and Technology, Gaithersburg, Md. :, 2002.