Using Reinforcement to Strengthen Users' Secure Behaviors

Ricardo M. Villamarín-Salomón, José C. Brustoloni
Computer Science Department, University of Pittsburgh
rvillsal@cs.pitt.edu, jcb@cs.pitt.edu

ABSTRACT
Users have a strong tendency toward dismissing security dialogs unthinkingly. Prior research has shown that users' responses to security dialogs become significantly more thoughtful when dialogs are polymorphic, and that further improvements can be obtained when dialogs are also audited and auditors penalize users who give unreasonable responses. We contribute an Operant Conditioning model that fits these observations, and, inspired by the model, propose Security-Reinforcing Applications (SRAs). SRAs seek to reward users' secure behavior, instead of penalizing insecure behavior. User studies show that SRAs improve users' secure behaviors and that behaviors strengthened in this way do not extinguish after a period of several weeks in which users do not interact with SRAs. Moreover, inspired by Social Learning theory, we propose Vicarious Security Reinforcement (VSR). A user study shows that VSR accelerates SRA benefits.

ACM Classification Keywords
D.4.6 Security and protection, H.1.2 User / Machine systems, H.5.2 User interfaces

General Terms
Human Factors, Security

INTRODUCTION
The designer of security dialogs faces a difficult problem: users tend to ignore such dialogs and accept risks imprudently. Earlier security warnings often used language users didn't understand and delegated to users decisions they were ill-prepared to make. Researchers demonstrated that dialogs that instead disclose threats in plain language and strongly suggest a preferred course of action can lead to significantly more prudent user decisions [9]. Dialogs in recent versions of applications such as Internet Explorer and Firefox have been accordingly modified [19]. Researchers have also found that user training before application use, possibly employing games [20], or training embedded in the application itself, especially in the form of cartoons [15], can also help users make more prudent security decisions. Nonetheless, even well-trained and informed users can be caught dismissing warnings and accepting risks imprudently. (For example, an amusement at several security conferences has been to disclose passwords of attending experts, caught by exploiting well-known avoidable vulnerabilities.)

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.

This paper advances the notion that this problem is also a behavioral one. Users acquire the habit of ignoring security dialogs partly because they find more rewarding to do so. Like other undesirable habits, learning that this habit may be harmful, and even conceptually understanding why it may be harmful, can be insufficient to quit it.

If ignoring security dialogs is at least partly a behavioral problem, then it ought to respond to behavioral interventions. We show that this is indeed true. In particular, we introduce Security-Reinforcing Applications (SRAs), a novel class of applications that can reliably deliver rewards when users accept justified risks or reject unjustified ones. Deploying SRAs, system administrators can manipulate users' reward matrix such that users find more advantageous to heed security dialogs and make more prudent decisions. We also contribute Vicarious Security Reinforcement (VSR), a form of training that is well suited for SRAs. SRA and VSR are intended for social contexts (e.g., work, school, and home contexts) where some individuals (e.g., managers, coaches, teachers, and parents) are tasked with supervising and positively affecting the behavior of others. We report user studies that show that SRAs are effective and continue to be effective even after users have not interacted with them for more than a month, and that VSR significantly accelerates learning of desired security behaviors in SRA users.

The next section provides an overview of psychological theories inspiring SRAs and VSR, respectively Operant Conditioning and Social Learning. We then argue why users may find it more rewarding to ignore security dialogs, and reinterpret behaviorally two previous interventions, polymorphic and audited dialogs [12]. Next, we describe the techniques we designed based on aforementioned theories. Afterward, we present our methodology for evaluating these techniques, and experimental results. We then discuss related work and present our conclusions.

THEORETICAL BACKGROUND
This section summarizes theories inspiring SRAs and VSR.

According to Operant Conditioning (OC), an individual acquires or maintains a behavior (or fails to do so) as a result of the behavior's consequences to the individual (especially consequences that are immediate and clearly contin-

gent on the behavior). A behavior's strength is measured by how often it's emitted [6]. Consequences that strengthen a behavior are called reinforcers (or rewards or incentives). Reinforcement is called positive or negative, respectively when it presents something pleasing or withdraws something displeasing to the individual. Consequences that weaken a behavior are called punishments. They present something displeasing or withdraw something pleasing to the individual.

Stimuli present in the environment only immediately before behaviors that are reinforced are called antecedents (or discriminative stimuli). Antecedents cue such behaviors [18], making them more likely to occur. Behaviors can be weakened by removing from the environment the respective antecedents, instead of or in addition to punishing them.

Behaviors can alternatively be weakened by extinction, i.e., ignoring them and making sure they are not reinforced. The schedule of reinforcement has large impact on how resistant a behavior is to extinction (more than does the magnitude of reinforcement). Continuous reinforcement occurs every time the individual exhibits the desired behavior. It is appropriate only for new or infrequent behaviors. Continuous reinforcement of high-frequency behaviors leads to early satiation and quick extinction when reinforcement is absent. On the contrary, intermittent reinforcement occurs only some of the times the individual exhibits the desired behavior [7]. It can preclude satiation in high-frequency or stable behaviors. An intermittent schedule is called fixed rate, variable rate, fixed interval, or variable interval, respectively, if the desired behavior must be observed a fixed or variable number of times or a fixed or variable time interval must pass before the desired behavior is reinforced. Behaviors reinforced according to variable rate or variable interval schedules tend to be particularly resistant to extinction.

Social Learning (SL) theory extends OC by noting that individuals can also acquire and maintain behaviors by observing their consequences in others (called models). This is known as observational learning (OL), modeling, or vicarious learning. OL can be quicker or less costly than an equivalent personal experience. A modeling intervention may conclude with reinforcement of the model if the behavior is desired. This process is called *vicarious reinforcement*. OL is governed by four sub-processes: attention, retention, reproduction and motivation [2]. These are further discussed in section "Vicarious Security Reinforcement."

PRELIMINARIES

This section discusses from a behavioral viewpoint why users ignore security dialogs, and the effectiveness of two previous interventions, polymorphic and audited dialogs.

Many users ignore security dialogs because they find it more rewarding to do so. Users typically view an application as a tool used to achieve some goal. Securely using the application rarely is a conscious or high-priority part of that goal. When a user dismisses a security dialog and accomplishes his goal, the latter accomplishment usually reinforces the user's behavior of ignoring security dialogs. If a risk that is object of the dialog materializes, security is compromised. However, often security breaches are not immediately apparent, or causal links between them and the user's dismissal of a security dialog are unclear. In any case, users rarely are held accountable for security breaches. Thus, security breaches are usually ineffective as punishments for ignoring security dialogs.

On the contrary, when a user heeds a security dialog and abandons his goal, he gets no reinforcement for his decision. OC predicts that lack of reinforcement tends to extinguish the user's behavior of heeding security dialogs. Worse, in some cases the user may be punished for abandoning his goal. A net result of these perverse incentives is that users learn to ignore security dialogs.

In previous research [12], we proposed two techniques to improve users' behaviors. First, we showed that users make significantly more prudent security decisions when presented with polymorphic instead of fixed dialogs. Polymorphic dialogs have intentionally variable form to make it harder for users to dismiss them. This result can be explained by OC. Fixed dialogs can be interpreted as antecedents of users' dismissive behaviors. Polymorphic dialogs weaken those behaviors by removing their antecedents. Second, we showed that users make even more prudent security decisions when logs of users' responses to security dialogs are available to auditors, and the latter suspend or fine users who respond inappropriately. However, suspensions or fines can leave users confused or upset. These results can also be explained by OC. Audited dialogs weaken the behavior of dismissing security dialogs by enabling reliable detection of such behavior and delivery of consequent punishments (e.g., suspensions or fines). However, punishments do not per se teach the desired behaviors. Without other interventions to achieve the latter, many individuals find punishment unfair.

Interventions with better effectiveness and user acceptance than those of audited dialogs would be highly desirable. SL and OC suggest that modeling and reinforcing desired behaviors and extinguishing undesired behaviors, in preference to punishing the latter, could achieve desired effects. Embedding such interventions in computer applications enables immediate feedback, maximizing intervention effectiveness. We accordingly designed SRAs and VSR.

SECURITY-REINFORCING APPLICATIONS

In this section we define security-reinforcing applications and present our research hypotheses about their properties.

Definitions and Efficacy

A security-reinforcing application (SRA) is a computer application that can reinforce its users' secure behaviors (e.g., with praise or notification of a prize the user will receive). An organization can initiate such reinforcement manually or automatically. In the former case, special entities (e.g., a company's security auditors) possess the privilege of instructing the application to apply reinforcement. In the

latter case, the application itself delivers the reinforcement when conditions specified by a policy are met. For instance, an SRA may be configured to reward employees automatically each time they reject three risks flagged as unjustified.

We define secure behavior as either rejection of *unjustified risks* or acceptance of *justified risks*. Insecure behaviors are defined as acceptance of unjustified risks (UR) and rejection of justified risks (JR). A security risk is justified if its acceptance is allowed by a security policy of the organization that a user is member of. In this paper, the policy we use is that a risk may be accepted only if (i) it is necessary to do a user's primary tasks, (ii) there are no other, less risky, alternatives to accomplish such tasks, and (iii) there are no available means to mitigate the risk. Otherwise, accepting a security risk is unjustified. For example, in the case of email, a UR may be an email message containing an attachment that is unexpected, from an unknown sender, unnecessary to the user's job-related tasks, or of a type that may spread infections (e.g., .exe). In this case, the user may mitigate the risk by, e.g., asking the sender to retransmit the attachment in a less risky file format (e.g., .txt). A JR may be represented by an email that (a) the user was expecting and contains an attachment useful to complete a work-related task, or (b) was sent by a known member of the user's organization, with wording not appearing out of character for such sender, and explaining clearly why the recipient needs the attachment for her work.

Existing computer applications typically are not SRAs. However, SRAs could be advantageous in a wide variety of domains. In the case of email, for instance, companies could designate a security auditor who may send employees email messages intentionally including JRs or URs. The auditor would disguise her messages to look like other email messages. The auditor would instruct the SRA to reward the user for rejecting URs and accepting JRs, according to a reinforcement schedule. Security auditors would include in these messages a special email header line that they would sign with a private or secret key that attackers cannot obtain. The SRA verifies the auditor's header line using the corresponding public or secret key, and hides it from users. By selectively rewarding the employees' secure behaviors, the auditor can increase the likelihood of secure behaviors, as predicted by OC. More specifically, our hypothesis is:

H1. When users interact with SRAs, they have lower UR acceptance than when they interact with conventional applications, whereas their JR acceptance and time required to complete tasks remain similar.

Reinforcing Stimuli

Little is known about what rewards would work well in a software environment such as SRAs. It is not possible to know a priori if a particular stimulus will be reinforcing for a user under specific circumstances. Auditors cannot simply ask users either, as self-reporting may be unreliable, especially if contingencies are complex [5]. Our hypothesis is:

H2. A combination of praise and prizes is an effective positive reinforcer in a security-reinforcing application.

A SRA can deliver different types of rewards to users after they emit secure behaviors. For instance, praise rewards can be easily presented as congratulatory messages. A prize reward can be delivered, e.g., by announcing that a bonus will be added to the employee's paycheck, or by showing a coupon code redeemable in authorized online merchants. Figure 1 shows a praise reward that an email client could be configured to show to users when they reject a UR. To help users who don't know what kinds of risk their organization deem acceptable, the software would provide a "[what's this]" link. If the user clicks on that link the software presents an explanation (figure 2). It's important that the user not simply learn to avoid all risks. Had the user accepted a justifiable risk, the software would present a dialog similar to figure 3. The dialog in figure 1 also announces that monetary rewards can be forthcoming if the user keeps handling her email securely. The user can get more information about the latter by clicking on the "[more info]" link (figure 4). Figure 5 shows a notification of a prize reward.

Security auditors who use SRAs can measure if a reward is reinforcing, and adjust it accordingly, by performing a direct test. If the frequency of a desired behavior increases when the presentation of a stimulus is made contingent upon the behavior, then the stimulus is considered reinforcing. Prizes and praise are generalized reinforcers [6] that are commonly used to strengthen a wide range of behaviors necessary to maintain productivity. Thus, it is plausible that they can be also effective in strengthening secure behaviors, though this has not been experimentally tested before.

Schedules of Reinforcement

Security auditors that employ SRAs need guidance on when to provide reinforcement. In general, reinforcement can be given continuously or intermittently. Auditors can arrange to provide reinforcement continuously during an initial learning phase, to promote user's acquisition of new behaviors. However, continuous reinforcement cannot be provided long-term. In production, only a small percentage of messages received by a user could be realistically expected to be tagged by auditors for reinforcement. Only intermittent reinforcement can be maintained long-term. Previous results from OC suggest that behaviors intermittently reinforced are resistant to extinction. However, this has not been verified in software applications. Our hypothesis is:

H3. Intermittent reinforcement schedules are effective in a security-reinforcing application.

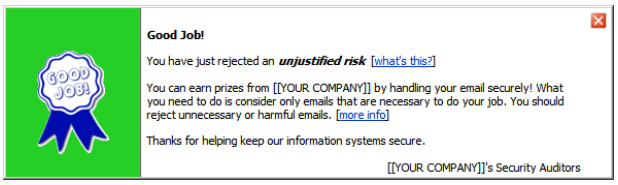

Figure 1. Example of a praise reward

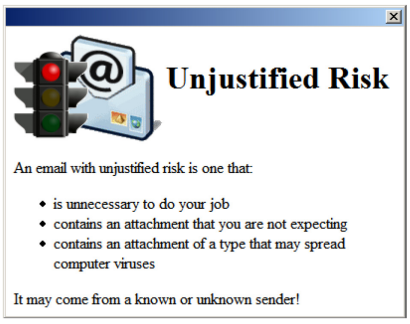

Figure 2. Information about URs

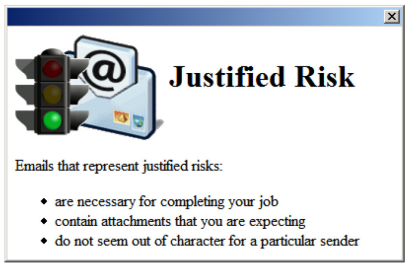

Figure 3. Information about JRs

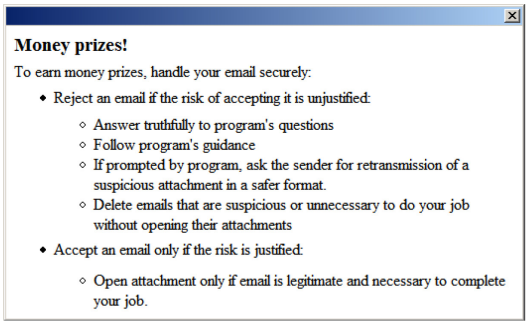

Figure 4. Info. on how to earn prize rewards

Resistance to Extinction

Users may not use SRAs during, e.g., weekends or vacations. Thus, security auditors cannot provide reinforcement every day or even every month. If users' secure behavior extinguishes during these absences, security auditors would need users to go through a learning phase after they return. We hypothesize that this will not be usually necessary:

H4. After a user's secure behaviors have been strengthened by interacting with a security-reinforcing application using intermittent reinforcement schedules, those behaviors remain strong after a period of several weeks during which the user interacts only with conventional applications.

Risk Identification

It may not be initially apparent to users why security auditors reward some decisions and not others. If users find an SRA's rewards unpredictable or unfair, they may reject the SRA, even if the SRA objectively improves security. To help users understand what is rewarded (and ultimately accept SRAs), all SRA's notifications include links that users can click to obtain plain-language explanations. During the initial learning phase, SRAs can also display notifications explaining what is not rewarded (e.g., figure 6). Users can ignore these notifications, and SRAs never penalize users for insecure behaviors.

Implementation

For testing our hypotheses, we extended the email client Mozilla Thunderbird 1.5 to convert it into a SRA, as described next. First, the application uses the same polymorphic dialogs as [12], to eliminate the discriminative stimulus of insecure behaviors which compete with secure behaviors [18]. A SRA with polymorphic dialogs asks the user to provide context information necessary for a security decision, and then suggests an appropriate course of action [12]. Second, we incorporated the praise and prize dialogs shown in figures 1 and 5. The praise dialog is shown non-modally and embedded as part of the application's chrome (just below its standard toolbar). The dialog in figure 6 is also shown this way. We did so to allow users to continue interacting with the program without having to explicitly dismiss the dialog first (as a modal dialog would force them to do). The prize notification is shown as a floating balloon above the application's status bar. A status message informs the user of the rewards he has accumulated for behaving securely. Both dialogs disappear whenever the user selects another message. Figure 11 shows an instance when both praise and prize rewards are given to the user at the same time. However, in general, each reward could be presented alone according to a reinforcement schedule. The tight integration of the reinforcing stimuli with the email client's chrome makes it difficult for attackers to imitate such stimuli. (They could try to do so to fool users into behaving insecurely.) Third, we implemented the continuous and fixed-ratio schedules of reinforcement, with the ability of presenting either praise or prize rewards just described. An arbitrary number of schedules can be active at the same time forming a combined schedule. When the requirements of the active schedule(s) are met, the appropriate stimuli are displayed immediately.

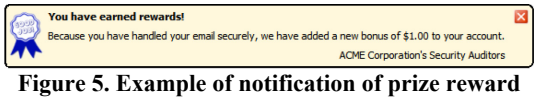

Figure 5. Example of notification of prize reward

VICARIOUS SECURITY REINFORCEMENT

In this section we first describe our rationale for complementing SRAs with vicarious security reinforcement (VSR) and our hypothesis about the latter. We then describe the design of a VSR intervention that we evaluated.

Rationale

SRAs can be effective in strengthening secure behaviors. However, when interacting with SRAs, users need to actually experience a situation in which they will be reinforced after securely handling a security risk. Thus, s/he may accept several URs or reject several JRs before s/he receives a reward. There are at least two undesirable implications of this. First, it may take some time for a user to understand the association between secure behavior and reward. Second, given the sheer number of risky situations affecting security, a user may get reinforced for securely handling some of them, but may miss others. A possible solution could be to include in instruction manuals or help messages rules for discriminating between types of risk, and the consequences of accepting and rejecting instances of each type. However, users may fail to read these materials, and even if they do read them, they may fail to see the

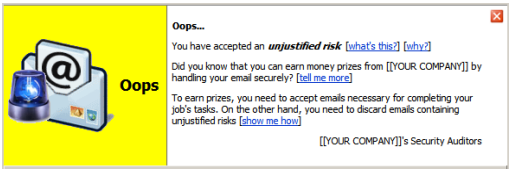

Figure 6. Dialog shown by a SRA whenever users behave insecurely in a learning phase

benefit of applying those rules and could simply ignore them [15]. Vicarious security reinforcement (VSR) can model secure behaviors and present their desirable consequences without waiting for users to emit fortuitously such behaviors and stumble upon their consequences. This use of vicarious reinforcement for strengthening secure behaviors is new. It is also worthwhile since faster improvement of security behaviors may help users avoid unnecessary errors. Specifically, our hypothesis is:

H5. While learning to use an SRA, users who previously had VSR training have lower UR acceptance than and similar JR acceptance and time to complete tasks as do users who did not have such training.

Design

In this section, we describe the design of our VSR intervention in which secure behavior of a model is reinforced. We describe the features included in our intervention grouped by the four sub-processes that govern vicarious learning. The produced intervention can be watched at [21].

Attention

Three aspects have been identified as influential in getting and maintaining observers' attention: the model, the observers' characteristics, and the modeling display [3].

First, regarding the models, there are two types frequently used: coping models and mastery models [17]. The former is a model whose initial behavior is flawed, but that gradually improves to the desired level of performance. The latter is a model that acts flawlessly from the beginning. Given that very proficient security behavior from a person (i.e., a mastery model) often has negative connotations (the person is seen as "anal" or "paranoid" [1, 13]), we chose to use a coping model. Other recommendations in the relevant literature about models and their characteristics, are that several different models be utilized, and that at least one "high status" model be included. We heeded such advice as follows. We used two extra models acting as co-workers of the main model. When interacting with the latter, they emphasized the desirability of behaving securely. This was also intended to convey the idea that secure behavior can be socially acceptable [13]. Also, we included a model portraying the coping model's boss. The latter's status is distinguished by more formal clothes. Second, the characteristics of the observer must be taken into account. Hence, we tested our intervention only with people having no computer-technical background but who had work experience, and who use or have used an employer-assigned email account to complete their work-related tasks. (Very technical people may not feel very inclined to pay attention to a person with limited technical skills such as the model in our video [1].) We conjectured that people with the selected profile would be more predisposed to empathize with the model, and thus to pay attention to him and his behavior. Third, there are several ways to display a VSR intervention such as live performances and videos. Since we could schedule only one person at a time, we chose to portray our intervention using a video, which is easily reusable. Experts (e.g., [17, 3]) argue that, for maximizing a vicarious-learning intervention's effectiveness, the modeling display should portray behaviors to be modeled: (a) vividly, and in a detailed way, (b) from least to most difficult, (c) beginning with a little stumbling, followed by self-correction, and with a strong finish, (d) with enough frequency and redundancy to facilitate retention, (e) keeping the inclusion of non-target behaviors to a minimum, (f) with a length of between 5 and 20 minutes, among others.

Based on these criteria, we implemented the intervention as a video with 4 scenes, and running time of approximately 10.5 minutes. Scene 1 first introduces Jack Smith, the main model in the video, in his work environment (figure 7). Then, it shows him receiving an assignment from his boss, who (a) hands Jack printed information useful to complete the tasks assigned, (b) states that other information will be sent by email, and (c) presses Jack to complete the task as soon as possible. Scenes 2 to 4 each shows the model handling risks of increasing difficulty. In scenes 2 and 3 Jack handles URs, while in the last scene he handles a JR. To make the model's behavior appear respectively detailed and vivid, he "thinks aloud" when trying to determine whether a risk is justified, and gesticulates accordingly. In scenes 2 and 3, at first Jack appears to fall for the ploy in the emails, and he is seen about to open the attached file. However, he realizes that the emails possess suspicious characteristics, verbalizes them, and rejects that risk. Lastly, in scene 4, Jack is initially wary about the JR in his inbox because it was sent by somebody who doesn't work in his company, and who he doesn't remember. However, after reading the email, he recalls that he was expecting such email based on information given earlier by his boss, and finally accepts it. We included a JR to avoid having subjects simply learn to reject any risk regardless of it being justified or not.

Retention

Several studies (e.g., [16, 8]) have shown that the inclusion of a list of "learning points" about the main ideas presented in a modeling intervention (e.g., video) enhances observers' retention. We implemented that suggestion by showing,

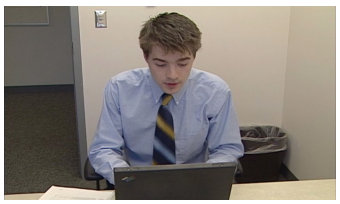

Figure 7. Jack Smith, the main model

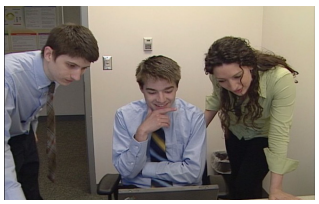

Figure 8. Model with co-workers seeing the reinforcing stimuli

Figure 9. Boss congratulates model for behaving securely

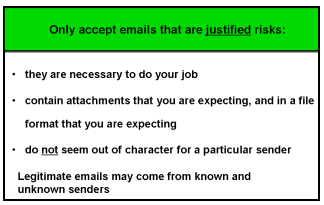

Figure 10. Clues shown after scene 4

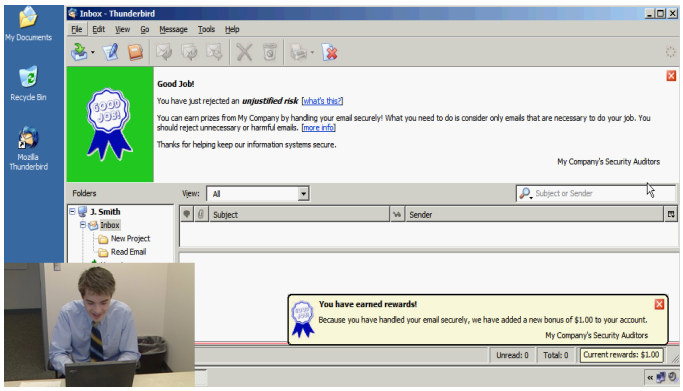

Figure 11. Model is reinforced for behaving securely

after scenes 2-4, a summary of the clues that the model used to identify the type of risk (e.g., figure 10 for scene 4) plus additional clues that an observer could use for the same purpose. The clues were shown and narrated one by one. Several clues were shown in all three summaries, thus providing the redundancy that facilitates learning [17].

Reproduction

Observers must be able to enact the behavior modeled in a vicarious intervention. In our experiments, the tasks assigned didn't require more skills than handling emails using an email client, opening attachments, and editing documents using Microsoft Word. Our eligibility criteria during recruitment ensured that subjects already had these abilities.

Motivation

Social Learning theory draws a distinction between acquisition and behavior since observers will not apply everything they learn [2]. To ensure enactment of modeled behaviors, it's necessary to make desired consequences contingent upon such behaviors. We incorporated this important point in our intervention, as explained next. The model receives the praise and prize rewards implemented for the SRA (see figure 11). These rewards are presented every time the model behaves securely, namely, after rejecting an UR in the scenes 2 and 3, and accepting a JR in the fourth scene. In addition, at the end of scene 2 after receiving the rewards, the model invites two coworkers, a female and a male, to see such on-screen rewards (figure 8). The former expresses surprise and satisfaction for the company's new practice of rewarding employees for managing their email securely, and asks the latter if he also considers such practice "cool." The male coworker model agrees, and mentions that he was rewarded earlier too. Then he states that he'll definitively be handling his email account more carefully. The boss model, who overheard part of the conversation when transiting through the hallway, enters into Jack's office and congratulates him for behaving securely (figure 9). He does the same with the male coworker, and states he is sure the female coworker will behave securely too. Before leaving, he encourages the models to keep up the good work. After discussing with Jack how to use the rewards they will get for behaving securely, the coworkers leave.

EVALUATION METHODOLOGY

This section presents the methodology we used to test our hypotheses. We performed a user study, called the SRA study, to test hypotheses 1-4, and another user study, called the VSRA study, to test hypothesis 5. No user participated in both studies.

Experiment Design

Each study used a within-subjects design, as recommended for reinforcement experiments [10]. Each subject role-played an employee in two similar scenarios, A and B, under a single study's different conditions. The first condition in each study was the same (control condition). Its goal was to measure each subject's performance when using conventional security dialogs. The control condition used a randomly selected scenario and the unmodified Mozilla Thunderbird 1.5 email application, while subsequent conditions used the other scenario and the SRA email application. Note that the control condition did not teach anything that might affect the subject's performance in subsequent conditions because (1) before the study, selected subjects were already familiar with email programs and conventional security dialogs, and (2) subsequent conditions use a different scenario and our modified security dialogs. In each subsequent condition, we compared the subject's performance following our interventions to the respective performance under the control condition.

The SRA study had four conditions: control, learning, maintenance, and extinction. The learning condition differs from the subsequent conditions by offering more frequent reinforcement. Continuous or very frequent reinforcement is often necessary for acquisition of new behaviors, according to OC, but in the long term makes those behaviors more susceptible to extinction. Moreover, very frequent reinforcement usually cannot be maintained in production. Accordingly, the maintenance condition offers less frequent reinforcement. Subjects progressed from learning to maintenance condition when their measured ability to discriminate JRs and URs was considered adequate. Proficiency is required before the maintenance condition because maintenance reinforcement could be insufficient for acquiring new behaviors. Subjects performed under the first three conditions in a single session. The extinction condition differs from the maintenance condition only in that it was performed in another session five weeks later. Between sessions, subjects did not use SRAs. The purpose of the extinction condition was to test whether acquired security behaviors would extinguish after long periods without reinforcement (as might occur, e.g., in an employee's vacation).

The learning condition used a combined schedule of reinforcement. Its component schedules were (a) continuous with praise reward, and (b) fixed ratio with a prize reward (money) every other secure behavior emission. The dialog in figure 6 was shown only during learning. As explained earlier, we do this to help users understand what behaviors are not rewarded (e.g., rejection of justified risks). The maintenance and extinction conditions used a different

combined schedule whose components were (i) fixed ratio with praise reward every other secure behavior emission, and (ii) fixed ratio with monetary reward every third secure behavior emission. Each prize reward consisted of $0.70.

The VSRA study differed from the SRA study in only two ways. First, between the control and learning conditions, subjects watched our VSR intervention. Second, there was no extinction condition, because the study's goal was simply to measure any speed up in learning caused by VSR.

Evaluation Metrics

We used metrics from signal detection theory [14] to quantify subjects' performance. In a signal-detection task, a certain event is classified as signal and a subject has to detect if the signal is present. Noise trials are those in which the signal is absent. The hit rate (HR) is the proportion of trials in which the signal is correctly identified as present. The false alarm rate (FA) is the proportion of trials in which the signal is incorrectly identified as present. A measure of detectability, known as sensitivity, is defined as $d'=z(\text{HR})-z(\text{FA})$, where z is the inverse of the normal distribution function. d' near 1 corresponds to moderate performance, while higher values correspond to better performance in distinguishing signal from noise. In our user studies, the signals were JR email messages, while the noise were UR email messages. We defined a hit to be user acceptance of a JR (signal present and correctly identified), and a false alarm to be user acceptance of a UR (signal absent and incorrectly identified as present).

Scenarios and Email Sets

We used the same scenarios in random order in the two studies. In scenario A, an employee is selecting applicants for a job at her company. In scenario B, an employee needs to process customers' insurance claims [12]. In both cases, the role-played characters work for fictitious companies and know specific people inside them.

We created 4 sets of emails per scenario. Each set consisted of 10 emails, half of which represented JRs and the rest URs. We will refer to these sets as Learning-I, Learning-II, Maintenance, and Extinction. There were two learning sets because some users may require more practice and reinforcement to learn to distinguish justified and unjustified risks. Each email in these sets contained a Word attachment. The arrangement of risks in each set is shown in table 1. We created these emails inspired on messages received in our email accounts (mainly URs) and emails in the Enron corpus [4] (mainly JRs). Each email contains a header that identifies the type of risk it represents, and which is signed by a security auditor.

Recruitment and Eligibility

We advertised the study with flyers around our university's campus, and with electronic posts in online websites. We announced that the study was related to email clients' usability, not security. Once interested people contacted us, we asked them to fill out a short web-based questionnaire to determine their eligibility. Subjects had to be at least 20 years old and native or proficient English speakers. They had to have work experience of at least one year in organizations that assigned them an email account which they had to use for job-related purposes. They had to have experience with desktop email applications, not just webmail. Finally, they could not hold or be currently pursuing a degree in computer science or electric engineering. The latter criterion was intended to avoid testing people who were already security proficient. Compensation per session was $15-$22.

	Learning-I	Learning-II	Maintenance	Extinction
1.	JR	JR	JR	JR
2.	JR	JR	JR	JR
3.	JR	UR	UR	UR
4.	UR	UR	UR	UR
5.	UR	JR	UR	UR
6.	UR	JR	JR	JR
7.	JR	UR	JR	JR
8.	UR	UR	UR	UR
9.	JR	JR	JR	JR
10.	UR	UR	UR	UR

Table 1. Risks arrangement in each set

Figure 12 presents the criteria we used for subjects to pass between conditions in a study. Only subjects whose sensitivity was $d'\leq\gamma$ during the control condition were selected for participating in the learning condition. We set cut-off $\gamma=1.02$ (moderate performance). Remaining subjects' security behavior was deemed as already strong, and unlikely to significantly benefit from our reinforcement interventions. If a subject's sensitivity was $d'>\gamma$ after handling the risks in the Learning-I set, the SRA pushed the entire Maintenance set into her inbox and activated the corresponding combined schedule. However, if the subject's sensitivity was $d'\leq\gamma$, the SRA kept pushing subsets $s_i \subset$ Learning-II into the subject's inbox and waited for her to handle the risks in those subsets. The SRA only pushed subset s_{i+1} if the subject's sensitivity was still $d'\leq\gamma$ after handling the risks in her inbox. Otherwise, the subject was switched to maintenance condition. The number of risks in the pushed subsets s_1, s_2, and s_3 was respectively 4, 4, and 2. Each subset contained an equal number of JRs and URs. If, after processing the entire Learning -I and -II sets, the subject's sensitivity hadn't exceeded the cutoff γ, her participation was terminated to limit the session's length. Subjects who progressed to maintenance were eligible for another session to test if their secure behaviors extinguished.

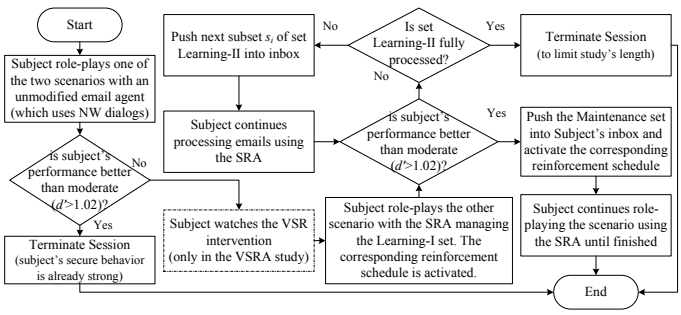

Figure 12: Criteria for passing between conditions

Laboratory Sessions

In the first session of the SRA study, and only session of the VSRA study, subjects received a handout that briefly described the scenario they were about to role-play, and were given the opportunity to ask questions about it. We told subjects that the main objective of the study was to evaluate the usability of email programs when used in a corporate setting. We didn't tell subjects that we were studying security of email clients because we didn't want to prime them to security concerns before the control condition. We asked them to behave as close as possible as they would if they were at work, considering the scenario they were about to role-play. We explicitly instructed subjects not to request information from us regarding what to do with the emails they were processing. We then had subjects sit at a desk in our laboratory, which we told them to be the office of the role-played employee. The desk was equipped with a laptop (running Windows XP Pro), a pen, and a phone in case the person wanted to make calls. Subjects were told they were allowed to call the fictitious company's technical support referred to in the handout, or to any other phone number they desired in relation to the experiment.

After finishing the scenarios, subjects who interacted with the SRA were asked to complete an exit survey. Then, during debriefing, we asked them to share with us some insights about their decisions of accepting or rejecting specific risks. They were also encouraged to provide feedback about our interventions. We didn't tell subjects in the SRA study whether they had qualified for a second session. Four weeks after the first session, we asked only those subjects who proceeded to maintenance in the SRA study to come for a second laboratory session during the subsequent week. When they came back, they received the handout of the last scenario they role-played. After they read it, we emphasized again that subjects should behave as closely as possible as they would do at work considering the role-played employee. After processing the extinction set, subjects were asked to complete the same exit survey of the first session.

Only in the VSRA study, just before qualified subjects proceeded to the learning condition, we told them that they were going to watch a video, and that it was up to them to decide what to do, when role-playing the described scenario, with the information presented. They could either apply the information given in the video or ignore it if that was what they would do if they were at work. To evaluate retention, after watching the VSR video, subjects took an on-screen quiz consisting of 4 questions. The quiz was previously unannounced to avoid biasing subjects to pay more attention than might otherwise be the case. Before starting the quiz, a message box was shown instructing users that, while taking the quiz, they should imagine they were Jack Smith, the model just observed in the video, and providing Jack's employer name and email address. Each question showed a snapshot of an email message, gave context information related to that email, and asked the user to identify whether it is a JR or a UR. Half of the questions were about URs and the other half about JRs. After a subject answered each question, a message box was shown telling her whether the answer she picked was correct and why. If the answer was correct, the subject was also congratulated. Once a subject finished the quiz, a short video [21] was shown explaining subjects that they shouldn't worry if they didn't remember all the rules shown in the VSR video, because they'd be interacting with an email program that uses context-sensitive guidance with polymorphic dialogs [12] to help users apply such rules. Then, a short video presenting a brief overview of such guiding interface was shown. Finally, subjects role-played the other scenario with our SRA.

EXPERIMENTAL RESULTS

A total of 37 people participated in our studies, but 13 of them did only the control condition because their behavior was deemed already secure enough (d'>1.02). We do not consider their results any further. Of remaining subjects, 12 participated in the SRA study (8 females and 4 males), and 12 in the VSRA study (7 females and 5 males). We scheduled an equal number of subjects of each gender, but absenteeism was higher among males. Most of these subjects had two or more years of work experience (10 and 12 respectively in the SRA and VSRA studies).

Tables 2 and 3 respectively show summary statistics of subjects' performance in the SRA and VSRA studies. To compare each reinforcement condition to the respective study's control condition, we calculated p-values with Wilcoxon's signed-ranks test. This non-parametric test is appropriate for comparing averages of related samples of any size without assuming that the averages have any particular distribution (the more commonly used t-test, on the contrary, assumes that averages have normal distribution, which is true only if the sample size is large or the underlying metric is known to have normal distribution). We used a one-sided test to compare UR acceptance and two-sided tests to compare JR acceptance and time to complete tasks, because we expected relationships as specified in hypotheses 1-4. Noted effect sizes are Cohen's d; values of (a) 0.2 to 0.3, (b) around 0.5, and (c) > 0.8, respectively correspond to small, medium, and large effects. One of the subjects in the SRA study didn't progress past the learning condition because the subject's behavior improvement was insufficient. The other 11 subjects were invited for a second session, and 7 of them did so after about 40 days.

We first describe results of the SRA study. As hypothesized, subjects had equivalent JR acceptance (essentially the same) in control as in learning (p=1.0, n=12), maintenance (p=1.0, n=11), and extinction (p=1.0, n=7). Also as hypothesized, there was a significant (and large) reduction in the acceptance of URs in the learning (p=0.002, d=2.85), maintenance (p=0.001, d=1.95), and extinction (p=0.008, d=4.29) conditions relative to the control condition. We observed that the acceptance of URs declined as subjects progressed from learning to maintenance to extinction, as might be expected from a learning effect, as subjects were reinforced for secure behaviors using the same scenario and

security dialog in those conditions (but not in the control condition). However, this improvement was not part of our hypotheses, and didn't reach statistical significance at the sample size considered. Also, the persistence of improvements in the maintenance and extinction conditions can be attributed to the use of intermittent reinforcement schedules, which make behavior resistant to extinction. Compared to the control condition, subjects spent less time completing tasks in the learning (p=0.04), maintenance (p=0.01), and extinction (p=0.016) conditions. These reductions were medium from control to learning (d=0.5), and large from control to maintenance (d=1.03) and control to extinction (d=1.94). In the SRA conditions, the reduction in task completion time was because subjects spent little or no time reviewing the attachments of UR emails. These results confirm hypotheses 1-4.

We now present results of the VSRA study. All subjects correctly answered all post-VSR quiz questions. We found that subjects had equivalent JR acceptance in learning (p=1.0, n=12) and maintenance (p=0.5, n=12) conditions as in the respective control condition. Also, there was a statistically significant (and large) reduction of UR acceptance, from respective control condition, in the learning (p=0.00024, d=3.8) and maintenance (p=0.00024, d=4.05) conditions. Moreover, there was a statistically significant and large reduction in time spent completing the assigned tasks when subjects used the SRA, during the learning (p=0.00097, d=1.71) and maintenance (p=0.00048, d=1.595) conditions than in the respective control.

To test hypothesis 5, we compared results of the SRA and the VSRA experiments. We could do such comparison because the tasks assigned to subjects in both cases were the same (they role-played the same scenarios). To make fair comparisons, we subtracted the rates obtained in the SRA-Learning, SRA-Maintenance, VSRA-Learning, and VSRA-Maintenance conditions from the rates in their respective control conditions. This was done to avoid possible biases because of a priori differences between groups (e.g., more skilled or risk averse subjects in one study than the other). We compared these differences with Mann-Whitney tests to determine which interventions achieved the largest improvements, according to the criteria stated in hypothesis 5. Times to complete tasks were compared directly without any adjustment. We did a one-sided test for comparing acceptance of unjustified risks and two-sided tests for comparing acceptance of justified risks and times. We computed effect sizes using pooled standard deviations. We found that there was no significant difference in acceptance of JRs or time to complete assigned tasks between subjects who interacted with the SRA with or without previous VSR training. Additionally, there was a significant and large improvement in rejection of URs in subjects in the VSRA-Learning condition relative to subjects in the SRA-Learning condition (p=0.033, d=0.89), but a non-significant difference between subjects in the SRA-Maintenance and VSRA-Maintenance conditions (p=0.074). These results verify hypothesis 5.

Subjects' opinions about the interventions were uniformly positive. This is reflected in the scores (worst=1, best=5) they gave in the exit survey that they took after the first session of the SRA study (we found no significant difference between these scores and those given by subjects in the second session), and after the only session in the VSRA study. Subjects found the SRA's user interface easy to understand ($\bar{x}_{SRA}$=4.4, $\bar{x}_{VSRA}$=4.4), and that it provided good guidance ($\bar{x}_{SRA}$=3.8, $\bar{x}_{VSRA}$=4.3). They moderately followed the guidance ($\bar{x}_{SRA}$=3.2, $\bar{x}_{VSRA}$=3.3), and found the questions somewhat helpful ($\bar{x}_{SRA}$=3.1, $\bar{x}_{VSRA}$= 3.3). Subjects would be comfortable with the SRA's guidance in the future ($\bar{x}_{SRA}$=3.7, $\bar{x}_{VSRA}$=4.0), and would give friends a mildly positive recommendation about it ($\bar{x}_{SRA}$=3.4, $\bar{x}_{VSRA}$ =3.4). A two-sided Mann-Whitney test found no significant difference between the scores of the two experiments.

RELATED WORK

Kumaraguru et al. [15] designed Phishguru (PG), a training system to educate users about phishing. PG sends users special phishing email messages with links to a website with cartoons that teach users how to avoid falling for phishing attacks. Unlike PG, SRAs embed rewards in the client such that it can deliver rewards immediately after

	Control	Learning	Maintenance	Extinction
# subjects	12	12	11	7
Hit (justified risk acceptance) rate				
mean	1.00	0.98	1.0	1.0
std. dev	0.0	0.06	0.0	0.0
False alarm (unjustified risk acceptance) rate				
mean	0.82	0.20	0.15	0.00
std. dev	0.16	0.24	0.30	0.00
Time to complete tasks (minutes)				
mean	26.23	19.97	15.99	12.96
std. dev	9.26	7.89	5.87	2.19

Table 2. Summary statistics of conditions in the SRA study

	Control	Learning	Maintenance
# subjects	12	12	12
Hit (justified risk acceptance) rate			
mean	0.95	0.93	0.88
std. dev	0.09	0.1	0.18
False alarm (unjustified risk acceptance) rate			
mean	0.88	0.083	0.05
std. dev	0.16	0.16	0.09
Time to complete tasks (minutes)			
mean	30.71	15.42	16.67
std. dev	9.62	4.58	5.79

Table 3. Summary statistics of conditions in the VSRA study

desired user behaviors. This could give an advantage to SRAs because OC suggests that the behavior learning effect is much stronger when rewards are immediate. When used to educate about organization-specific security policies and targeted attacks, PG and SRAs are likely to require similar supervisory effort (i.e., supervisors who know users' context well, set security policies, and help the system select and label instances of JRs and URs). However, to the extent that PG seeks to educate only about generic threats, it has the advantage that it can benefit individuals without supervision. Another difference is that users need to learn security concepts from the PG site and then remember and apply them unaided. Unlike PG, SRAs embed an organization's security policy in the application, guide users, and require only that users provide truthful context information. By reducing cognitive load, SRA may facilitate decisions involving complex policies. PG-trained users might be quicker applying simpler policies. Considering such tradeoffs, an organization might use both PG (e.g., for managers or simpler policies) and SRAs (e.g., for staff or more complex decisions).

Sunshine et al. [11] performed a usability study of web browsers' SSL warnings, and found that a large number of subjects ignored these warnings when using Firefox v2 (90%), v3 (55%), and Internet Explorer v7 (90%). They then designed two different warnings with an overall better effectiveness (45% and 60% of subjects ignored their first and second warning respectively). They concluded that warnings alone are insufficient to deter users from behaving insecurely. Their findings are consistent with our results in [12], where we found that polymorphic dialogs alone, although effective, need to be complemented with punishment or reinforcement to achieve larger improvements.

CONCLUSIONS

We evaluated employing reinforcement for strengthening secure behaviors through security-reinforcing applications (SRAs) and vicarious security reinforcement (VSR). SRAs reward users for accepting justified risks (JRs) and rejecting unjustified risks (URs). We tested a SRA in the context of email where a security auditor sends to end-users email messages with JRs and URs. The reinforcers used were praise and prize rewards. In a user study, users who interacted with a SRA behaved significantly more securely than when they interacted with a conventional application, and there was no adverse effect on time needed to complete tasks. Subjects were first conditioned using continuous reinforcement, and then their behavior was maintained with intermittent reinforcement. The strengthened secure behaviors didn't extinguish after a period of 40 days in which users didn't interact with SRAs. In another user study, before using the SRA, users observed a model being reinforced for secure behavior (VSR). These users improved their security behavior faster than did the first study's users (SRAs without VSR). VSR can help users avoid unnecessary errors while learning to distinguish JRs and URs and the consequences of accepting or rejecting risks of each type.

REFERENCES

1. A. Adams, and M.A. Sasse, "Users are not the enemy. Why users compromise computer security mechanisms and how to take remedial measures," Communications of the ACM, vol. 42, no. 12, 1999, pp. 40-46.
2. A. Bandura, Social learning theory, Prentice-Hall, 1977.
3. A.P. Goldstein, and M. Sorcher, Changing supervisor behavior, Pergamon Press, 1974.
4. B. Klimt, and Y. Yang, "Introducing the Enron corpus," in Proc. CEAS, 2004.
5. B.F. Skinner, "Operant behavior," American Psychologist, vol. 18, no. 8, 1963, pp. 503-515.
6. B.F. Skinner, Science and human behavior, Macmillan Pub Co, 1953.
7. C.B. Ferster, and B.F. Skinner, Schedules of reinforcement, Appleton-Century-Crofts, 1957.
8. G.P. Latham, and L.M. Saari, "Application of social-learning theory to training supervisors through behavioral modeling," Journal of Applied Psychology, vol. 64, no. 3, 1979, pp. 239-246.
9. H. Xia, and J.C. Brustoloni, "Hardening Web browsers against man-in-the-middle and eavesdropping attacks," in proc. WWW, ACM, 2005, pp. 489-498.
10. J. Cameron, & W.D. Pierce, Rewards and intrinsic motivation: Resolving the controversy, Bergin & Garvey, 2002
11. J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, & L. Cranor, "Crying Wolf: An Empirical Study of SSL Warning Effectiveness," in Proc. USENIX Security 2009
12. J.C. Brustoloni, and R. Villamarín-Salomón, "Improving security decisions with polymorphic and audited dialogs," in Proc. SOUPS, 2007, pp. 76-85.
13. M.A. Sasse, and I. Flechais, "Usable Security: Why do we need it? How do we get it," in Security and Usability: Designing Secure Systems That People Can Use, L. Cranor, and S. Garfinkel eds., O'Reilly, 2005, pp. 13-30.
14. N.A. Macmillan, and C.D. Creelman, Detection theory: A user's guide, Cambridge University Press, 1991.
15. P. Kumaraguru, Y. Rhee, S. Sheng, S. Hasan, A. Acquisti, L.F. Cranor, and J. Hong, "Getting users to pay attention to anti-phishing education: evaluation of retention and transfer," in Proc. APWG's annual eCrime researchers summit, 2007, pp. 70-81.
16. P.J. Decker, "The enhancement of behavior modeling training of supervisory skills by the inclusion of retention processes," Personnel psychology, vol. 35, no. 2, 1982
17. P.W. Dowrick, Practical guide to using video in the behavioral sciences, Wiley New York, 1991.
18. R.G. Miltenberger, Behavior modification: Principles and procedures, Cole Publishing Company, 1997.
19. S. Egelman, L.F. Cranor, and J. Hong, "You've Been Warned: An Empirical Study of the Effectiveness of Web Browser Phishing Warnings," in Proc. CHI, 2008.
20. S. Sheng, B. Magnien, P. Kumaraguru, A. Acquisti, L.F. Cranor, J. Hong, and E. Nunge, "Anti-Phishing Phil: the design and evaluation of a game that teaches people not to fall for phish," in Proc. SOUPS 2007, pp. 88-99.
21. VSR intervention; http://vsr.securityconditioning.org