Adaptive Security Dialogs for Improved Security Behavior of Users

Frederik De Keukelaere[1], Sachiko Yoshihama[1], Scott Trent[1], Yu Zhang[2], Lin Luo[2] and Mary Ellen Zurko[3]

[1] IBM Tokyo Research Laboratory, Kanagawa, Japan
[2] IBM China Research Laboratory, Beijing, China
[3] IBM Lotus, Massachusetts, USA
{frederik, sachikoy, trent}@jp.ibm.com {zyucrl, luolin}@cn.ibm.com mzurko@us.ibm.com

Abstract. Despite the increasing awareness of the importance of security for daily computer users, we see that many users still fail to behave securely when confronted with a security-related decision. In this paper, we introduce a new approach to security-related dialogs called Adaptive Security Dialogs (ASD). This approach is a combination of a new architecture and a new way of interacting with users to provide them with appropriate and effective security dialogs. ASD realizes this goal by matching the complexity and intrusiveness of security-related dialogs to the risk associated with the decision the user is making. This results in an architecture in which users can focus on their tasks, get (immediate) feedback on their decisions, and interact with dialogs with an appropriate complexity and appearance for the decision's associated risk. This paper makes the following three contributions. First, we introduce a general architecture for handling security-related decisions. Second, through an empirical user study using a web-based e-mail client, we show significant improvement in the care exercised by our participants without sacrificing usability. Third, we describe how the different pieces of existing research fit into the bigger picture of improving users' behavior.

1 Introduction

In their daily life, computer users are frequently confronted with security-related decisions. However, even though most computer users are aware of the existence of security risks when using a computer, many do not make safe security decisions [15].

The underlying causes of this have been identified as: (1) most users are task driven and do not want to be bothered by distracting security questions [15], (2) dangerous security decisions usually go unpunished and undetected for long periods of time [2], (3) many security-related dialogs are too complicated for typical users [7, 16], and (4) it hard for the users to estimate the level of risk, since the dialogs look basically the same regardless of their security implication. For example, a dialog for saving a file and a dialog for running active content in a browser often look much alike [15]. We illustrate this with real world examples in Sec. 2.1.

To battle these problems, several approaches have preceded ours. To counter task-driven behavior, [7] concludes that the user's primary task should be clearly interrupted. To ensure users feel the impact of their security decisions, users were immediately punished in [2]. To simplify the user experience, a complex operation with many dialog boxes, required to connect to a secured wireless network, was simplified to a three-click operation [1]. In [5], dynamic security skins make a dialog or even an entire application appear differently to prevent phishing attacks.

However, to the best of our knowledge, none of these proposals have addressed the different levels of user risk and correspondingly adapted their dialogs. Nor did they change the appearances of the dialogs based on user behavior or external factors such as recent attacks. In this paper, we introduce an architecture which has a general model of a user with respect to security decisions. This model, when confronting a user with a security decision, takes into account: (1) the security risk associated with the decision, (2) the user's recent security performance, and (3) the external security situation and related factors. We show how adapting the security dialogs to these factors significantly improves security behavior. To test the effectiveness of our model, we implemented a prototype web mail client and performed a user study.

This paper makes the following three contributions. First, we introduce a general architecture for handling security-related decisions (Sec. 2). Second, through an empirical user study, we show significant improvement in user behavior when opening e-mail attachments in a web-based e-mail client. This improvement was quantified by a reduction in opened attachments, and an increase in time spent on the decision whether or not to open the attachments (Sec. 3 and 4). Third, we describe how the different pieces of existing research fit into the general architecture and what work is needed to complete the puzzle (Sec. 5 and 6).

2 Adaptive Security Dialogs

2.1 An intuitive feel

To get a better feel for the current user experience with security dialogs, let us briefly consider opening attachments in an e-mail program. When trying to open an attached text file the typical warning users are confronted with is “Opening attachments can be dangerous to your system. Click OK to open the attachment.” When Bob, an inexperienced user, first sees this, he probably thinks “Is that so? Eh, what does that mean? What should I do?” and he quickly discovers that the message goes away by simply clicking OK. Alice, a more experienced user, will most likely think “I already know that, you have told me a thousand times. And by the way, text files are safe.”

Now suppose they are confronted with a dangerous attachment, an executable file disguised as a text file. Bob, who has gained some experience by now, has learned that if he clicks OK, he can quickly open the attachment. “Why not?” he thinks, “The other times I clicked OK nothing bad happened.” Alice, who is really in a hurry today, did not pay attention to the extension of the file name and is fooled by the icon of the attachment (which of course looked just like a text file). While unknowingly

installing the latest malware on her system, she thinks, “Is it really necessary to have these useless warnings for text files?” and continues her work.

Our goal with Adaptive Security Dialogs (ASD) is to change the outcomes of these all-too-common scenarios. Since ASD adapts dialogs to the risk a user is exposed to, the appearance of the dialog when opening a text file and when opening an executable file will be obviously different. For Alice, this would have been a clear indicator that she was not opening a text file. Confronting Bob with a new type of dialog along with explanatory information, allows him to learn that different files have different risks.

Since ASD learns about the performance of each user as they are confronted with security dialogs, Alice, who always behaves securely, will receive fewer complex dialogs. This allows her to complete her tasks with minimal disturbance. For example, after behaving safely for some time she will no longer be informed of potential risks when opening a text file. But she will receive an appropriate dialog box for high risk attachments, warning that attention should be paid. In contrast, Bob will get the maximum of information about the risks he is being exposed to and more feedback on the choices he makes so that he can learn how to behave properly. As his knowledge evolves, so will his user performance and the ASD system will become less intrusive.

A third aspect of ASD is adaptation to external factors. Suppose, for example, a worm is spreading as a macro in an MS Word file. Typically these attachments can be identified by a certain fingerprint (e.g., the file name). By using an external source (e.g., an anti-virus program), ASD could identify high risk attachments and warn the user in the attachment-opening dialog. ASD could even offer to inform the sender of the possible infection. Other external sources influencing the security risk level could include compliance with company security policy, current best practices, and so on.

The core goal of ASD is therefore to provide an effective dialog with an appropriate level of intrusiveness to the user's tasks while providing a personally optimized user experience with the necessary feedback.

2.2 Architecture

In this section we describe the components of the ASD architecture. Since the focus of this paper is to demonstrate the feasibility of ASD using an empirical study of its effectiveness in a web mail application, we leave a detailed explanation of the algorithms to a future paper. This section provides the high level overview required to understand the overall system and interpret the results of the user experiment.

To realize the goal of the ASD, we designed the architecture in Fig. 1. The architecture is composed of several data stores, engines, and an end-user application. Security information is collected and stored in the data stores. The engines use this information to alter the behavior of the dialogs. The different types of data stores are:

- Decision Risk Data (DRD): This data store contains information that links program execution points and decision types to risk levels. E.g., a mail client would have a corresponding record stating that opening an attachment (a program execution point) that contains an executable file (a decision type) is a high risk operation.
- User Performance Data (UPD): This data store contains user performance regarding security dialogs. E.g., if a user opens any attachment regardless of type, this incautious security behavior would be stored here.

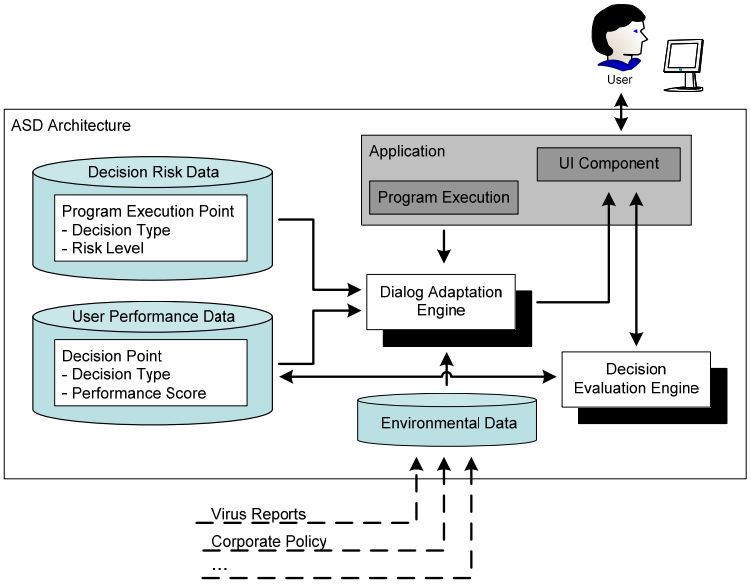

Fig. 1. ASD Architecture *The Dialog Adaptation Engine combines data sources to provide the user with an appropriate dialog. The user's decision is evaluated by the Decision Evaluation Engine which gives feedback to the user and loops the performance score back into the system.*

– Environmental Data (ED): This data store contains information from external sources which influence the engines. E.g., information about best practices, recent virus and malware trends, corporate policy, other users' behavior, and so on.

Based on information in these data stores, the engines will instruct the UI component to generate an appropriate dialog, track users' behavior, and provide user feedback. Feedback is used to train the user to make better decisions. The engines are:

– Dialog Adaptation Engine (DAE): This engine selects the appropriate dialog for a given security risk during program execution. E.g., for a normally cautious user who tries to open an text file attachment, a minimal dialog will be displayed.
– Decision Evaluation Engine (DEE): Users make decisions about each dialog, e.g., whether or not to open an attachment. The DEE tracks this and generates a user performance score, which is stored, and used to select dialogs. This engine guides the user by providing information, penalties, and or rewards.

2.3 Dialog Types

Although theoretically any type of dialog box can be used within our ASD system, we limited the dialog types in our empirical study to the following five classes of dialogs. These dialogs were selected based on the authors' estimation of three criteria: 1) understandability, 2) interference with the user's task, 3) estimated handling time.

– Warn-and-Continue (W&C) [2]: This is a dialog that warns the user and asks the user to confirm an action. This is currently the most used dialog in e-mail software.

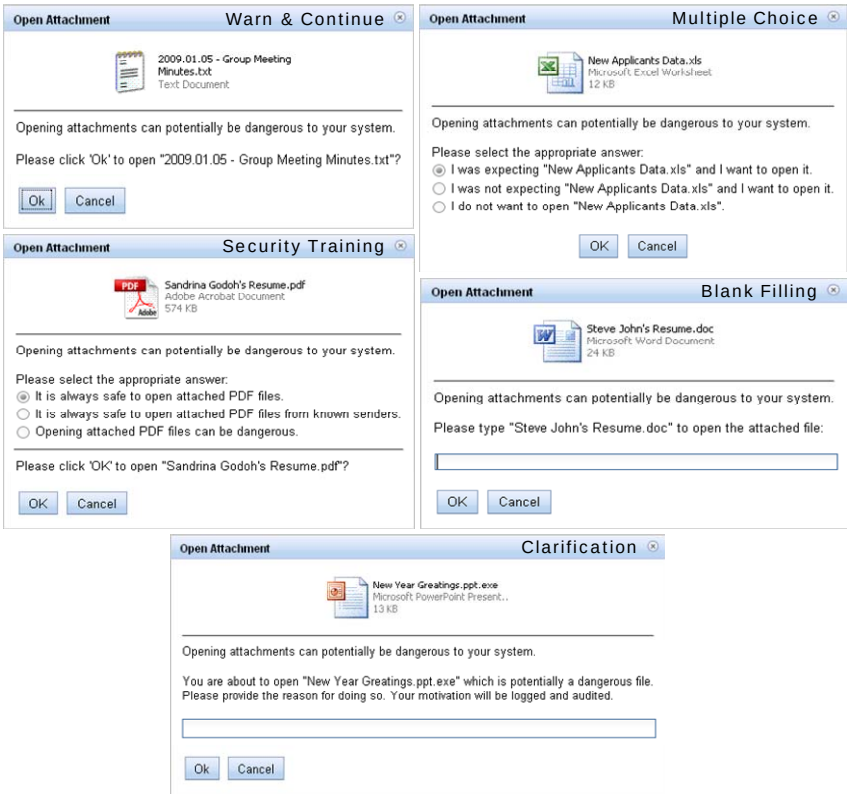

Fig. 2. Different types of dialog boxes *Warn-and-Continue, Multiple Choice, Security Training, Blank Filling, and Clarification Dialogs for attachments of different file types.*

In the upper left of Fig. 2 a sample W&C dialog is shown for the situation in which a user wants to open an e-mail attachment containing a text document.

– Multiple Choice Dialog (MC): This type of dialog provides the user with different options from which to choose the correct one to open the attachment. At the upper right side of Fig. 2 a sample multiple-choice dialog is shown for the situation in which a user wants to open an e-mail attachment containing an MS Excel File[1].
– Security Training (ST): This type of dialog combines a W&C dialog with a related security question. This question is taken from a set of security training questions and selected based on the topic of the W&C dialog. With the ST dialog the user learns more about the background of the question and can therefore make more informed decisions. At the bottom left of Fig. 2 a sample ST dialog is shown for the situation in which a user wants to open a PDF file.

[1] As an alternative a polymorphic dialog [2] could be used. Such a dialog changes the order of its elements each time it is presented, and provides step by step context-based guidance to the user. In our user experiment, we opted for the simpler multiple choice dialog to ensure that decisions can be made in one step.

– Blank Filling (BF): To complete this dialog the user is asked to confirm the file to be opened by typing in the file name. This makes sure the user is fully aware of which file he/she is being opened. At the bottom right of Fig. 2 a sample blank filling dialog is shown for the situation in which a user wants to open an e-mail attachment with the filename "Steve John's Resume.doc".
– Clarification Dialog (CD): In this dialog a user is required to type in the reason for opening the attachment. This reason is stored and later audited. This makes the user more aware of what he is doing and requires him to think about the motivation for doing it. At the bottom of Fig. 2 a dialog is shown for a situation in which a user is trying to open an executable attachment pretending to be an MS PowerPoint presentation[2].

3 Testing methodology

3.1 Experiment set-up

To test the impact of ASD on the user tendency to open all attachments, we implemented a web-based e-mail client simulating ASD and performed user trials. To make sure that none of our participants would perform better than others due to familiarity with a certain e-mail client, we built a new mail application based on the DOJO Dijit Mail demo [14]. As a result, our participants performed the tests on a browser-based AJAX application similar to currently popular online mail clients such as GMail [8], Yahoo! Mail [18], and Hotmail [10].

We created three different versions of the e-mail application for this experiment: (1) a version in which all of the dialogs are W&C dialogs, referred to as W&C throughout the rest of this paper, (2) a version in which dialogs are selected using the ASD system, except that feedback is given to the participant, called ASD throughout the rest of this paper, (3) a version in which dialogs are selected using ASD and automatic feedback is provided to the participant, called ASDF throughout the rest of this paper. Feedback was given by using virus warnings and explanatory dialogs to help users to make better choices for later attachments.

The questions we wanted to answer with this experiment are:

1. Do our participants behave more carefully when using ASD or ASDF versus W&C?
2. Does the use of ASD or ASDF come with a cost in usability?
3. How do the participants experience each type of dialog as regards complexity and interference with their tasks?
4. What is the effectiveness of immediate feedback on a participants' tendency to open attachments?

We divided users into three equal groups to observe any differences in the behaviors of the participants using W&C, ASD, or ASDF. One group used the W&C dialog application, the second group ASD, and the third used ASDF. We tracked the decisions of each participant on whether or not to open attachments. In addition, we

[2] This dialog can be considered a variation on the audited dialog introduced in [2]. Once again, we selected a simplification to maintain a single step decision process.

measured the time it took the user to answer each dialog. This time was measured by taking the difference between the time at which the dialog appeared and the time of the user decision. Due to the length of the experiment, it was not possible to evaluate the influence of repeated appearances of the same dialog on the decision time. Therefore, the measured time should only be seen as a first indicator towards the amount of attention the participant paid to the dialog when confronted with the choice to open an attachment or not. We used an *unpaired t-test* to evaluate the significance of the differences between the three groups.

At the end of the test, we presented the participants with a feedback form in which they were asked to evaluate our application. The W&C group was presented with a general questionnaire about the overall usability. The ASD group received the same questionnaire with additional questions regarding the difficultly of understanding the dialogs and the interference of the dialogs with completing their tasks. The ASDF group was asked the same questions with one additional question about the usefulness of the feedback. These final results were later compared with the objective effects observed during the tests.

3.2 Role Playing Scenario

Our participants were provided with a URL which gave them access to our experiment website. On the first page they received instructions on how to complete the experiment. The instructions explained (1) that we were testing a new web-based mail client for its usability, (2) that they were supposed to play the role of Chris Baker, an office worker at a credit card application company[3], and (3) a set of tasks that Chris needed to complete. Note that we chose not to disclose that we were actually measuring the efficacy and usability of ASD to make sure that our participants were not security-biased while performing the experiment.

Chris' task list was given to the participants to make sure that everyone would perform the same actions. A welcomed side-effect of a task list is that having the tasks distracts people from the actual dialogs, more as if they were doing their daily tasks in their normal working environments, which made the test more realistic. The tasks required each participant to read the mails in the Inbox and to possibly open the attachments. The following email messages were present:

- A request from a coworker to extract deadlines from some meeting notes. The notes were attached as a text file.
- A request from a coworker to search for the applicant with the highest annual income in a new applicants list. The list was attached as an MS Excel file.
- A request from a coworker to check out a cool New Year's card. The card was attached as an executable disguised as an MS PowerPoint file.
- A resume of an unknown job applicant. Attached as a PDF file.
- A resume of an unknown job applicant. Attached as an MS Word file.

[3] The authors acknowledge that providing participants with a scenario can influence their security behavior, as shown by Schechter et al. [11]. However, it was our intention to control the content of the messages and their attachments and therefore we could not allow our test subjects to answer their personal messages.

For the W&C group, all dialogs were of the W&C type. The dialogs for the ASD and ASDF groups are shown in Fig. 2. Due to the length of the test, it was not possible to study all aspects of the adaptive behavior of ASD. We studied adaptation based on the type of attachment (decision risk data) and the simulation of a recent outbreak of PDF exploits (environmental data). However, even with these limitations, we will show a significant improvement in behavior in Sec. 4. It is the authors' opinion that experimentally observing the educational benefit of ASD on user's behavior would improve the results even more. However, to obtain significant results regarding long-term user behavior improvement would require a longer observation period than in this experiment.

3.3 Participant Information

The participants in the experiment were all familiar with e-mail clients. However, we excluded potential participants with a background in computer science or engineering, or any person that had significant computer security expertise. Their security behavior is fundamentally different from the general computer users, for whom ASD was designed. After filtering, we had a total of 32 participants. From these, the results of 8 participants were excluded as incomplete. This could have been avoided if the tests were performed in a more controlled environment such as a fixed lab setting. However, for more natural behavior, we allowed participants to perform the tests in their own environments.

Table 1 summarizes the characteristics of the final participants and the survey summary results. A majority of our participants were female. This was not the authors' intention and we do not assign any significance to it. The other results depicted in Table 1 will be further discussed in the next section.

Table 1. Participants characteristics *Participant numbers, genders, average ages, and their feedback on the system.*

	W&C	ASD	ASDF
Participants (#)	8	8	8
Female (#)	5	6	6
Male (#)	3	2	2
Age (avg.)	30	32	28
Overall Usability	2.75 / 5	2.62 / 5	2.75 / 5
std.dev.	0.71	0.74	0.89
Usefulness Feedback			2.37 / 5
std.dev.			0.52

4 Experimental Results

Tables 2 and 3 contain the main results of our experiment. We indicate the mean, standard deviation, effect size, and p-value of our experimental results. The effect sizes are expressed as Cohen's d-values [4, 13] which indicate the strength of the

observed effect. A value larger than 0.8 is considered a large effect. The p-value of our *unpaired t-test* [3, 4] is used to indicate the statistical significance of our results. A value smaller than 0.05 indicates that our results are statistically significant and that we accept our null hypothesizes that with ASD fewer people immediately open attachments and with ASD people spend more time considering their decisions.

Table 2. Comparison between W&C and ASD (unpaired t-test, n=8) *ASD shows a significant increase in the time participants spent analyzing the dialogs and a reduction in the percentage of attachments opened.*

	mean	std. dev.	eff.size	p-value
attachments opened (%)				
W&C	92.50%	10.35%		
ASD	87.50%	10.35%		
difference	-5.00%	17.73%		not sig.
dialog response time (sec)				
W&C	3.85	1.76		
ASD	29.49	29.88		
difference	25.64	30.56	1.3	0.00004

Table 2 shows the comparison between W&C and ASD. It indicates that ASD causes an average 5% reduction in the number of attachments that our participants opened. The average time spent by users making a decision increased by an average of 25.64 seconds. This result was statistically significant and had a large effect (p=0.00004, d=1.3). Although the result for file openings was not statistically significant, combined with the decision times it indicates that our participants were more careful in ASD than in W&C when deciding to open an attachment. The large value for the standard deviation on the time measurements, 29.88 seconds, is caused by having five different types of dialogs with different average decision times, and representing a large variance in times.

Table 3. Comparison between W&C and ASDF (unpaired t-test, n=8) *ASDF shows a significant increase in the time participants spent analyzing the dialog and a significant reduction in the percentage of attachments opened.*

	mean	std. dev.	eff.size	p-value
attachments opened (%)				
W&C	92.50%	10.35%		
ASDF	75.00%	14.14%		
difference	-17.50%	12.82%	1.51	0.01707
dialog response time (sec)				
W&C	3.85	1.76		
ASDF	35.80	29.28		
difference	31.96	30.23	1.65	< 0.00001

Table 3 shows the comparison between W&C and ASDF. In this case, the results indicate that ASDF causes an average 17.50% reduction in the number of attachments opened by our participants. The average time spent making decisions went up by 31.96 seconds. In this comparison both values were statically significant and had a large effect (p=0.01707, d=1.51 and p<0.00005, d=1.65 respectively). The large standard deviation for time measurements has the same cause as in Table 2.

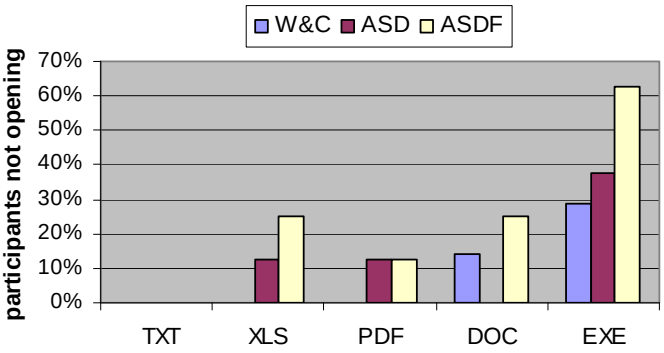

Fig. 3. Percentage of participants not opening attachments of varying file types *For W&C most participants opened all of the attachments, while for ASDF their behavior was more careful.*

Tables 2 and 3 indicate that ASDF causes an improvement in the care taken by our participants when opening attachments. Figure 3 reinforces this by showing the number of participants who did not open certain types of attachment using W&C, ASD, and ASDF. A percentage of zero indicates that all of the participant opened that attachment. With W&C, the participants tended to open almost every type of attachment. We see this is not the case for ASD and even more so for ASDF. An interesting result is that for BF (the dialog box used for opening a doc file) the performance of ASD is worse than W&C. The feedback we received from our participants is that they were not motivated to think about security when confronted with a blank to fill in. They simply performed the task they were asked to do, which was to type the file name so that they could open the attachment. In addition, interviews with our participants showed that to them file types had little security meaning. More education on the risks associated with certain file types could potentially improve the performance of this dialog box.

Figure 4 displays the participants' reported scores for difficulty and interference with their tasks. These values are expressed on a scale from 1 to 5, ranging from “easy to understand” to “difficult to understand” and “low interference” to “high interference.” In addition, this figure also contains the relative time participants spent on different dialogs. These values are expressed on a scale from 1 to 5, where 1 is the dialog that took the least time to respond to and 5 is the one that took the longest time. The reported values become larger as the dialogs were more demanding of the users. Again, the score for BF is unusual, since it indicates that in spite of the fact that it is considered more complex and more intrusive than dialogs such as MC, it does not result in more careful security behavior. However, it does have a relative shorter decision time.

The bottom of Table 1 contains feedback we received from our participants regarding the relative overall usability of the application (compared to their current mail client) and on the usefulness of the feedback in ASDF. These values are expressed on a scale from 1 to 5, ranging from “much less usable” to “much more usable” and “not useful” to “very useful” respectively. A score of 3 should be considered equally usable and to have neutral usability, respectively.

The participants rated our mail application as slightly less usable (a score less than 3) compared to their current mail clients. The main complaint we received was

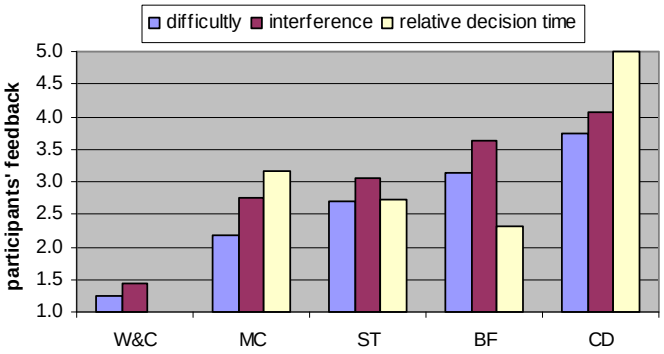

Fig. 4. Participants' experienced difficulty, interference and relative decision time for the different dialog types. *Paradoxically, little time was spent on the Blank Filling dialog despite the high difficulty and interference rating.*

that our application was missing functionality for forwarding mail and did not have a folder for sent mail. A more interesting aspect of the results was that they were all very similar to each other. This suggests that the dialogs are not a significant factor when deciding on the overall usability of a web based mail client. An alternative explanation could be that the dialogs in ASDF do not negatively influence the usability. In either case, the participants did not consider the ASDF versions to be (significantly) worse than the W&C version.

The participants in the ASDF test did not consider the feedback useful (a score less than 3). The main complaint we received was that the feedback came too late and had no impact on opening the attachment. For example, when a user was presented with a multiple choice question, he received feedback after making a decision. In this prototype, we did not provide any options to the user to change the decision. As a result, most of our participants spent little time reading the feedback and considered it to be interfering with their tasks. Despite this, the results for ASDF are slightly better than those for ASD.

5 Future work

The biggest challenge in evaluating the efficacy of ASD(F) is to carry out a long-term experiment that fully resembles the day to day environment in which participants are confronted with security dialogs. Due to the limited timeframe in which we evaluated our system, we were only able to measure our system in its initial state. Therefore, it was not possible for us to test the effectiveness of using dialogs to educate the users and improve their long-term security behavior. The already promising ASD(F) could be further improved by such a study. Interesting aspects to measure would be: 1) the impact of using different machine-learning algorithms to profile the security awareness of the participant, 2) the improvement of the perceived usability of a system that adapt to the user's security knowledge, 3) the extent to which a user learns to associate risk levels with the appearance of certain types of dialogs.

As a result of various studies [6, 7, 9, 17], we are beginning to understand the factors that influence a user's interaction with a security dialog. Also in this experiment, we found that there is no single factor that influences users. Further study of the correlation of the different aspects of a security dialog and users' security behavior (such as the impact of difficulty, interference, decision time, layout, and attachment file type) could be used as a basis for improving the base set of dialogs in our ASD architecture.

6 Related Work

Although security usability is still a field in which a lot of future research must be done, several relevant experiments have preceded ours. In addition to the references to related work throughout the paper, we list some of the most relevant studies here and consider how they compare to our experiments. A recent publication by West [15] tries to answer the question: “Why do well-intending users make dangerous decisions?” His analysis looks at the psychology behind user behavior when confronted with security decisions. One of the main thoughts in this paper is that people are generally unmotivated when confronted with security decisions. This was confirmed in our experiments as our participants often choose the quickest path to complete their tasks. Paradoxically, making them spend more time on some of the security dialogs did not have a negative impact on their perception of the overall usability of our application. This hints that, for certain decisions, users find interruptions justified.

Another aspect contributing to users making dangerous decisions is that it is generally hard for security-naive users to understand security precautions. In [16] Whitten evaluates how difficult it is for ordinary users to integrate encryption and signing of e-mails into everyday tasks. This work shows that security user interfaces need their own design principle and should be considered separately from normal user interface design. This was confirmed by Zurko in [19], where she evaluated how users respond to dialogs in Lotus Notes and noted the difficulty they had in understanding and correctly evaluating the content of the dialogs. In our experiment, ASDF ensured that the security dialogs were treated differently from other dialogs. We designed the dialogs so that they provide the user with the necessary information to facilitate decision making. A thought-provoking finding was that making users spent time on a dialog can provide improved security behavior. There appears to be a correlation between the number of actions required to process a dialog and security performance.

Dangerous decisions are also often triggered when users do not spend sufficient time evaluating a security decision. Users tend to be task driven and any dialog that stands between them and the completion of their task is generally considered an obstacle that needs to be overcome as quickly as possible. In [12], Sharek wanted to evaluate if users differentiate between real pop-up messages and fake pop-up messages. Perhaps the most interesting result from this study was that up to 40% of the test users just wanted to get rid of a dialog as quickly as possible and had very little or no concern about the content or authenticity of the pop-up. By giving our test

users a task driven scenario we evaluated the effectiveness of our solution against this potential problem.

In most applications, security dialogs are just one of many types of dialogs. This leads to the current situation where users tend to consider each dialog they encounter to be of equal importance. To the user, all dialogs appear the same [7]. For example, a warning that opening a suspicious attachment is dangerous and a dialog used for paragraph formatting often appear visually the same to users [15]. In [2] Brustoloni successfully introduced polymorphic and audited dialogs to make security dialogs stand out clearly from other dialogs. This idea was applied in Firefox 3 and Internet Explorer 7 to handle certificate problems more carefully. In our ASD architecture, we extended this idea to make the appearance of all security dialogs fundamentally different depending on the attachment type, user performance, and so on.

In [7], Egelman differentiates between active warnings and passive security indicators. Active warnings can be regarded as a form of dialog since they interrupt the user's task. Passive notices are just indicators on the users' screen of dangerous states. Egelman found that passive notices have very little effect on user behavior, so we designed our ASD architecture and experiment to use active security interruptions.

7 Conclusion

In this paper we introduced Adaptive Security Dialogs (ASD), a new architecture and approach to improve the security behavior of computer users. We made the following three contributions.

First, we introduced ASD, a general architecture for handling security-related decisions. We described the different components within ASD and illustrated how ASD adapts the type of dialog to (1) the risk associated with the security decision the user is about to make, (2) the user performance regarding previous security decisions, and (3) environmental factors such as virus reports, company policies, and so on.

Second, we studied the feasibility of our approach. We created several versions of a web-based e-mail prototype which we used to compare the current practice with ASD. We observed the security behavior of 24 participants while they were performing a set of e-mail-related tasks. With this empirical study we showed how our ASD prototype provides a significant improvement in the care exercised by the participants regarding opening attachments. Despite the high intrusiveness of our dialogs in certain risky situations, our participants rated the usability of all of the prototypes similarly, illustrating that ASD does not add significant overhead.

Third, the elements of our ASD architecture were compared to the current state of the art and we described how the different pieces of existing research fit into a bigger picture. In addition, we identified what work is needed to fill the gaps to build a fully adaptive security dialogs framework.

Acknowledgements. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. Their time and efforts have helped improving this paper. In addition, we would like to thank J. C. Brustoloni and R. Villamarin-Salomon for their permission to reuse their user scenario [2] in our experiments.

References

1. D. Balfanz, G. Durfee, R. E. Grinter, D. K. Smetters, and P. Stewart. Network-in-a-box: how to set up a secure wireless network in under a minute. In SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium, page 15, Berkeley, CA, USA, 2004. USENIX Association.
2. J. C. Brustoloni and R. Villamarin-Salomon. Improving security decisions with polymorphic and audited dialogs. In SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security, pages 76-85, New York, NY, USA, 2007. ACM.
3. R. Burton. Using Excel to do basic statistical tests. http://depts.alverno.edu/nsmt/stats.htm, 2002.
4. J. Cohen. Statistical Power Analysis for the Behavioral Sciences (2nd Edition). Lawrence Erlbaum, January 1988.
5. R. Dhamija and J. D. Tygar. The battle against phishing: Dynamic security skins. In SOUPS '05: Proceedings of the 2005 symposium on Usable privacy and security, pages 77-88, New York, NY, USA, 2005. ACM.
6. 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, pages 581-590, New York, NY, USA, 2006. ACM.
7. S. Egelman, L. F. Cranor, and J. Hong. You've been warned: an empirical study of the effectiveness of web browser phishing warnings. In CHI '08: Proceeding of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems, pages 1065-1074, New York, NY, USA, 2008. ACM.
8. Google. Gmail. http://mail.google.com.
9. P. Kumaraguru, Y. Rhee, A. Acquisti, L. F. Cranor, J. Hong, and E. Nunge. Protecting people from phishing: the design and evaluation of an embedded training email system. In CHI '07: Proceedings Of the SIGCHI Conference on Human Factors in Computing Systems, pages 905{914, New York, NY, USA, 2007. ACM Press.
10. Microsoft. Hotmail. http://www.hotmail.com.
11. S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer. The emperor's new security indicators - an evaluation of website authentication and the effect of role playing on usability studies. In Security and Privacy, 2007. SP '07. IEEE Symposium on, 2007.
12. D. Sharek, C. Swofford, and M. Wogalter. Failure to recognize fake internet pop-up warning messages. In Proceedings of the Human Factors and Ergonomics Society 52nd Annual Meeting, pages 557-560, 2008.
13. W. Thalheimer and S. Cook. How to calculate effect sizes from published research articles: A simplifed methodology. http://worklearning.com/effect sizes.htm, 2002.
14. The DOJO Foundation. Dijit mail demo. http://dojotoolkit.org/demos.
15. R. West. The psychology of security. Communications of the ACM, 51(4):34-40, 2008.
16. A. Whitten and J. D. Tygar. Why Johnny can't encrypt: a usability evaluation of pgp 5.0. In SSYM'99: Proceedings of the 8th Conference on USENIX Security Symposium, pages 14-14, Berkeley, CA, USA, 1999. USENIX Association.
17. M. S. Wogalter. Handbook of Warnings, Communication-Human Information Processing (C-HIP) Model, page 5161. Lawrence Erlbaum Associates, 2006.
18. Yahoo! Yahoo! mail. http://mail.yahoo.com.
19. M. E. Zurko, C. Kaufman, K. Spanbauer, and C. Bassett. Did you ever have to make up your mind? What Notes users do when faced with a security decision. In ACSAC '02: Proceedings of the 18th Annual Computer Security Applications Conference, page 371, Washington, DC, USA, 2002. IEEE Computer Society.