# IMPROVING COMPUTER-SYSTEM SECURITY WITH POLYMORPHIC WARNING DIALOGS AND SECURITY-CONDITIONING APPLICATIONS

by

**Ricardo Mark Villamarín Salomón**

B.S., Universidad Católica de Santa María, Perú, 1998

M.S., University of Pittsburgh, USA, 2009

Submitted to the Graduate Faculty of

Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2009

UNIVERSITY OF PITTSBURGH

ARTS AND SCIENCES

This dissertation was presented

by

Ricardo Mark Villamarín Salomón

It was defended on

October 29th, 2009

and approved by

José Carlos Brustoloni, Department of Computer Science

Adam J. Lee, Department of Computer Science

G. Elisabeta Marai, Department of Computer Science

James B. D. Joshi, Department of Information Science & Telecommunications

Dissertation Advisers: José Carlos Brustoloni, Department of Computer Science,

Adam J. Lee, Department of Computer Science,

# IMPROVING COMPUTER-SYSTEM SECURITY WITH POLYMORPHIC WARNING DIALOGS AND SECURITY-CONDITIONING APPLICATIONS

Ricardo Mark Villamarín Salomón, PhD

University of Pittsburgh, 2009

Many computer security decisions depend on contextual information that computer systems cannot automatically obtain or verify. Users need to supply such information through, e.g., computer dialogs. Unfortunately, users often do not provide true information to computer systems, but rather (intentionally or automatically) input whatever information will quickly dismiss security dialogs and allow users to proceed with their primary goal (which is rarely computer security). Obviously, such user behavior can compromise computer systems' security. With the generalized use of the Internet today, an individual's insecure behavior can have severe negative consequences to his organization, including financial losses, unintended release of private information, or an inability to operate normally in everyday activities. In spite of such potential consequences, users continue to behave insecurely. Industry surveys and security researchers still find users to be the weakest link in the computer security chain.

To address the aforementioned problems, we first propose a model that helps explain why users behave insecurely when operating computer systems. Then, based on that model, we propose and evaluate techniques that improve users' security behaviors by automatically manipulating antecedents and consequences of such behaviors. First, we propose the use of warning polymorphism, which randomizes options in security warning dialogs, and delays activation of some of those options, so as to avoid cuing automatic, possibly untrue user responses. Second, we contribute the notion of security-conditioning applications (SCAs), and implement and evaluate two types of such applications, namely, security-reinforcing applications (SRAs) and insecurity-punishing applications (IPAs). SRAs strengthen users’ secure behaviors by reliably delivering reinforcing stimuli contingently upon such behaviors, according to a specific reinforcement policy and schedule. IPAs weaken users’ insecure behaviors by reliably delivering aversive stimuli, pre-specified by a policy, contingently upon those behaviors. Finally, we devise vicarious security-conditioning interventions to prepare users for interaction with SCAs and accelerate the latter's security benefits and user acceptance.

Results of empirical evaluations of our proposed techniques show that they are, indeed, effective in improving users’ security behaviors, increasing computer systems’ security. Moreover, we show that, with appropriate schedules and stimuli, such improvements are resistant to extinction over time.

## TABLE OF CONTENTS

## LIST OF TABLES

# LIST OF FIGURES

# PREFACE

Completing this research work has demanded from me more focus and dedication than I could have ever imagined. All the hours that I have spent designing the techniques that are at the center of my dissertation, coding them, planning and performing the experiments to validate their effectiveness, analyzing the resulting data, and then writing about them, seemed endless. The fact that many of these activities had to be done with little supervision for some period of time was not very encouraging either. Along the way, the temptation of giving up surfaced many times. Fortunately, I persevered despite all the unexpected obstacles that I encountered, and successfully completed all the requirements associated with earning my Ph.D. degree. I am sure that the journey through the maze of completing this process is unique for every graduate student, but one has to experience it before one can truly appreciate it. Over the course of my own journey I have come to appreciate many things like the encouragement, efforts, financial support, advice, and guidance that I received from many people, all of whom I try to remember and thank in the next paragraphs.

First, I would like to thank my parents, Carlos and Irene, who inculcated in me the belief that education is an essential component of a person's development and growth. Furthermore, they always gave me the freedom to pursue any intellectual interest I wanted to put my mind into. Without their love, efforts, and support I would not have been able to earn my Ph.D. degree.

Second, I would like to thank my adviser, José Carlos Brustoloni, for providing me with guidance and financial support for most of the time I was in the Ph.D. program. Further, his advice on how to become a better researcher will remain valuable to me for the rest of my life. He spent a lot of time helping me polish the ideas I wanted to evaluate in my research and contributed many important suggestions to my research as well. Finally, he relentlessly encouraged me to stay focused on my research work and to complete my dissertation.

Third, my entire committee deserves to be acknowledged for the completion of my degree. Special thanks go to Adam Lee for his suggestions to improve the organization of several chapters in the dissertation as well as the overall quality of the dissertation's text, and for recommending me to discuss other computer security areas in which my research could be applicable. Liz Marai gave me interesting recommendations about future work and called to my attention important aspects of my dissertation that I needed to emphasize more. James Joshi gave me helpful advice on aligning my research with future research goals, and also provided useful suggestions on what content I had to include in the dissertation to make it easier for others to understand the experiments I performed.

Fourth, I would like to thank many other current and former members of the University of Pittsburgh in general, and from the Computer Science Department in particular, without whose valuable help I could not have completed my dissertation. Elaine Rubinstein, from the Office of Measurement and Evaluation of Teaching, suggested the statistical methods I used to analyze the data of my experiments. Fellow Ph.D. students Nicholas Farnan and Roxana Gheorghiu, Robert Hoffman from Tech support, and John Ramirez from the Computer Science department's faculty, all participated in the production of my vicarious-conditioning interventions. Rami Melhem, the chair of the Computer Science department for most of the time I was in the Ph.D. program, greatly facilitated the process of approval of my dissertation committee and even provided personal funds that allowed me to complete my experiments. Daniel Mosse, current chair of the Computer Science department, approved the teaching assistantship I received for my last semester in the Ph.D. program, which allowed me to complete my dissertation. Markus Mock, former member of the Computer Science department's faculty, encouraged me to apply to the University of Pittsburgh's Ph.D. program, and, once admitted, agreed to initially supervise my work until I could decide the area in which I wanted to concentrate on for the rest of my Ph.D. studies.

Fifth, I would like to thank my friend Ernesto Cuadros, founder of the Peruvian Society of Computing, who introduced me and many other students in Arequipa, Peru, to the idea and benefits of applying to a postgraduate program in Computer Science that focused on research.

Last, but not least, I give many thanks to my wife, Rosario Itzamara, my sisters, Raquel Helena, Pilar Katherine, and Yasmin Irene, and my brother, Carlos Felipe, for their love and interest, and to my daughter, Stephanie Ymasumac, whose presence gave me an interminable source of motivation to complete my research.

# 1.0 INTRODUCTION

In an ideal world, the security of computer systems would be transparent to users, and they would not have to be bothered with constantly making security choices. This grand challenge has encouraged computer scientists to devise fully-automated technical solutions [117] to security problems [87]. In reality, however, only some security problems can be solved algorithmically based on input that can be obtained automatically. For instance, many technologies, such as security kernels [24], handle security tasks with no human intervention [117]. On the other hand, some security decisions require computer programs to ask human operators for input (e.g., using security dialogs) before they can be solved algorithmically, because the software cannot automatically determine all the contextual information relevant to the security decision. A case in point is when a user interacting with an email program receives a message with an attachment of a type that may spread infections (e.g., .doc [112], .exe). If the user knows the message's sender and was expecting such attachment from him, then she may decide that it is justified to open it. However, the email program is oblivious to such information. Another example is the Windows XP SP2's firewall program asking users if they want to allow an application to access the Internet.

For the cases where it is not possible to entirely automate security decisions, the presence of humans in the security loop must be considered by computer scientists who want to devise technologies and computer systems that are *actually* secure rather than *theoretically* secure

[14][86]. However, making computer security usable and apparent without overwhelming or decreasing the overall security of computer systems is a formidable research challenge in and on itself [92][117][31]. To achieve that goal two main approaches have been tried: (i) improving the user interface around security technology, and (ii) educating users on security issues and mechanisms. However a surprising number of difficulties have been found.

First, attempts to improve the usability of security technologies by bettering the user-interface around them have accomplished mixed results. On the one hand, empirical evaluations have shown that, in general, simply using nicely designed user-interfaces does not guarantee that a security mechanism will be used effectively [7][72][28]. On the other hand, although the usability of many security systems has improved as a result of modifications made at the user-interface level [103][94], many problems still remain. For instance, many security indicators that warn users about potential security hazards are not heeded by users [72][103]. This happens when users either do not understand the risks that such indicators try to convey or their importance, or have become habituated to ignore them. The latter is especially true when (i) user-interface elements get on the way of users completing their primary (or "production") tasks [8][74][98], (ii) the hazard that the security indicator warns of never materializes [77][8][49], or (iii) the causal link between the user's behavior and the adverse consequence is not perceived by the user [10][79]. In many of these cases, the security indicators themselves may become cues to insecure behavior. We were able to corroborate this problem when we evaluated a guiding user-interface designed to help users decide whether it was safe to open an attachment [98]. We noticed that participants quickly learned the option in the guidance that allowed them to open an attachment they wanted, however questionable, and ignored other, more suitable, options. When debriefed, they stated that they wanted to complete the tasks assigned in the study more quickly.

Second, many researchers have proposed educating users in security matters such as how to better use security technologies [47] or identify security risks (e.g., [106]). These approaches have focused primarily on both acquisition of knowledge [88][106][47] about security risks and retention of such knowledge [89]. However, it has been shown that users may successfully acquire and retain knowledge, and yet not apply such knowledge and behave insecurely. For instance, Dhamija and Perrig [94, p. 11][74, p. 27] found that some users who did know how to construct strong passwords, chose not to comply with the request to construct them, despite having received relevant training.

The non-intuitive human behaviors just described are often unexpected by computer scientists and software engineers, and make it difficult to write computer applications that are secure. Hence, a growing chorus of computer security researchers [117][76][42][31][28][8][92] is arguing that, in the same way that security must be designed into systems from the ground up: i.e., usability and the role of humans in a system cannot be an afterthought, but must also be considered from the very beginning when designing systems that need to be actually secure. Adding an attractive user-interface on top of an existing security system, or simply educating users in security matters may not be enough to achieve such a goal. As attackers pay more and more attention to the human element in computer systems [59], there is a pressing need that computer scientists provide guidance and principles to software engineers to produce applications that are secure when the human in the loop is considered. Such principles need to be devised and evaluated in a scientific way.

In this dissertation we aim to show that it is possible to make applications usable and secure if certain principles are considered from the ground up. These principles leverage earlier results from behavioral sciences. Specifically, we show that by manipulating the antecedents and consequences of security behaviors in an automated way, we can improve such behaviors, and thus increase the overall security of computer applications.

## 1.1 THESIS STATEMENT

In this dissertation we propose and evaluate approaches to strengthen users secure behaviors and weaken insecure behaviors. Our thesis statement is:

**Computer applications that (i) disrupt antecedents of insecure behavior, (ii) reliably convey reinforcers or punishments contingent upon security decision behavior, and (iii) whose use may be anteceded by interventions that vicariously condition such behavior, can improve computer systems' security.**

In this statement, a security decision behavior can be either an insecure or a secure behavior. An insecure behavior is any user action that causes a computer system to enter into an insecure state [66, p. 95]. For instance, accepting security risks that are considered unjustified in a security policy. A secure behavior, instead, would be user actions considered justified with respect to a security policy. User actions in these definitions are restricted, in the present research, to the set of actions possible during the interaction between end-users and computer programs. Opening an email attachment that a user is expecting and that is necessary to do her job is an example of an action in this set. However, actions such as stealing a computer or writing a password down on a piece of paper are not addressed in this thesis, because these actions do not involve explicit human-software interaction.

## 1.2 RESEARCH QUESTIONS

To demonstrate our thesis, we designed automated security techniques and evaluated them in five user studies that answer the following questions.

1. Does disrupting the usual antecedents of insecure behaviors cause a significant reduction in their frequency? If so, does it also significantly reduce the frequency of secure behaviors? Does such disruption significantly increase the time needed to complete production tasks compared to cases where that disruption does not occur? (Chapter 4)
2. Do computer applications that make positive consequences contingent upon users' secure behaviors cause a significant increase in the frequency of such behaviors? If so, do they also significantly affect the time needed to complete production tasks compared to computer applications that do not employ such positive contingency? What positive consequences are effective for such purpose? When and how often is necessary to make those consequences contingent upon users' secure behaviors to increase and maintain their frequency? Does the lack of application of such positive contingency for a period of many hours, days, or weeks cause the frequency of secure behaviors to decrease significantly or at all? (Chapter 5)
3. Do computer applications that make negative consequences contingent on users' insecure behaviors cause a significant reduction in their frequency? If so, compared to computer applications that do not employ such negative contingency, do they also significantly reduce the frequency of secure behaviors, or affect the time needed to complete production tasks? (Chapter 6)
4. Are computer applications that make positive consequences contingent upon users' secure behaviors as effective in promoting such behaviors and discouraging insecure ones as com-

puter applications that make negative consequences contingent on users' insecure behaviors? Are the former computer applications more acceptable to users than the latter? (Chapter 6)

5. If end-users vicariously learn that positive consequences will be made contingent upon their secure behaviors before interacting with computer applications that actually employ such positive contingency, does their secure behavior improve more than that of users who interact with the aforementioned applications without first observing vicarious reinforcement? If end-users vicariously learn that negative consequences will be made contingent on insecure behaviors before interacting with computer applications that actually employ such negative contingency, does their insecure behavior weaken more than that of users who interact with the aforementioned applications without first observing vicarious punishment? (Chapter 7)

Answering these questions will provide the necessary evidence to demonstrate our thesis. Addressing the first question will allow us to support point (i) of the thesis statement, whereas answering questions two to four will demonstrate part (ii). Finally, part (iii) is validated by addressing the fifth question.

## 1.3 OVERVIEW OF CONTRIBUTIONS

In answering the above questions, we make the following contributions.

First, we contribute a new model of why users may emit insecure behaviors rather than secure ones when interacting with computer applications to perform production tasks. In our model, the frequency of security decision behavior is regulated by its antecedents and consequences. Moreover, based on the model, we devise several principles and techniques to help improve users' security behaviors and implemented and evaluate such techniques, as detailed in the remaining contributions.

Second, we contribute polymorphism in warning dialogs (PD), a technique aimed at disrupting antecedents of insecure behaviors. Studies with human participants demonstrate that PD makes those behaviors less likely to occur, as predicted by the model.

Third, we contribute the notion of security-conditioning applications (SCAs) and develop and test two types of them, namely security-reinforcing applications and insecurity-punishing applications. These applications deliberately and reliably make respectively positive and negative consequences contingent upon secure and insecure behaviors. User studies show that, consistently with the model, SCAs effectively improve users' security behavior. Moreover, additional empirical evaluations show that, when using specific schedules and stimuli, improvements obtained with our techniques are resistant to extinction.

Fourth, we contribute vicarious security conditioning (VSC), a novel technique that can accelerate the attainment of benefits of SCAs and their user acceptance. We implement two types of VSC interventions: vicarious security reinforcement and vicarious insecurity punishment, to condition users to behave more securely without having to be respectively directly reinforced or punished as with SCAs. User studies confirm the stated advantages of our VSC interventions.

Finally, we demonstrate that our proposed security warnings, conditioning applications, and vicarious interventions do not negatively impact productivity since they neither reduce number of production tasks completed by users nor increase the time needed to complete them.

## 1.4 OUTLINE

This first chapter has provided an introduction to the dissertation research. The rest of the dissertation is organized as follows. Chapters 2 and 3 describe respectively theoretical background and related work. Chapter 4 discusses the deficiencies in existing security dialogs, describes how such dialogs become antecedents that cue insecure behaviors, and present our proposed solutions. In chapters 5 and 6 we focus on manipulating the consequences of secure and insecure behaviors for respectively strengthening and weakening them. Chapter 7 describes and evaluates the use of vicarious conditioning to complement the techniques presented in chapters 4-6. Chapter 8 reviews and summarizes the findings of this dissertation. Chapter 9 suggests future research that could extend the work of this dissertation.

# 2.0 THEORETICAL BACKGROUND

This chapter reviews frameworks for analyzing and changing human behavior. Section 2.1 presents an overview of operant conditioning, a theory of human behavior. This dissertation will show that operant conditioning can be used to increase the frequency of secure behaviors and decrease the incidence of insecure ones. However, operant conditioning may not always be acceptable or efficient. Section 2.2 discusses a complementary theory: observational learning. According to observational learning, behavior can be acquired not only as described by operant conditioning, but also vicariously. This happens when people observe other people's behaviors and the consequences they have. The present dissertation evaluates the efficiency and acceptability of observational learning and compares it to that of operant conditioning.

## 2.1 OPERANT CONDITIONING

Operant conditioning was introduced by B.F. Skinner [19]. He defines an operant as a type of behavior that occurs spontaneously, i.e., without being *caused* by stimuli located in the environment [19]. The term stresses the fact that the behavior operates on the environment to generate consequences [18]. Operant conditioning includes several methods to change operant behavior.

By Skinner's law of conditioning of operants, the strength of an operant is increased when its occurrence is followed by the presentation of a reinforcing stimulus (positive "reinforcer”) [19]. A reinforcer can instead be negative, in which case it involves the withdrawal of an aversive stimulus [18]. In both cases, the reinforcer strengthens the preceding operant. Positive reinforcers are more common, and are usually associated with the lay term “reward”. A reward can be properly considered a positive reinforcer only if it increases the strength of the operant upon which it is made contingent. According to Skinner’s law of extinction of operants, when no (positive) reinforcer is presented (or no negative reinforcer is withdrawn) after emission(s) of an already strengthened operant, such operant’s strength is reduced [19]. An operant’s strength is measured by observing the rate at which it is emitted [19][15]. Skinner argued that such a rate of operant emission “comes close to our preconception of the learning process. As the organism learns, the rate rises.” [15].

When or how often a behavior is reinforced is determined by the schedule of reinforcement used. If a behavior is reinforced each time it is emitted, the schedule is called continuous. However, other schedules, known as intermittent, are possible [18]. The simplest are (1) fixed ratio schedule, (2) variable ratio schedule, (3) fixed interval schedule, and (4) variable interval schedule [18][23]. In a fixed ratio schedule, every *nth* (e.g., third; fifth) behavior emission is reinforced [23]. In a variable ratio schedule, every *nth* emission is reinforced *on average*, but the gap between two reinforced emissions of a behavior can be small or large [23]. In the fixed interval schedule, the first behavior emitted after an interval of time (e.g., 90 seconds) is reinforced [23]. In a variable interval schedule, the first behavior emitted after a certain interval of time (e.g., 90 seconds) is reinforced, but the actual interval is sometimes larger and sometimes shorter than that [23]. Numerous experiments have shown that behaviors reinforced intermittently extinguish more slowly than those reinforced continuously [18]. It is also possible to combine two or more schedules [38]. In many real life situations it is more appropriate to use a combined schedule instead of a simple one. For instance, if a salesperson is paid partly on salary (fixed interval) and partly on commission (fixed ratio), the combination can correct the abulia that might follow after reinforcement at a high ratio [18].

Operant behavior can occur in the absence of any correlated stimulus [19]. However, if a stimulus antecedes and accompanies an operant every time it is subsequently reinforced, then such stimulus, when present, can increase the probability of occurrence of the operant. This so-called *discriminative stimulus* does not elicit an operant [38], rather, it merely cues it [99]. Consequently, if the environment is manipulated such that the discriminative stimuli of a certain behavior are no longer present, such behavior will be less likely to occur [99].

Punishment is another effective technique in operant conditioning for changing behavior [67], and can be positive or negative [18]. Positive punishment involves presenting an aversive stimulus after an operant occurrence, while negative punishment consists in withdrawing a positive reinforcer [67][18][36]. With appropriate procedural parameters, undesired behaviors can be

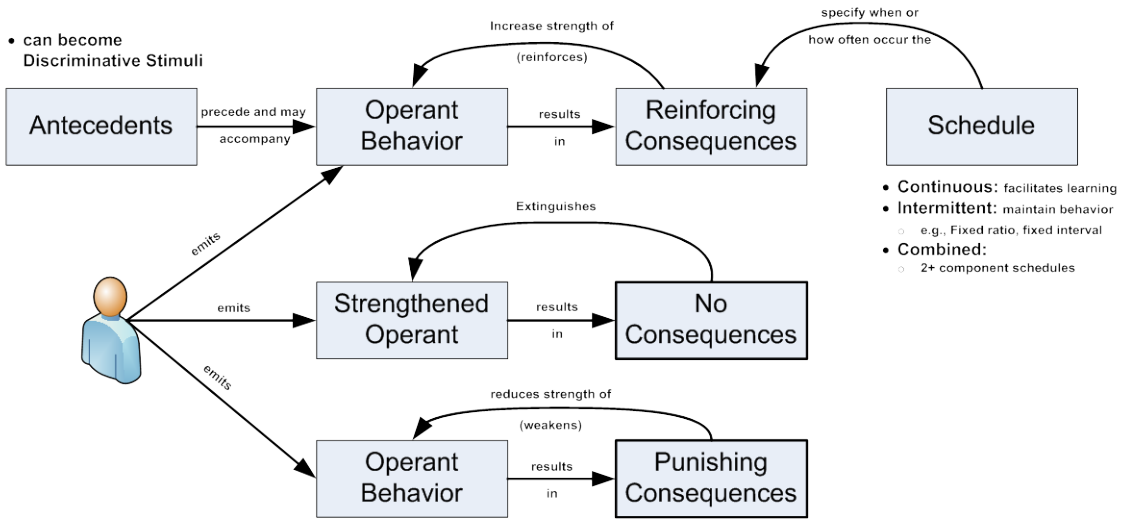


**Figure 2.1**: Operant Conditioning Concepts

nearly totally suppressed in as few as one or two trials [67]. For instance, ensuring little or no delay between punishment and the target behavior, and disallowing unauthorized escape from punishment are two conditions that maximize punishment's effectiveness [123].

Figure 2.1 illustrates the Operant Conditioning concepts presented so far.

### 2.1.1 APPLICATION TO SECURITY BEHAVIORS

In this dissertation, we explore the application of operant conditioning (OC) methods to two types of security related human end-users' behaviors (henceforth referred to simply as security behaviors): secure and insecure behaviors. What is considered secure and insecure behavior in an organization depends on the security policy used in that organization. A security policy is defined as "a statement of what is, and what is not, allowed" [66] to do by users. However, there are factors that may prevent a security policy from being completely enforceable in an automatic manner. First, policies are rarely expressed in a precise way (e.g., mathematically, as a list of allowed and disallowed system states [66]). Instead they normally describe in natural language what end-users are allowed to do [66]. Second, the security mechanism trying to enforce a policy (or part of it) may be unable to determine the inputs necessary to determine a secure course of action consistent with the policy. Therefore, the mechanism may need to rely on end-users feeding it with truthful inputs.

The ambiguity associated with using non-precise ways of expressing a policy and the reliance on end-users to provide truthful inputs may result in behaviors not always in compliance with a security policy. On the one hand, users may intentionally refuse to provide the aforementioned inputs. On the other hand, even if a user is motivated to try to comply with his organization's security policy, there is a risk that the inputs that such user provides to a mechanism may lead to insecure states. For example, a user may receive an email with a file in Word format attached that appears to have been sent by a known co-worker. Given that Word files can contain malicious code, opening the attachment carries the risk that the user's computer may get compromised if the file is infected (e.g., with a very recent virus strain not yet detected by antivirus software). If the infection were certain to occur then opening the attachment wouldn't be just a risk, it would be a security problem [24]. On the contrary, if it were impossible for an infection to occur (e.g., if Microsoft Word were not installed in the users' PC), then there wouldn't be any risk associated with opening the attachment (e.g., the user may instruct Windows to open the attachment with some other, less vulnerable, editor).

One way for the user to try to mitigate these risks is by attempting to follow his organization's security policy as faithfully as possible. In those cases where users has made such attempt we say that taking (or accepting) a security risk is *justified*. Otherwise, accepting a security risk is *unjustified*. Note, however, that risks considered justified under one policy may be considered unjustified under another [66]. A very lax security policy may allow a user to accept any risk. On the contrary, some policies may seem to be so strict to users that they may decide to reject any risk. The latter may be undesirable if it causes users to not take (i.e., reject) even those risks that are necessary for completing their primary work. To avoid the case where users attempt to behave securely simply by rejecting any risk, we define secure behavior as not only rejection of unjustified risks but also acceptance of justified risks. Insecure behaviors are accordingly defined as acceptance of unjustified risks and rejection of justified risks. In this dissertation, the policy we use is that a risk may be accepted only if (i) it is necessary to do a user's primary productive tasks, (ii) there are no other, less risky, alternatives to accomplish such tasks, and (iii) there are no available means to mitigate the risk. For example, consider the case of email:

- An unjustified risk (UR) may be an email message containing an attachment that is unexpected, from an unknown sender, unnecessary to the user's job-related tasks, of a type that may spread infections (e.g., .exe), or some combination of these factors. In this case, the user may mitigate the risk by asking the sender to either retransmit the attachment in a less risky file format (e.g., .txt) or include the contents of the attachment in the message body.
- A justified risk (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.

Both secure and insecure behaviors are operants.

Traditionally, a company's employees are held accountable for getting their production tasks done, but not for whether they behave securely while interacting with the computer applications needed to complete these tasks. Users typically view these applications as tools used to achieve their primary goals. Securely using computer applications 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. Figure 2.2 presents a model of the end-users' security-related behaviors just described.

OC also suggests that several types of interventions can be used to change computer users' behaviors. First, if application developers eliminate the antecedents of insecure behaviors, those behaviors would be less likely to occur. Second, if application developers enable computer systems' administrators to make positive consequences contingent upon users' secure behaviors, they can strengthen such behaviors. Third, if application developers enable computer systems' administrators to make adverse consequences contingent on users' insecure behaviors, they can weaken such behaviors' strength. Chapters 4, 5, and 6 respectively explore and evaluate these three approaches to change end-users' security behaviors when users are interacting with com-

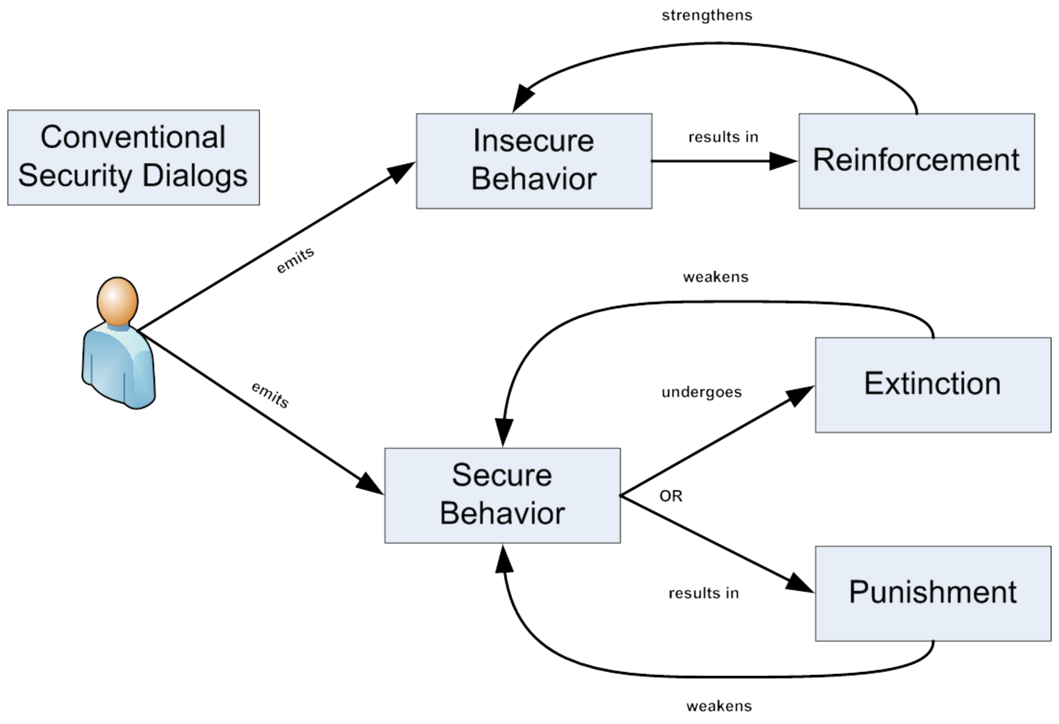


**Figure 2.2:** Model of computer users' current security behaviors

puter applications necessary to complete their primary tasks.

An issue regarding the applicability of OC methods in the context of end-user security is the way in which users need to learn how to discriminate between justified and unjustified risks. OC suggests that users learn the associations between a response and either a reinforcing stimulus (reinforcement) or a punishing stimulus when these are made contingent upon the behavior. That may not be the most efficient and user-acceptable way to promote such learning. Articulating and conveying to the user the rules for discriminating between types of risk may speed up learning and improve user attitudes toward the interventions. One possible efficient way of conveying these rules is the use of observational learning, which considers the role not only of OC procedures, but also that of vicarious processes, as described next.

## 2.2 OBSERVATIONAL LEARNING

Observational learning (OL), also known as modeling or vicarious learning, can be defined as learning happening as a result of observing the behavior of another person (called “model”). The presentation of a model’s behavior to an observer may conclude with (1) reinforcement if the behavior is desired, (2) punishment if the behavior is undesired, or (3) neither reinforcement nor punishment. Albert Bandura, who introduced OL, reported that learning occurs in subjects observing models in all those three cases [4][2]. Several studies have shown that this type of learning is as effective as learning that involves actually performing a specific behavior [4][118][33][78].

Modeling differs from operant conditioning in that, for learning to occur, a person (an “observer”) does not need to experience a situation where behavior is emitted and subsequently reinforced or punished. According to Bandura, observational learning may be more efficient than operant conditioning in many cases: "learning would be exceedingly laborious … if people had to rely solely on the effects of their own actions to inform them what to do" [4]. It may also be less error-prone: "because people can learn from example what to do, at least in approximate form, before performing any behavior, they are spared needless errors" [4]. As with operant conditioning, behaviors learned using modeling can be maintained using several schedules of reinforcement.

Observational learning is governed by four sub-processes: attention, retention, reproduction and motivation [4], as described next.

- **Attention.** A person must first pay attention to a model engaging in a certain behavior. This is determined by several factors including models' "interpersonal attraction", since "models who possess engaging qualities are sought out, while those lacking pleasing characteristics are ignored or rejected" [4].
- **Retention**. Once attending to the observed behavior, the observer must be able to effectively remember what the model has done. This is important, as explained next. First, users need to remember that many behaviors which are punished should not be reproduced. Second, in the absence of the model, they have to recall the behavior that will lead to reinforcement.
- **Reproduction**. The observer must be capable of replicating the behavior being observed. When deficiencies in ability exists, "the basic subskills for complex performances must first be developed by modeling and practice" [4].
- **Motivation**. People do not enact every behavior they learn but they "are more likely to adopt modeled behavior if it results in outcomes that they value than if it has unrewarding or punishing effects" [4]. Reinforcement can be a source of motivation to reproduce a learned be-

havior in the following ways: (1) replication of the modeled behavior results in desired rewards for the observer, (2) the observer sees the model receive desirable rewards for performing the target behavior, and (3) the observer finds the performance of the target behavior self-rewarding [4][33]. On the contrary, observed negative consequences (e.g., punishment) reduce the tendency to enact a model's behavior [4].

### 2.2.1 APPLICATION TO SECURITY BEHAVIORS

Modeling has been successfully applied to elicit behavior related to production activities [4][33][91], and to raise awareness about other, non-primary, aspects of the work life (e.g., sexual harassment awareness [120]). In this dissertation we explore its application to security related activities, and evaluate its efficiency and acceptability when used to not only convey rules necessary for discriminating between types of security risks, but also to encourage people to apply them. This carries several benefits. On the one hand, people do not have to be put into risky security situations for them to learn (1) how to discriminate between types of security risks, and (2) what the desirable consequences of behaving securely are (e.g., reception of rewards). This can save time compared to a pure OC intervention. On the other hand, people do not need to actually be punished to learn about the consequences of insecure behaviors. In that sense, modeling may be more acceptable than OC.

There are some considerations for using observational learning to induce enactment of secure behaviors in computer users, such as:

- The characteristics that models must possess for observers to be inclined to imitate the models' secure behavior.
- The medium used to display modeling (e.g., live performance, videos).

- The rewards that would be more effective for inducing users to reproduce secure behaviors.
- The punishments that would be effective without being excessively aversive.

Relevant literature (e.g., [4][33][91]) provides information about what strategies, with respect to the above considerations, have worked to ensure enactment of behaviors related to production activities. However, security is not a production, but a supporting or enabling task [74][102]. As such, the present research is needed to evaluate the applicability of modeling to facilitate users' enactment of secure behaviors when they interact with computer applications necessary to do their primary tasks. In chapter 7 we describe the design and evaluation of two interventions that we created based on the observational learning principles just described.

# 3.0 RELATED WORK

This chapter presents and discusses related work. In section 3.1 we review research documenting insecure behaviors of users when interacting with computer security mechanisms. Sections 3.2-3.4 review approaches that have been proposed to diminish the frequency of such behaviors, respectively warning, educating (or training), and conditioning users. Section 3.4 summarizes the chapter's contents.

## 3.1 INSECURE BEHAVIORS

The need to account for human actions in the security loop has been acknowledged very early by the computer-security research community. For instance, in their influential 1975 paper [52], Saltzer and Schroeder identified the principle of *psychological acceptability*, which states that it is essential that user interfaces "be designed for ease of use" so that humans are able to "apply the protection mechanisms correctly". However, until recently, little work was done to make security technology more usable and friendly to humans [76]. This situation has started to change in no small part due to the growing number of security risks that end-users face when operating computer systems connected to the Internet. With the generalized use of the Internet today, an individual's insecure behavior can have severe negative consequences to his organization, including financial losses, unintended release of private information, or an inability to operate normally in everyday activities. In spite of such potential consequences, users continue to behave insecurely. As evidence of this, recent surveys (2009) cite human failures as either the main cause of security breaches [32] or one of the most important contributors [27] to the occurrence of such breaches.

In some cases, users are not purposely emitting these behaviors. For instance, in their seminal paper, Whitten and Tygar [7] document a study where only one-third of twelve participants were able to adequately encrypt email communications using PGP 5.0 even when the participants made honest efforts to do so. The researchers concluded that, although the user-interface model used by that software was attractive, it was not enough to make security usable, and that correct conceptual models of security need to be communicated to users. In another study [28], Balfanz and Grinter document their attempt to set up a public key infrastructure (PKI). The study's subjects had advanced degrees (including PhDs in Computer Science and related disciplines) and were willing to finish the assigned task with an understanding of the protection they were gaining. However, it was reported that, along the way, users found the task very complex and finally resorted to follow instructions mechanically without comprehending what they were doing. Part of the problem was that the user-interface was poorly designed and required users to make numerous complicated decisions.

In many more instances, however, users deliberately circumvent security mechanisms or learn to automatically ignore them, even if the mechanism provides adequate protection when receiving truthful inputs. Several researchers have documented this behavior and we mention some of their findings. DeWitt and Kuljis [8] evaluated the usability of a protection mechanism called Polaris, which used the *principle of least authority* (POLA) to prevent malware from damaging users' files. They found that despite having the product documentation at hand, and knowing how to use the user-interface of Polaris to operate programs more securely, users chose not to do it and were willing to perform potentially harmful actions, thus compromising security. During debriefing, participants stated that they did so to complete their primary tasks sooner. Zurko et al. [77] evaluated the security behavior of users of Lotus Notes when confronting a security warning. The tested warning dialog asked users whether they wanted to allow unsigned code, which they received in an email message, to execute. The researchers conducted a field study in a software development company and reported that 59% of users who saw the warning allowed the code to run. This happened despite the fact that a trustworthy co-worker (a company's security maven) had asked these (and other) users to apply a set of security settings that would have automatically disallowed the execution of the code without any warning. The researchers predicted that "the more frequently security warnings appear in everyday use, the more users will learn to click 'OK' without thinking or even remembering that they have done so" [77]. Nodder [22] corroborated such prediction when, after doing numerous usability tests at Microsoft Research, observed that users have the tendency to simply dismiss any dialog that gets in their way, and that they seldom read the main text shown in computer warnings [22, pp. 594-595]. Wu et al. [73] conducted a user study to evaluate the effectiveness of anti-phishing toolbars in preventing users from getting conned by interacting with spoofed websites. Before starting, all the participants in the study were informed that the study's main focus was security. However, several subjects decided, after noticing and understanding the warning shown by the toolbars, not to heed such warnings. Users explained that this was because they either wanted to complete the tasks of the study they deemed as "primary", or decided that certain aspects of the website were more worthy of their attention than the toolbar's warning.

All these studies show that, even if a security solution can work flawlessly from a technical perspective, it can be sabotaged by users who refuse to provide accurate inputs. This results in an overall insecure state of computer systems. In a few cases this is not done on purpose, though, in many cases, computer users are willing to intentionally cut corners to complete the job that they consider to be primary [74]. For instance, users feed false inputs to security mechanisms even if a friendly user interface is provided [74]. In some cases, the latter happens unconsciously as users have become increasingly habituated to ignore warnings and select a default choice [22]. In the rest of this chapter we take a look at what approaches have been researched to minimize the occurrence of insecure behaviors.

## 3.2 WARNING USERS

When computer programs cannot automatically make a security decision, they typically warn users and ask them how to proceed. However, researchers have found that there are several pitfalls associated with security warnings, as described next. First, security warnings that resemble computer dialogs shown for low- or no- security risk conditions (e.g., connection problems) are usually not heeded by users [49][103]. It has been pointed out (e.g., [77][22]) that users become habituated to ignore such notifications and pick the option that allows them to proceed with their primary task. Second, conventional security warnings do not offer users suitable options to overcome the security risk, but merely ask users whether they want to proceed or not [39]. Third, the wording used in many security warnings includes concepts that non-technical users find hard to understand [22]. Fourth, some warnings do not explain the consequences of not heeding them.

We review approaches that have attempted to address such concerns and make security warnings more useful and effective.

Xia and Brustoloni [39] proposed hardening web browsers against man-in-the-middle (MITM) and eavesdropping attacks through techniques that used security warnings and dialogs. Their warnings used “plain language” that “non-specialists” could understand. The first technique, Specific Password Warnings (SPW), showed a sequence of warnings when users were about to submit passwords over an unencrypted connection. SPW explains the dangers of sending passwords unencrypted, and strongly discourages users to continue. Nonetheless, the user is allowed to proceed if she so chooses. This approach was dubbed “Guidance with override” (G+O). The second technique, Context-Sensitive Certificate Verification (CSCV), shows a sequence of warnings every time the user navigates to a website with a certificate that the browser is unable to verify because the public key of the certificate issuer is unknown. A dialog first asks users if they have the certificate issuer’s public key in removable media. If users do not have it, CSCV shows unverified contact information, included in the site’s certificate, necessary to acquire the CA’s certificate. Moreover, if the user self-reports that she is not a member of the organization that owns the website, CSCV warns her that the situation is unusual and risky (a potential MITM attack). CSCV does not allow, in any case, the user to proceed to the website until she acquires, out-of-band, the necessary tokens. This approach was dubbed “Guidance without override” (G–O). In a user study with 17 computer science students, Xia and Brustoloni found that, compared with the performance of a control condition where users interacted with an unmodified web browser (Internet Explorer 6.0), users behaved more securely with SPW and CSCV.

There are a few issues associated with CSCV and SPW. CSCV does not allow users to proceed until they eliminate conditions of a particular risk. In the situation in which CSCV was evaluated (forcing internal members of an organization to acquire the organization's root certificate out-of-band), G–O may be a reasonable approach, as the software can tell what the users need to do to proceed securely. However, in many other situations where the software cannot tell, G–O may be too restrictive, and prevent users' from completing their tasks, ultimately leading to users' irritation. On the other hand, it is more reasonable to ask users to accept dealing with a mechanism, such as SPW, that implements G+O since it allows override. Nonetheless, because the users in the SPW study only had to make a simple security decision for one single site and presumably only one time, it is uncertain how applicable to more complex decisions and how resistant to habituation SPW is. Moreover, SPW was tested with computer science students, who are not "non-specialists" and may not behave the same as the general population [1].

Egelman et al. [103] conducted a user study to evaluate the effectiveness of anti-phishing warnings in Microsoft Internet Explorer 7 (IE7) and Mozilla Firefox 2.0 (FF2). They classified a warning as *active* if it interrupted the user's primary task and as *passive* if it did not. An active warnings is shown for websites that are included on a blacklist, while a passive warning (only displayed by IE7) is only shown when the browser deems a site suspicious but has not yet completely verified it as fraudulent. They found that their participants tended to ignore passive warnings, as there was no significant difference in the number of users who entered information in a forged website when a passive warning was shown and when no warning (neither passive nor active) was displayed at all (control group). On the contrary, it was more likely that participants heeded active warnings. However, although IE7's active warning dialog was significantly more effective than IE7's passive warning, it was significantly less effective than FF2's active warning. None of the participants who saw FF2's dialog got phished but 45% of IE7's active warnings did. The researchers then analyzed all the warnings using principles from warning sciences [61] and concluded that the FF2's active warning's appearance, which distinguished it from less severe warnings, was the reason why users did not get habituated to it in the experiment. On the contrary, the active warnings in IE7 resembled other less severe errors that users in the experiment had become accustomed to routinely ignore.

We consider the latter results interesting, but point out some limitations. First, one group was already familiar with the warnings tested (IE7) while the other was not (80% of users in the FF2 group had not seen the anti-phishing warning before). Thus, the novelty of the warnings might have played a role in the results. Moreover, during the experiments the users in the FF2 group were exposed to that warning just two times. That may not be enough to unequivocally judge whether FF2's active warning's characteristics make it really resistant to habituation. As its novelty wears down, its effectiveness may decrease, but how much or if at all is not known. Second, FF2's and IE7' active warnings are only shown when a site has been deemed fraudulent with some certainty. This leaves users unprotected against fraudulent websites that have not been yet incorporated into a browser's blacklist. A user accustomed to being warned when visiting fraudulent websites could easily fall for the latter websites. In essence, the warnings based on blacklists cue users that they should behave securely, but in the absence of such warnings such behavior is less likely to occur as predicted by Operant Conditioning.

To assess the likelihood of users visiting fraudulent, but not yet blacklisted, websites, Sheng et al. [107] conducted during four days in December 2008, a study of several blacklists of phishing websites, including those used by Firefox v2 and v3 (FF3)'s anti-phishing warnings. They found that (a) at "hour 0" (i.e., when they initially started checking phishing websites), FF2's and FF3's anti-phishing filter missed on average about 60% of fresh phishing websites the researchers had in their phishing feed, (b) at hour 1 it missed 30%, and (c) 44% of the phishing campaigns last only two hours [107]. This means that it is likely that users visiting non-blacklisted forged websites will fall for the attack when no warning is shown. Ultimately, simply warning users when it is likely that the security mechanism is correct (e.g., FF2's active warnings) may not be enough, and other measures to ensure enactment of secure behavior need to be employed. Educating users and conditioning their behaviors are two measures that have been proposed for such purpose and we review them in subsequent sections.

In a follow up study to the anti-phishing warnings experiment, Sunshine et al. [49], evaluated the effectiveness of three current (FF2's, FF3's, and IE7's), and two proposed warnings used for Secure Sockets Layer (SSL)-related failures in web browsers. According to the researchers, the latter were designed using principles from warning sciences. One of such warnings consisted of a single page whose appearance conveyed a high-risk security situation (white text over a dark red background). The user could proceed past this warning to the website only by clicking on a difficult to see hyperlink at the bottom of the page. The other warning consisted of two pages. The first page asked whether the website she was visiting was either, (a) a bank/financial institution, (b) an online store/e-commerce website, (c) other, and (d) unknown. The second page was the same as the single-page warning, and was only shown when the participant selected one of the first two choices in the previous page. In a laboratory study, they evaluated how likely users were to heed all five SSL warnings when visiting a bank website, and a university's library website. They found that participants, when visiting the bank website, were more likely to heed their two proposed warnings (55%, and 40% respectively for the single- and multi-page warnings) and the FF3's warning (45%) than the FF2's and IE7's warnings. They also found that, except for the FF3's warning, all other warnings were equally ignored in order to proceed to the library's site. The authors attributed the results obtained with their warnings to awareness they raised in users, and that they communicated clearly what users could do to mitigate the risk. They also note that the results related to FF3's warning could be due to the complexity of overriding it.

There are several points to note with regard to the latter study. First, results of an exit survey indicated that the color and text of the proposed warnings was not significantly influencing users to make a decision compared to existing warnings. This is interesting, as the results of the anti-phishing warnings study suggested that such features (which can be used to convey the degree of risk faced) could be crucial factors in the effectiveness of the FF2's anti-phishing warning. However, users' answers to surveys are not necessarily reliable, so this may simply be not the case. Second, participants in their study were all "CMU faculty staff or students", and thus they might have been already familiarized with the library's website and corresponding error, and knew the site was legitimate. They could have decided to proceed based on such knowledge rather than the warning's characteristics in the single-page dialog, or, to a lesser degree, to the options in the multi-page dialog (to which 75% of users provided the correct choice, "other"). Thus, participants who heeded the warning when trying to visit the website may have done so simply because they were more alarmed, unlike when they saw the warning for a known website. Second, in the FF3, FF2, and IE7 conditions, respectively 40%, 35%, and 50% of users claimed to have seen the warning before (when asked about their visit to the bank website). On the contrary, in the proposed multi-page, and single-page conditions, respectively 5%, and 20% of users claimed to have seen the warning before. Since the latter cannot be possible, users probably confused them with similar dialogs (e.g., the first of the page of the multi-page warning is similar to the “page not found” message used by Firefox). Hence, the novelty factor could have been paramount on their proposed warning success (when shown in the bank website), and since users were exposed to the warnings only twice, its resistance to habituation may not have been unmistakably demonstrated.

In Chapter 4 we propose and evaluate security warnings designed using principles from operant conditioning, to help users follow a secure course of action in the context of email communications. Our results do not suggest that the effectiveness of our warnings decreases over time when users are managing up to eight unjustified risks, well after the warning’s novelty may no longer play a role. While it is not possible to tell what number of exposures to a warning is a definitive indicator to its resistance to habituation, we think that our experiments can be a more realistic benchmark than the studies presented in this section. The latter tested the effectiveness of warnings only once or twice.

## 3.3 EDUCATING USERS

In this section, we review research literature about the effectiveness of security education for improving users’ security behaviors. First, we present results from the literature that suggest that, in general, security education does not work, but point out problems with their conclusions. Second, we review studies finding that security education can be effective when designed according to principles of learning sciences. Third, we describe research that indicates that users who have received proper training may not necessarily apply it. Finally, we present our conclusions.

Anandpara et al. [113] evaluated the effectiveness of anti-phishing educational material provided by the Federal Trade Commission (FTC). Participants in their study took a test before and after reading such material to measure their "Phishing IQ". The researchers found that the number of stimuli misclassified as phishing was substantially large in the post-test compared to that of the pre-test. They concluded that the educational material simply made the participants more cautious rather than improving their judgment.

Jackson et al. [21] observed a similar phenomenon as Anandpara et al. when conducting a user study to evaluate extended certificate indicators in Internet Explorer 7 (IE7). They performed a user study with three groups of users. In one group, users read the IE7 help manual that explained the features of extended validation and phishing filter (trained group), in another group users did not read such file but saw the IE7 security indicators (untrained group), and in a control group, users did not interact with the extended certification features. In their experiments they found that participants in the trained group were more likely to classify both legitimate and fake websites as legitimate.

Kumaraguru et al. [89] disputed that the results in [21] and [113] were evidence that user education for security issues is not effective. They noted that in the "Phishing IQ" experiment [113], the instructional material was ineffective because it was not specific to phishing but to online fraud in general. They also note that in the IE7 extended certificate indicators experiment [21], the help file did not provide specific clues to distinguishing phishing from legitimate websites [60]. Thus, Kumaraguru et al.'s conclusion was that the ineffectiveness of inadequate instructional materials does not imply that security education does not work. They then devised training (the terms security education and security training are used interchangeably in their work) interventions that were more successful. They designed and evaluated Phishguru, an "embedded-training" intervention in a comic-strip format to combat phishing attacks [89]. Embedded training is a methodology in which educational materials are integrated as part of users' primary tasks. In their experiments, users were divided into three groups, depending on the training materials provided. One group received no training (control group), another was provided with "non-embedded" training (existing training materials sent by email), and the other received embedded training (displaying a comic strip intervention only when the user clicked on a link to a phishing website in an email message). Their main finding was that participants in the embedded training condition showed a significantly better performance than those in the other groups with respect to correctly identifying phishing emails during a first session and another occurring one week after.

Sheng et al. also used principles of learning sciences to design a computer game, "Anti-phishing Phil", to educate users about phishing and how to avoid falling for it [106]. Their game was claimed to allow a user to acquire conceptual (declarative) and procedural knowledge about phishing and to use them both for recognizing spoofed websites in an interactive way. They evaluated whether three groups of people were able to better distinguish phishing from legitimate websites before and after respectively (i) interacting with the computer game, (ii) reading the conceptual knowledge of the game presented in color-printed form, and (iii) reading traditional educational material. They found a non-significant reduction in false negatives (phishing websites *not* classified as such) from pre- to post-tests in all three groups. On the other hand, they found that the number of false positives (legitimate websites erroneously identified as phishing websites) increased in the third group but decreased in the others, with the game showing the biggest (and statistically significant) reduction.

The work by Sheng et al. and Kumaraguru et al. just described demonstrated that it is possible to design interventions that allow users to effectively acquire and retain knowledge about security issues. However, other researchers have found that, even when users possess security knowledge, that does not ensure they will use it, as detailed next. Dhamija and Perrig [94] discovered, in an experiment to evaluate the effectiveness of password mechanisms that people who had received training about how to create strong passwords, decided to choose trivial ones. Apparently they preferred convenience over security, and were not willing to do the extra effort to create a secure password. Sasse et al. [75] performed both a field study in a large UK company, and interviews with users (half of which were also employees of that company), to analyze their security behaviors. They found that users were not very concerned about the security of their organization's computer systems despite the fact they were trained in security procedures [75]. Often they "choose to follow existing policies only selectively or not at all" [75]. Interviewees also stated that they believed that if their insecure behaviors caused a security breach, they would not be held accountable. However, all users who participated in the interviews reported that they would behave more securely if their personal information (e.g., their health records, payroll information, and personal email) could be compromised by their insecure actions. Interestingly, though, later research suggests that sometimes not even in cases where personal assets are at risk users will necessarily be inclined to acquire security knowledge or to use it. For instance, in [119], Herley concludes that, in cases where the losses for falling for phishing attacks are covered by a user's financial institution, users may not find it worthy to spend time or effort in receive or apply anti-phishing education.

To summarize, the results of these studies suggest that while it is not accurate to conclude that security education is ineffective based on results obtained using inadequate educational materials, and that training interventions designed properly can help users acquire knowledge about how to behave more securely, security education does not guarantee that users will put acquired knowledge into practice. Basically, it is up to the user to decide whether to do so, and in many cases, as described earlier, they will not [75][94]. In the case of the effective anti-phishing training interventions described above (Phishguru for instance), they not only provide clues on how to identify phishing, but also make it clear the consequences of falling for such attacks (e.g., that the user's "identity" could be stolen). That may give users the necessary motivation to apply their acquired knowledge since it can be used, after the experiments, to protect *personal* information or assets that they consider important (as noted by Sasse et al. [75]). Whether users will make the same effort of applying their security knowledge for protecting others' (e.g. their company's) assets, especially if they will not be held accountable, is a different issue [75][30]. In the final analysis, training users may not be enough to guarantee secure behavior in many circumstances, and needs to be coupled with, e.g., monitoring and reinforcement, to encourage users to apply their knowledge [2]. Moreover, in case of willful non-compliance to security policies, holding trained users accountable for their insecure actions could be a reasonable measure in certain environments (as suggested in [74][75]).

To address the concerns stated above, we evaluate, in Chapter 7, two interventions, designed following the theory of observational learning, that not only provide clues on how to identify potentially dangerous email communications, but also encourage users to apply such knowledge, and discourage users to behave insecurely when managing their employer-assigned email accounts.

## 3.4 CONDITIONING USERS' BEHAVIOR

In this section we review two studies that make opposing claims regarding the effectiveness of conditioning users' behaviors. One suggests that measures to condition users' behavior may be effective. The other suggests that attempting to manipulate consequences of behaviors in order to condition users would be unsuccessful. However, neither study evaluated their claims satisfactorily.

Gonzalez and Sawicka created, using causal diagrams, theoretical models of compliance to computer security measures according to operant conditioning principles, such as reinforcement and extinction [53][55][54][56]. A central tenet of their models is that risk *perception* regulates security compliance, i.e., a person's perception of risk is updated by security incidents: their occurrence increases risk perception whereas their absence decreases it. In the first case the person conditions to comply with security measures and the reinforcement is "the well-being associated with feeling protected from external risks". In the latter case the prolonged absence of security problems extinguishes compliance. In their models, the conditioning period/zone tends to last less than does the extinction period. This difference is attributed to two elements. On the one hand, conditioned operant behaviors last longer if the schedule of reinforcement is not continuous. They argue that this is the case in modern work milieus where several demands and time pressures may interfere with the delivery of reinforcement (awareness of averted risk according to them). On the other hand, the effectiveness of modern security mechanisms causes noncompliance to security policies to occur for long periods of time without negative consequences. These extinction periods facilitate "superstitious learning" (i.e., making incorrect inferences about risk, its consequences, and the impact of non-compliance). This alternation of conditioning and extinction periods (influenced by the fluctuations in the frequency of security incidents), may continue indefinitely. Gonzalez and Sawicka hypothesized that, to prevent computer systems to become too vulnerable and to avert superstitious learning, organizations could implement "risk perception renewals" (through, e.g., training, publications, seminars, and reminders) at the start of periods of decaying risk perception (extinction zone) [54]. However, they did not provide details about the implementation of such measures, so their utility and effectiveness remain unproven. For example, it is possible that people may be alarmed initially as a consequence of exposure to such measures, but that over time the absence of security incidents will cause their efficacy to wane (the "cry wolf" effect). To counteract this, they also propose that security incidents, even small ones, be documented and shared across an organization (i.e., the "wolf" did arrive, even if no impact was sensed by specific portions of the workforce). Nonetheless, no empirical data was provided to validate the effectiveness of such measure, and the proposed countermeasures' *actual* impact on productivity is unknown. Thus, user studies are required to answer questions about the efficacy and side-effects of their proposed solutions.

Pahnila et al. proposed a theoretical model containing several factors that could impact *intention* to comply with, *actual* compliance to, and *attitudes* toward complying with information systems (IS)'s security policies [105]. Their model was based on principles from several sources such as General Deterrence Theory, and behavioral frameworks. Some of the factors in their model were sanctions (punishment), rewards, and information quality. They used a web-based questionnaire to collect data to validate the impact of the factors in their model. Only employees from a Finnish company completed the questionnaire. Their results suggested, among others findings, that sanctions had insignificant effect on intention to comply with IS security policies, and that rewards did not have effect on actual compliance with IS security policies. However, it is well known that the most reliable ways to find whether interventions involving rewards are effective are (i) performing direct tests, (ii) observing what people find reinforcing, and (iii) analyzing historical records of reinforcement [17][33]. Self-reporting (e.g. surveys), in general, should only be used as reference [33][16]. In Nodder's words: "what users say they'll do and what they actually do often differ" [22].

In Chapters 5 and 6 we empirically test whether the operant conditioning techniques of reinforcement and punishment can help improve users' secure behaviors. We found that they can be, indeed, effective.

## 3.5 SUMMARY

In this chapter, we first reviewed existing literature that documents users' insecure behaviors. Afterward we presented and commented on approaches that have taken concepts from warning, learning, and behavioral sciences, to improve users' security behavior. Some approaches are theoretical and/or suggest untested measures [56][105]. The remaining approaches have been empirically evaluated with regard to their applicability and usefulness to improve users' security behavior. However, we have identified some issues that the latter approaches do not convincingly solve. First, security education does not guarantee that users will apply their acquired knowledge about security risks [74]. Second, the reviewed warnings' resistance to habituation the has not been demonstrated convincingly [103][49].

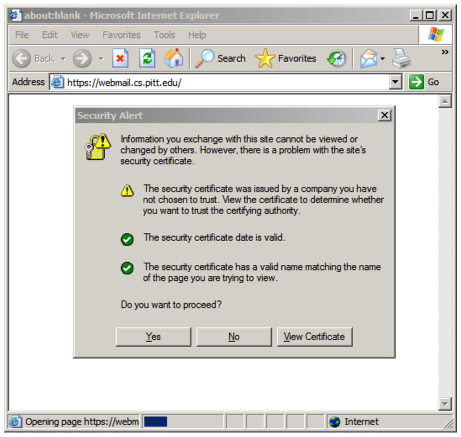
We believe that carefully leveraging principles from behavioral sciences, detailed in Chapter 2, can overcome the limitations of current approaches. We report our findings in subsequent chapters.

# 4.0 CONTEXT-SENSITIVE GUIDING POLYMORPHIC DIALOGS

Conventional security dialogs suffer from several drawbacks. First, many of these dialogs do not provide suitable options to help users take a secure action. Second, these dialogs are designed in such a way that users can easily dismiss them by selecting any, possibly insecure, option in the dialog, so that users can continue with their workflow. To address the former problem, in this chapter we contribute context-sensitive guiding (CSG) dialogs, a novel type of dialogs that can help users behave more in compliance with a security policy. Applications with CSG dialogs ask the user to provide context information relevant to the security decision. Based on such information, these applications then decide or suggest an appropriate course of action. To combat the latter problem, we also contribute polymorphism for security dialogs, which is based on principles of operant conditioning. This technique is used to harden security dialogs, including CSG dialogs, against automatic and false user answers. Polymorphic dialogs continuously change the form of required user inputs and intentionally delay the latter, forcing users to pay attention to security decisions. We implemented context-sensitive guiding polymorphic dialogs, CSG-PD, against email-borne viruses on the Thunderbird email agent. In a user study, we found that users with weak security behaviors accept significantly fewer unjustified risks with CSG-PD than with conventional dialogs. Moreover, CSG-PD had insignificant effect on acceptance of justified risks and time to complete tasks.

## 4.1 INTRODUCTION

Computer applications often need to make context-dependent security decisions. One example of this situation is when a user interacting with an email client receives a message with an attachment of a type that may spread infections. If the user both knows the message's sender and was expecting such attachment then she may decide that the risk inherent in opening the attachment is justified. However, the email program cannot automatically determine such contextual information. Another example is when a user interacting with a web browser navigates to a site with a SSL certificate that the browser is unable to verify (Figure 4.1). The user may be member of the organization that owns the website and that is the certification authority that issued and signed the certificate, but it is often infeasible for software to determine such information automatically. In these situations, an application usually needs user input, because the application cannot determine automatically all the context relevant to the security decision.


about:blank - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address https://webmail.cs.pitt.edu/ Go
Security Alert
Yes No View Certificate
Opening page https://webm
Internet

Information you exchange with this site cannot be viewed or changed by others. However, there is a problem with the site's security certificate.

The security certificate was issued by a company you have not chosen to trust. View the certificate to determine whether you want to trust the certifying authority.

The security certificate date is valid.

The security certificate has a valid name matching the name of the page you are trying to view.

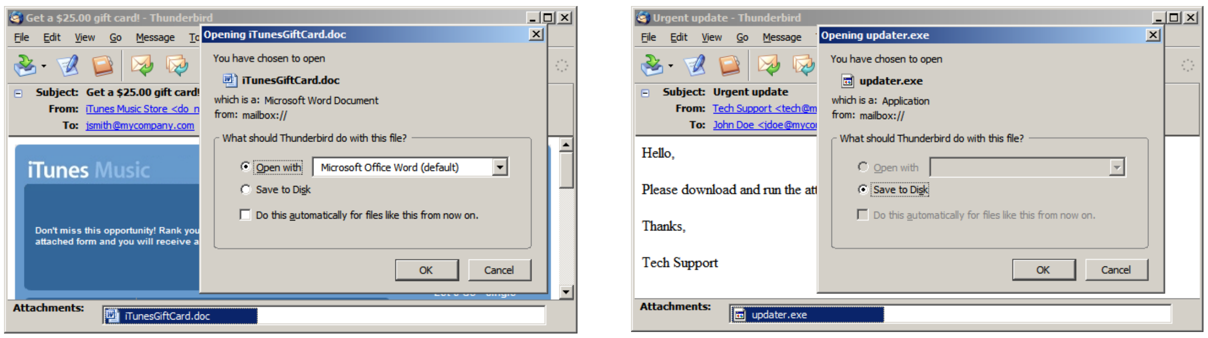
Do you want to proceed?

**Figure 4.1**: Dialog following the warn-and-continue (W&C) approach

Internet Explorer 6.0 shows this warning when a server certificate cannot be verified because the public key of issuing Certification Authority is unknown.

Many applications translate this need into complete delegation of the security decision to the user. On the one hand, some applications warn users of the risk and ask them whether they want to accept it, which is an approach that we call *warn-and-continue* (W&C). Figure 4.1 shows an example of such approach. On the other hand, several applications do not warn users when they are facing a security risky situation. For example, Mozilla Thunderbird v1.5 allows the user to cancel or save an attachment instead of opening it, but does not warn the user if such attachment is unsafe (Figure 4.2). We call this approach *no-warning* (NW).

The approaches just covered are problematic for several reasons. First, warning dialogs often use language that users do not understand, and request from users decisions whose consequences they do not fully appreciate. To many users, these dialogs are meaningless "dilemmas" rather than choices [22][49]. Second, even when such dialogs offer understandable and suitable options, their repeated presentation and lack of any noticeable consequence causes users to start responding to them unthinkingly. Users quickly learn dialog options that allow them to continue on their primary tasks without interruption by the security decision [77]. Thus, users acquire the habit of choosing dialog options that may be insecure and possibly untruthful [98], without ac-


Get a $25.00 gift card! - Thunderbird
File Edit View Go Message To
iTunes Music
OK
Cancel
Urgent update - Thunderbird
File Edit View Go Message
OK
Cancel

**Subject:** Get a $25.00 gift card!
**From:** iTunes Music Store <do_n
**To:** jsmith@mycompany.com

Don't miss this opportunity! Rank you
attached form and you will receive a

**Attachments:** iTunesGiftCard.doc

**Opening iTunesGiftCard.doc**

You have chosen to open
**iTunesGiftCard.doc**
which is a: Microsoft Word Document
from: mailbox://

What should Thunderbird do with this file?

[✓ Open with] Microsoft Office Word (default)
Save to Disk
☐ Do this automatically for files like this from now on.

**Subject:** Urgent update
**From:** Tech Support <tech@m
**To:** John Doe <jdoe@myco

Hello,

Please download and run the at

Thanks,

Tech Support

**Attachments:** updater.exe

**Opening updater.exe**

You have chosen to open
**updater.exe**
which is a: Application
from: mailbox://

What should Thunderbird do with this file?

Open with
[✓ Save to Disk]
☐ Do this automatically for files like this from now on.

**Figure 4.2**: Examples of dialogs following the no-warning (NW) approach

Mozilla Thundebird v1.5 shows this kind of dialogs when a user attempts to download an attachment tually considering other choices that may be more appropriate [22]. Third, as applications with NW rely entirely on external security utilities (such as antivirus software), they neither include safeguards to protect users nor warn them about possible security risks. If security utilities were both always completely accurate in their detection (i.e., their results are only true positives and true negatives) and deployed in every user's computer, then NW could be a very acceptable approach from the users' point of view. However, this is not the case.

To mitigate these problems, we contribute two improvements to conventional dialogs. First, *context-sensitive guiding* (CSG) dialogs ask the user to provide context information necessary for a security decision. Based on such information, CSG dialogs decide or suggest an appropriate course of action. However, if CSG dialogs accept unverified user inputs that enable users to continue, CSG dialogs can become as insecure as W&C and NW. To avoid this, we also contribute a technique for hardening CSG against automatic and false user answers. *Polymorphic dialogs* (PD) are dialogs that, every time they are displayed to users, change the form of required user inputs and intentionally delay the latter, forcing users to pay attention to security decisions. We call the combination of these two techniques *context-sensitive guiding polymorphic dialogs* (CSG-PD). Operant Conditioning would interpret non-polymorphic dialogs in NW and W&C as antecedents that, when present, make it very likely that users will select a habitual option. The latter behavior is reinforced by users obtaining what they want, thus the likelihood of it occurring in the future increases. Moreover, the behavior rarely causes punishing consequences, such as security breaches. Therefore, the behavior is learned and associated with its usual antecedent, the non-polymorphic dialog. Polymorphic dialogs disrupt the antecedent making the behavior less likely.

In this chapter, we illustrate the use of CSG-PD against email-borne virus propagation. We implemented CSG-PD on Mozilla Thunderbird, and evaluated its effectiveness. In a user study, we found that users with initially weak security behaviors accept significantly fewer unjustified risks with CSG-PD than with NW dialogs. These effects are not due to simple risk aversion: CSG-PD had insignificant effect on acceptance of justified risks. Experimental results also suggest that CSG-PD has insignificant effect on task completion time relative to NW.

The rest of this chapter is organized as follows. Sections 4.2 and 4.3 respectively describe our novel techniques, context-sensitive guiding dialogs and polymorphic dialogs. Section 4.4 explains how to apply our techniques to improve users' security decisions about email attachments. Section 4.5 details our threat model. Sections 4.6 and 4.7 present our evaluation methodology and experimental results. Finally, Section 4.8 summarizes this chapter.

## 4.2 CONTEXT-SENSITIVE GUIDANCE

*Context-sensitive guidance* (CSG) consists in designing software such that before it presents to a user an option for acceptance of a security risk, the software asks the user relevant context information. The software then excludes insecure options and presents remaining options with context-appropriate interpretation and guidance. CSG dialogs are designed such that if the user provides correct context information and follows the software guidance, he or she will make a justified security decision [98]. What security decisions are justified depend on a security policy. CSG enables an organization to embed in its members' computer applications the organization's policies for classifying risks. Members should accept only risks that the organization's policies consider *justified*, and avoid *unjustified* ones. The organization may be, e.g., a governmental, military, or commercial entity. Risks may originate from outside or inside the organization. CSG handles primarily risks whose evaluation requires inputs that computers cannot obtain automatically, and obtains those inputs from users directly. CSG therefore complements more automated defenses, such as firewalls, anti-virus software, and intrusion detection systems. Given that none of these defenses is infallible, they often need to be combined in a layered security approach [122].

There are several types of risky situations where the use of CSG would be beneficial. We present two examples of such situations. First, consider the case of email attachments. A software engineer employing CSG would design an email program such that an organization that installs it can define certain attachment types to be risky. For instance, an organization may consider Word files risky because many viruses exploit Word vulnerabilities to propagate [112]. For each risky attachment type, CSG would allow the organization to define a decision tree for classifying the risk as justified or not. The email agent would transform this decision tree into dialogs that are presented on a sidebar when the user clicks on a risky attachment. The email agent allows the user to open or save the attachment only if, according to the organization's decision tree and the user's answers, the risk is justified.

Second, consider the case of users trying to navigate to potentially harmful websites (e.g., by following a hyperlink sent by email, or displayed on another website). Application developers can design a web browser that can deem some websites as suspicious based on specific criteria or heuristics. Since detection of malicious websites (e.g., phishing websites) based on heuristics is not highly accurate [107], the browser needs contextual information from users to better assess whether the website is really harmful. The browser can incorporate CSG dialogs that obtain this information from users, and allow the user to continue to the website if doing so is considered justified based on the user's answers and his organization's policy.

## 4.3 POLYMORPHIC DIALOGS

Software engineers conventionally insert fixed dialogs (i.e., dialogs that have the same appearance, options, and layout every time they are displayed) where context-dependent security decisions are needed in computer systems. The security of such decisions depends on the truthfulness of users' answers, which often can be compromised in one of two ways. First, after a hyperlink arouses a user's interest, the user often regards any intervening dialogs as meaningless formalities. The user will often give any responses that seem necessary to get the target object, even responses that may be false. This occurs if emitting those responses does not demand an effort greater than what the user usually expends (or is willing to expend) on obtaining objects similarly interesting (or valuable) to him. Second, many dialogs are such that users almost always need to give the same answer. Repetition can condition users to give that same answer automatically, even when it is false. Automatic answers reduce user effort and reinforce perceptions of dialogs as mere formalities.

Polymorphism for computer dialogs attempts to improve the truthfulness of users' answers by combating automatic responses. In a polymorphic dialog, each option's form is changed every time it is presented to the user. These changes can make effects of automatic selection of options less predictable and force users to respond more attentively. Polymorphic dialogs also delay and increase effort necessary for response. Greater effort may moderate users' interest and propensity to give false answers.

The design space for polymorphic dialogs is vast. This research considers only two dialog changes. First, when a dialog includes two or more options, they are displayed in random order. This ensures that users cannot automatically find a certain option always at the same place in the dialog. Second, the final option that confirms an operation (e.g., an option for opening an email attachment, or an option to allow a user to proceed to a website) becomes active only after the respective dialog has been displayed for some period of time. This delay encourages users to consider the dialog's other options.

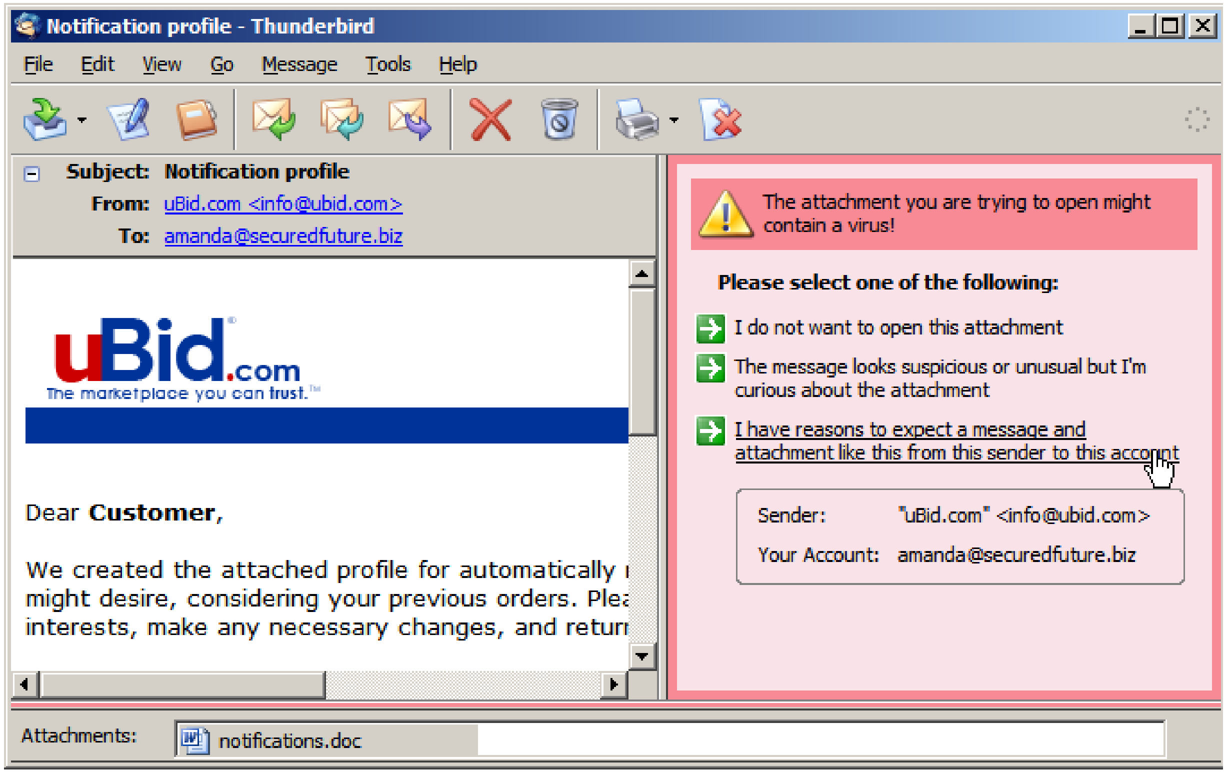
Any conventional security dialog may benefit from incorporating polymorphism into its design. For instance, context-sensitive guiding dialogs may be hardened by incorporating polymorphism so that users pay more attention to the options in such dialogs, which are presented based on a company's security policy. Another case in point are the dialogs that inform users of important security updates available for the user's installed software and that urge users to apply such updates. For instance, Microsoft found, based on feedback from their customers, that the dialog shown by Windows Update suffered the drawbacks mentioned above and was routinely dismissed by users [121]. In the present research, we explore adding polymorphism to context-sensitive guiding dialogs, and evaluate the effectiveness of this technique in that setting.

Despite the benefits that polymorphism provides to conventional, non-polymorphic, dialogs, two potential drawbacks are that the careful selection process imposed by this technique might slow users down or discourage users to take even risks that are really necessary to complete their primary tasks. This leads to the following hypothesis, which we evaluate empirically:

*Hypothesis 1. Users who interact with context-sensitive guiding polymorphic dialogs accept as many justified risks and fewer unjustified risks as users who interact with conventional dialogs, and complete tasks in the same amount of time.*

## 4.4 AN EMBODIMENT OF CONTEXT-SENSITIVE GUIDING POLYMORPHIC DIALOGS FOR E-MAIL APPLICATIONS

This section illustrates the use of CSG and polymorphic dialogs against email-borne virus propagation. Typically, an organization would implement its policies by modifying a CSG template that comes with the email application. We implemented such a template (Figure 4.11, p. 52) for Mozilla Thunderbird, which condenses advice from several sources [64][112]. According to our template policy, when a user clicks on an attachment that is a risky document (e.g., Word file), Thunderbird notifies the user that the attachment might contain a virus. The dialog asks whether the user: (a) does not wish to open the attachment; (b) finds the attachment suspicious but is cu-



Subject: Notification profile
From: uBid.com <info@ubid.com>
To: amanda@securedfuture.biz

Dear **Customer**,

We created the attached profile for automatically
might desire, considering your previous orders. Plea
interests, make any necessary changes, and retur

The attachment you are trying to open might contain a virus!

**Please select one of the following:**

I do not want to open this attachment

The message looks suspicious or unusual but I'm curious about the attachment

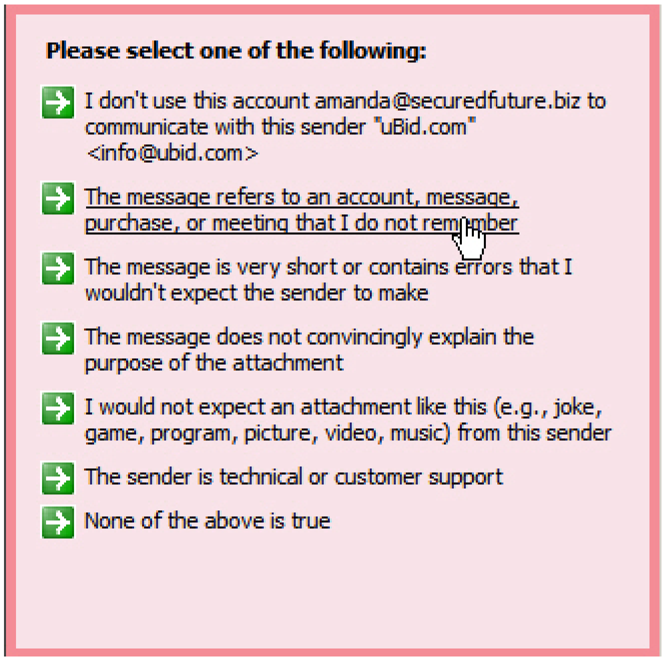
I have reasons to expect a message and attachment like this from this sender to this account

Sender: "uBid.com" <info@ubid.com>
Your Account: amanda@securedfuture.biz

**Figure 4.3:** CSG alerts user that an attachment might be infected

rious about it; or (c) has reasons to expect a message and attachment like those from that sender to that account (see Figure 4.3).

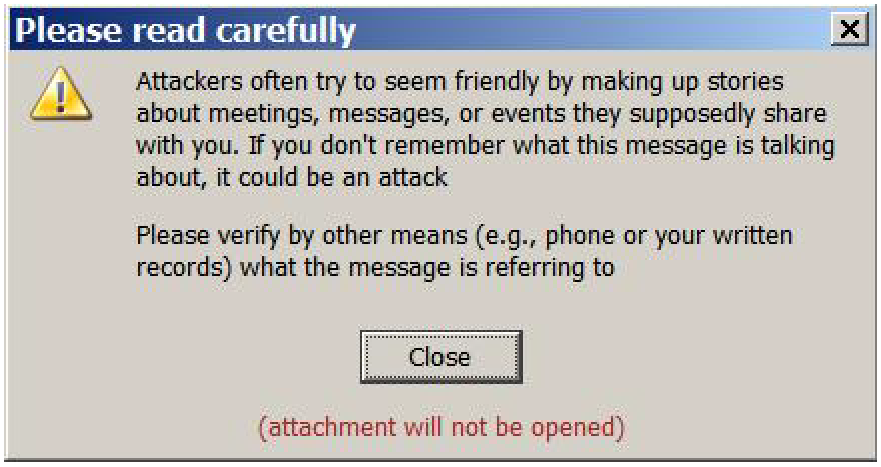
If the user selects (a), our template policy aborts the operation immediately. If the user selects (b), the template also eventually aborts the operation. However, it first asks context information that enables it to suggest what the user can do to better evaluate the risk, while reinforcing alignment between the user's and the organization's understanding of unjustified risks. The template asks the user whether: (b1) he does not use that account to communicate with that sender; (b2) the message refers to something, such as a meeting, that the user does not remember; (b3) the message is unusually short or contains errors that the user would not expect the sender to make; (b4) the message does not convincingly explain the purpose of the attachment; (b5) the attachment seems out of character for the sender; (b6) the sender is technical or customer support; or (b7) other (see Figure 4.4).



**Please select one of the following:**

- I don't use this account amanda@securedfuture.biz to communicate with this sender "uBid.com" <info@ubid.com>
- The message refers to an account, message, purchase, or meeting that I do not remember
- The message is very short or contains errors that I wouldn't expect the sender to make
- The message does not convincingly explain the purpose of the attachment
- I would not expect an attachment like this (e.g., joke, game, program, picture, video, music) from this sender
- The sender is technical or customer support
- None of the above is true

**Figure 4.4:** CSG options when user is curious about the attachment



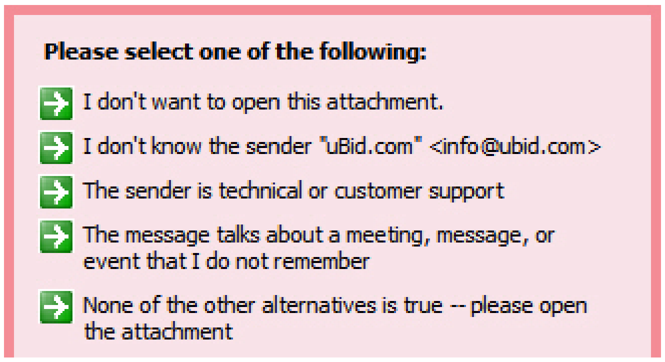
Attackers often try to seem friendly by making up stories about meetings, messages, or events they supposedly share with you. If you don't remember what this message is talking about, it could be an attack

Please verify by other means (e.g., phone or your written records) what the message is referring to

(attachment will not be opened)

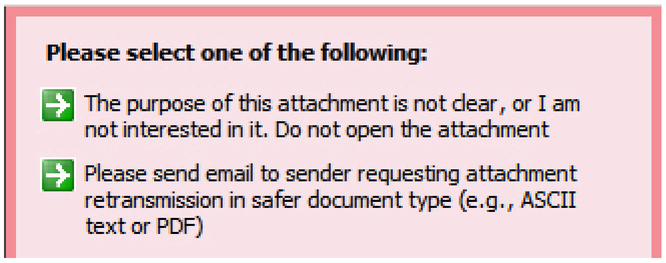
**Figure 4.5:** CSG dialog when message references something that user does not remember

According to the user's answer, the template explains why the organization considers the risk unjustified and suggests what the user can do to better evaluate the risk. For example, if the user does not remember a reference in a message (option b2), the template explains in simple language that this is a common ploy that attackers use, and asks the user to verify the reference by other means (see Figure 4.5). After user confirmation, the template aborts the operation. The user can later retry the operation, hopefully after following the received guidance.


**Please select one of the following:**

- I don't want to open this attachment.
- I don't know the sender "uBid.com" <info@ubid.com>
- The sender is technical or customer support
- The message talks about a meeting, message, or event that I do not remember
- None of the other alternatives is true -- please open the attachment

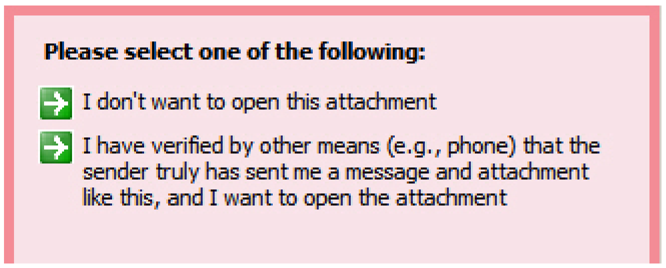
**Figure 4.6:** CSG options when user expected attachment from sender to account


**Please select one of the following:**

➔ The purpose of this attachment is not clear, or I am not interested in it. Do not open the attachment

➔ Please send email to sender requesting attachment retransmission in safer document type (e.g., ASCII text or PDF)

**Figure 4.7:** CSG options when user does not know sender


**Please select one of the following:**

➔ I don't want to open this attachment

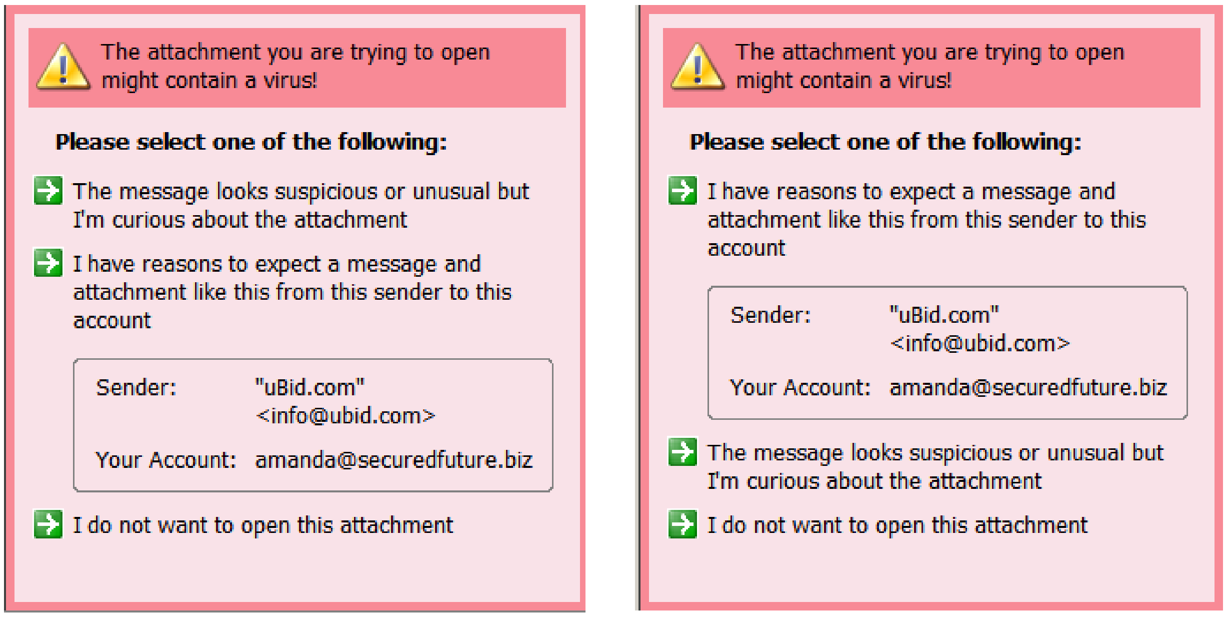
➔ I have verified by other means (e.g., phone) that the sender truly has sent me a message and attachment like this, and I want to open the attachment

**Figure 4.8:** CSG options when sender is technical or customer support

If the user selected instead (c), our template policy still attempts to ensure that the user did not make a mistake. The template asks if: (c1) the user does not wish to open the attachment; (c2) does not know the sender; (c3) the sender is technical support or customer service; (c4) the message refers to something, such as a meeting, that the user does not remember; or (c5) the user does wish to open the attachment (see Figure 4.6). If the user selects (c1), the template aborts the operation immediately. If the user selects instead (c2), the template asks whether the user: (c2a) does not wish to open the attachment, or (c2b) would like the application to ask the sender to re-transmit the attachment in a safer document type (e.g., ASCII text – see Figure 4.7). If the user selects instead (c3), the template asks whether the user: (c3a) does not wish to open the attachment, or (c3b) has verified by other means (e.g., phone) that the sender did send an attachment like that (attackers often impersonate technical or customer support, even though the latter usually avoid sending risky attachments – see Figure 4.8). If the user selects instead (c4) or (c5), the template respectively presents the dialog in Figure 4.5 or opens the attachment.

The entire decision tree of our template policy is presented in Figure 4.11 (p. 52). Different organizations would modify such a template to implement their own policies.

We now illustrate how to add the two types of polymorphism discussed in Section 4.3 to the CSG dialogs just described. First, the options in the CSG dialogs are presented in random order every time the respective dialog is shown to the user. For instance, Figure 4.9 illustrates two different presentations of the dialog in Figure 4.3 with its options in different order. Second, we delay the activation of option c5 in Figure 4.6, which confirms the operation of opening the attachment, for a few seconds (see Figure 4.10).


The attachment you are trying to open might contain a virus!

**Please select one of the following:**

The message looks suspicious or unusual but I'm curious about the attachment

I have reasons to expect a message and attachment like this from this sender to this account

| | |
|---|---|
| Sender: | "uBid.com" <info@ubid.com> |
| Your Account: | amanda@securedfuture.biz |

I do not want to open this attachment

The attachment you are trying to open might contain a virus!

**Please select one of the following:**

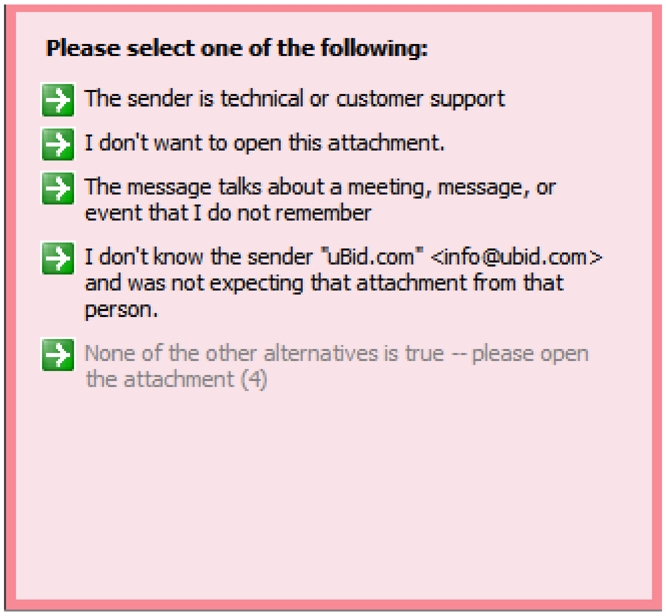
I have reasons to expect a message and attachment like this from this sender to this account

| | |
|---|---|
| Sender: | "uBid.com" <info@ubid.com> |
| Your Account: | amanda@securedfuture.biz |

The message looks suspicious or unusual but I'm curious about the attachment

I do not want to open this attachment

**Figure 4.9:** One kind of polymorphic dialog varies the order of its options each time the dialog is displayed


**Please select one of the following:**

- The sender is technical or customer support
- I don't want to open this attachment.
- The message talks about a meeting, message, or event that I do not remember
- I don't know the sender "uBid.com" <info@ubid.com> and was not expecting that attachment from that person.
- None of the other alternatives is true -- please open the attachment (4)

**Figure 4.10:** Another kind of polymorphic dialog activates an option only after the respective dialog has been displayed for some time

To test our hypothesis (Section 4.3), we prototyped the defenses described in this section (CSG against email-borne viruses and polymorphic dialogs) as a Mozilla extension for Thunderbird on a PC running Windows XP. A configuration option selects NW (Thunderbird's default dialogs) or CSG-PD (CSG with polymorphic dialogs). We implemented user interfaces in XUL, Mozilla's XML User Interface Language, and processing logic in JavaScript, which facilitates porting our extension to other platforms where Thunderbird runs.

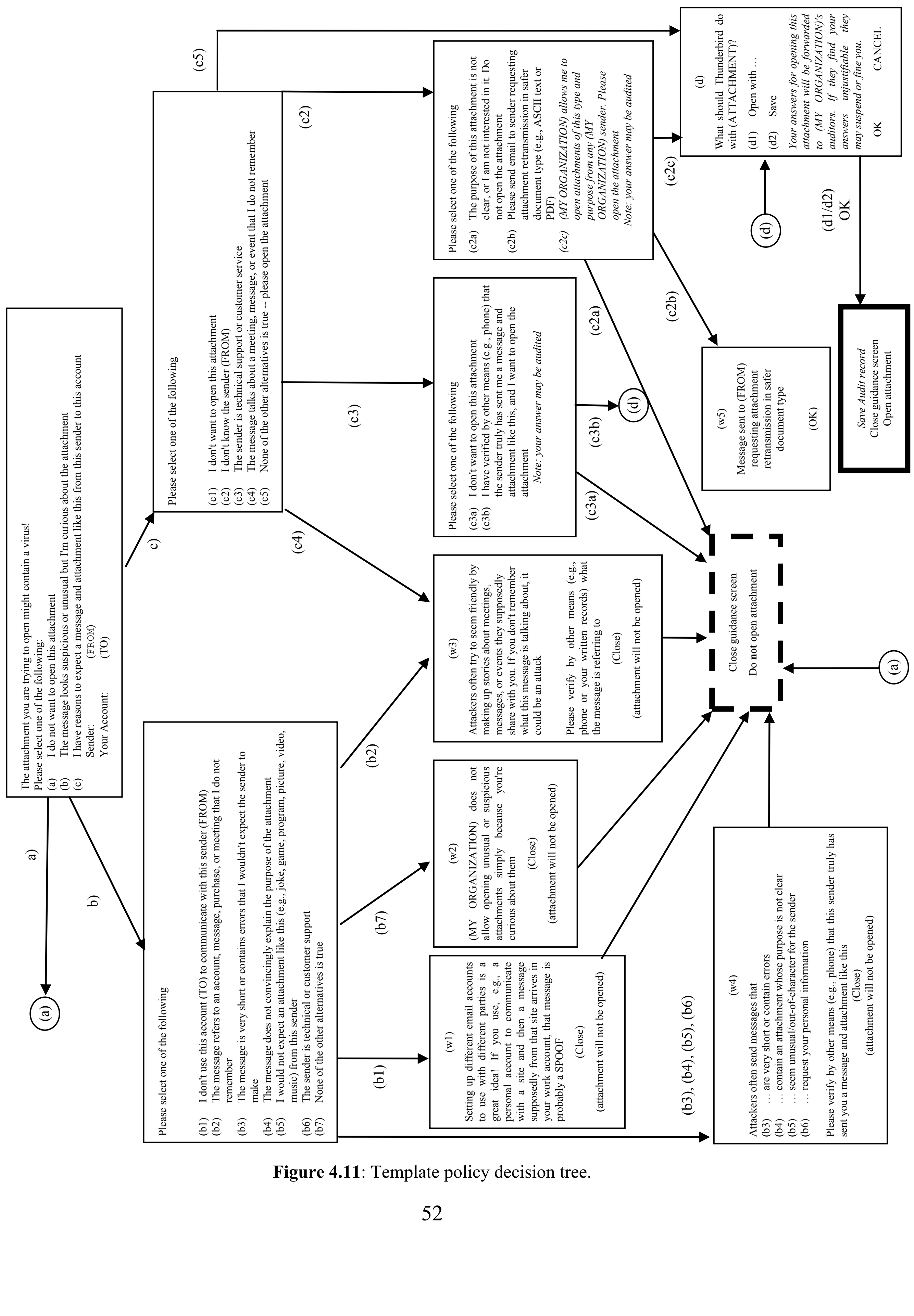

The attachment you are trying to open might contain a virus!
Please select one of the following:
(a) I do not want to open this attachment
(b) The message looks suspicious or unusual but I'm curious about the attachment
(c) I have reasons to expect a message and attachment like this from this sender to this account
Sender: (FROM)
Your Account: (TO)
a)
b)
c)
(a)
Please select one of the following
(b1) I don't use this account (TO) to communicate with this sender (FROM)
(b2) The message refers to an account, message, purchase, or meeting that I do not remember
(b3) The message is very short or contains errors that I wouldn't expect the sender to make
(b4) The message does not convincingly explain the purpose of the attachment
(b5) I would not expect an attachment like this (e.g., joke, game, program, picture, video, music) from this sender
(b6) The sender is technical or customer support
(b7) None of the other alternatives is true
(b1)
(b7)
(b2)
(b3), (b4), (b5), (b6)
(w1)
Setting up different email accounts to use with different parties is a great idea! If you use, e.g., a personal account to communicate with a site and then a message supposedly from that site arrives in your work account, that message is probably a SPOOF
(Close)
(attachment will not be opened)
(w2)
(MY ORGANIZATION) does not allow opening unusual or suspicious attachments simply because you're curious about them
(Close)
(attachment will not be opened)
(w3)
Attackers often try to seem friendly by making up stories about meetings, messages, or events they supposedly share with you. If you don't remember what this message is talking about, it could be an attack
Please verify by other means (e.g., phone or your written records) what the message is referring to
(Close)
(attachment will not be opened)
(w4)
Attackers often send messages that
(b3) … are very short or contain errors
(b4) … contain an attachment whose purpose is not clear
(b5) … seem unusual/out-of-character for the sender
(b6) … request your personal information
Please verify by other means (e.g., phone) that this sender truly has sent you a message and attachment like this
(Close)
(attachment will not be opened)
Please select one of the following
(c1) I don't want to open this attachment
(c2) I don't know the sender (FROM)
(c3) The sender is technical support or customer service
(c4) The message talks about a meeting, message, or event that I do not remember
(c5) None of the other alternatives is true -- please open the attachment
(c5)
(c2)
(c3)
(c4)
Please select one of the following
(c3a) I don't want to open this attachment
(c3b) I have verified by other means (e.g., phone) that the sender truly has sent me a message and attachment like this, and I want to open the attachment
Note: your answer may be audited
(c3a)
(c3b)
(d)
Please select one of the following
(c2a) The purpose of this attachment is not clear, or I am not interested in it. Do not open the attachment
(c2b) Please send email to sender requesting attachment retransmission in safer document type (e.g., ASCII text or PDF)
(c2c) (MY ORGANIZATION) allows me to open attachments of this type and purpose from any (MY ORGANIZATION) sender. Please open the attachment
Note: your answer may be audited
(c2a)
(c2b)
(c2c)
(w5)
Message sent to (FROM) requesting attachment retransmission in safer document type
(OK)
(d)
What should Thunderbird do with (ATTACHMENT)?
(d1) Open with …
(d2) Save
Your answers for opening this attachment will be forwarded to (MY ORGANIZATION)'s auditors. If they find your answers unjustifiable they may suspend or fine you.
OK CANCEL
(d)
(d1/d2) OK
Save Audit record
Close guidance screen
Open attachment
Close guidance screen
Do not open attachment
(a)

**Figure 4.11**: Template policy decision tree.

## 4.5 THREAT MODEL

The primary threat against CSG is that users may not provide legitimate inputs. Users often deem security dialogs irrelevant to the tasks they are performing and try to evade them [39]. Polymorphic dialogs seek to mitigate this risk.

CSG and polymorphic dialogs assume that neither attackers nor users can disable or spoof them, e.g., by reconfiguring, modifying, substituting, or extending applications. An organization may, e.g., use operating system protection mechanisms to reserve such privileges to system administrators. Additionally, an organization may use mechanisms such as Trusted Network Connect [111][40] to verify the configuration of a member's computer whenever the latter attempts to connect to the organization's network.

In the case of email attachments, we assume that attackers may wish to infiltrate an organization's computers with malware that firewalls and anti-virus software do not detect (e.g., see [26]). Such malware may be new and specially targeted against an individual within the organization, thus thwarting signature- or heuristic-based detection.

## 4.6 EVALUATION METHODOLOGY

### 4.6.1 EXPERIMENTAL DESIGN

We performed a user study to compare CSG-PD to NW dialogs (the latter are used in unmodified Thunderbird v1.5). The user study involved two similar scenarios, A and B. Each scenario comprises the same counts of justified and unjustified risks. The study used a within-subjects design with two conditions: control and CSG-PD. In the control condition, participants performed one of the scenarios (randomly selected) using NW dialogs. In the CSG-PD condition, participants role-played the other scenario while interacting with context-sensitive guiding polymorphic dialogs. We randomly selected the order in which participants performed the two scenarios to avoid order-induced biases. However, participants always used NW first to avoid learning effects. Participants were already familiar with NW at the beginning of the study. Consequently, there is nothing new that participants might have learned from NW and applied to CSG-PD.

We measured (1) the counts of justified and unjustified risks each participant accepted with each type of dialog, and (2) the time each participant took for completing a scenario's tasks with each type of dialog. We used Wilcoxon's signed-ranks test to compare the participants' performance in the control and CSG-PD conditions. This non-parametric test is used when comparing related samples without making assumptions about their statistical distributions. We did a one-sided test for comparing acceptance of unjustified risks and two-sided tests for comparing acceptance of justified risks and time to complete tasks, because we expected relationships as specified in Hypothesis 1.

CSG-PD requires more effort from users than does NW for accepting an unjustified risk. Users need to realize this when interacting with CSG-PD. However, after repeated use, users might also try to figure out ways to reduce such higher effort so as to behave in the same way as when confronted when NW. To shed light on these processes, we grouped unjustified risks by their order of appearance in each scenario. We calculated the frequency with which risks appearing in a certain order were accepted by participants using each type of dialog. We define *net acceptance frequency* (NAF) as the difference between corresponding acceptance frequencies with CSG-PD and NW. As users process the unjustified risks, NAF would take values that are either

(i) around zero if a UR is accepted as frequently when users interact with CSG-PD as when they interact with NW (e.g., in the case of the very first risk processed, the user has not seen CSG-PD often enough to note that its options are shown in random order every time CSG-PD is displayed), (ii) negative if the user realizes the higher effort imposed by CSG-PD for accepting URs and does it less frequently or at all, or (iii) positive values if a UR is accepted more often with CSG-PD than with NW (e.g., if users figured out how to accept an UR with less effort with CSG-PD than with NW). Thus, NAF can be used as a proxy of users' effort estimation. We plot the net acceptance frequencies vs. order of unjustified risk to investigate how effort estimates by users evolve with continued use of CSG-PD.

### 4.6.2 SCENARIOS

In each scenario, a participant is asked to role-play an employee of a fictitious company. Initially, the participant receives a handout that briefly describes the employee and coworkers (including some personal details), the company, and tasks the employee is involved in. In scenario A, the employee is selecting applicants for a job at the company. In scenario B, the employee needs to process customers' insurance claims. After the participant has read a scenario's handout, we ask the participant to process the respective employee's email messages. In each scenario, the employee's inbox contains 10 unread messages, each containing a Word attachment. The first and sixth messages' attachments are needed in work-related tasks and therefore pose justified risks. The remaining messages' attachments pose unjustified risks. For consistency, we used the same order of risks in both scenarios. We consider that a participant accepts or rejects an attachment's risk by respectively opening the attachment or not.

Appendix A contains the handouts for the scenarios. Section B.3 of Appendix B contains details about all the emails used in the present experiment.

### 4.6.3 PARTICIPANTS

Participants in the user study were at least 20 years old and had previous work experience in which they needed to use an email agent, such as Outlook or Thunderbird. We considered such experience necessary for participants to be able to faithfully play the assigned roles. We required participants to be native English speakers or have similar proficiency, so as to rule out linguistic difficulties that might, e.g., cause a participant to miss nuances or errors that suggest that a message is spoofed. We excluded from the study Computer Science and Electrical Engineering students or graduates, whose greater familiarity with computers might cause them to process email differently from the general population. We recruited participants by distributing flyers around the University of Pittsburgh's campus, posting in Pittsburgh jobs newsgroups, posting in the http://pittsburgh.craigslist.org and http://pittsburgh.backpage.com volunteering sections, and publishing a printed ad in Pittsburgh's City Paper.

After recruitment, we excluded from the study participants who accepted less than half of the unjustified risks in the scenario they performed with NW dialogs. These participants did not perform a second scenario with CSG-PD. These participants' performance suggests that their secure behavior was already strong before the user study. CSG is not intended for such users. Instead, CSG is designed to help users whose security behaviors are weak, to take more secure actions. This criterion excluded 5 of 18 participants recruited for the study (27.8%). Excluded participants accepted on average 30% of unjustified risks ($\sigma$= 11.2%, min=12.5%, max=37.5%).

**Table 4.1:** Characteristics of participants who progressed past control condition

| | |
|---|---|
| # Participants | 13 |
| # Female | 10 |
| # Male | 3 |
| Familiarity with email agents (self-reported) | 4.1 / 5 |
| Ease of user study tasks (self-reported) | 4.5 / 5 |
| # Unjustified risks accepted in control condition | 78.9% |

Table 4.1 summarizes characteristics of participants who interacted with CSG-PD. The number of people noted was necessary to reach statistically significant results. A majority of participants were female. This fact was unplanned and we do not assign any particular significance to it.

### 4.6.4 LABORATORY SESSIONS

Each participant role-played the two scenarios described in Section 4.6.2 in an individually scheduled laboratory session using the prototype described in Section 4.4. Each participant's session lasted between 26 and 92 minutes. Participants received between \$15 and \$22 compensation for their time. We took notes and recorded the participant's computer screen, face, and voice. These recordings helped us confirm counts and tasks completion times. We did not record participant names or other personal information, and we report only aggregate results.

## 4.7 EXPERIMENTAL RESULTS

Table 4.2 summarizes the main results of our user study. The p-values were calculated using a Wilcoxon's signed-ranks tests. The noted effect sizes are Cohen's d; values of (a) 0.2 to 0.3, (b)

**Table 4.2:** Comparison between CSG-PD and control

p-values were calculated using Wilcoxon's signed-ranks test (* = significant)

| | **Control** | **CSG-PD** |
|---|---|---|
| # participants | 13 | |
| | *# of justified risks accepted* | |
| Mean | 2.00 | 1.92 |
| Std. Dev | 0.00 | 0.28 |
| p-value | 1.00 | |
| | *# of unjustified risks accepted* | |
| Mean | 6.31 | 4.15 |
| Std. Dev | 1.11 | 1.86 |
| p-value | 0.003* | |
| Effect size (Cohen's d) | 1.08 | |
| | *Time to complete tasks (minutes)* | |
| Mean | 25.34 | 20.35 |
| Std. Dev | 12.02 | 8.75 |
| p-value | 0.11 | |

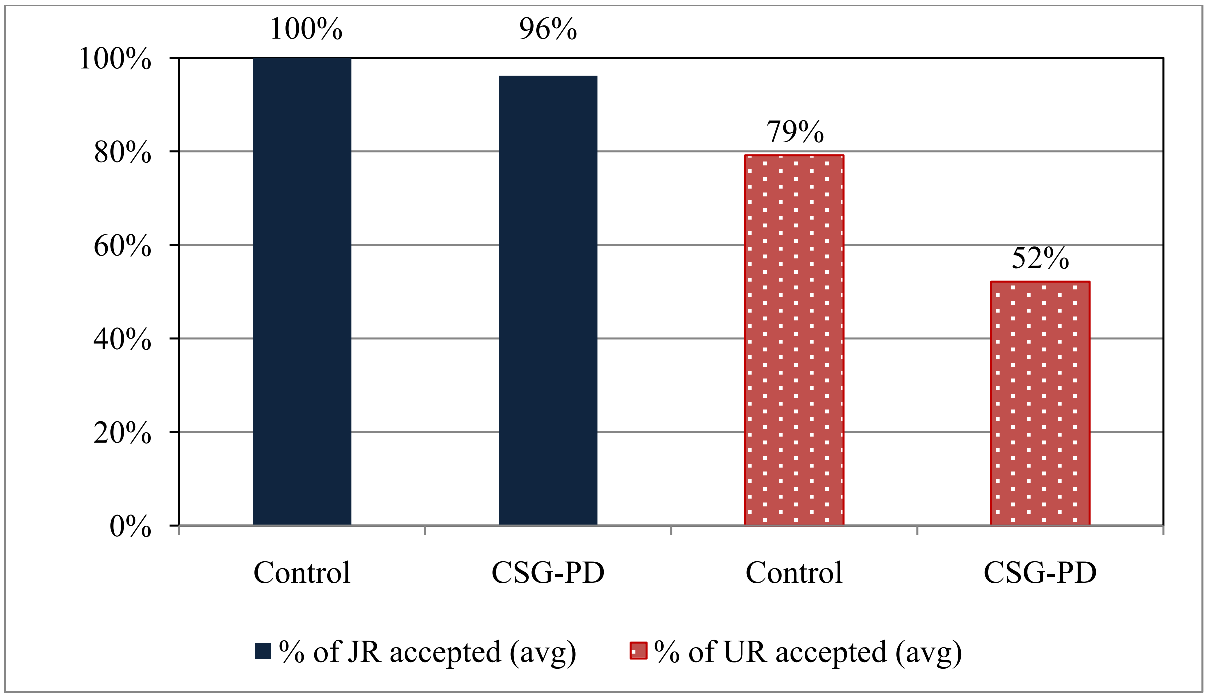


**Figure 4.12:** Average percentage (%) of justified risk (JR) and unjustified risk (UR) accepted during both conditions

around 0.5, and (c) 0.8 to infinity, are indicative of small, medium, and large effects, respectively [44]. Table 4.2 shows that, compared to conventional (NW) dialogs (control condition), CSG-PD provides a statistically significant and large reduction in the number of unjustified risks accepted (p-value = 0.003, d = 1.08). There was also an insignificant difference (p-value = 1.0) in the number of justified risks accepted. We plot average percentages of justified risk and unjustified risk accepted in Figure 4.12. Table 4.2 also shows that compared to control, CSG-PD provides an insignificant reduction in tasks completion time (p-value = 0.11). These results verify Hypothesis 1.

Figure 4.13 shows how the net acceptance frequency of unjustified risks evolved with continued use of CSG-PD. The graph shows that, after having accepted 3 unjustified risks with CSG-PD, users apparently realized that unjustified risks require higher efforts than with NW.

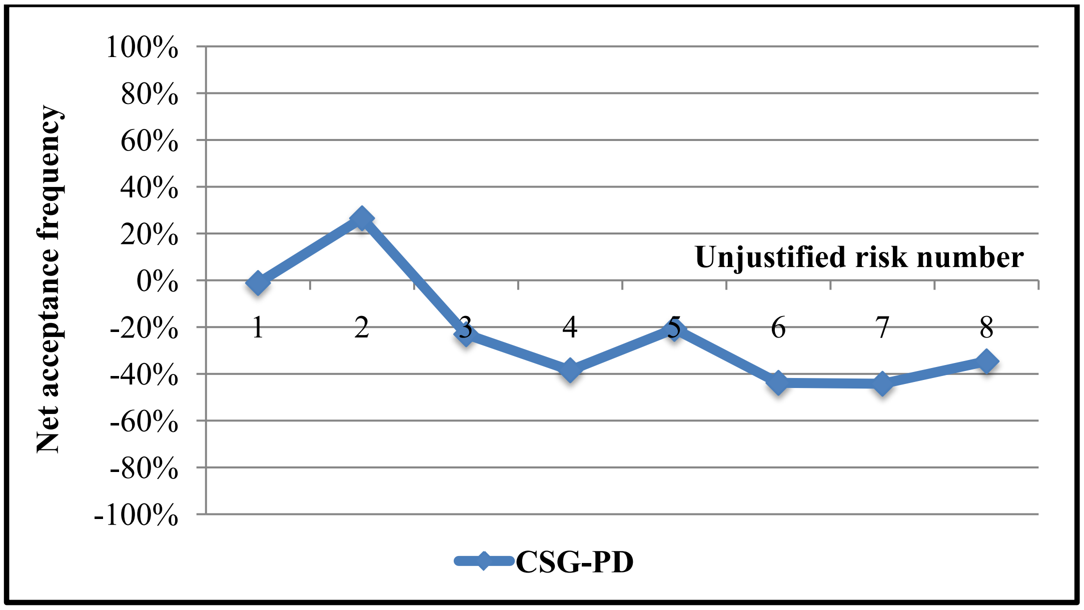


**Figure 4.13:** Unjustified-risk net acceptance frequency decreases after user learns amount of effort required by CSG-PD

CSG-PD's higher amount of effort decreases net acceptance frequency for the remaining unjustified risks, on average, by 36%.

**Table 4.3:** Participant perceptions of CSG-PD

(worst = 1, best = 5)

| | |
|---|---|
| Dialogs are easy to understand | 3.9 |
| Questions are helpful | 2.4 |
| Interface provides good guidance | 3.6 |
| Participant followed guidance | 2.5 |
| Would feel comfortable receiving such guidance in future | 3.7 |
| Would recommend to friend | 3.1 |

Table 4.3 shows the results of a survey completed by participants at the end of their sessions. They would be fairly comfortable with CSG-PD in the future, but would give friends a neutral recommendation. Participants found CSG-PD easy to understand and that it provides fairly good guidance, but they did not always find the questions helpful or follow guidance. The latter may be because, after being exposed a few times to the dialogs, users start to make better decisions on their own, by looking for the cues suggested in the guidance.

## 4.8 SUMMARY

Some policies may be flexible but insecure (e.g., Thunderbird's NW), while others can be inflexible but highly secure (e.g., Microsoft Outlook blocking attachments of a specific type without warning users [114]). To be both secure and flexible, policies often need to consider context information that can be obtained only from the user. However, designing effective dialogs for eliciting such information can be a formidable challenge. Many users view such dialogs as meaningless obstacles and do not hesitate to give false answers.

In this chapter we proposed and evaluated a new technique for improving the truthfulness of user answers and the consequent quality of security decisions. Polymorphic dialogs continuously change the form of required user inputs, preventing automatic answers. To illustrate the use of such technique, we designed a policy and corresponding context-sensitive guidance (CSG) for avoiding virus infection from email attachments.

We implemented CSG with polymorphic dialogs (CSG-PD) within the context of the Thunderbird email client. Results from a user study show that users accept significantly fewer unjustified risks with CSG-PD than with conventional dialogs (NW). Moreover, CSG-PD has insignificant effect on acceptance of justified risks. Users quickly adapted to the new dialogs, and we found no evidence of loss of effectiveness after continued use (Figure 4.13). Users' perception of the new dialogs was mostly positive (Table 4.3), and it appears that users would accept them if adopted by the organization that users are members of.

# 5.0 SECURITY-REINFORCING APPLICATIONS

In the previous chapter, we explored the use of *antecedent* invalidation to reduce the likelihood of emission of insecure behaviors. Complementary effects can be achieved by manipulating the *consequences* of security behaviors. In this chapter, we introduce and evaluate one such approach, security-reinforcing applications (SRAs). In the next chapter, we explore an alternative approach, insecurity-punishing applications (IPAs). Collectively, we call these two types of applications security-conditioning applications (SCAs).

SRAs allow system administrators to reward users for using software securely while completing production tasks. These rewards are administered close in time after the behavior and according to a specific schedule. We prototyped an Email-SRA on top of the Mozilla Thunderbird email client, and empirically evaluated whether users managed their email accounts more securely with the SRA than with the unmodified Thunderbird client. Results of a user study show that users indeed improve their security behavior and reject more unjustified risks with the SRA than with the original email program, without affecting acceptance of justified risks or time to complete tasks. Moreover, secure behaviors strengthened by using the SRA did not extinguish after a period of several weeks in which users only interacted with conventional applications.

## 5.1 INTRODUCTION

Most organizations already give their members rewards to shape members' behavior and align them with organizations' primary goals. Common rewards include recognition, special meals or retreats, promotions, individual or team bonuses, profit sharing, stock options, and so on. However, since security is often seen as a secondary goal [74][102], behavior that helps maintain the security of the company's information systems is rarely, if ever, rewarded. At most, the benefits for behaving securely might consist in avoiding penalties, or being accepted by peers in a particular "security conscious" group [102]. These benefits are not enticing if insecure behavior is never penalized in the user's organization, or if applying security interferes with completion of production tasks (which, on the contrary, does result in adverse consequences). Operant conditioning predicts that lack of rewards for particular behaviors tends to extinguish such behaviors and that using rewards that people do not find reinforcing does not strengthen behaviors [18].

Including how securely members use computer systems in the objectives of current incentive programs would be useful to address the aforementioned issues. However, achieving this in practical and effective ways is not straightforward. First, it is not known what rewards can be effective in promoting secure behaviors, and in what amount these should be delivered. Second, it is unknown what schedules for giving rewards would be most effective. Third, it is unknown if secure behaviors will extinguish during periods of time in which they are not reinforced (e.g., when employees leave the workplace for the day, the weekend, or vacations).

In this chapter, we show how, based on principles of operant conditioning, computer scientists can design and use applications to successfully address each of the above concerns. To this end, we contribute security-reinforcing applications (SRAs), which reward users for complying with their organization's security policies while they also perform their primary tasks. We empirically evaluate SRAs using as a case study the handling of email attachments and demonstrate that SRAs are indeed effective in improving users' secure behaviors without affecting their productivity. In our experiments with human subjects, we show that users reject significantly more unjustified risks with an SRA than with a conventional email application. Also, users neither take more time to finish assigned tasks nor reject risks that are needed to complete them. Finally, the secure behaviors strengthened with SRAs do not extinguish after a period of several weeks in which users only interact with conventional applications.

The rest of this chapter is organized as follows. Section 5.2 defines SRAs, describes their properties, and presents our hypotheses about them. Section 5.3 explains how to apply our technique to strengthen users' security behaviors related to opening email attachments, and provides details about the implementation of an Email-SRA for this purpose. Section 5.4 discusses SRA's assumptions, threat model, and security analysis. Section 5.5 presents the methodology we used to evaluate SRA. Section 5.6 presents our experimental results, and Section 5.7 summarizes the chapter.

## 5.2 SECURITY-REINFORCING APPLICATIONS

In this section we present specific details about security-reinforcing applications (SRAs), their properties, and our hypotheses about them. We first define SRAs and state our first hypothesis about their effectiveness. We then discuss what stimuli can be effectively used for reinforcing users' secure behavior with SRAs, what scheduling could be most appropriate to use with these applications, and whether behaviors conditioned with SRAs are resistant to extinction, and state our hypotheses about these topics. Finally, we present examples of computer security mechanisms where SRAs could be successfully used.

### 5.2.1 DEFINITION AND EFFECTIVENESS

A security-reinforcing application (SRA) is a computer application that can reinforce its users' secure behaviors. An SRA does this by delivering reinforcing stimuli (e.g., a praise reward or notification of a prize reward that the user would receive in the future) contingently upon emission of such behaviors, according to a reinforcement schedule. An organization can initiate the reinforcement automatically or manually. In the former case, the application itself applies the reinforcement when conditions specified by a policy are met. For instance, an SRA may be configured to reward an employee automatically if she rejects a risk that the application deems unjustified. In the latter case, special entities, such as an organization's security auditors, possess the privilege of instructing the application to apply reinforcement. Security auditors can do this by first sending to a user's computer risks that they a priori consider justified or unjustified. The auditors can then instruct the SRA to reinforce the user when she either rejects the unjustified risks or accepts the justified ones. By selectively rewarding the employees' secure behaviors, the auditor can increase the likelihood of such behaviors, as predicted by Operant Conditioning.

*Hypothesis 2. Users who interact with security-reinforcing applications accept as many justified risks and fewer unjustified risks as users who interact with conventional applications, and complete tasks in the same amount of time.*

### 5.2.2 REINFORCING STIMULI

Little is known about what types of 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. A software engineer cannot simply ask users either, as self-reporting has been found to be unreliable, especially if contingencies are complex [16, p. 8]. An 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.

*Hypothesis 3. A combination of praise and prizes is an effective positive reinforcer in a security-reinforcing application.*

To measure if a reward is reinforcing, and adjust it accordingly, security auditors who use SRAs can perform a direct test. If the frequency of a desire 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 [18] 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. We experimentally test their effectiveness in this chapter.

It may not be initially apparent to users why the SRA rewards 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 improve security. To help users understand what is rewarded (and ultimately accept SRAs), all SRA's notifications should include links that users can click to obtain plain-language explanations.

### 5.2.3 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 using 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 Operant Conditioning suggest that behaviors intermittently reinforced are resistant to extinction. However, this has not been verified in software applications.

*Hypothesis 4. Intermittent reinforcement schedules are effective in a security-reinforcing application.*

During the initial learning phase, SRAs can also display notifications explaining what behaviors are not rewarded (e.g., insecure behaviors). Users should be able to ignore these notifications, and SRAs never penalize users for emitting those behaviors.

### 5.2.4 RESISTANCE TO EXTINCTION

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

*Hypothesis 5. After a user's secure behaviors have been strengthened by interacting with a security-reinforcing application using intermittent reinforcement schedules, those behaviors re-*

*main strong after a period of several weeks during which the user interacts only with conventional applications.*

### 5.2.5 POTENTIAL AREAS OF APPLICATION

Existing computer applications typically are not SRAs. However, SRAs could be advantageous in a wide variety of domains. We now briefly discuss the way SRAs can be used in some domains.

First, in the case of email, companies could designate a security auditor who may send employees email messages intentionally including justified or unjustified risks. The auditor would disguise her messages to look like other email messages. The auditor would instruct an Email-SRA, previously deployed to the organization's computers, to reward users for rejecting unjustified risks and accepting justified risks, according to a reinforcement schedule. The Email-SRA could recognize the type of risk in each message based on a special email header signed by the company's security auditors. This header would not be visible to users.

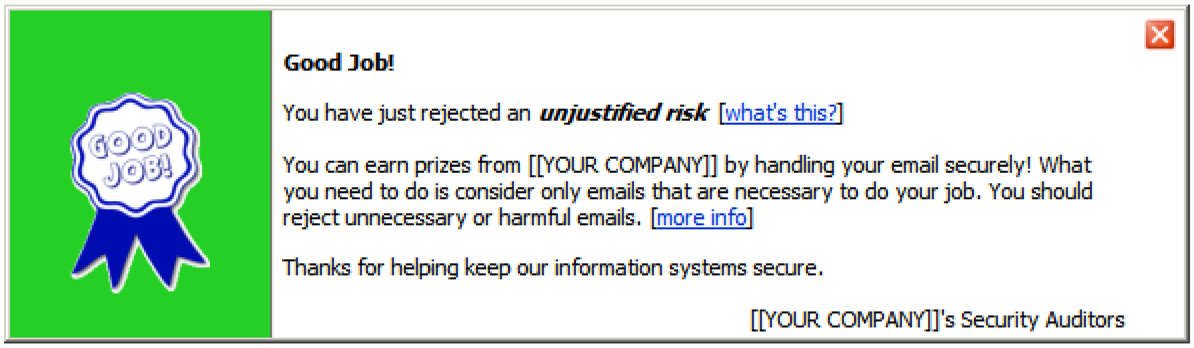
Second, consider the case of navigating to potentially harmful websites that we discussed in the previous chapter. Recall that current browsers cannot determine for sure which websites, not present in a blacklist, are really malicious based just on heuristics, and thus need to rely on users' judgment. An organization seeking to reward users that do not navigate to such sites can monitor and strengthen their members' secure behavior with a Browser-SRA. For instance, the organization's information technology department may intentionally insert or replace links on white-listed webpages that users are visiting (e.g., a search engine's results page or a newspapers' webpage, which usually include advertising links that may lead to malicious websites). The wording of the links may be carefully chosen to mimic the language that attackers use to lure users into visiting their sites. If the user clicks on the link and is presented with a security dialog that warns him that the site may be potentially dangerous and the user heeds it, the user can be rewarded by the Browser-SRA. In a learning stage, every time the user heeds the warning he would be rewarded, whereas in a maintenance stage only some of these decisions would be rewarded.

Third, protection mechanisms trying to enforce certain security principles, such as the principle of least authority, can be converted into SRAs. For instance, the software Polaris (which we mentioned in Chapter 3) is used to ‘polarize’ an application, i.e., to create a ‘tamed’ version (known as ‘pet’) of that application which is immune to viruses. In a user study, it was shown that users displayed “apathy” towards such mechanisms [8]. To overcome this, Polaris could be converted into a SRA that would initially reward users every time they create a “pet” of an application to do activities that carry some risk (e.g., a Word processor for opening an attachment received by email). Once users have been conditioned to perform in this way with the SRA, the behavior can be maintained employing intermittent schedules.

## 5.3 AN EMBODIMENT OF SECURITY-REINFORCING APPLICATIONS ON E-MAIL CLIENTS

This section elaborates on how to employ the concepts of security reinforcement to create applications that help users combat email-borne virus propagation.

A feasible way to condition users in this particular domain could involve security auditors who send an organization’s members email messages representing justified and unjustified risks. These auditors can instruct a deployed Email-SRA to reward users when they accept the former risks and reject the latter ones. Rewards can include praise and prizes. For the first case, Figure 5.1 illustrates a praise reward that an email application could be configured to show to the user when she rejects an unjustified risk. To help users who do not 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, illustrated in Figure 5.2. It is 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 the one in Figure 5.3. The information about risks shown in these two dialogs is consistent with the policy we mentioned in Chapter 2. The dialog in Figure 5.1 also announces that monetary rewards can be forthcoming if the person continues handling her email securely. The user can get more information on such prizes by clicking on the "[more info]" link (Figure 5.4). For the second case, Figure 5.5 illustrates a notification of a prize reward.



**Good Job!**

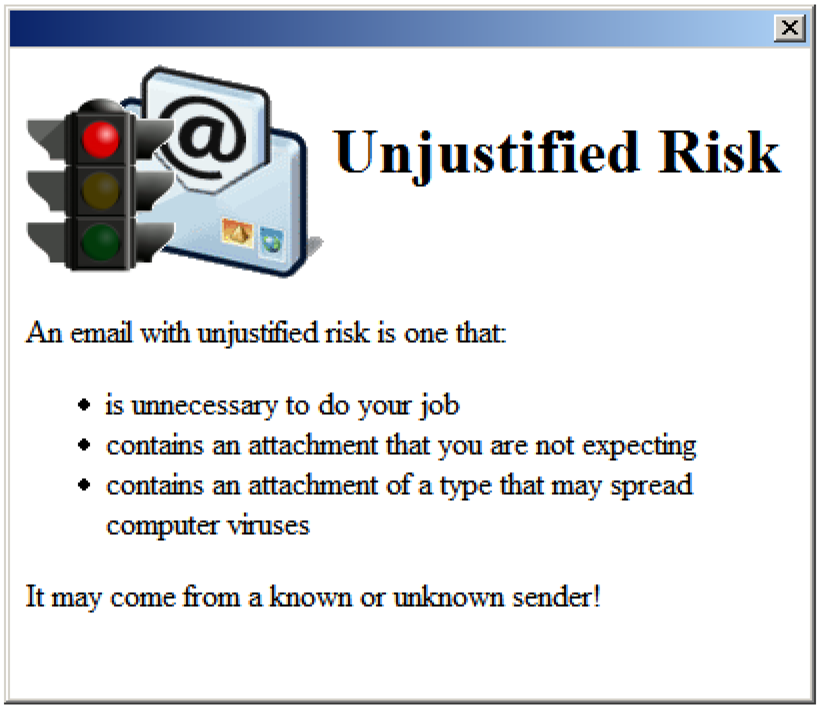
You have just rejected an ***unjustified risk*** [what's this?]

You can earn prizes from [[YOUR COMPANY]] by handling your email securely! What you need to do is consider only emails that are necessary to do your job. You should reject unnecessary or harmful emails. [more info]

Thanks for helping keep our information systems secure.

[[YOUR COMPANY]]'s Security Auditors

**Figure 5.1**: Example of a praise reward



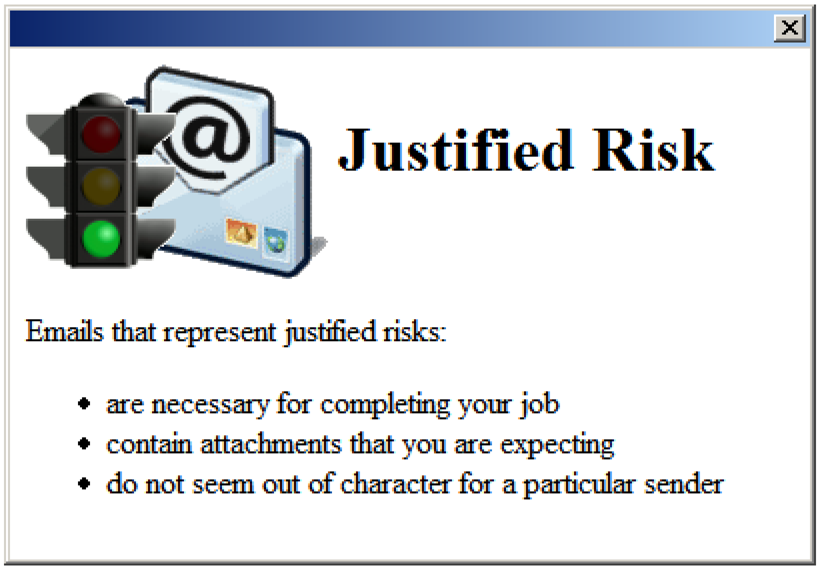
Unjustified Risk

An email with unjustified risk is one that:

- is unnecessary to do your job
- contains an attachment that you are not expecting
- contains an attachment of a type that may spread computer viruses

It may come from a known or unknown sender!

**Figure 5.2:** Information about unjustified risks

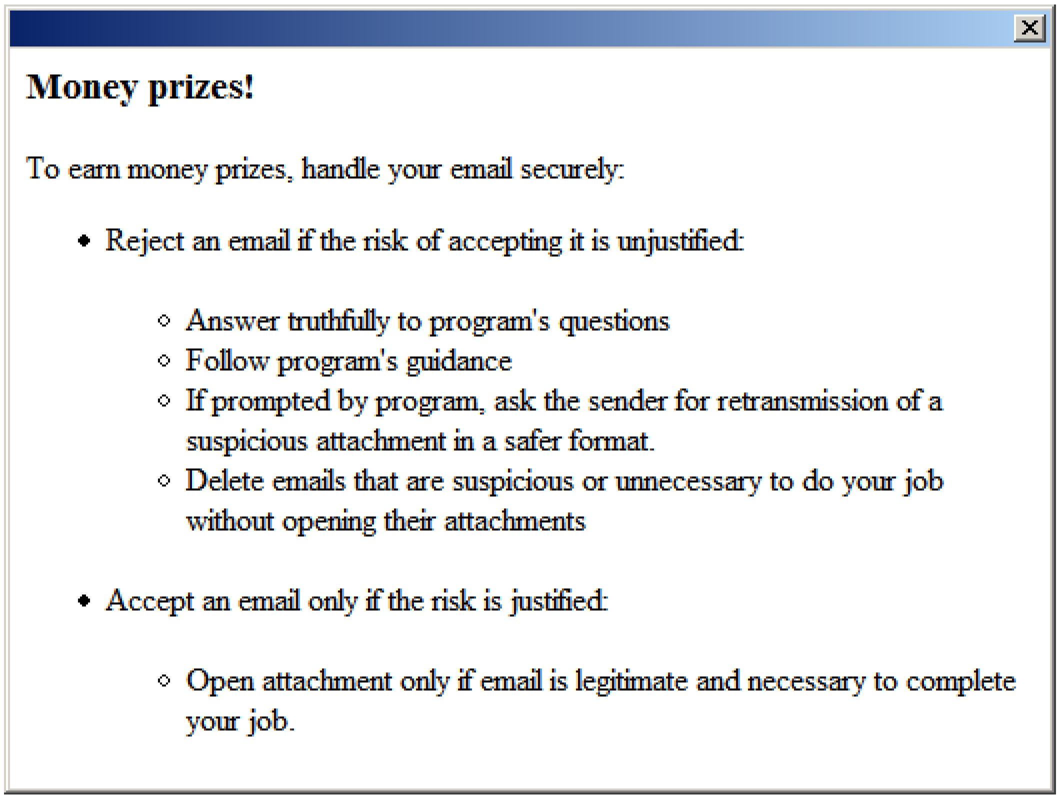


Justified Risk

Emails that represent justified risks:

- are necessary for completing your job
- contain attachments that you are expecting
- do not seem out of character for a particular sender

**Figure 5.3.** Information about justified risks

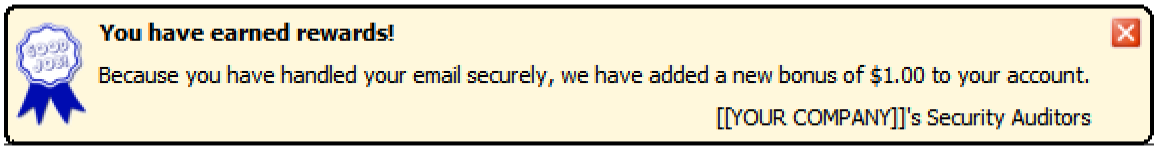


**Money prizes!**

To earn money prizes, handle your email securely:

- Reject an email if the risk of accepting it is unjustified:
    - Answer truthfully to program's questions
    - Follow program's guidance
    - If prompted by program, ask the sender for retransmission of a suspicious attachment in a safer format.
    - Delete emails that are suspicious or unnecessary to do your job without opening their attachments
- Accept an email only if the risk is justified:
    - Open attachment only if email is legitimate and necessary to complete your job.

**Figure 5.4:** Information about how to reject unjustified or accept justified risks to earn prize rewards

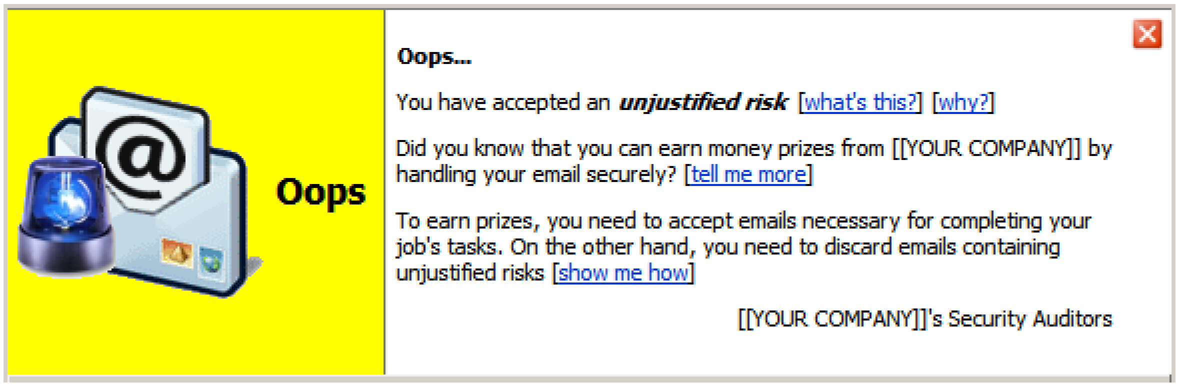


**You have earned rewards!**

Because you have handled your email securely, we have added a new bonus of $1.00 to your account.

[[YOUR COMPANY]]'s Security Auditors

**Figure 5.5.** Example of notification of prize reward



**Oops...**

You have accepted an ***unjustified risk*** [what's this?] [why?]

Did you know that you can earn money prizes from [[YOUR COMPANY]] by handling your email securely? [tell me more]

To earn prizes, you need to accept emails necessary for completing your job's tasks. On the other hand, you need to discard emails containing unjustified risks [show me how]

[[YOUR COMPANY]]'s Security Auditors

**Figure 5.6:** Dialog shown by a SRA whenever users behave insecurely in a learning phase

In an initial learning stage, the Email-SRA can be configured to reward the organization's members using a continuous schedule. Additionally, during that stage the dialog in Figure 5.6 can be displayed every time users accept unjustified risks and reject justified risks. Once users have been conditioned, they would proceed to a maintenance stage, where intermittent schedules could be employed.

### 5.3.1 IMPLEMENTATION DETAILS

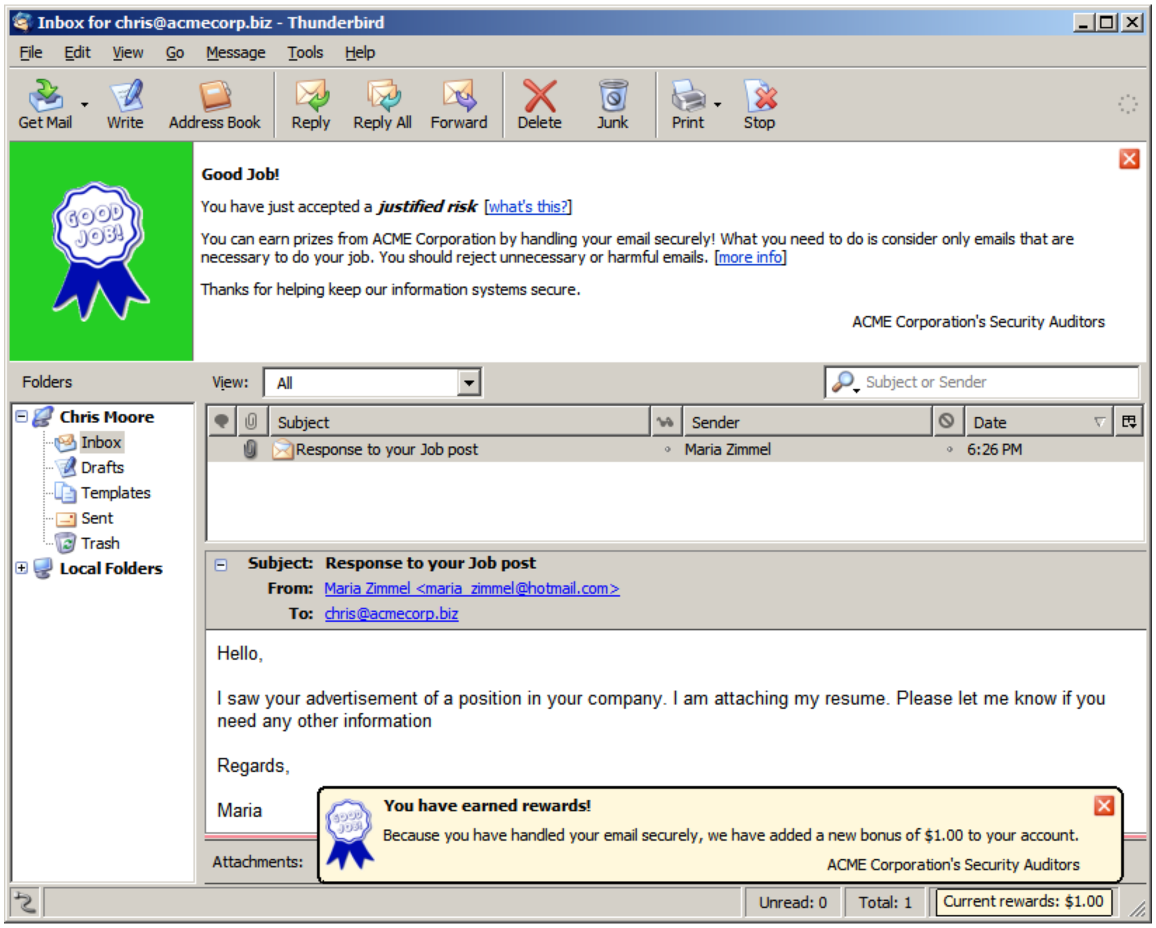
For testing our hypotheses, we extended the email client Mozilla Thunderbird 1.5 to convert it into a SRA, as described in this section. We first describe the features implemented in the prototype, and then explain the prototype's main components.

#### 5.3.1.1 FEATURES

First, the application uses CSG-PD (Chapter 4) to eliminate the discriminative stimulus of insecure behaviors which compete with secure behaviors [99]. These dialogs also help users follow our policy before emitting a behavior.

Second, we incorporated the praise and prize dialogs shown in Figures 5.1 and 5.5. The praise dialog is shown non-modally and embedded as part of the application's user-interface (just below its standard toolbar). The dialog in Figure 5.6 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. Prize and praise reward dialogs disappear whenever the user selects another message. Figure 5.7 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.

Third, we implemented the continuous and fixed-ratio schedules of reinforcement, with the ability of presenting either the praise or the 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 dialog(s) are displayed immediately (e.g., dialogs for rewarding secure behaviors).


Inbox for chris@acmecorp.biz - Thunderbird
File Edit View Go Message Tools Help
Get Mail Write Address Book Reply Reply All Forward Delete Junk Print Stop
GOOD JOB!
Folders
View: All
Subject or Sender
Chris Moore
Inbox
Drafts
Templates
Sent
Trash
Local Folders
Subject Sender Date
Attachments:
Unread: 0 Total: 1 Current rewards: $1.00

Good Job!

You have just accepted a ***justified risk*** [what's this?]

You can earn prizes from ACME Corporation by handling your email securely! What you need to do is consider only emails that are necessary to do your job. You should reject unnecessary or harmful emails. [more info]

Thanks for helping keep our information systems secure.

ACME Corporation's Security Auditors

Response to your Job post — Maria Zimmel — 6:26 PM

**Subject: Response to your Job post**
**From:** Maria Zimmel <maria_zimmel@hotmail.com>
**To:** chris@acmecorp.biz

Hello,

I saw your advertisement of a position in your company. I am attaching my resume. Please let me know if you need any other information

Regards,

Maria

**You have earned rewards!**

Because you have handled your email securely, we have added a new bonus of $1.00 to your account.

ACME Corporation's Security Auditors

**Figure 5.7:** SRA showing both praise and prize rewards

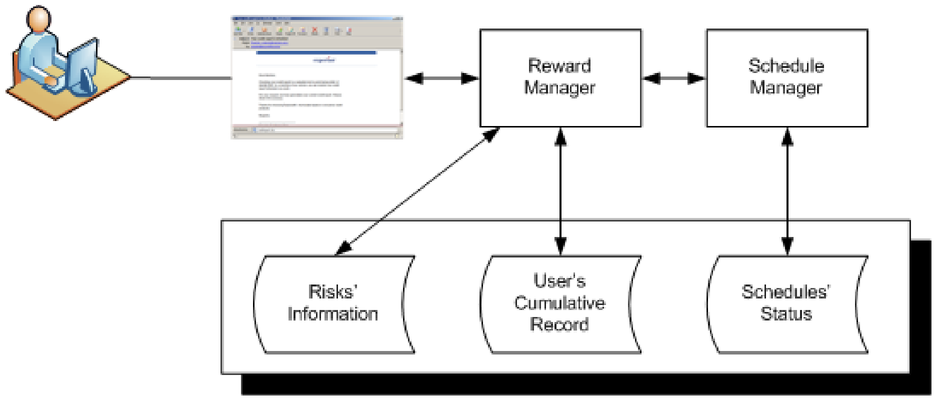


**Figure 5.8:** SRA prototype's components

### 5.3.1.2 COMPONENTS

Figure 5.8 provides a basic overview of the main components of the implemented prototype, which we describe next. Whenever the user handles a risk, a component called *reward manager* (RM) is notified about the users' action (i.e., acceptance or rejection of a risk). Then, the RM takes into account the status of such risk (e.g., whether it is unhandled, or if the user has already been rewarded for handling it appropriately) to determine what to do. If the user's action qualifies for a reward (i.e., rejection of a UR or acceptance of a JR), and such reward has not already been given to the user for that specific risk, the RM consults a component called *schedule manager* (SM) to determine if the conditions of any active schedule have been met. If so, the RM then rewards the user according to such schedules. Each of the schedules can be configured with a different type of reward (in our case, only praise and prize rewards). The status information of each schedule is also stored so that it is not lost when the user closes or restarts the email client. In Operant Conditioning, a *cumulative record* includes a subject's responses, the moment when they occurred, and which responses were reinforced. In our case, the RM also stores in a user's cumulative record what rewards the user has received, the phase she is currently in (e.g., initial learning phase, maintenance phase –Section 5.2.3), and other information.

Taken together, the parts of Figure 5.8 that are inside a shaded rectangle can be considered as a user's profile. In our implementation, such profile stores neither users' personal information, nor information about email communications that security auditors did not send.

## 5.4 THREAT MODEL AND SECURITY ANALYSIS

The assumptions and threats described in Section 4.5 for CSG apply to SRAs as well. In addition, SRAs assume that system administrators sign tagged messages with a private or secret key that attackers cannot obtain. The SRA verifies tagged email messages signed by the company auditor using the corresponding public or secret key. We assume that neither attackers nor users can disable or spoof SRAs, e.g., by reconfiguring, modifying, substituting, or extending these applications. An organization may, e.g., use operating system protection mechanisms to reserve such privileges to system administrators. Additionally, an organization may use mechanisms such as Trusted Network Connect [111][40] (TNC) to verify the configuration of a member's computer whenever the latter attempts to connect to the organization's network.

Attackers could try to imitate the SRA stimuli to fool users into behaving insecurely. Operating system protection mechanisms and TNC coupled with the tight integration of the reinforcing stimuli with the email client's chrome (e.g., Figure 5.7) make it difficult for attackers to do so.

This chapter illustrates the use of SRAs against email-borne malware propagation. There are several ways an organization's members may want to evade or trick SRAs in this context.

First, such members might attempt to evade SRAs by using instead external (e.g., Web-based) email accounts. We assume that an organization's firewalls can block direct communication between members' computers and external email servers. Such blocking is common in corporate environments. Second, users may want to trick the system by sending to themselves messages that can be considered unjustified or justified risks according to the organization's policy and then reject or accept them to get rewarded. SRAs thwart these attempts by rewarding users only when the message is digitally signed by a security auditor.

## 5.5 EVALUATION METHODOLOGY

In this section we present the methodology used to test our hypotheses. We first describe the scenarios used and the sets of emails that participants handled. Then we briefly cover the metrics employed to measure participants' performance. Afterward, we describe our experimental design. Finally, we explain our recruitment procedures, and give an outline of each session.

### 5.5.1 SCENARIOS AND EMAIL SETS

We used the same scenarios of our evaluation of CSG-PD. In scenario A, our subjects role-play an employee who is selecting applicants for a job at her company. In scenario B, an employee needs to process customers' insurance claims. The latter scenario was slightly modified for this study. The first alteration consisted in specifying that Amanda, the fictitious employee that was going to be role-played by participants, was single. The second change was to specify that Theresa, one of the people known by Amanda, worked in Human Resources. These additional details facilitated the creation of additional legitimate emails.

We created four sets of emails per scenario. Each set consisted of ten emails, half of which represented justified risks and the rest unjustified risks. We will refer to these sets as Learning-I, Learning-II, Maintenance, and Extinction. We created two learning sets to account for users who need longer training periods than others. The Maintenance set is used during a maintenance stage as described in section 5.2.3 to evaluate whether the strength of the secure behavior of participants acquired during learning can be maintained with intermittent reinforcement. Finally, the Extinction set is used to test whether the secure behavior of participants extin-

**Table 5.1:** Risks arrangement in each set

UR = unjustified risk, JR = justified risk. Boldface emails were used also in the CSG-PD evaluation

| | **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 |

guishes after a period of time. Section B.2 of Appendix B contains details about the emails in each of the four sets, but such emails' arrangement in each set can be seen in Table 5.1. Risks highlighted in boldface were also used for evaluating CSG-PD. Many of the new emails we used in this study were inspired in messages we received in our email accounts (mainly unjustified risks) and emails in the Enron corpus [11] (mainly justified risks). Each email set contains the same types of risks (Appendix B). Our SRA recognizes the type of risk that each email represents based on a special header in the email message. This header is signed by the company's security auditors, but it is not visible to users.

### 5.5.2 EVALUATION METRICS

We use concepts from Signal Detection Theory (SDT) to quantify participants' performance when handling a particular set of emails. In a signal-detection task, a certain event is classified as signal and a participant has to detect if the signal is present [50]. Noise trials are those in which the signal is absent. The noise trials form a probability of states, as do the signal trials.

There are several metrics associated with SDT as described next. First, the hit rate (HR) is the proportion of trials in which the signal is correctly identified as present. Second, the false alarm rate (FA) is the proportion of trials in which the signal is incorrectly identified as present. Third, a measure of detectability (known as sensitivity) is defined as $d'=z(\mathrm{HR})–z(\mathrm{FA})$, where $z$ is the inverse of the normal distribution function [85]. A moderate performance means that $d'$ is near unity [85]. Higher sensitivity values mean better performance in distinguishing signal from noise.

In our case, the signals in each email set are the justified risks while the unjustified risks are considered noise. We define a hit when the user accepts a justified risk (signal present and correctly identified), and a false alarm when the user accepts an unjustified risk (signal absent and incorrectly identified as present). To avoid infinite values in calculating *d'*, we convert proportions of 0 and 1 respectively to 1/(2N) and 1-1/(2N) [85], where N is the total number of either justified or unjustified risks.

### 5.5.3 STUDY DESIGN

The present study uses a within-subjects design with four different conditions. The conditions were control, learning, maintenance, and extinction, performed in this order. The first three conditions were tested in one laboratory session. In the control condition, a participant interacted with an unmodified version of Mozilla Thunderbird 1.5 (which used NW dialogs) and role-played one of our scenarios (randomly selected to avoid bias due to any differences between scenarios). In the remaining conditions the participant role-played the other scenario interacting with the email client converted to an SRA. To avoid learning effects between the control and subsequent conditions, the control condition always used NW dialogs. Such dialogs were already familiar to participants before the study and did not teach anything new that might have affected participants' performance in subsequent conditions. The learning condition used a combined schedule of reinforcement. Its component schedules were (i) continuous with praise reward, and (ii) fixed ratio with a prize reward (money) every other secure behavior emission. The dialog in Figure 5.6 was shown only during learning condition. 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. This different schedule is necessary because the frequency of reinforcement during learning is not sustainable in a production environment. The extinction condition was tested in a second session more than five weeks after the first one. We did this to emulate the situation where employees are distant from their computers for extended periods of time (e.g., vacations, attendance to conferences). Each prize reward consisted of \$0.70.

Only subjects whose sensitivity was $d' \leq \gamma$ during the control condition were selected for participating in the learning condition. We set cut-off $\gamma=1.02$, which indicates moderate performance [85]. Remaining participants' security behavior was deemed as already strong, and unlikely to significantly benefit from our reinforcement interventions. In our experiments, this resulted in the exclusion of 6 out of 18 participants. The use of the sensitivity metric ($d'$) allows performance to be measured by considering both hit and false alarm rates. This prevents participants from progressing to the next stage by simply accepting or rejecting all the risks.

In a production environment some people may need extra reinforcement to learn how to behave securely. Thus, to accommodate those users, we also applied the criterion just described to determine whether participants would pass from the learning to the maintenance stage, as described next. If the participant's sensitivity was $d' > \gamma$ after she finished 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 participant's sensitivity was $d' \leq \gamma$, the application kept pushing subsets $s_i \subset$ Learning-II into the participant's Inbox and waited for her to handle the risks in those subsets. The SRA only pushed subset $s_{i+1}$ if the participant's sensitivity was still $d' \leq \gamma$ after handling the risks in her Inbox. Otherwise the participant was switched to Maintenance. The cardinalities of the pushed subsets were $|s_1|=|s_2|=4$ and $|s_3|=2$. Each subset contained an equal number of JRs and URs. If after processing the entire Learning-I and Learning-II sets, the

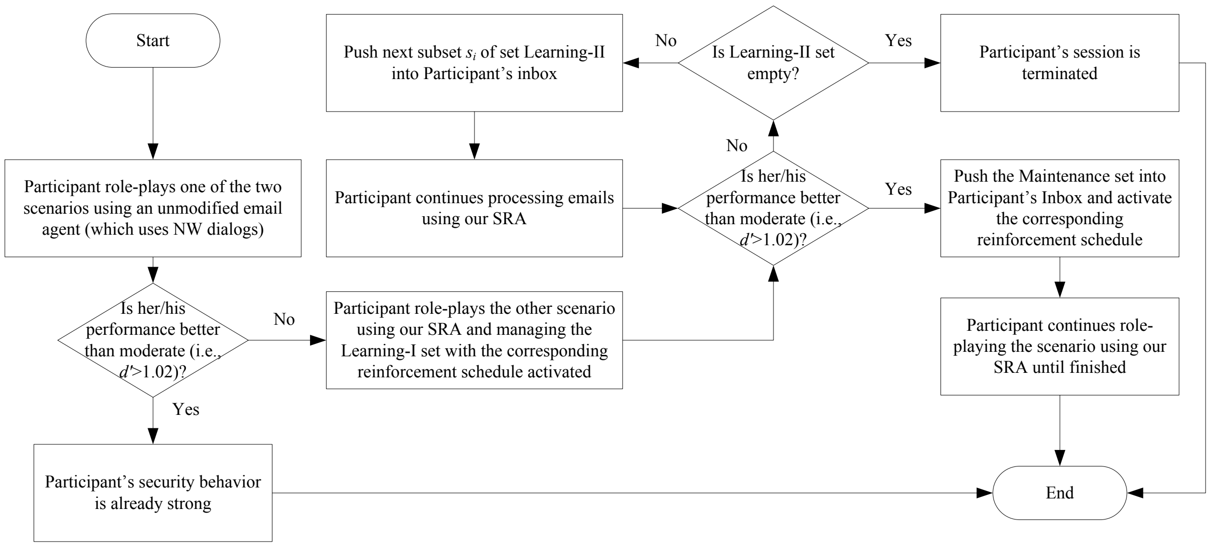

Start
Participant role-plays one of the two scenarios using an unmodified email agent (which uses NW dialogs)
Is her/his performance better than moderate (i.e., d'>1.02)?
No
Yes
Participant's security behavior is already strong
Participant role-plays the other scenario using our SRA and managing the Learning-I set with the corresponding reinforcement schedule activated
Push next subset s_i of set Learning-II into Participant's inbox
Participant continues processing emails using our SRA
Is Learning-II set empty?
No
Yes
Participant's session is terminated
Is her/his performance better than moderate (i.e., d'>1.02)?
No
Yes
Push the Maintenance set into Participant's Inbox and activate the corresponding reinforcement schedule
Participant continues role-playing the scenario using our SRA until finished
End

**Figure 5.9:** Criteria for passing from control to learning condition, and from learning to maintenance condition

participant's sensitivity had not exceeded the cutoff γ, her participation was terminated in order to limit the experimental sessions' length. In our experiments, only one participant did not improve her security behavior during the learning condition as judged with the aforementioned criterion, and thus she did not proceed to maintenance. Figure 5.9 depicts the way we applied the passing criteria just described in the SRA study. Participants who progressed to maintenance were eligible for a second session to test whether their secure behaviors extinguished.

### 5.5.4 PARTICIPANTS

We advertised the study with flyers around the University of Pittsburgh's Oakland campus, and with electronic posts in pittsburgh.backpage.com and pittsburgh.craigslist.org. We announced that the study was related to email clients' usability, not security. Once interested people contacted us, we directed them to fill out a short web-based questionnaire to determine their eligibility. Participants had to be at least 20 years old and native or proficient English speakers. They had to have a minimum of one year of work experience in companies that assigned them an email account which they had to use for job-related purposes. They had to have experience with desktop email clients and not simply with webmail. Finally, they could not hold or be currently pursuing a degree in Computer Science or Electrical Engineering. This requirement was intended to avoid testing people who were already computer-security proficient. People who participated in our other experiments (Chs. 4, 6, and 7) were not eligible for this study.

Table 5.2 summarizes characteristics of participants who interacted with the SRA. Most of the participants had two or more years of work experience. The majority of participants were female. We scheduled an equal number of participants of each gender, but absenteeism was higher among males.

### 5.5.5 LABORATORY SESSIONS

In the first session, participants received a handout that briefly described the scenario they were

**Table 5.2.** Characteristics of participants who interacted with the SRA

| | |
|---|---|
| # Participants | 12 |
| # Female | 8 |
| # Male | 4 |
| Familiarity with email agents (self-reported) | 4.0 / 5 |
| Ease of user study tasks (self-reported) | 4.2 / 5 |
| # Unjustified risks accepted in control condition | 82% |
| # Had two or more years of work experience | 10 |

about to role-play, and were given the opportunity to ask any question about it. Subsequently, we told participants that the main objective of the study was to evaluate the usability of email programs when used in a corporate setting by people who possessed real work experience. We did not tell participants that we were studying security of emails clients because we did not want to prime them to security concerns. 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 participants not to request information from us regarding what to do with the emails they were processing.

We then had participants sit at a desk in our laboratory, which we told them to be the office of the role-played fictitious employee. The desk was equipped with a laptop, a pen, and a phone in case the person wanted to make calls. Participants 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. The testing laptop was running Windows XP, Professional edition.

After finishing the scenarios, participants 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 did not tell participants whether they had qualified for a second session.

Four weeks after the first session, we asked only those participants who proceeded to maintenance 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 one more time to participants, before they started, that they should behave as close as possible as they would do at work considering the role-played employee in the scenario. After finishing processing the extinction set, participants were asked to complete the same exit survey they did in the first session.

As compensation for their time, participants received, after the first session, \$15 if they performed only the first scenario, and up to \$22 if they role-played both scenarios. Compensation was up to \$22 in the second session as well.

## 5.6 EXPERIMENTAL RESULTS

In total, eighteen people participated in this study but six of them did only the control condition and we do not consider their results any further.

Table 5.3 shows summary statistics about the remaining twelve participants' performance in each condition. One of these participants did not progress beyond the learning condition. Also, only seven of the other eleven participants returned for a second session after an average of 40 days. Table 5.4 presents comparisons between participants' performances in different conditions using p-values calculated with Wilcoxon's signed-ranks test. This non-parametric test is used when comparing related samples without making assumptions about their statistical distributions. 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 2-5. Noted effect sizes are Cohen's d.

As hypothesized, participants accepted as many justified risks (essentially all) in control as in learning (p-value=1.0, n=12), maintenance (p-value=1.0, n=11), and extinction (p-value=1.0, n=7). Also as hypothesized, there was a significant (and large) reduction in the acceptance of unjustified risks from the control to the learning (p-value=0.002, d=1.37), maintenance (p-value=0.001, d=1.95), and extinction (p-value=0.008, d=4.29) conditions. In these cases the decrease in acceptance of unjustified risk was large. We observed that the acceptance of unjustified risks declined as participants progressed from learning to maintenance to extinction, although this improvement did not reach statistical significance at the sample size considered.

One of the twelve participants in the learning condition did not progress to maintenance in our experiment. It is possible that longer learning periods might be necessary for such participants. In addition, other types of interventions, such as punishment [74, pp. 15 and 27] might be considered for noncompliant users. We explore the latter intervention in the next chapter.

We plot averages of hit (justified risk accepted) and false alarm (unjustified risk accepted) rates in Figure 5.10. When interacting with our SRA, participants accepted far fewer unjustified risks while continuing to accept essentially all justified risks. This improvement is due to the reinforcement given to participants when they behaved securely. In addition, the persistence of improvements in the maintenance and extinction conditions can be attributed to the use of intermittent schedules of reinforcement, which make behavior resistant to extinction.

**Table 5.3:** Summary Statistics of SRA conditions

| | Control | Learning | Maintenance | Extinction |
|---|---|---|---|---|
| # participants | 12 | 12 | 11 | 7 |
| *# of justified risks accepted* | | | | |
| Mean | 5.00 | 5.33† | 5.00 | 5.00 |
| Std. Dev | 0.00 | 1.50 | 0.00 | 0.00 |
| *# of unjustified risks accepted* | | | | |
| Mean | 4.08 | 1.33 | 0.73 | 0.00 |
| Std. Dev | 0.79 | 2.23 | 1.49 | 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 |

Compared to the control condition, participants spent less time completing tasks in the learning (p-value=0.04), maintenance (p-value=0.01), and extinction (p-value=0.016) conditions. These reductions in time spent were medium from control to learning (d=0.5) condition, and large from control to maintenance (d=1.03) condition and from control to extinction (d=1.94) condition. In the SRA conditions, the reduction in task completion time was because participants spent little or no time reviewing the attachments of unjustifiably risky email.

† Of the twelve participants who progressed from control to SRA-Learning, one did not progress from SRA-Learning to SRA-Maintenance. Such participant accepted 10 justified risks (5 from the Learning-I email set and 5 from the Learning-II email set). This causes the average number of risks accepted to be more than 5.

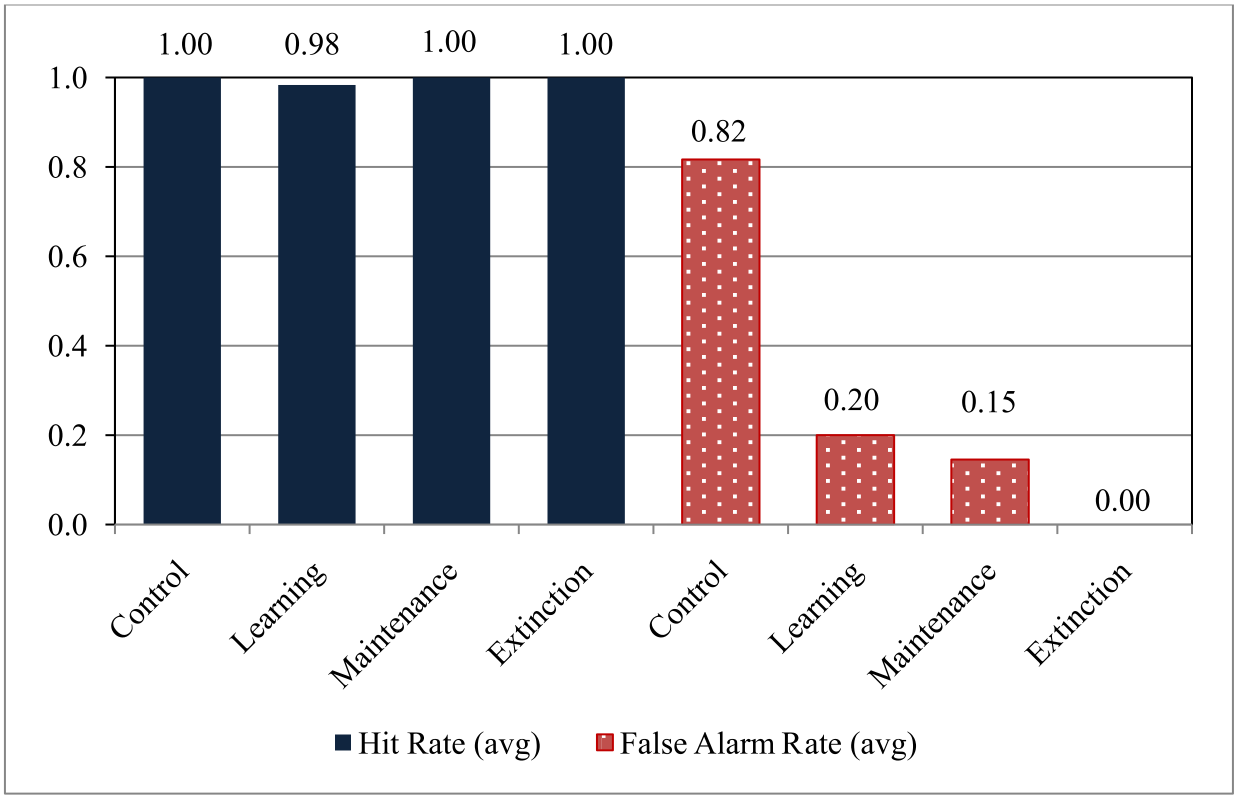


**Figure 5.10:** Average Hit and False alarm rates in the control and SRA conditions

These experimental results fully verify our hypotheses 3, 4, and 5, but partially confirmed hypothesis 2. As expected, UR acceptance declined in the SRA conditions compared to control, and JR acceptance was not different in these conditions. In addition, and unexpectedly, SRA significantly reduced task completion time.

Average results (n=12) of the exit survey are shown in Table 5.5 for the first session of the present experiment (we found no significant difference between these scores and those given by participants in the second session). Participants found SRA's user interface easy to understand, and reported that it provided good guidance. They moderately followed the guidance, and found the questions somewhat helpful. Participants would be comfortable with the SRA's guidance in the future, and would give friends a mildly positive recommendation about it.

**Table 5.4:** Comparisons with control condition

p-values were calculated using Wilcoxon's signed-ranks test (*=significant)

| | **SRA-Learning** | **SRA-Maintenance** | **SRA-Extinction** |
|---|---|---|---|
| | *Acceptance of Justified Risks (JRs)* | | |
| p-value | 1.00 | 1.00 | 1.00 |
| effect size | -- | -- | -- |
| | *Acceptance of Unjustified Risks (URs)* | | |
| p-value | 0.002* | 0.001* | 0.008* |
| effect size | 1.37 | 1.95 | 4.29 |
| | *Time to complete tasks* | | |
| p-value | 0.04* | 0.01* | 0.016* |
| effect size | 0.50 | 1.03 | 1.94 |

**Table 5.5.** Average perceptions of SRA (n=12)

(worst = 1, best = 5)

| | |
|---|---|
| Dialogs are easy to understand | 4.4 |
| Questions are helpful | 3.1 |
| Interface provides good guidance | 3.8 |
| Participant followed guidance | 3.2 |
| Would feel comfortable receiving such guidance in the future | 3.7 |
| Would recommend to friend | 3.4 |

## 5.7 SUMMARY

In this chapter, we evaluated the use of reinforcement for strengthening secure behaviors through the use of security-reinforcing applications (SRAs). These applications reward users for accepting justified risks (JR) and rejecting unjustified risks (UR) according to a specific schedule of reinforcement.

We tested SRAs in the context of email communications where a security auditor sends to end-users email messages that represent JRs and URs. Such messages included a special header that the auditor signed and that indicated the type of risk the email represented. The reinforcing stimuli used were praise and prize rewards. In our experiments with human participants, users who interacted with a SRA behaved significantly more securely than they did when they interacted with a conventional application, and there was no adverse effect on time need to complete tasks. Participants were first conditioned using a continuous schedule of reinforcement, and then their behavior was maintained with intermittent reinforcement.

Conditioned secure behavior, as any other type of behavior, can extinguish if reinforcement for desirable actions is discontinued. Our results suggest that secure behaviors strengthened with SRAs can be very resistant to extinction: the strengthened secure behaviors persisted after a period of several weeks in which users did not interact with SRAs.

# 6.0 INSECURITY-PUNISHING APPLICATIONS

In this chapter we concentrate on the manipulation of consequences of insecure behaviors and evaluate the use of punishment to weaken them. For this, we contribute the concept of an insecurity-punishing application (IPA). IPAs first warn users that they may be penalized if they select untruthful alternatives in security dialogs (e.g., to accept unjustified risks). IPAs then deliver punishing stimuli to users if their choices are found unjustifiably risky with respect to a security policy. We experimentally evaluate IPAs and compare them to SRAs. The IPA used in our user studies was a modified Mozilla Thunderbird v1.5 email client. Our results show that IPAs are effective, but users may not like them. In our user studies, IPA's acceptability and effectiveness were significantly lower than SRAs'.

## 6.1 INTRODUCTION

As with reinforcement, organizations customarily punish undesirable employee behavior related to production tasks. Punishments can include admonitions, demotions, termination, less priority in parking, shift, or vacation allocation, or fines. However, such aversive consequences typically are not made contingent on employees' failures related to computer security, which is regarded as a secondary task [74][102].

If organizations decide to use punishment for users' insecure behavior, they might face several challenges because the conditions that make punishment effective [123] may be difficult to achieve in a software environment. First, punishment must be delivered very close in time after the undesired behavior. However, in a software setting there can be a substantial delay between the time an insecure behavior occurs and the time it is discovered [10][79]. By that time, punishment may have little effectiveness [36][123] or may be infeasible to apply (e.g. if the offending employee already left the company). Second, no unauthorized escape from punishment should be allowed. In practice, however, users are able to escape from being punished in different ways. For example, users frequently share password with others or write them down and store them in easily accessible places (e.g., their desk drawers [1]), but avoid punishment if this behavior is not monitored by information technology departments [1]. Third, punishment stimulus given for undesirable behavior should be sufficiently intense. For production-related tasks this may include the employee's temporary suspension or even termination, but what punishments could be both intense and acceptable for security-related failures is unclear. Finally, punishment for insecure behavior must not only be announced (unwarned punishment can cause frustration) but also enforced. Some organizations do the former but not the latter and it has been shown that when threatened punishment for insecure behavior does not materialize, users "lose respect for the security in general", resulting in a "declining security culture" [74].

To address these challenges, we propose the use of insecurity-punishing applications (IPAs), which penalize users for accepting risks considered unjustified according to users' organization's security policy. We implemented an IPA that uses punishing stimuli that comply with the conditions specified above and delivers them contingently upon users' insecure behavior. In a user study, we show that participants accept significantly fewer unjustified risks with our IPA than with a conventional application. This finding is consistent with Operant Conditioning which predicts that aversive stimuli, such as those used by IPAs, decrease the frequency of emission of a target behavior (insecure behavior in this case). Moreover, when interacting with the IPA, participants neither accept fewer justified risks nor take more time to complete assigned tasks than with the conventional application. However, we also compared IPAs and SRAs and found that IPA's acceptability and effectiveness were significantly lower than SRAs'. Collectively, we call these two types of applications security-conditioning applications (SCAs).

The rest of this chapter is organized as follows. Section 6.2 defines IPAs and presents our hypothesis about their effectiveness and impact on productivity. Section 6.3 explains how to apply our technique to weaken users' insecure behaviors related to opening email attachments, and provides details about the implementation of an Email-IPA for this purpose. Section 6.4 presents our hypothesis about comparing IPAs and SRAs. Sections 6.5 and 6.6 respectively detail our evaluation methodology and experimental results. Finally, Section 6.7 summarizes the chapter.

## 6.2 INSECURITY-PUNISHING APPLICATIONS

An insecurity-punishing computer application (IPA) is one that, as part of its specification, possesses the following capabilities. First, it warns its users before they perform a potentially insecure action (using the application) that they will be penalized if that action is found to be unjustifiably risky (e.g., by the users' organization's security auditor). Second, it can actually deliver a punishment to its users. For example, an IPA may punish users who behave insecurely by allowing them to access only limited functionality of the application during some punishment period. Third, an IPA is equipped to avoid users' circumvention of the applied punishment. The punishment can be initiated manually (e.g., by the users' organization's security auditor) or automatically (e.g., by the application, according to a pre-specified policy). An IPA makes it clear that the punishment is contingent upon insecure behavior because it punishes users as soon as they behave insecurely.

The aforementioned use of punishment as a way to decrease the frequency of insecure behaviors has not been evaluated before. Validating its effectiveness is worthwhile, as it can help reduce the number of future security incidents caused by organizations' members' noncompliance. However, a potential complication is that some users who non-intentionally accepted unjustified risks could become overly averse to handling risks after being punished, rejecting even justified risks. In this chapter, we investigate whether this is the case.

*Hypothesis 6. Users who interact with insecurity-punishing applications accept as many justified risks and fewer unjustified risks as users who interact with conventional applications, and complete tasks in the same amount of time.*

Many security areas can benefit from using IPAs, including those using SRAs, which we described in the previous chapter. First, an Email-IPA may automatically penalize users for accepting email attachments that are considered risky by the users' organizations. An organization's security auditor may periodically send such emails to users to test their preparedness. Second, a Browser-IPA can punish users who click on links leading to potentially dangerous websites after ignoring that application's warnings. The links could have been purposefully inserted by the user's organization on webpages that users visit often. Third, a Pola-IPA (see section 5.2.5) can penalize users who use conventional applications to perform risky actions instead of versions of such applications with restricted capabilities as advocated by the principle of least authority.

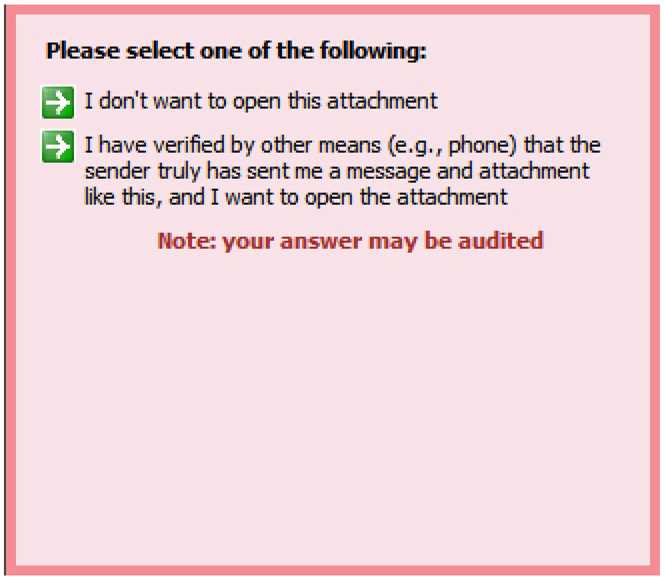
## 6.3 AN EMBODIMENT OF INSECURITY-PUNISHING APPLICATIONS ON E-MAIL CLIENTS

This section elaborates on how to employ the concepts of insecurity punishment to create applications that help users prevent email-borne virus propagation. In this context, an organization's policy specifies which attachments are considered unacceptably risky, and a deployed Email-IPA punishes users who open those attachments by selecting untruthful answers on warning dialogs. Such answers can be reviewed by the organization's security auditors who then make the decision about punishing the user.

### 6.3.1 AUDITED DIALOGS

Our Email-IPA first tries to help users behave securely by guiding them in the process of identifying unjustified risks. For this purpose, it uses context-sensitive guiding dialogs (CSG). However, we extend such dialogs to also notify users about the consequences of providing untruthful answers. First, each dialog that accepts user input is modified to notify users that their answers may be audited. For example, the dialog shown in Figure 6.1 is the audited version of the dialog in Figure 4.8. Second, when appropriate, a final confirmation dialog is added (see Figure 6.2). This dialog notifies the user that confirmation of the operation will cause the user's answers and its context (e.g., message and attachment in case of e-mail) to be forwarded to the organization's auditors. This dialog also summarizes possible consequences to the user if auditors find that the user's answers are unjustified in the respective context. For example, auditors may suspend the user, require the user to pay a fine, or require the user to pass remedial training.

We refer to any security dialog that incorporates the latter two measures as an *audited dialog*. Audited dialogs alone suffer from the same drawbacks of any fixed dialog. Thus, we leverage the results discussed in previous chapters by adding polymorphism; we call the resulting dialogs *polymorphic audited dialogs* (PAD). PADs are important components of IPAs for several reasons. First, punishments applied to users without previous notification that they will occur as consequence of users' inputs to security dialogs would look arbitrary to users and result in frustration. This is clearly an undesirable outcome. Second, the audit trail created by PADs provides information that enables auditors or automated auditing software to determine whether users' inputs to dialogs are unjustified with respect to the security policy. When polymorphic audited dialogs also incorporate CSG, we refer to them as CSG-PAD.

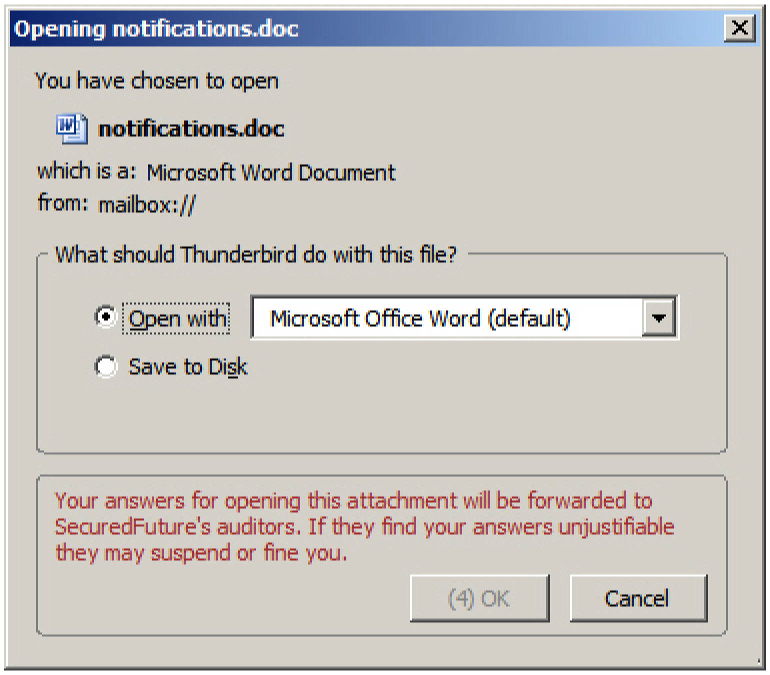

**Please select one of the following:**

I don't want to open this attachment

I have verified by other means (e.g., phone) that the sender truly has sent me a message and attachment like this, and I want to open the attachment

**Note: your answer may be audited**

**Figure 6.1:** Audited version of the dialog in Figure 4.8


Opening notifications.doc

You have chosen to open

notifications.doc

which is a: Microsoft Word Document

from: mailbox://

What should Thunderbird do with this file?

Open with [✓] Microsoft Office Word (default)

Save to Disk

Your answers for opening this attachment will be forwarded to SecuredFuture's auditors. If they find your answers unjustifiable they may suspend or fine you.

**Figure 6.2:** Final confirmation dialog for operation and forwarding user's answers to organization's auditors

### 6.3.2 PUNISHING STIMULI

Auditors can use different types of penalties to punish the user if auditors find the user's behavior unjustifiably insecure. In our case, the auditor instructs the IPA to suspend a user's access to email for a specified amount of time. While a user is suspended, the user cannot use the application normally (see Figure 6.3). The Email-IPA will only display the auditors' notice and explanation of failed audit and penalties (see Figure 6.4). Penalties for accepting unjustified risks monotonically increase with each subsequent violation. For example, the user may be suspended for increasing periods, and after a certain number of violations may also need to pay increasing fines. The latter is a form of *response cost*, which is a type of punishing stimuli that has been proved to be as effective as physically-intense stimuli [123, p. 392]. Thus, we fulfill the intensity

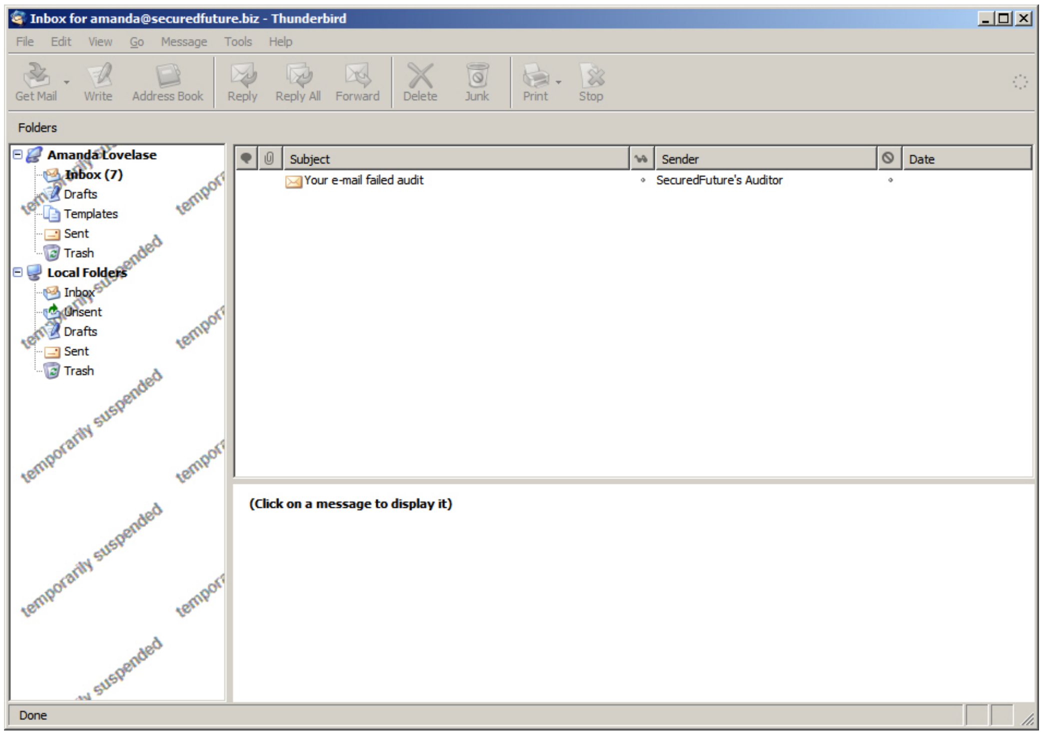
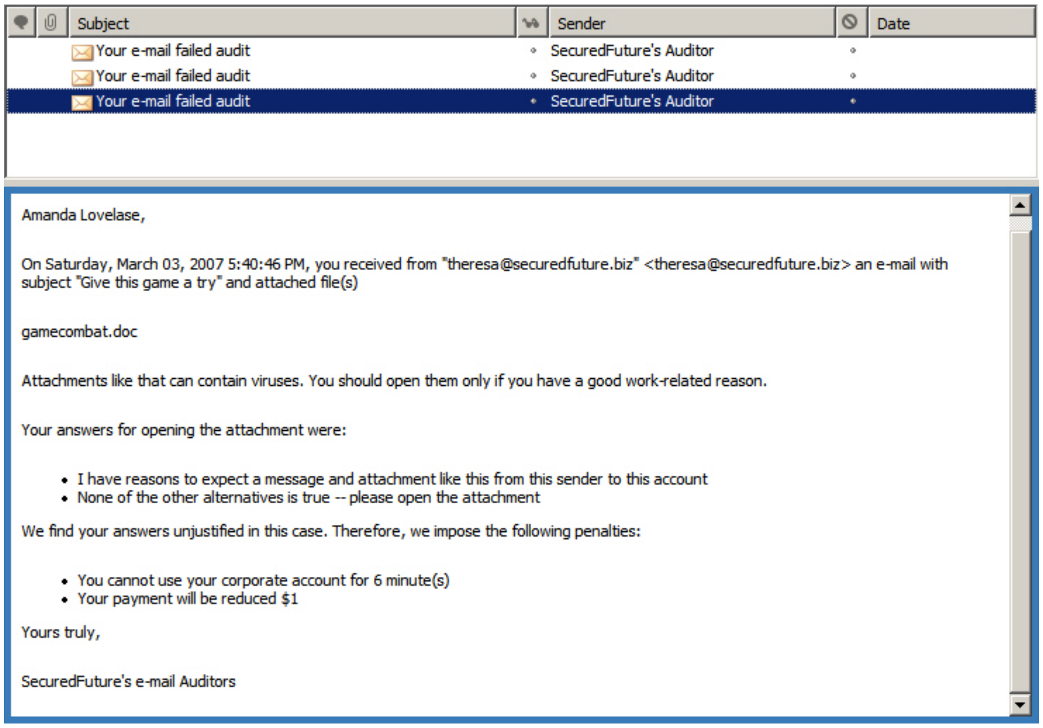


**Figure 6.3:** Thunderbird's screen while user is suspended. The user can access the auditors' notice of failed audit and penalties, but no other messages

requirement mentioned earlier that is necessary for stimuli to be actually punishing.

### 6.3.3 OPERATION

IPAs may require an organization's privacy policies to grant the organization's auditors the right to read members' answers and context information relevant to security decisions (e.g., email messages and attachments). An organization's members might attempt to evade auditing by using instead external (e.g., Web-based) email accounts. We assume that an organization's firewalls block direct communication between members' computers and external email servers. Such privacy policies and blocking are quite common in corporate environments. The latter ensures that users cannot escape punishment if they behave insecurely. Moreover, we implemented



| Subject | Sender | Date |
|---|---|---|
| Your e-mail failed audit | SecuredFuture's Auditor | |
| Your e-mail failed audit | SecuredFuture's Auditor | |
| Your e-mail failed audit | SecuredFuture's Auditor | |

Amanda Lovelase,

On Saturday, March 03, 2007 5:40:46 PM, you received from "theresa@securedfuture.biz" <theresa@securedfuture.biz> an e-mail with subject "Give this game a try" and attached file(s)

gamecombat.doc

Attachments like that can contain viruses. You should open them only if you have a good work-related reason.

Your answers for opening the attachment were:

- I have reasons to expect a message and attachment like this from this sender to this account
- None of the other alternatives is true -- please open the attachment

We find your answers unjustified in this case. Therefore, we impose the following penalties:

- You cannot use your corporate account for 6 minute(s)
- Your payment will be reduced $1

Yours truly,

SecuredFuture's e-mail Auditors

**Figure 6.4:** Notice and explanation of failed audit and penalties in Thunderbird, after user's acceptance of unjustified risks for a third time

mechanisms for preventing punishment circumvention by restarting the IPA.

The processes of auditing users' answers to security dialogs and enforcing penalties require an authenticated channel between the organization's auditors and the IPAs installed at each computer of organization members. An email agent can implement such a channel by automatically adding or verifying a signature (if using public-key cryptography) or message authentication code (if using shared secrets) to the messages sent between the application and auditors.

Manually assessing whether the organization's members' answers to security dialogs are aligned with the organization's policy can be labor-intensive for auditors if the amount of email traffic that members generate is large. Auditors can, instead, send their organization's members *test messages* containing attachments that auditors *a priori* consider unjustified risks. Judging members' responses to test messages can be automated and therefore may be easier than evaluating responses to other messages. For instance, auditors can send test messages including a header indicating the type of risk that the email message represents. An IPA can recognize such header and take an appropriate action based on whether the user accepts the message. Test messages also encroach less on users' privacy and facilitate delivering punishing stimuli very soon after the insecure behavior. The latter characteristic has been found to be important for the effectiveness of punishing stimuli [123].

## 6.4 COMPARISON TO SECURITY-REINFORCING APPLICATIONS

A priori, insecurity-punishing applications could be expected to be about as effective as security-reinforcing applications. Organizations could use SRAs or IPAs depending on organization or member peculiarities. For example, some organizations might prefer to use IPAs for users whose insecure behaviors persist despite use of SRAs. Previous results in Operant Conditioning suggest that members would prefer reinforcement to punishment. However, this has not been verified before in a software setting. To this end, we test the following hypothesis:

*Hypothesis 7. Users who interact with security-reinforcing applications have similar rates of both justified-risk acceptance and unjustified-risk rejection as users who interact with insecurity-punishing applications, complete tasks in the same amount of time, and are more satisfied with the user interface.*

## 6.5 EVALUATION METHODOLOGY

We performed a user study to compare IPAs with (i) a conventional application using no-warning (NW) dialogs, and (ii) SRAs. This study shares the same scenarios and email messages (section B.3 of Appendix B) that our user study for testing CSG-PD employed, as it was performed closely afterward. For the present study we used a within-subjects design with two conditions, control and IPA. In the control condition, participants performed one of the scenarios (randomly selected) using NW dialogs (unmodified Mozilla Thunderbird 1.5). In the IPA condition participants role-played the other scenario while interacting with an insecurity-punishing application. The latter was implemented on top of Mozilla Thunderbird 1.5. Participants always used NW first to avoid learning effects. Participants were already familiar with NW at the beginning of the study. Consequently, there is nothing new that participants might have learned from NW and applied to IPA.

We measured (1) the counts of justified and unjustified risks each participant accepted in

**Table 6.1:** Characteristics of the participants

| | |
|---|---|
| # Participants | 7 |
| # Female | 6 |
| # Male | 1 |
| Familiarity with email agents (self-reported) | 3.9 / 5 |
| Ease of user study tasks (self-reported) | 4.3 / 5 |
| # Unjustified risks accepted in control condition | 66% |

each condition, and (2) the time each participant took for completing a scenario's tasks during each condition. We recruited participants for this experiment using the same communication channels as the CSG-PD's experiment and employed the same eligibility criteria. We did not schedule people who participated in any of our other experiments (Chapters 4, 5, and 7).

Each participant role-played the two scenarios described in Section 4.6.2 in an individually scheduled laboratory session. Each participant's session lasted between 31 and 103 minutes. Only those participants who accepted at least half of the unjustified risks in the scenario role-played in the control condition progressed to the IPA condition. This criterion excluded 1 of 8 participants recruited for the study (12.5%). The excluded participant accepted 37.5% of the unjustified risks. Table 6.1 summarizes characteristics of the participants who progressed to the IPA condition. After finishing the scenarios, participants who interacted with the IPA were asked to complete an exit survey.

We tested our IPA with the following penalty policy. On the first violation, the participant was suspended for 3 minutes. On the second violation, the participant was suspended for 6 minutes. For each subsequent violation, the participant was suspended for 6 minutes and $1 was subtracted from the participant's compensation. For consistent testing conditions, we programmed our IPA to automatically detect acceptance of an unjustified risk and generate the corresponding user suspension message 7 seconds thereafter. The template used for the auditors' messages can be found in section C.1 of Appendix C.

## 6.6 EXPERIMENTAL RESULTS

Table 6.2 summarizes the main results of our user study. It shows that, compared to NW dialogs (control condition), IPA provides a statistically significant and large reduction in the number of unjustified risks accepted (p-value = 0.008, d = 2.60). In addition, IPA had no effect on the number of justified risks accepted and had no significant effect on tasks completion time. We plot average hit (justified risk accepted) and false alarm (unjustified risk accepted) rates in Figure 6.5.

Figure 6.6 shows how the net acceptance frequency of unjustified risks evolved with continued use of IPA (which uses CSG-PAD). Net acceptance frequency of CSG-PD is included for reference. The graph shows that, after having accepted 2 unjustified risks with the IPA, users realized that unjustified risks require higher efforts: users have to pay more attention to the dialogs' options if they do not want to be penalized. This finding is consistent with current literature [67] that states that punishment can be effective in weakening undesired behavior in as few as two trials. Interaction with the IPA decreased net acceptance frequency for the remaining unjustified

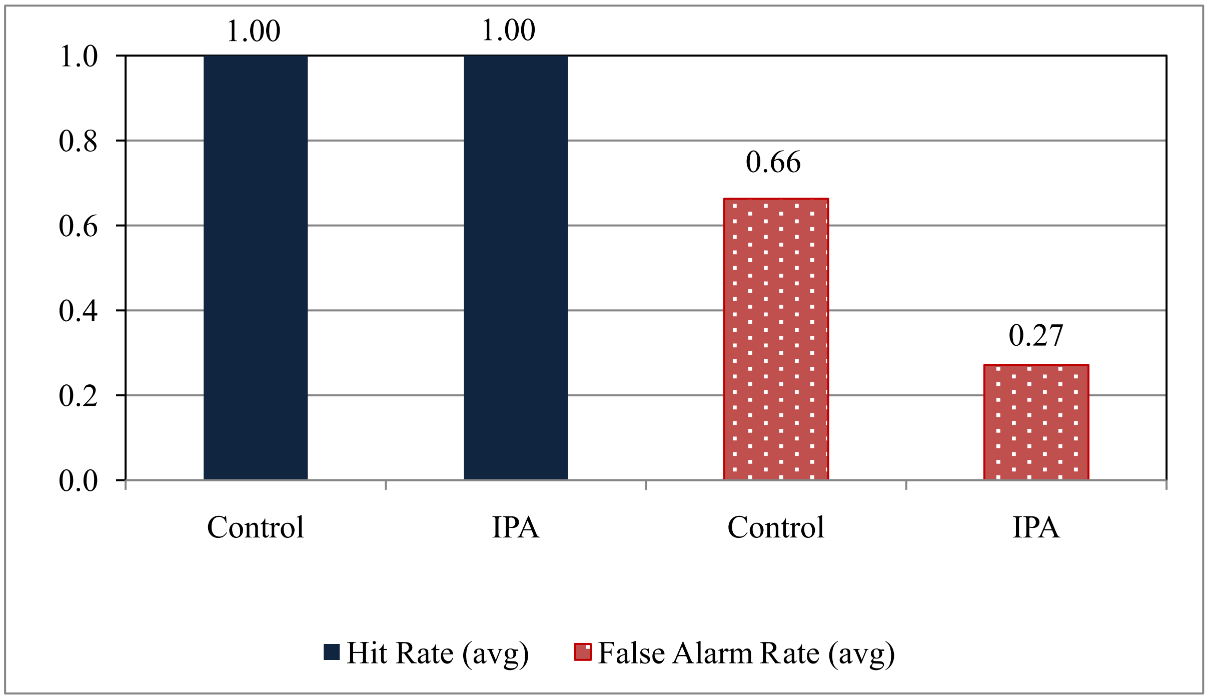


**Figure 6.5:** Average Hit and False alarm rates

risks on average by 58% (compared to CSG-PD's 36%).

These results confirm Hypothesis 6.

Table 6.4 shows the results of a survey completed by participants at the end of their sessions. They would be neutral about receiving IPA's guidance in the future, but would be unlikely

**Table 6.2:** Comparison between IPA and control

p-values were calculated using Wilcoxon's signed-ranks test (* = significant)

| | **Control** | **IPA** |
|---|---|---|
| # participants | 7 | |
| | *# of justified risks accepted* | |
| Mean | 2.00 | 2.00 |
| Std. Dev | 0 | 0 |
| p-value | 1.000 | |
| | *# of unjustified risks accepted* | |
| Mean | 5.29 | 2.14 |
| Std. Dev | 0.76 | 1.46 |
| p-value | 0.008* | |
| Effect size (Cohen's d) | 2.60 | |
| | *Time to complete tasks (minutes)* | |
| Mean | 29.72 | 27.60 |
| Std. Dev | 19.21 | 9.94 |
| p-value | 0.81 | |

**Table 6.3:** Comparisons with SRA's conditions

p-values were calculated using a two-sided Mann-Whitney test (*=significant)

| | IPA vs SRA-Learning | IPA vs SRA-Maintenance | IPA vs SRA-Extinction |
|---|---|---|---|
| | *Acceptance of Justified Risks* | | |
| p-value | 1.00 | 1.00 | 1.00 |
| effect size | -- | -- | -- |
| | *Acceptance of Unjustified Risks* | | |
| p-value | 0.015* | 0.014* | 0.004* |
| effect size | 1.18 | 0.97 | 2.43 |
| | *Time* | | |
| p-value | 0.036* | 0.004* | 0.001* |
| effect size | 0.93 | 1.61 | 2.19 |

to recommend the IPA to a friend. Further questioning revealed that some participants disliked IPA's penalties or found that they had been applied unfairly. For example, some participants automatically trusted messages supposedly sent by a coworker and found it hard to conceive that such messages might be forged. The auditors' messages did not explain sufficiently well to these participants why they failed audit.

To test Hypothesis 7, we compared results of the present experiment with results of our experiment with a security-reinforcing application (SRA). We were able to compare the two experiments because the tasks assigned to participants in both cases were the same (participants role-played the same scenarios).

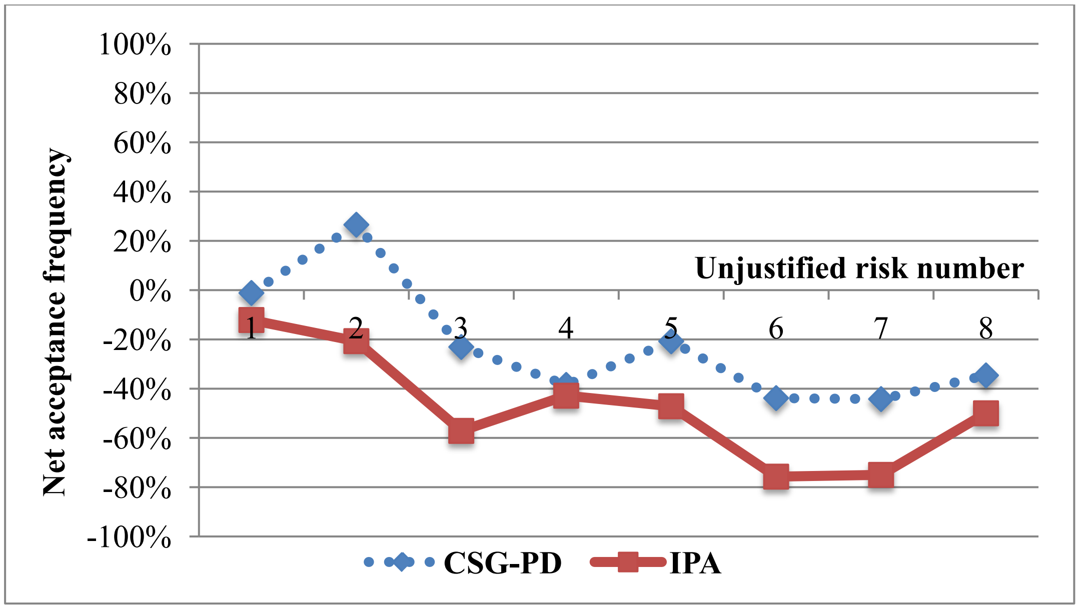


**Figure 6.6:** Unjustified-risk net acceptance frequency decreases after user figures out higher efforts imposed by CSG-PD or IPA

**Table 6.4:** Participant perceptions of IPA

(worst = 1, best = 5)

| | |
|---|---|
| Dialogs are easy to understand | 3.7 |
| Questions are helpful | 2.1 |
| Interface provides good guidance | 2.6 |
| Participant followed guidance | 2.4 |
| Would feel comfortable receiving such guidance in future | 3.0 |
| Would recommend to friend | 1.9 |

**Table 6.5:** Comparison of average participants' perceptions about IPA and SRA

(worst = 1, best = 5) p-values were calculated using a one-sided Mann-Whitney test (*=significant)

| | **IPA** | **SRA** | **p-value** | **eff. size** |
|---|---|---|---|---|
| Dialogs are easy to understand | 3.7 | 4.4 | 0.08 | -- |
| Questions are helpful | 2.1 | 3.1 | 0.039* | 1.03 |
| Interface provides good guidance | 2.6 | 3.8 | 0.02* | 1.40 |
| Participant followed guidance | 2.4 | 3.2 | 0.16 | -- |
| Would feel comfortable receiving such guidance in the future | 3.0 | 3.7 | 0.22 | -- |
| Would recommend to friend | 1.9 | 3.4 | 0.006* | 1.53 |

When evaluating SRA, the number of both justified and unjustified risks was 5, whereas in the present experiment they were 8 and 2 respectively[1]. In addition, different participants were involved. To make fair comparisons, we made the following adjustments. First, we calculated false alarm (unjustified risks accepted) rates for each of the conditions in both experiments, including control conditions. Second, we subtracted false alarm rates of the IPA, SRA-learning, SRA-maintenance, and SRA-extinction conditions from the false alarm rates in their respective control conditions. This second step was performed to avoid possible biases because of a priori differences between groups (e.g., more skilled or risk averse participants in one group than the other). Third, we did a similar adjustment for hit rates.

We compared the differences in rates using a two-sided Mann-Whitney test. This

[1] This change was because in the case of IPAs, (punishing) consequences were made contingent only upon acceptance of URs, whereas in the case of SRAs, (reinforcing) consequences were made contingent upon both rejection of URs and acceptance of JRs. Thus, in the latter case, more JRs were needed to more faithfully assess the effect of reinforcement on the acceptance of that type of risks.

non-parametric test is appropriate for comparing non-related samples without making assumptions about their underlying distribution or direction of improvement. Noted effect sizes are Cohen's d, and were calculated using pooled standard deviations. The results of the test for differences in false alarm rates (unjustified risk accepted) revealed that SRA's learning (p-value=0.015, d=1.18), maintenance (p-value=0.014, d=0.97), and extinction (p-value=0.004, d=2.43) conditions provided significantly greater (and large) improvements than did IPA. These results are illustrated in Table 6.3. We found insignificant differences when we applied the two-sided test to the differences in hit rates (justified risks accepted) in both experiments, as we expected.

We also compared the time it took for participants in the IPA and SRA conditions to complete the assigned tasks. To do this comparison we used a two-sided Mann-Whitney test. We found that there was a significant and large reduction in time spent completing tasks in the SRA-learning (p-value=0.036, d=0.93), SRA-maintenance (p-value=0.004, d=1.61), and SRA-extinction (p-value=0.001, d=2.19) conditions compared to the IPA condition. Table 6.3 also shows these findings.

Average results of the exit survey are shown in Table 6.5 for both IPA and the first session of the SRA experiment (we found no significant difference between the latter scores and the second SRA session's). Based on these averages, it can be observed that participants' reactions to the SRA interventions were more favorable than to IPA. Those improvements were expected and thus we determined their significance using a one-sided Mann-Whitney test. The results reveal that participants thought that the questions in the guidance were significantly more helpful (p-value=0.039), and that the user interface provided significantly better guidance (p-value=0.02) than did participants who used IPA. Moreover, participants who interacted with SRA were significantly more inclined to recommend it to a friend (p-value=0.006) than were participants who interacted with IPA. All these improvements in participants' perceptions were large (d=1.03, d=1.40, and d=1.53 respectively).

Experimental results were therefore better than expected in Hypothesis 7. As expected, SRA's user acceptance was significantly better than IPA's, and JR acceptance was not different. In addition, and unexpectedly, SRA significantly reduced acceptance of unjustified risks and task completion time. The latter result can be due to the fact that participants did not spend time during suspensions in the SRA study unlike in the IPA experiment.

## 6.7 SUMMARY

The use of punishment is common in organizations, but only for behaviors that negatively affect production tasks which are of primary concern. In this chapter we explored the feasibility of using punishment for behaviors that compromise information systems' security. The latter is seen by users as a secondary task. Toward that goal, we proposed and evaluated insecurity-punishing applications (IPAs). IPAs allow organizations to hold users accountable by applying penalties to those who emit unjustifiably insecure behaviors. We implemented an IPA with CSG and audited dialogs (CSG-PAD) on Mozilla Thunderbird.

Results from a user study show that users accept significantly fewer unjustified risks with IPA than with an application that uses conventional dialogs. Furthermore, we found that IPA has insignificant effect on acceptance of justified risks and time to complete tasks. Users' perception of IPA (Table 6.4) was lukewarm and it appears unlikely that users would adopt them spontaneously. However, it appears that users would accept them if adopted by an organization.

Audit decisions that users do not understand can generate resentment. The security concepts underlying an audit decision need to be explained in plain language, so that users can learn from the notification itself. We take these recommendations into account to design an improved version of the present chapter's auditors' message for a user study described in the next chapter about vicarious-conditioning of security behaviors.

On the other hand, the reduction of unjustified risk acceptance achieved with IPAs was smaller than that of security-reinforcement applications (SRAs). Although neither technique negatively impacted acceptance of justified risks or time needed to complete tasks, SRAs enjoyed better user approval than IPAs. In light of the latter results, organizations are advised to use punishment only as a last resort when CSG-PD alone or in conjunction with SRAs does not thwart certain insecure behaviors.

# 7.0 VICARIOUS CONDITIONING OF SECURITY BEHAVIORS

Previous chapters covered two techniques, security-reinforcing (SRAs) and insecurity-punishing applications (IPAs), that computer scientists and software engineers can apply to make computer systems more secure. Vicarious conditioning (VC) interventions can be used to minimize users' errors when conditioning users' behavior with SRAs or IPAs and to accelerate the conditioning process. We will explore two types of VC: vicarious security reinforcement (VSR), and vicarious insecurity punishment (VIP). In the former, secure behavior of a model is reinforced, while in the latter, insecure behavior of a model is penalized. These interventions use modeling not only to help users learn how to behave more securely, but also to encourage them to apply their acquired skills. We empirically tested these interventions in two user studies and found that participants who watched the VSR and VIP interventions before interacting with SRAs and IPAs, respectively, improved their security behaviors faster than participants who interacted with those applications alone. Moreover, we found that (a) the VIP intervention improved IPA's user acceptance, and (b) although training with VSR before a user interacts with a SRA speeds up learning, once a user has learned the desired behaviors, the VSR advantage vanishes.

## 7.1 INTRODUCTION

Previous chapters have shown that security-reinforcing applications (SRAs) are effective tools for strengthening secure behaviors. However, when interacting with SRAs, computer users need to actually experience a situation in which they will be reinforced after securely handling a security risk. That is, users are not reinforced until after they have behaved securely. Thus, users may accept several unjustified risks or reject several justified risks before they receive a reward. This has at least two undesirable implications. First, it may take some time for users to understand the association between secure behavior and reward. For instance, after rejecting an unjustified risk and being rewarded, the user may not realize that she can also be rewarded for rejecting all risks of the same type. Another case is when a user accepts an unjustified risk but doesn't realize that she can still be rewarded if she rejects that same risk afterward. 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 for these problems 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, or may fail to see the benefit of applying those rules and could simply ignore them [89]. Vicarious security reinforcement can help emphasize the desirable consequences of secure behavior without waiting until a user realizes that those consequences exist when he emits the secure behavior. 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.

Likewise, when interacting with insecurity-punishing applications (IPAs), a person will need to actually behave insecurely to learn that a behavior is punishable. This may unnecessarily slow down the learning process and can cause frustration in people who, unaware of the consequences, inadvertently behave insecurely. Vicarious insecurity punishment can help people learn what behaviors they should not enact or imitate. When observed before a person interacts with an IPA, it should reduce the likelihood of emitting insecure behaviors and might reduce frustration caused by sanctions otherwise seen as arbitrary. However, the effectiveness of vicarious learning for speeding up the process of learning to avoid insecure behaviors has not been evaluated before in a software context. If confirmed, this result would be worthwhile since it can help reduce security incidents caused by users' security failures. It may also reduce reluctance to interact with IPAs.

This chapter introduces and evaluates principles and techniques for creating vicarious-conditioning interventions to promote secure behaviors and discourage insecure behaviors. When models are reinforced, we call the intervention vicarious security reinforcement (VSR), and when they are punished, we refer to the intervention as vicarious insecurity punishment (VIP). We empirically tested whether participants who watch a VSR intervention before interacting with an SRA and participants who watch a VIP intervention before using an IPA learn how to behave securely faster than those who interact with the applications alone. We found that, indeed, these interventions accelerate the two applications' security benefits and, in the case of IPAs, the user acceptance.

Interventions like VSR and VIP can be deployed in organizations before users interact with SRAs or IPAs. It is already customary for information technology departments to use collateral resources such as instructional materials and tutorials to help users learn how to operate deployed computer applications. Among other uses, the techniques and principles provided in this chapter can help these departments to design equivalent collateral materials for SRAs or IPAs when deploying these applications. However, unlike traditional collaterals, our vicarious interventions can also help organizations (i) shape their members' security behaviors, and (ii) align such behaviors to the organizations' security policies.

The remainder of this chapter is organized as follows. Section 7.2 presents recommendations for creating vicarious interventions that have been successful in past research about non-security behaviors. Section 7.3 illustrates how to tailor this advice for creating interventions to effectively condition security behaviors vicariously. Section 7.4 presents our hypotheses about the effectiveness, impact on productivity, and user acceptance of SRAs and IPAs when users watch vicarious security-conditioning interventions before using such applications and when users do not. Sections 7.5 and 7.6, respectively, detail our evaluation methodology and experimental results. Section 7.7 discusses those results and, finally, section 7.8 summarizes the chapter.

## 7.2 CHARACTERISTICS OF VICARIOUS-CONDITIONING INTERVENTIONS

In this section, we describe the features that have been successfully used to create vicarious interventions for non-security behaviors, grouped by the four sub-processes that govern vicarious learning: attention, retention, reproduction, and motivation (Chapter 2). For this, we condensed advice from research works about vicarious interventions that promoted learning of general behaviors [2][4] and production-related behaviors [91][9][90][37].

**Attention**. Three aspects have been identified as being influential in getting and maintaining an observer's attention, namely, the model, the observer's characteristics, and the modeling display [9]. These have been called "modeling enhancers", and we examine each in turn. First, regarding the models, there are two types used frequently in vicarious learning interventions: coping models and mastery models [91]. 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. Interventions using each type of model have produced desired results. Other recommendations in the relevant literature about models and their characteristics, are that the model should preferably be of the same sex and race than the observer [9], that several different models be utilized, and that at least one "high status" model be included. Second, the characteristics of the observer must be taken into account. Individuals' characteristics may inhibit or facilitate the model's ability to "first, gain and hold the observer's attention and, second, influence his or her behavior in a constructive direction" [33]. Third, there are several ways to display a vicarious-conditioning intervention. For instance, live performances and videos. Experts (e.g., [91][9]) argue that, for maximizing a modeling intervention effectiveness, the modeling display should portray behaviors to be modeled (a) vividly, and in a detailed way, (b) from least to most difficult behaviors, (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, and (f) with a length of between 5 and 20 minutes.

**Retention**. There is a natural lag between the time a person observes a models' behavior and the time she has the opportunity to either engage in the behavior (if reinforced), or refrain from doing so (if punished). Thus, retention of details is important for the person to remember which behaviors she can enact and which she shouldn't. Several studies (e.g., [90][37]) 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.

**Reproduction**. It is crucial that observers be able to enact the behavior modeled in the vicarious intervention. If the individual lacks basic sub-skills necessary to enact a modeled behavior, this needs to be detected and remedied before the person is able to reproduce such behavior.

**Motivation**. As mentioned in Chapter 2, social learning theory (which observational learning is part of) draws a distinction between acquisition and behavior since observers will not apply everything they learn [4]. To ensure enactment of modeled behaviors, it is necessary to make desired consequences (reinforcement) contingent upon such behaviors. Likewise, to deter undesired behaviors, aversive consequences (punishment) should be made contingent upon such behaviors.

## 7.3 VICARIOUS SECURITY-CONDITIONING FOR EMAIL PROCESSING

In this section we explain how to apply the recommendations outlined in the previous section for creating effective vicarious interventions for security behaviors, using handling email securely as a test case. For this, we also considered security research that examined the effect that the behavior of security-minded people has in others [30][1]. Based on all these criteria, we created two vicarious interventions, one showing vicarious security reinforcement (VSR), and the other showing vicarious insecurity punishment (VIP). Henceforth we will refer to our interventions simply as vicarious security-conditioning interventions. We first describe the content of these interventions, and then how they implement the advice given in the previous section about the four processes that govern vicarious learning. Appendix G contains a brief description of the production stage of these interventions, which can be watched at [97].

**Table 7.1:** List of scenes in the vicarious security-conditioning interventions

| **Vicarious Reinforcement** | **Vicarious Punishment** |
|---|---|
| *Scene 1* | |
| Introduces Jack Smith, the main model, and his work environment. Also, Jack's boss is seen giving him documents needed to complete an assignment. The boss mentions that other necessary information will be sent by email and emphasizes that the work needs to be done soon | |
| *Scene 2 (unjustified risk)* | |
| Jack receives an email from Buy.com with which he has an account. The email warns him that his account was compromised and that he needs to fill out the attached form to avoid permanent suspension of his account. Jack rejects the risk after realizing that he did not sign up using his corporate account. | Jack receives an email from iTunes offering a $20 gift card if he completes the attached survey. Jack realizes that this is suspicious since he has never bought anything from iTunes. However, he states that if the file is infected, Tech support should be blamed. Jack then opens the attachment. |
| *Scene 3 (unjustified risk)* | |
| Jack receives an email supposedly coming from a co-worker. The message urges him to open the attached file containing minutes of a meeting that he needs to review. Jack realizes that he has not attended any meeting with the co-worker and then refrains from opening the attachment. | |
| *Scene 4 (justified risk)* | |
| Jack receives an email that mentions that the attached file has been commissioned by Jack's company. Jack states he does not know the person and he is seen about to discard the message. Suddenly, he remembers that his boss notified him about the file, and since he was expecting it, he opens it. Once opened, Jack corroborates that the file is a document related to his job. | |

Our two interventions were produced using videos having four scenes each, and running times of approximately 7 (VIP intervention) and 10.5 (VSR intervention) minutes. Table 7.1 lists the scenes and gives a brief summary of each. The first scene first introduces Jack Smith, the main model in the video, in his work environment (Figure 7.1). Then, it shows him receiving an assignment from his boss (Figure 7.2). The latter hands Jack printed information useful to complete the tasks assigned, and states that other information will be sent by email. Finally, the boss character presses Jack to complete the task as soon as possible. Scenes two, three, and four, each shows the model handling risks of increasing difficulty. In scenes two and three the model handles unjustified risks, while in the last scene he handles a justified risk.

In the second and third scenes, 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 takes an action. In the second scene, such action is rejecting

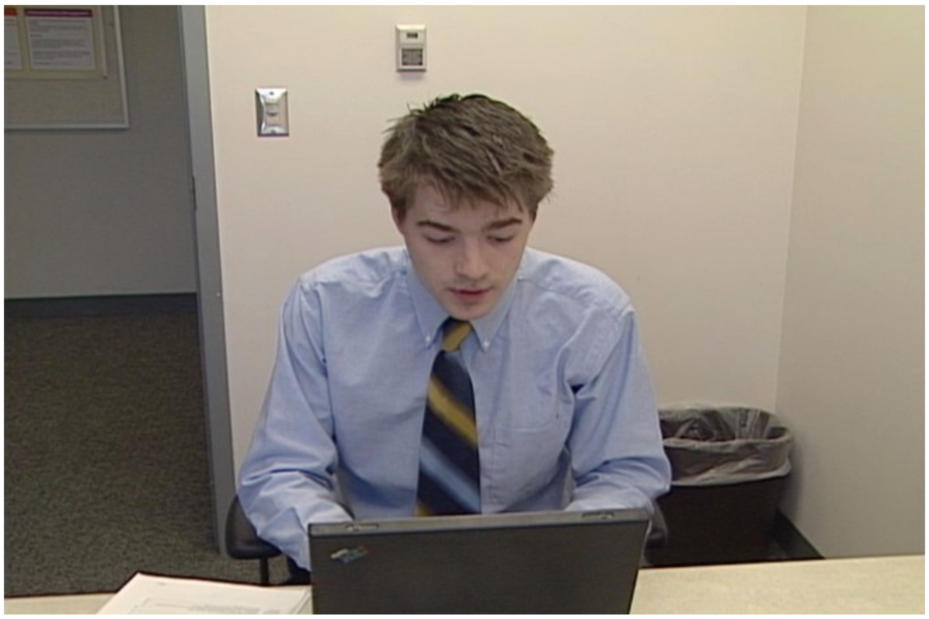


**Figure 7.1: Jack Smith, the main model**

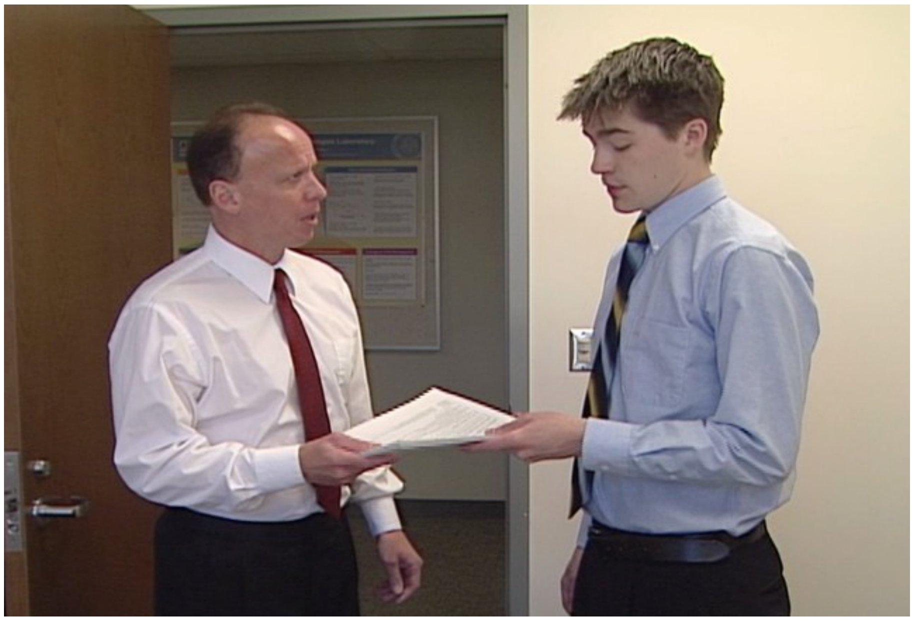


**Figure 7.2: Main model receives an assignment from his boss**

the risk in the vicarious-reinforcement intervention, and accepting it in the vicarious-punishment intervention. In the third scene of both interventions the model rejects the risk.

Finally, in the fourth scene, Jack is initially wary about the justified risk in his Inbox because it was sent from somebody who does not work in Jack's company, and who he does not 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 justified risk to avoid having participants simply learn to reject any risk regardless of it being justified or not.

### 7.3.1 ATTENTION

To maximize observers' attention we chose to use a coping model in our interventions 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][74]). Using coping models has the added advantage that they are usually seen as to have, initially, similar behavior to that of people who may learn from them. Thus, observers can relate to such models and pay attention. The model "thinks aloud" when trying to determine whether a risk is justified or not, and gesticulates accordingly (Figure 7.3). This was designed to make the model's behavior appear detailed and vivid. We did not heed the advice about using a model of same race and gender as observers because we did not want to introduce variability in our interventions depending on the participant being tested. We did, however, use several models, and included at least one high status model. First, in the vicarious reinforcement interventions, we included two extra models acting as the main model's co-workers. When interacting with the main model, they emphasized the desirability of behaving securely. The latter also was intended to convey the idea that secure behavior can be socially acceptable [74]. Second, in both vicarious interventions, we included a model portraying the coping model's boss. The latter is distinguished by age and more formal clothes.

For testing our interventions, we only accepted participants with no computer-technical background, but who had work experience and 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]. On the contrary, it is plausible that non-technical people would be more predisposed to empathize with a coping model, and thus to pay attention to him and his behavior.

Given that in our studies we could schedule only one participant at a time, we chose to portray our interventions using a video medium that is easily reusable.

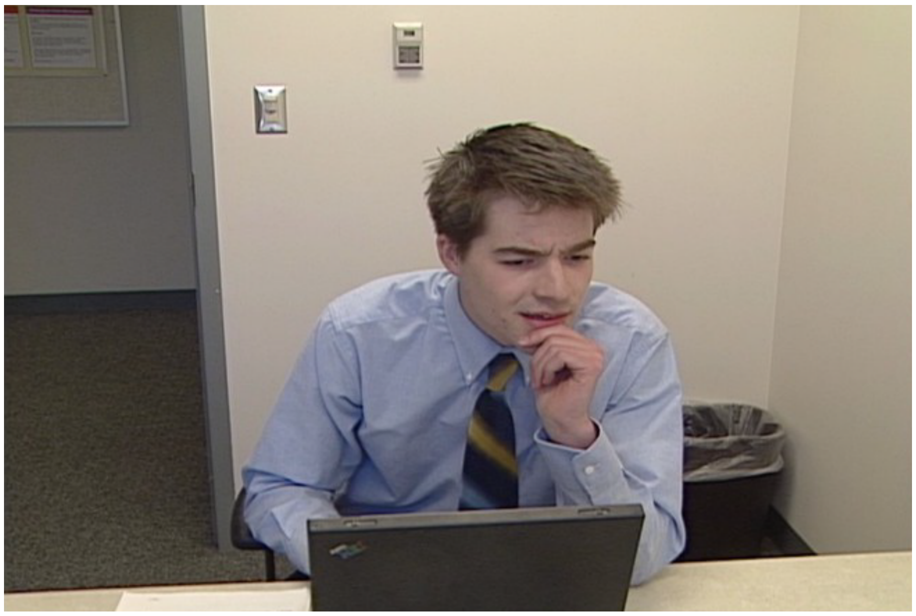
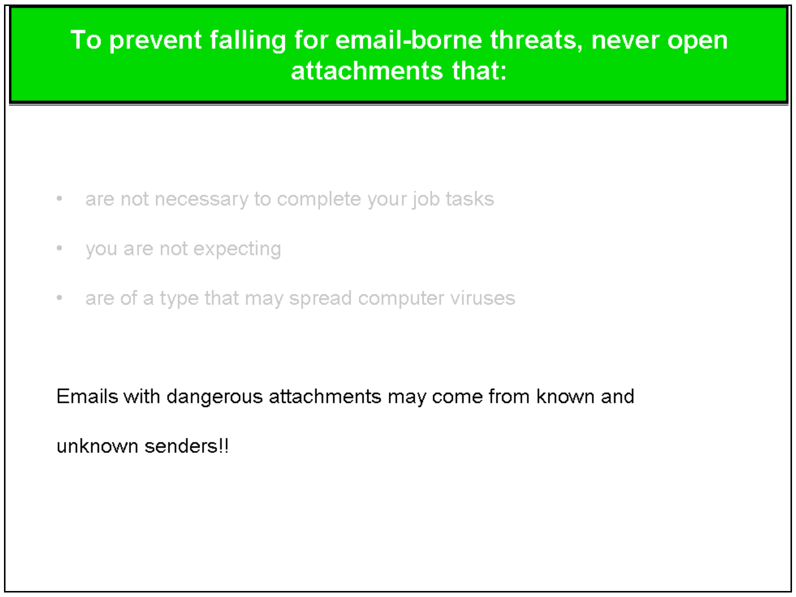


**Figure 7.3:** Model trying to decide whether to accept or reject a risk

### 7.3.2 RETENTION

We implemented the suggestion about the list of "learning points" by showing, after the second, third, and fourth scenes, a summary of the clues that the model used to identify the type of risk, plus additional clues that an observer could use for the same purpose. The clues were shown and narrated one by one, and each became dimmed when the next clue appeared. This was done to help participants focus on the current clue, but without risking forgetting the previous clues. For instance, Figure 7.4 shows the last summary screen shown after the second scene in the vicarious insecurity-punishment experiment. Several of the clues were shown in all three summaries, thus providing the repetitiveness that facilitates learning. Appendix D contains the complete list of clues shown at the end of each scene.

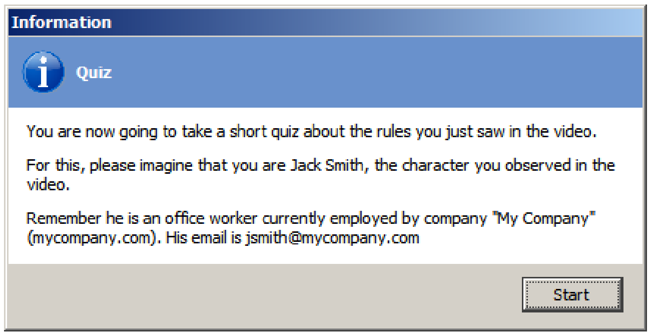

To prevent falling for email-borne threats, never open attachments that:

- are not necessary to complete your job tasks
- you are not expecting
- are of a type that may spread computer viruses

Emails with dangerous attachments may come from known and unknown senders!!

**Figure 7.4:** Last summary screen shown at the end of the second scene of the vicarious insecurity-punishment intervention

To evaluate retention, after the participant finished watching the video, an on-screen quiz, consisting of four questions, was administered. Before starting, 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 remind them that he was an office worker, the name of the company that employed him, and his email address (Figure 7.5). Each question showed a snapshot of an email message, gave some context information related to that email, and asked the user to identify whether it was a justified or unjustified risk. Figure 7.6 shows Question #1, which was the simplest; the complete list of questions is in Appendix E. Half of the questions were about unjustified risks and the other half about justified risks. After the participant answered each question, a message box was shown telling her whether the answer she picked was correct or not, and the reason why. In the former case, the participant was also congratulated (e.g., Figure 7.8 for question #1). In the latter case the message stated that the course of action the user chose was not

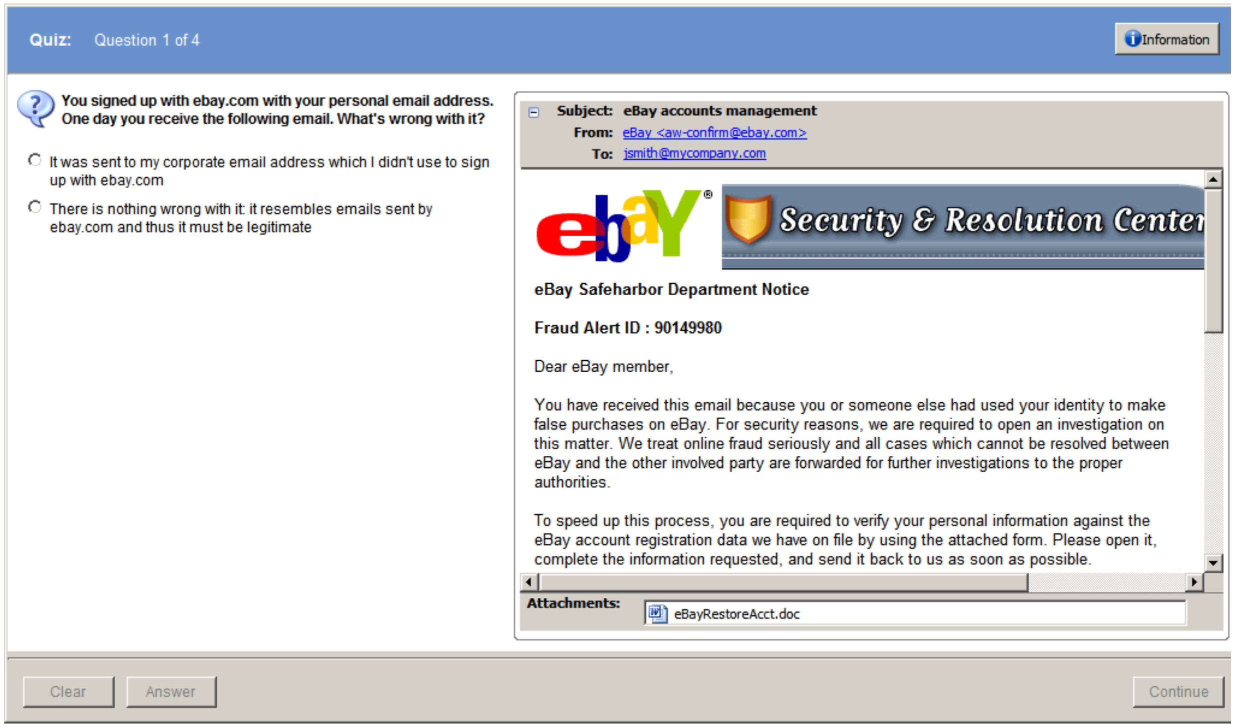


You are now going to take a short quiz about the rules you just saw in the video.

For this, please imagine that you are Jack Smith, the character you observed in the video.

Remember he is an office worker currently employed by company "My Company" (mycompany.com). His email is jsmith@mycompany.com

**Figure 7.5:** Dialog box shown after the video announcing the participant about the quiz



You signed up with ebay.com with your personal email address. One day you receive the following email. What's wrong with it?

- It was sent to my corporate email address which I didn't use to sign up with ebay.com
- There is nothing wrong with it: it resembles emails sent by ebay.com and thus it must be legitimate

Subject: eBay accounts management
From: eBay <aw-confirm@ebay.com>
To: jsmith@mycompany.com

eBay Safeharbor Department Notice

Fraud Alert ID : 90149980

Dear eBay member,

You have received this email because you or someone else had used your identity to make false purchases on eBay. For security reasons, we are required to open an investigation on this matter. We treat online fraud seriously and all cases which cannot be resolved between eBay and the other involved party are forwarded for further investigations to the proper authorities.

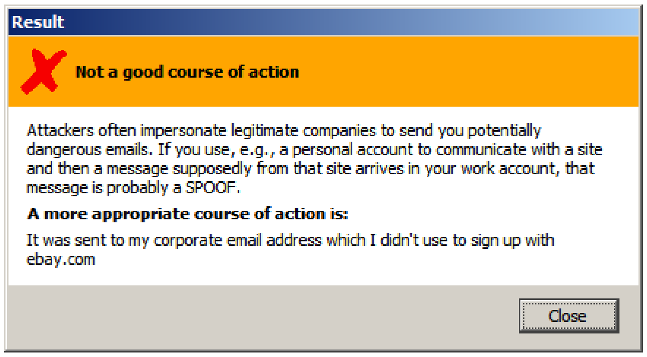
To speed up this process, you are required to verify your personal information against the eBay account registration data we have on file by using the attached form. Please open it, complete the information requested, and send it back to us as soon as possible.

Attachments: eBayRestoreAcct.doc

**Figure 7.6:** First question in the quiz. The button labeled "Information" displays a dialog similar to Figure 7.5

appropriate (e.g., Figure 7.7), but the user was not penalized in any way.

After the participant finished the quiz, a short video was shown explaining users that they should not worry if they did not remember all the rules shown in the video, because they will be



Not a good course of action

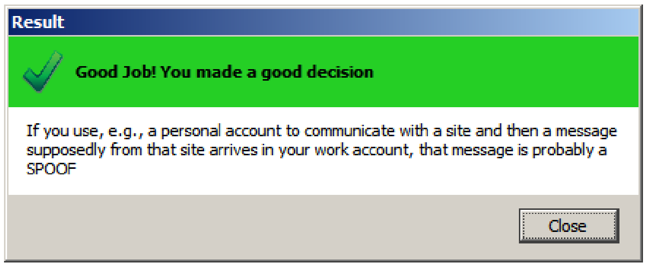
Attackers often impersonate legitimate companies to send you potentially dangerous emails. If you use, e.g., a personal account to communicate with a site and then a message supposedly from that site arrives in your work account, that message is probably a SPOOF.

**A more appropriate course of action is:**

It was sent to my corporate email address which I didn't use to sign up with ebay.com

**Figure 7.7:** Dialog shown when participant answered a question in the quiz incorrectly

In this case the first question was answered incorrectly



Good Job! You made a good decision

If you use, e.g., a personal account to communicate with a site and then a message supposedly from that site arrives in your work account, that message is probably a SPOOF

**Figure 7.8:** Dialog shown when participant correctly answered a question in the quiz.

In this case, the first question was answered correctly

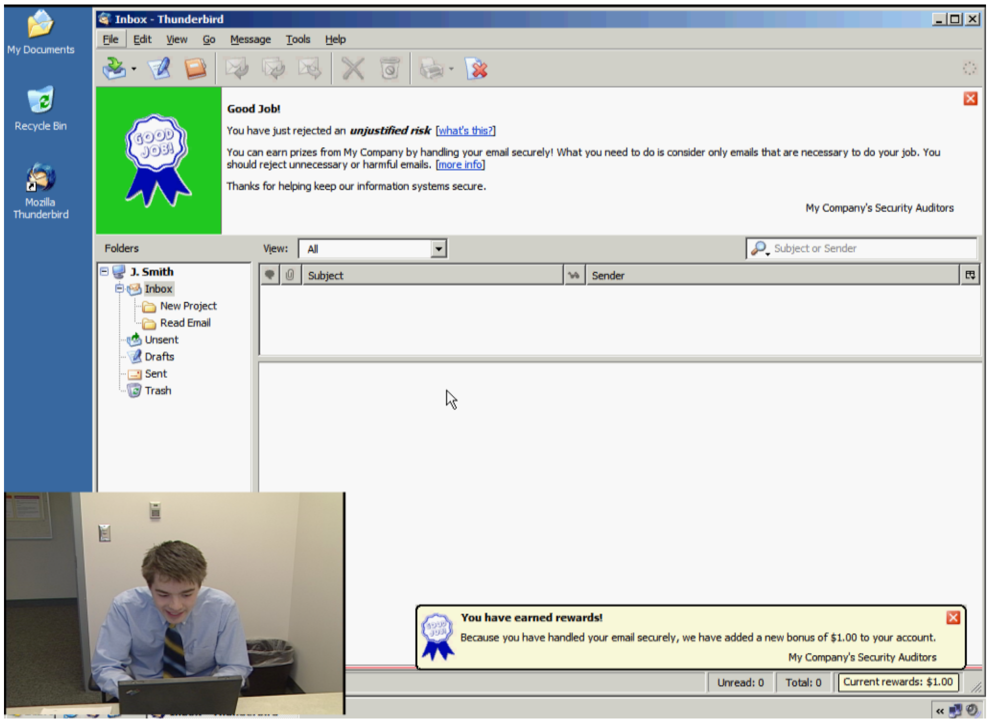
interacting with an email program that used context-sensitive guidance to help the user to apply such rules. Then, a brief tour of the CSG interface was shown (see [97]).

### 7.3.3 REPRODUCTION

In our experiments, the assigned tasks did not require more skills than handling email communications using an email client program, opening attachments, and editing documents using Microsoft Word. Our eligibility criteria during recruitment ensured that participants already had these abilities.

### 7.3.4 MOTIVATION

In the vicarious-reinforcement intervention the model receives the praise and prize rewards (see Figure 7.9) implemented for the security-reinforcing application (SRA) that we evaluated in Chapter 5. These rewards are presented every time the model behaves securely, namely, after rejecting an unjustified risk in the second and third scenes, and accepting a justified risk in the fourth scene. In addition, after receiving the rewards at the end of the second scene, the model invites two co-workers, one female and one male, to see such on-screen rewards (Figure 7.10).



**Good Job!**

You have just rejected an ***unjustified risk*** [what's this?]

You can earn prizes from My Company by handling your email securely! What you need to do is consider only emails that are necessary to do your job. You should reject unnecessary or harmful emails. [more info]

Thanks for helping keep our information systems secure.

My Company's Security Auditors

**You have earned rewards!**

Because you have handled your email securely, we have added a new bonus of $1.00 to your account.

My Company's Security Auditors

Unread: 0 Total: 0 Current rewards: $1.00

**Figure 7.9:** Model is reinforced for behaving securely

(second scene of the vicarious security-reinforcement intervention)

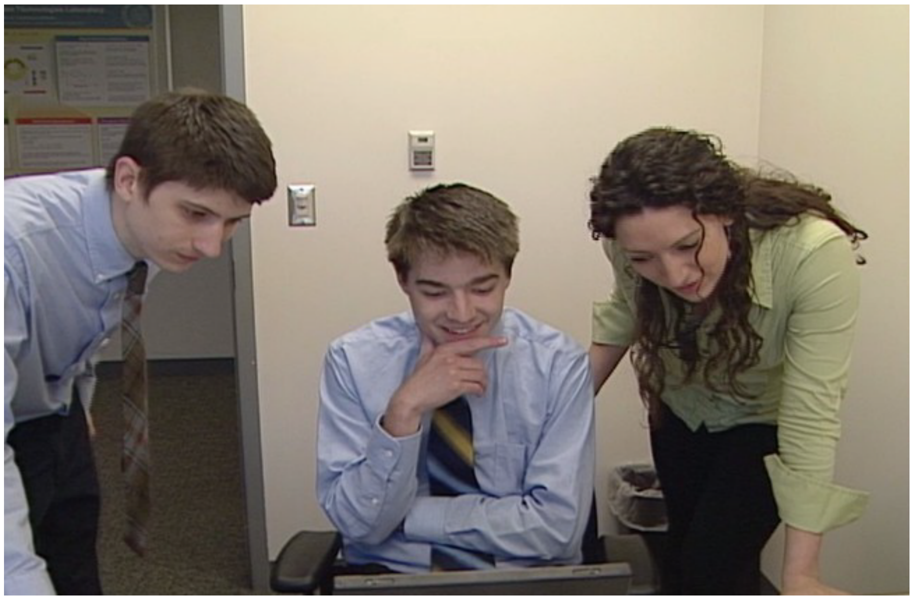


**Figure 7.10:** Model with co-workers seeing the on-screen reinforcing stimuli

(second scene of the vicarious security-reinforcement intervention)

The female model expresses satisfaction and surprise for the company's new practice of rewarding employees for managing their email accounts securely, and asks the male coworker model if he also considers such practice "cool". The male coworker model agrees with that assessment and mentions that he was also rewarded earlier. The main model and the female model show surprise, and the male coworker model reinforces the notion that he will definitively be handling his email account with more care. He also asserts that he will only open attachments that are necessary for doing his job.

The main model's boss character, who has been overhearing part of the conversation when transiting through the hallway, enters into Jack's office and congratulates him for behaving securely (Figure 7.11). He does the same with the male coworker model, and states he is sure the female model will behave securely as well. He finally encourages the models to keep up the good work and leaves. After a brief conversation with the main model about how to use the re-

**Figure 7.11:** Boss congratulates model for behaving securely

(second scene of the vicarious security-reinforcement intervention)

wards they will get for behaving securely, the other two models leave the office.

In the case of the vicarious-punishment intervention, during the second scene, the model correctly identifies that the email he is handling is an unjustified risk. However, he states that it will not be his fault if the attachment he opens is infected. He further states that tech support personnel should ensure that no dangerous emails reach his Inbox, and that they will be blamed if a security breach happens as result of the model's insecure behavior. He then proceeds to open the attachment (a Word document in the video). When he is reviewing it, the email program he was using, which is the insecurity-punishing application evaluated in Chapter 6, enforces the penalty for behaving insecurely. The imposed penalty is a suspension of the model's email use for 3 minutes (Chapter 6). After reading the auditors' email informing him of the suspension, the model verbalizes his concern that he may not be able to finish his work on time. Suddenly, the model playing Jack's boss, who was in the hallway, enters into the office, looks at the screen and realiz-

**Figure 7.12:** Boss verbally reprimands model for behaving insecurely

(second scene of the vicarious insecurity-punishment intervention)

es that Jack has been suspended. The boss then reprimands Jack and tells him that he must be careful with what he opens at work (Figure 7.12). Jack says that he is sorry and that he will be more careful. The boss leaves and after a brief pause the suspension ends. Jack promises he will be more careful from then on, and the on-screen summary appears. In the third scene the model avoids penalties by rejecting an unjustified risk, while in the fourth scene he accepts a justified risk necessary for doing his job. The model neither receives rewards nor is punished in these two scenes.

In our vicarious interventions, the model does not interact with our CSG interface because we preferred to focus on the desired behaviors rather than teaching the user how to use our guiding interface. However, as mentioned earlier, a short video explained the participant that he was going to interact with CSG-PD (in the case of the SRA) or CSG–PAD (in the case of IPA), and that the objective of the guidance was to help him remember the learning points shown in the video (see [97]).

## 7.4 HYPOTHESES

In this section we present hypotheses regarding our vicarious security-conditioning interventions, and explain the rationale behind them.

*Hypothesis 8. When interacting with security-reinforcing applications, users who have previously observed a vicarious security-reinforcement intervention accept as many justified risks and reject more unjustified risks than users who did not observe such intervention, and complete tasks in the same amount of time.*

Social learning theory (which vicarious learning is part of) predicts that a person who observes a model behaving in a specific way and is then rewarded for doing so, can learn to behave in that same way as the model. For this, the model must possess engaging qualities and the observer not only must be capable of emitting the behavior but also must consider the reward desirable. In addition, empirical studies have determined that when a list of "learning points" is made available to the observer when he is trying to reproduce the behavior, retention of what is learned increases [90]. By pairing a vicarious-reinforcement intervention with security-reinforcing applications, we are using social learning theory in a novel way. CSG-PD plays the role of learning points, while the reinforcement delivered by the application can help strengthen the learned behavior.

*Hypothesis 9. When interacting with insecurity-punishing applications, users who have previously observed a vicarious insecurity-punishment intervention accept as many justified risks* *and reject more unjustified risks than users who did not observe such intervention, complete tasks in the same amount of time, and are more satisfied with the user-interface.*

According to social learning theory, observed negative consequences reduce people's tendencies to emit the behavior that was punished and similar ones. By using insecurity-punishing vicarious conditioning before users interact with an IPA, we can achieve a similar effect. This is a novel application of social learning theory to weaken the insecure behaviors that users emit when interacting with computer applications.

## 7.5 EVALUATION METHODOLOGY

In this section we describe the methodology we used to perform two user studies to evaluate our interventions. We used the same procedures to recruit participants described for the evaluation of CSG-PD, SRA, and IPA, and employed the same eligibility criteria.

One user study, which we will refer to as VC_SRA (vicarious conditioning before interacting with a SRA), evaluated the vicarious-reinforcement intervention, and had the same design, first three conditions (control, learning, and maintenance), scenarios, email sets, and passing criteria from control to learning, and from learning to maintenance stages as the experiment we performed to evaluate SRAs (Chapter 5). As the intention of the study was simply to measure any speed up in learning compared to using a SRA alone, no participant was scheduled for a second session, and thus there was no extinction condition. The outline of each session is described next.

First, after signing the consent form, participants received a handout describing one of our scenarios and role-played it with an unmodified email client (Mozilla Thunderbird 1.5).

**Table 7.2:** Characteristics of participants who progressed past control condition

| | VC_SRA | VC_IPA |
|---|---|---|
| # Participants | 12 | 8 |
| # Female | 7 | 5 |
| # Male | 5 | 3 |
| Familiarity with email agents (self-reported) | 4.2 / 5 | 3.9 / 5 |
| Ease of user study tasks (self-reported) | 4.3 / 5 | 4.0 / 5 |
| # Unjustified risks accepted in control condition | 88% | 72% |
| # Had two or more years of work experience | 12 | 8 |

Second, we gave participants a handout that described the other scenario and then asked them to role-play the character described there. Just before participants did so, 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 in there. 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. However, we did not tell participants that they were going to be evaluated with a quiz. This was done to avoid biasing them to pay more attention to the video than they would normally do if there were no quiz. Third, participants watched our vicarious security-reinforcement video, took the aforementioned on-screen quiz, and then watched a short video explaining them that they were going to interact with CSG-PD. Finally, users role-played the scenario using our SRA.

The other user study, which we will refer to as VC_IPA (vicarious conditioning before interacting with an IPA), evaluated the vicarious-punishment intervention, and had the same design, conditions, scenarios, email sets, and passing criterion from control to IPA conditions as the experiment we performed to evaluate our insecurity-punishing application (chapter 6). Participants first role-played one scenario, then watched the video, took the quiz, and were informed about CSG-PAD. We gave them the same instructions about the application of concepts in the video before watching it that we gave in the case of the vicarious-reinforcement study. Likewise, we did not inform them beforehand about the quiz. Finally, participants role-played the second scenario with the same IPA used for previous experiments but with a different auditor email. The new auditor's email incorporated two main changes. First, it gave a brief explanation to users about the type of attachments that it was justified to open. Second, it stated the reason why the specific attachment the user opened was an unjustified risk. With these changes we hoped to avoid the frustration that arose when users deemed their suspension as something arbitrary (Chapter 6). Section C.2 in Appendix C contains the template used for the improved auditors' email.

In both studies, participants who role-played a second scenario were asked to complete an exit survey. Table 7.2 summarizes characteristics of these participants. Before they completed the survey, we stressed that it was anonymous and encouraged them to be as honest as possible. The survey was the same one used for the CSG-PD, SRA, and IPA experiments, but we added some questions to collect participants' opinions of the video interventions. Finally, we debriefed participants, and paid them between \$15 (if they did just the control condition) and \$22. Participants of our other experiments (Chapters. 4, 5, and 6) were not eligible for these studies.

## 7.6 EXPERIMENTAL RESULTS

This section present the results obtained in two user studies that evaluated our vicarious conditioning interventions. First, we present results related to the interventions' effectiveness in improving end-users' security behaviors, and their impact on users' productivity. Second, we present and comment about participants' impressions of such interventions. Third, to test hypotheses 8 and 9, we compare these quantitative and qualitative results to those obtained with our SRA and IPA interventions respectively.

### 7.6.1 EFFECTIVENESS AND IMPACT ON USERS' PRODUCTIVITY

Twenty seven people participated in our studies, but only 20 progressed beyond the control conditions, 12 in the VC_SRA experiment and 8 in the VC_IPA experiment. In the remainder of this section we consider only results obtained with these 20 participants. Table 7.3 shows summary statistics of participants' performance in both studies. As before we used Wilcoxon's signed-ranks test to compare the participants' performance in the control and VC_SRA's (learning and maintenance) and VC_IPA conditions. We used a one-sided test to compare UR acceptance, and two-sided tests to compare JR acceptance and time to complete tasks. Table 7.4 shows the results of this comparison. Noted effect sizes are Cohen's d [44].

Results show that, as expected, participants accepted as many justified risks in VC_SRA-Learning (p-value=1.0, n=12), VC_SRA-Maintenance (p-value=0. 5, n=12), and VC_IPA (p-value=1.0, n=8) as in their respective control conditions. In addition, and as hypothesized, there was a statistically significant (and large) reduction in unjustified risks accepted from respective control conditions in VC_SRA-Learning (p-value=0.00024, d=3.83), VC_SRA-Maintenance (p-value=0.00024, d=4.05), and VC_IPA (p-value=0.004, d=3.36) conditions. Moreover, there was a statistically significant and large reduction in time spent completing the assigned tasks when participants used the SRA, during the learning (p-value=0.00097,

**Table 7.3:** Summary Statistics of conditions in the VC_SRA and VC_IPA studies

| | **VC_SRA** | | | **VC_IPA** | |
|---|---|---|---|---|---|
| | **Control** | **Learning** | **Maintenance** | **Control** | **IPA** |
| # participants | 12 | | | 8 | |
| | *# of justified risks accepted* | | | | |
| Mean | 4.75 | 4.67 | 4.42 | 2 | 2 |
| Std. Dev | 0.45 | 0.49 | 0.90 | 0 | 0 |
| | *# of unjustified risks accepted* | | | | |
| Mean | 4.42 | 0.42 | 0.25 | 5.75 | 0.375 |
| Std. Dev | 0.79 | 0.79 | 0.45 | 1.39 | 0.52 |
| | *Time to complete tasks (minutes)* | | | | |
| Mean | 30.71 | 15.42 | 16.67 | 26.17 | 13.36 |
| Std. Dev | 9.62 | 4.58 | 5.79 | 13.56 | 4.72 |

d=1.711) and maintenance (p-value=0.00048, d=1.595) conditions, after watching the vicarious-reinforcement video, and when interacting with the IPA (p-value=0.016, d=1.24) after watching the vicarious punishment video, than in the respective control conditions.

### 7.6.2 PARTICIPANTS' IMPRESSIONS

We show average results of the exit survey in Table 7.5 and Table 7.6. The former contains, as in Chapters 5 and 6 respectively, results of questions designed to gauge participants' impressions

**Table 7.4:** Comparisons of VC_SRA-Learning, VC_SRA-Maintenance, and VC_IPA to respective control conditions p-values were calculated using Wilcoxon's signed-ranks test (*=significant)

| | **VC_SRA-Learning** | **VC_SRA-Maintenance** | **VC_IPA** |
|---|---|---|---|
| *Acceptance of Justified Risks (JRs)* | | | |
| p-value | 1.0 | 0.5 | 1.00 |
| effect size | -- | -- | -- |
| *Acceptance of Unjustified Risks (URs)* | | | |
| p-value | 0.00024* | 0.00024* | 0.004* |
| effect size | 3.83 | 4.05 | 3.36 |
| *Time to complete tasks* | | | |
| p-value | 0.00097* | 0.00048* | 0.016* |
| effect size | 1.711 | 1.595 | 1.24 |

about features of the SRA and the IPA, while the latter summarizes their impressions about the vicarious-conditioning interventions (videos).

Table 7.5, on the one hand, shows that participants found the dialogs in the SRA and the IPA easy to understand, considered that these applications provided good guidance, and would be fairly comfortable to receive such guidance in the future. On the other hand, participants would give a neutral recommendation to a friend, were neutral about the helpfulness of the questions in the guidance, and not always followed the guidance.

Participants' impressions in Table 7.6 are grouped with regard to the four sub-processes involved in observational learning. An additional group shows other impressions that participants had about the videos. On the whole, participants in both studies strongly agreed that they paid attention to the videos. Regarding retention, participants agreed and strongly agreed that they had enough time to read and understand the rules respectively in the vicarious-punishment intervention and vicarious-reinforcement intervention. Corroborating this opinion, results of the on-screen quiz taken after watching the video show that in the VC_SRA study all participants correctly answered all the questions, whereas all but one participant in the VC_IPA study did. The one exception gave a wrong answer only to question three. Thus, both interventions were effective in ensuring that participants retained the main ideas acted by the models and enumerated in the summaries.

With respect to reproduction, participants in the two experiments were able to apply their recently acquired skills when identifying risks in the quiz. This is also consistent with their performances handling justified and unjustified risks, which was presented in the previous section. Concerning motivation, participants opined that they were likely and somewhat likely, in the

**Table 7.5:** Average perceptions of VC_SRA and VC_IPA

| | **VC_SRA** | **VC_IPA** |
|---|---:|---:|
| Dialogs are easy to understand | 4.4 | 4.3 |
| Questions are helpful | 3.3 | 3.0 |
| Interface provides good guidance | 4.3 | 3.9 |
| Participant followed guidance | 3.3 | 3.4 |
| Would feel comfortable receiving such guidance in the future | 4.0 | 3.8 |
| Would recommend to friend | 3.4 | 3.1 |

VC_SRA and VC_IPA studies respectively, to imitate the secure behavior of the model in the videos when handling their corporate email account. Put another way, participants answers suggest that, to handle their employer-assigned email account more securely, being reinforced would be a more persuasive measure than avoiding being punished for failing to do so (though a Mann-Whitney test comparing these scores did not detect a significant difference). Other users' opinions indicate that they considered the length of both videos reasonable, and that the videos' models, environment, and situations looked realistic for an office setting.

The exit survey for the VC_SRA study also asked users to provide any comments about the video and the experiment itself. For the VC_IPA study, however, we replaced that question with two questions, one that asked what participants liked about the video and other that asked what they thought needed improvement. We did this based on the observation that most participants in the VC_SRA study gave either a negative or a positive comment. On the whole, most of the comments were fairly positive. Participants thought that the video depicted real-life situations, got the point across, was easy to understand, was aimed at the right level of user (neither novice not advanced), and that the guidance was helpful. We got a few negative comments mostly about the models "overacting", and in one case, a participant stated that the reinforcement video was a bit lengthy and repetitive (which was intentional). Other participants' remarks could be qualified as constructive criticisms, such as that the video should include more situations of suspicious emails aimed at the novice user, and that the video shown after the quiz (introducing the guidance) should be of longer length to allow observers to read all the guidance's options. Finally, some of the comments were neutral, e.g., "the video and experiment were fine", and "nothing". Appendix F contains the complete list of comments given by users in both experiments.

**Table 7.6:** Average perceptions of vicarious interventions

| | | VC_SRA | VC_IPA |
|---|---|---|---|
| *Attention* | | | |
| | I paid attention to the video | 4.60 | 4.50 |
| *Retention* | | | |
| | I had enough time to read and understand the rules presented (in the video) | 4.67 | 4.25 |
| *Reproduction* | | | |
| | [the quiz] helped me to practice my understanding of the rules presented in the video | 4.42 | 4.25 |
| | [the quiz] evaluated fairly my understanding of the most important points presented in the video | 4.60 | 4.13 |
| *Motivation* | | | |
| | When handling my corporate email account, I am very likely to imitate the secure behavior of the person in the video | 4.08 | 3.50 |
| *Others* | | | |
| | The video's length was reasonable | 3.92 | 4.00 |
| | The people in the video looked like real office workers | 4.08 | 4.00 |
| | The video depicted a realistic work environment and work-related situations | 3.67 | 3.75 |

### 7.6.3 COMPARISON TO SRA AND IPA

In this section, we compare the quantitative and qualitative results presented in the two previous sections to those obtained in the SRA and IPA experiments (described in Chapters 5 and 6 respectively).

To compare quantitative results, we first transformed counts of justified and unjustified risks accepted to hit and false alarm rates respectively, for all conditions in the experiments com-

**Table 7.7:** Comparisons of VC_SRA's and VC_IPA conditions to respective SRA's and IPA's conditions

p-values were calculated using Mann-Whitney tests (*=significant)

| | **VC_SRA vs SRA** | | **VC_IPA vs IPA** |
|---|---|---|---|
| | **Learning** | **Maintenance** | |
| *Acceptance of Justified Risks* | | | |
| p-value | 0.78 | 0.478 | 1.000 |
| effect size | -- | -- | -- |
| *Acceptance of Unjustified Risks* | | | |
| p-value | 0.03* | 0.074 | 0.016* |
| effect size | 0.89 | -- | 1.66 |
| *Time* | | | |
| p-value | 0.1 | 0.786 | 0.004* |
| effect size | -- | -- | 2.02 |

pared. Then, we subtracted the rates obtained in our security-conditioning experimental conditions (SRA-Learning, SRA-Maintenance, IPA, VC_SRA-Learning, VC_SRA-Maintenance, and VC_IPA) from the rates in their respective control conditions. As mentioned in Chapter 6, this is necessary to avoid possible biases because of a priori differences between groups (e.g., more skilled or risk averse participants in one group than the other). Afterward, we compared these differences in rates using Mann-Whitney tests to determine which interventions achieved the largest improvements, according to the criteria stated in hypothesis 8. We did a one-sided test for comparing acceptance of unjustified risks and a two-sided test for comparing differences in acceptance of justified risks. Times to complete tasks were compared directly without any adjustment using a two-sided test. We also computed effect sizes (Cohen's d) using pooled standard deviations. Table 7.7 shows comparisons of quantitative results using these tests.

In the case of SRAs, the results indicate that there was no statistically significant difference in acceptance of justified risks when participants interacted with the application when they first watched a vicarious security-reinforcement intervention and when they did not. However, there was a significant and large improvement in rejection of unjustified risks from SRA-Learning to VC_SRA-Learning (p-value=0.033, d=0.89), but a non-significant improvement from SRA-Maintenance to VC_SRA-Maintenance (p-value=0.074). The latter results indicate that presenting (i) the rules for discriminating between justified and unjustified risks, and (ii) the consequences of accepting each type of risk before a user interacts with a SRA, does speed up learning (as was conjectured in Chapter 2). However, they also indicate that, once a person has learned how to do this discrimination, her behavior can be maintained with an intermittent schedule as effectively as that of a person who was conditioned without observing a vicarious intervention. Finally, we found no significant difference between the reinforcement conditions regarding time to complete tasks.

In the case of IPAs, results show that there was no significant difference in acceptance of justified risks for participants who interacted with the application when they first watched a vicarious insecurity-punishment intervention and when they did not. However, there was a significant and large improvement in rejection of unjustified risks from IPA to VC_IPA (p-value=0.008, d=1.66) conditions. Thus, vicarious insecurity-punishment significantly accelerates the acquisition of skills for identifying unjustified risks without negatively affecting users' recognition of justified risks. Regarding time to complete assigned tasks, there was a reduction in time to complete tasks from IPA to VC_IPA that was significant and large (p-value=0.004, d=2.02). The latter is because in the VC_IPA condition, only three out of eight participants were suspended for accepting one unjustified risk each, whereas in the IPA condition participants accepted, on average, two unjustified risks.

These results verify Hypothesis 8.

To test Hypothesis 9, we compared qualitative scores given by participants to the exit survey they completed after the IPA and VC_IPA studies. The scores were compared directly without any adjustment, using a one-sided Mann-Whitney test. Table 7.8 summarizes the results of these comparisons. The noted effect sizes (Cohen's d) were calculated using pooled standard deviations. The results show that participants who observed the vicarious-punishment intervention were significantly more likely to recommend the guidance to a friend, and deemed the guidance provided by the application as significantly better than participants who did not observe such interventions. In both cases the improvements were large. Other scores show that participants in the VC_IPA study gave higher scores to the application, but these increments were not

**Table 7.8:** Average perceptions of VC_IPA ($n_1$=8) and IPA ($n_2$=7)

(worst = 1, best = 5) p-values were calculated using a one-sided Mann-Whitney test (*=significant)

| | **VC_IPA** | **IPA** | **p-value** | **eff. size** |
|---|---|---|---|---|
| Dialogs are easy to understand | 4.3 | 3.7 | 0.176 | -- |
| Questions are helpful | 3.0 | 2.1 | 0.125 | -- |
| Interface provides good guidance | 3.9 | 2.6 | 0.028* | 1.42 |
| Participant followed guidance | 3.4 | 2.4 | 0.056 | -- |
| Would feel comfortable receiving such guidance in the future | 3.8 | 3.0 | 0.190 | -- |
| Would recommend to friend | 3.1 | 1.9 | 0.036* | 1.24 |

statistically significant. These results show that hypothesis 9 was mostly verified except for the observed significant reduction in time to complete tasks in the VC_IPA experiments, which was unexpected. The latter is, nonetheless, a positive outcome.

## 7.7 DISCUSSION

Participants in the VC_SRA-Learning and VC_IPA conditions performed better than those in the SRA-Learning and IPA conditions, but those in the VC_SRA-Maintenance and SRA-Maintenance performed equally well. Based on these results, we conjecture that a similar phenomenon would have been observed if participants in the IPA and VC_IPA conditions would have been given a second set of emails to handle. In such a case, the performance of participants in these two conditions would probably become similarly good too.

Several studies have shown that merely announcing rewards for desirable behaviors and punishments for undesirable behaviors either without actually delivering them or delivering them non-contingently is not effective to respectively promote or deter such behaviors [108][74][41][98]. The effectiveness of our vicarious interventions and security-conditioning applications is due to the fact that they announce *and* deliver, in believable and feasible ways (e.g., monetary bonuses, email suspensions), respectively rewards and penalties contingent upon users' behaviors. Failure to provide rewards or enforce sanctions might lead to lack of motivation in users to apply what they observed in the interventions.

The reverse of the latter, however, is not necessarily true. Although the vicarious-conditioning interventions tested in the present study did speed up learning and helped users avoid unnecessary mistakes, organizations can still get benefits from deploying SRAs and/or IPAs without those interventions (e.g., because of lack of resources, or time constraints). In such a scenario, it is important that emails used for initially conditioning users be innocuous messages.

Candid feedback from participants appears to corroborate that they, indeed, felt identified with a coping model, and paid attention to his behavior. However, a few participants considered the model's acting a bit over the top. Despite this, some of them acknowledged that such attitude is probably necessary to get and maintain their attention, and concluded that the video got its point across. It would be interesting to experiment with models acting a bit less excessively, to see if the impact of the interventions is similar. Such kind of acting may be more appealing for senior or higher ranked organizations' member. On the other hand, some participants also considered that it would be good to show more cases of risks being handled by the model. Creating a longer intervention incorporating such suggestion can be difficult given that, after 10 minutes, people's attention may decline [91]. It could be interesting future work to explore ways to create longer interventions without significantly losing users' attention.

Peer or supervisor behavior can sabotage attempts to establish a computer security culture in an organization. Studies have shown that new hires who complete security training as part of their inductions can immediately began to behave insecurely as a consequence of observing their new work-team's behavior [41]. Also, managers who have the authority to fire computer security personnel sometimes ignore security policies they consider as "petty" [74], setting a bad example for subordinates. Therefore, when employing vicarious security-conditioning interventions, like the ones evaluated in this chapter, it is important to obtain commitment from upper-level management to ensure that the security behaviors conditioned vicariously will not weaken because of the aforementioned observational learning of *insecure* behaviors.

## 7.8 SUMMARY

In this chapter we designed and evaluated vicarious interventions intended to help conditioning security behaviors of end-users. Our results show that users can improve their security behavior after observing our interventions. Moreover, such improvement occurs faster than when using security-reinforcing and insecurity-punishing applications alone. Our results also suggest that, once learning has occurred, secure behaviors can be maintained equally well by using such applications without first having to observe vicarious interventions. However, the use of vicarious interventions is still useful as it can help users to avoid unnecessary errors while users are learning to distinguish justified and unjustified risks and the consequences of accepting or rejecting risks of each type. In the case of insecurity punishment, a vicarious intervention can also improve the users' acceptance of IPAs.

Results of a quiz and an exit survey suggest that our vicarious interventions were successful in capturing and maintaining users' attention, and that they were effective in facilitating users' retention of the most important ideas conveyed. Moreover, users felt motivated to behave more securely in order to either be reinforced or avoid being penalized in the case of the vicarious-reinforcement and vicarious-punishment interventions respectively. In both cases, average scores in an exit survey indicate that participants agreed that they would be comfortable when interacting with our applications in the future.

# 8.0 CONCLUSION

Many security decisions depend on contextual information that computer systems cannot obtain or verify automatically. Users need to provide such information. Unfortunately, users often do not provide true information to computer systems, due to unfriendly user interfaces, or lack of education in security matters. In many cases, however, users intentionally or automatically provide any (possibly incorrect) information that will quickly dismiss security dialogs and allow users to proceed with their primary goal (which rarely is computer security). Such user behavior can compromise computer systems' security. In this dissertation, we developed and evaluated software techniques that seek to reduce such insecure behavior.

## 8.1 SUMMARY OF CONTRIBUTIONS

The research in this dissertation has led to a number of novel contributions.

First, we contribute a model that helps explain why users behave insecurely when interacting with computer applications. According to our model, which is grounded on theories of human behavior, consequences and antecedents of security behaviors influence users' behaviors. On the one hand, when users accept risks, they usually get their job done and do not *perceive* immediate adverse consequences (e.g., punishment by their employer, or noticeable malware infections). On the other hand, when users reject risks, they may be unable to complete their primary tasks and may be punished for that. Given these possible consequences, users tend to accept risks without considering whether it is secure to do so. In this way, insecure behaviors are strengthened, while secure behaviors gradually extinguish. Moreover, as computer applications usually show some form of security dialog or warning before a user can accept a risk (i.e., *antecede* such behavior), users learn to bypass or dismiss them, even if that requires providing false inputs. Repetition of such insecure behavior causes that conventional warnings get associated with, and in effect, become cues of it.

Second, based on our model, we contributed a mechanism for disrupting the antecedents of insecure behaviors called *context-sensitive guiding polymorphic dialogs* (CSG-PD). Computer Scientists can make use of CSG-PD by applying the following three principles in conjunction. First, CSG-PD includes meaningful options pertinent to the context of the security decision and that guide users to take a secure course of action. Second, CSG-PD randomizes the order in which a warning's options are shown every time the warning is displayed. Third, CSG-PD delays making active the final option in a warning that confirms an operation until it has been displayed for some time. CSG-PD is effective in thwarting automatic answers since the location of options is not predictable by users as usual, and because when users finally locate a desired option, they may have to wait before they can select it. CSG-PD also diminishes the probability of users giving false answers by forcing them to pay attention to more truthful and relevant alternatives in the warning dialog. We conducted a user study in which we verified that CSG-PD is effective, and that it did not negatively impact users' productivity. Moreover, results of our study do not suggest that CSG-PD experience a loss in effectiveness after its repeated presentation to users.

Third, we contribute the notion of a *security-reinforcing application* (SRA), which software developers can implement to strengthen end-users' secure behaviors by manipulating their consequences. SRAs do this by making reinforcing stimuli (i.e., desirable consequences) contingent upon such behaviors. System administrators initially use SRAs to condition users to emit secure behaviors by reinforcing them continuously, i.e., every time they emit such behaviors. SRAs can later be employed to maintain the strength of such conditioned behaviors by reinforcing users intermittently, i.e., after they have emitted either a certain number of secure behaviors, or at least one secure behavior in a certain period of time. SRAs can use a variety of reinforcing stimuli such as prize and praise rewards. In a user study, we evaluated SRAs and found that they are effective in improving end-users' secure behaviors without being detrimental to their productivity. Moreover, the secure behaviors conditioned and maintained using SRAs did not extinguish after a period of several weeks.

Fourth, we contributed the concept of an *insecurity-punishing application* (IPA). Software engineers can implement IPAs to weaken end-users' insecure behaviors. IPAs achieve this effect by making aversive consequences contingent upon those behaviors. IPAs include *audited dialogs*, which are modified security dialogs that warn a user that the options she selects in such dialogs will be logged for later analysis by her organization's security auditors, and that these auditors may penalize her if they find that her selections are deemed unjustifiably risky according to the organization's security policy. Second, IPAs can punish users by applying the penalties specified in the organization's policy either automatically or after being remotely instructed by the auditors. Example punishments include suspending the ability of users to use email for monotonically nondecreasing periods of time, and fines. An empirical evaluation of IPAs demonstrated that they do, indeed, weaken users' insecure behaviors, while not suppressing production-related behaviors, or increasing the time to complete production tasks. However, we also found that users' acceptance of IPAs can be significantly lower than that of SRAs.

Fifth, we devised a method for vicariously conditioning users' security behaviors aimed to complement our security-reinforcing and insecurity-punishing applications. When end-users watch interventions created using our method before interacting with these applications, users' security behavior improves, and it only needs to be maintained using the applications. We implemented two such interventions, one that shows a model being reinforced after behaving securely when interacting with a SRA, and other that shows a model being penalized after behaving insecurely when interacting with an IPA. In both cases, several clues are given to help end-users spot features in email communications that may indicate whether they represent justified or unjustified security risks. We experimentally validated the effectiveness of our interventions with user studies. In them, the interventions maintained users' attention, facilitated retention of the important elements shown in them, and motivated users to apply the skills they acquired vicariously. Information Technology departments seeking to deploy security-conditioning applications can use this kind of interventions to ensure that members of an organization can use such applications in the most effective way.

## 8.2 VERIFICATION OF THESIS STATEMENT

In this section we explain how our contributions verify the thesis statement of the present dissertation. Recall that our aim is to provide satisfactory evidence in this dissertation to justify the following statement:

**Computer applications that (i) disrupt antecedents of insecure behavior, (ii) reliably convey reinforcers or punishments contingent upon security decision behavior, and (iii)**

**whose use may be anteceded by interventions that vicariously condition such behavior, can improve computer systems' security.**

Our second to fourth contributions above prove that, when users interact with computer applications that add polymorphism to security warnings (which are antecedents of security behaviors), and that make reinforcing and punishing consequences contingent upon secure and insecure behaviors respectively, computer users' security behaviors improve. This, in turn, improves the security of the computer systems that users employ. Moreover, our fifth contribution demonstrates that by making vicarious security-conditioning interventions antecede the use of such computer applications, the improvement in users' security behaviors is achieved faster.

# 9.0 FUTURE WORK

This thesis focused on leveraging earlier results from behavioral sciences to devise principles and techniques that enable computer scientists to design and implement computer applications that are both usable and secure. In this chapter we outline future work that can be done using as a starting point the results obtained in the present research. Some of the future work described is laboratory-based, some involve field trials, some are concerned with the deployment of computer applications that are created based on our contributed principles and techniques, and other explores the application of our contributions to areas outside computer security.

## 9.1 LABORATORY STUDIES

This dissertation leaves several questions open that could be addressed by future research, as described in the following subsections.

### 9.1.1 EFFECT ISOLATION

How much of the benefit of CSG-PD is due to CSG or to PD? Our pilot studies suggested that CSG alone is ineffective [98]. However, this was not verified in a full user study. Because of the difficulty of recruiting participants with the desired profile and time limitations, the user study that evaluated CSG-PD did not isolate the effects of CSG and polymorphic dialogs. We measured only their aggregate effect. Moreover, how does the effectiveness of PD depend on the type of polymorphism used? We explored only simple dialog polymorphism in our experiments. Overcoming these limitations would be interesting future work.

Also, how does SRA effectiveness depend on type, dosage, and scheduling of rewards? For the same reasons as above, our user study evaluating SRAs did not isolate the effects of each of the component schedules belonging to the combined schedules used in the learning, maintenance, and extinction stages. We also did not isolate the effect of using either praise of prize rewards alone. We measured only their aggregate effect. Further studies are needed to evaluate the effectiveness of each schedule and each type of reward.

### 9.1.2 DIFFERENT STIMULI

We implemented an SRA that included specific prize and praise rewards. However, different organizations could use different stimuli depending on their particularities. Organizations already use rewards such as tickets for lunches, stock options, movie passes, and coupons redeemable in stores (e.g., online) to reward production-related behaviors [33]. Some of those could be easily delivered by SRAs (e.g., by displaying these rewards onscreen in a printable form). According to a recent industry survey [27], security breaches can cost on average $230,000 per year to an organization. Therefore, it is not unreasonable to expect that companies would be willing to spend money on these kinds of rewards for employees, instead of waiting to suffer losses because of financial damages incurred as a result of breaches.

We tested an IPA only with two specific punishing stimuli, suspensions and fines, that were inspired mostly by limitations of our laboratory environment. A possibly less disruptive penalty would be to suspend the user's ability to open attachments while preserving the user's ability to receive message bodies and send messages without attachments. It would be interesting to determine in further trials what types of sanctions work best and respective dosage effects.

### 9.1.3 UNIFIED SECURITY-CONDITIONING APPLICATION

In order to better understand users' behavior when interacting with SCAs, we tested reinforcing and punishment independently. However, using a combination of these two approaches could be interesting future work. One possibility is that participants could be conditioned using reinforcement, but those who willfully refuse to improve their behavior could be penalized [74]. One penalty could be to reduce monetary rewards previously given. If the user's reluctance continues, then a more drastic sanction such as email suspension could be applied.

## 9.2 TOWARDS DEPLOYING SECURITY-CONDITIONING APPLICATIONS

Deploying SCAs to an organization would involve several activities and would surely face some challenges. We briefly discuss those in this section using email clients as example.

First, an SCA would need to be implemented and deployed into the organization's member's computers. Several sources [25][110][20] indicate that email clients from Microsoft are the most popular (Microsoft Outlook in their different versions). However, in this dissertation, we implemented the entire SCA functionality as an "add-on" or "extension" for Mozilla Thunderbird. Thus, unless the testing organization currently uses such email client or is willing to migrate to it, it may be necessary to develop a so called *add-in* for Microsoft Outlook [58] that replicates the current functionality of our Mozilla extension (this appears to be feasible). The advantage of using add-ins or extensions is that the original program's source does not have to be modified, and no new program needs to be deployed. In addition, both Thunderbird [82][83] and Outlook [58, p. 279] have mechanisms that prevent users from uninstalling or disabling these kinds of extensions, unless they have administrative privileges.

Second, it must be ensured that a tight coordination with the organization's tech support department (or equivalent) takes place. First of all, the test emails actually need to reach employees' Inboxes. Thus, the company's email filters will necessarily need to be modified for this purpose (this is feasible as the test emails possess identifiable headers). Also, employees may ask tech support about the interventions, and a consistent answer needs to be provided. Of course, system administrators, being aware of the interventions, need not reveal details that might jeopardize evaluation of employees' acquired security skills.

Third, test emails tailored for the organization's members need to be created. As this may be too time consuming, automated or semi-automated ways to achieve this could be employed. One such method would be to collect samples of suspicious email messages directed to organization's members' email addresses, but that were caught by automated tools, such as anti-malware filters, and anti-spam and anti-phishing tools. It is quite common that the latter tools use scores to qualify how likely it is that email messages are really malicious [34]. Thus, to simplify work, only the emails that were the highest scored by such tools would be used, in addition to emails unequivocally identified as dangerous by antivirus and other antimalware tools. Once collected, these samples then would need to be modified, for example, as follows. First, they need to be sanitized. One alternative is to strip them of their malicious elements such as attachments, links, and embedded images. Some of such elements then would need to be modified. For instance, links can be rewritten to point to safe URLs inside the company's network, and images can be automatically downloaded and hosted in company's internal web sites. Second, some elements would be added to such emails, such as the headers that the SCA uses to recognize emails, as well as innocuous attachments. All these operations can be done automatically. Optionally, security auditors would review a pool of emails collected and altered in the aforementioned way, before authorizing its automatic delivery, according to a specific schedule, to employees' email inboxes.

Fourth, privacy concerns need to be considered. By using test emails we are avoiding the problem of having to scoop into users' real communications. However, in order to deploy a system that will actually reward users, information about users' performance need to be stored. It is imperative that this be done in a secure way. In addition, depending on the organizations' policies, users might be able to "opt out" of a security-reinforcement program. However, it is also very likely that an organization's members need to be accountable for their behavior if the latter causes a security breach. Thus, punishment interventions probably would not be something that employees would be allowed to opt out from.

Finally, once an organization has decided to deploy SCAs, it would be advisable to do a pilot deployment first. During it, one or more test emails representing unjustified risks would be sent to several employees. Then, a subset of employees who accept the risk would be selected to progress to a second phase. In this second stage, they will interact with SCAs to see if they improve their behavior and to find out about their concerns. Such concerns can be addressed before doing a broader deployment. If the organization desires to use vicarious-conditioning interventions prior to such deployment, they may decide to tailor the visual materials to show email addresses of the organization, instead of email addresses of a fictitious company as in the evaluation described in this dissertation, and to use risks that are the most common in the company. Optionally, video materials can be recorded at the organization's headquarters.

## 9.3 BEYOND COMPUTER SECURITY

Our techniques are intended for social contexts where some individuals (e.g., managers, coaches, teachers, parents) are tasked with supervising and positively affecting the behavior of others. Supervisors are required to know supervisees' context well, set policies, and help the system select and label instances of justified and unjustified risks. In the present research, these policies and risks were related to computer security. However, our techniques are general enough to be applied to other software involving policy-driven decisions based on information that needs to be obtained from users.

For instance, consider the process of mortgage applications. There are inherent risks associated with decisions in such a domain, and many factors need to be considered before taking the risk of approving a particular individual's application. A financial supervisor may use our conditioning techniques for helping junior analysts make better decisions in this context. These analysts may use software containing the approval policy embedded into it in the form of computer dialogs. These may guide analysts into making an approval or rejection decision. To force analysts to pay attention to the decision process, the aforementioned dialogs can be made polymorphic. In addition, supervisors may add into the software's queue mortgage applications of fictitious people some of whom carry a high risk for defaulting on the loan, as well as applications of others carrying little such risk. The software may recognize such applications if, for instance, they are tagged as unjustified and justified respectively. An analyst may suffer some form of penalty for approving mortgage applications tagged as unjustifiably risky, but may be re-warded for approving applications that meet the lending institution’s policy.

# APPENDIX A

# USER STUDY HANDOUTS

**Scenario #1**

You will be role-playing Chris, an office worker at a company called ACME

Chris works in a group dedicated to evaluation of credit card applicants. The other members of his group are:

- Alex: always meticulous and precise in her writing
- Bob: Always serious
- Frank: happy and carefree

Chris has two email accounts, **chris@acmecorp.biz**, which he uses for work-related messages, and **chris679@gmail.com**, which he uses for private messages.

You are to check Chris' inbox and do the following tasks:

**Task 1**

Chris wants to hire another worker. He advertised the position in work websites and is expecting resumes from applicants (whom he does not know). Chris needs to pick the applicant with most years of experience and write down her/his name.

**Task 2**

Finish processing (delete right away, read, answer, etc. messages) Chris' inbox's messages.

**Additional information**

If Chris needs help with his computer, he can send a message to techsupport@acmecorp.biz or contact Tech Support by phone. Chris always uses GMail for his account at priceline.com and PNCBank and for any other private communication. Chris recently ordered a getaway weekend from priceline.com. He travels next week.

**Scenario #2**

You are going to role-play Amanda Lovelase, an accountant working for SecuredFuture (SF), an insurance company that accepts claims in electronic format.

Amanda's only known people at SF are:

- Henry Buffett, an insurance specialist, who communicates verbosely.
- Theresa Goodrich, a nice old lady (although pretty busy), who works at payroll.

You are to check Amanda's inbox and do the following tasks:

**Task 1**

SF offers forms on its website that a claimant must download and then send as attachments to your email address (amanda@securedfuture.biz). You have to review the forms and check if all the required fields contain the proper information and if so, you acknowledge receipt to the sender and forward the forms to Henry (henry@securedfuture.biz). Otherwise you ask the sender to retransmit with corrections.

**Task 2**

Finish processing (delete right away, read, answer, etc.) messages in Amanda's inbox.

**Background information**

Before joining SF, Amanda volunteered for free a charitable organization, since she always has been a goodhearted person. She managed her own website (lovelase.org) where she advertised volunteering opportunities.

She is paying less attention now to her website, but she still uses her email address (amanda@lovelase.org) for all kind of personal matters, like to manage her accounts at uBid.com and barnesandnoble.com

# APPENDIX B

# EMAILS USED IN EXPERIMENTS

In this appendix we present the emails used for evaluating the techniques devised in this dissertation. Section B.1 describes the types of unjustified risks used to guide the design of the emails for our experiments. Sections B.2 and B.3 provide details about the emails themselves.

## B.1 TYPES OF UNJUSTIFIED RISKS

In this section we describe the types of unjustified risks (URs) based on which we created the emails used in the experiments discussed in the present dissertation. These UR types are based on the same sources [64][112] we used to create our sample template policy (Figure 4.11). Section 4.2 (p. 42) provides additional information about these risks.

- ***Email refers to <u>u</u>nknown <u>a</u>ccount or <u>e</u>vent*** (UAE). The email message refers either to an account (e.g., with an online merchant) that the recipient has not opened, or to an event (e.g., message, purchase, meeting) that the recipient is not involved with.

- ***Email was sent to unexpected or wrong email address*** (UWA). The recipient does not use this account to communicate with the sender. For instance, an email message supposedly sent by the recipient’s bank to her corporate email account, despite her having signed up with the bank using her personal email account.

- ***Message is out of character (OC) for sender*** (OCM). The wording of an email message, sent from a spoofed email address, is out of character or atypical for the real sender. For instance, a message full of spelling mistakes from a person who is known to be meticulous in her writing, or a very short email sent by someone who usually communicates verbosely.

- ***Email contains an attachment that is unexpected or OC for the sender*** (OCA). The user usually does not receive a type of attachments like that from the sender. For example, a game supposedly sent by an old and very busy person, or a joke from a person who is usually serious.

- ***Email purportedly sent by customer service or technical support contains an attachment related to unknown account or event*** (CTS). Attackers often impersonate technical support or customer service, even though the latter usually avoid sending risky attachments.

- ***Email contains an attachment whose purpose is either mentioned vaguely, unconvincingly or not at all in the message's body*** (VNP). Attackers often send messages that contain attachments whose purpose is not clear.

## B.2 EMAILS USED IN EXPERIMENTS ABOUT SECURITY REINFORCEMENT

We used the same emails in the evaluation of both security-reinforcing applications (SRA, chapter 5) and vicarious security-conditioning (VC_SRA, chapter 7). These emails are grouped in four sets, namely, Learning-I (L1), Learning-II (L2), Maintenance (M), and Extinction (E). In this section we provide details about these sets of emails, which we created for each of our two scenarios (Appendix A).

### B.2.1 EMAILS IN THE FIRST SCENARIO

Tables B.1 to B.4 present the messages in the four email sets created for the first scenario.

### B.2.2 EMAILS IN THE SECOND SCENARIO

Tables B.5 to B.8 present the messages in the four email sets created for the second scenario.

**Table B.1:** Emails in set Learning-I (L1) used in the first scenario (S1)

| ID | Sender | Subject | Risk |
|---|---|---|---|
| S1-L1-01 | Alex Twain <alex@acmecorp.biz> | Hiring a new team's member | JR |
| S1-L1-02 | Frank O'Brien <frank@acmecorp.biz> | New credit card rates | JR |
| S1-L1-03 | Robert Gravis <ro-bert@acmecorp.biz> | Updated Code of Conduct. ACME Corp | JR |
| S1-L1-04 | Citizens in Action <act@citizensinaction.org> | Pennsylvania Flood Victims Relief | UR: VNP |
| S1-L1-05 | Avis <Avis@rent.avis.com> | Rent a car for your next trip (and 15% off coupon) | UR: UAE |
| S1-L1-06 | American Airlines <ticket-ing@aa.com> | We're sorry for the delay | UR: UAE |
| S1-L1-07 | Maria Zimmel <ma-ria_zimmel@hotmail.com> | Response to your Job post | JR |
| S1-L1-08 | Dan Faughnan <dan@financialdpt.com> | Presentation | UR: UAE |
| S1-L1-09 | Carson Wilson <cwil-son@gmail.com> | Resume for Job application | JR |
| S1-L1-10 | PNC Bank <custo-mersvc@pncbank.com> | PNC Rewards Program | UR: UWA |

**Table B.2:** Emails in set Learning-II (L2) used in the first scenario (S1)

| ID | Sender | Subject | Risk |
|---|---|---|---|
| S1-L2-01 | Chelsea Anderson <chel-seaandr@yahoo.com> | My resume for financial po-sition | JR |
| S1-L2-02 | Alex Twain <alex@acmecorp.biz> | Addendum to credit approval policies | JR |
| S1-L2-03 | Priceline.com <bill-ing@priceline.com> | Confirmation | UR: UWA |
| S1-L2-04 | techsupport@acmecorp.biz | Company Directory | UR: CTS |
| S1-L2-05 | Joseph Miller <jmbiz@aol.com> | Job position | JR |
| S1-L2-06 | Robert Gravis <ro-bert@acmecorp.biz> | Employee satisfaction survey | JR |
| S1-L2-07 | Sam Vincenzo <samvincen-zo@unlimitedbay.com> | Meeting next week | UR: UAE |
| S1-L2-08 | alex@acmecorp.biz | Extremly iomportant | UR: OCM |
| S1-L2-09 | Linda Jones <linda79@mail.com> | About your job position | JR |
| S1-L2-10 | robert@acmecorp.biz | check this joke out | UR: OCA |

**Table B.3:** Emails in set Maintenance (M) used in the first scenario (S1)

| ID | Sender | Subject | Risk |
|---|---|---|---|
| S1-M-01 | Frank O'Brien <frank@acmecorp.biz> | For preparation of your monthly report | JR |
| S1-M-02 | Alex Twain <alex@acmecorp.biz> | Non-disclosure Agreement | JR |
| S1-M-03 | Verizon <veri-zon.ecenter@verizon.com> | Business cell phone | UR: UAE |
| S1-M-04 | Gmail Team <mail-noreply@google.com> | Gmail - Mandatory Accep-tance of New Privacy Poli-cy | UR: UWA |
| S1-M-05 | USPS <U.S._Postal_Service@usps.com> | USPS - Authorize indirect delivery or reattempt | UR: UAE |
| S1-M-06 | Robert Gravis <robert@acmecorp.biz> | Mandatory survey | JR |
| S1-M-07 | Kenneth Reed <kenreed25@msn.com> | Applying to your job offer | JR |
| S1-M-08 | Customer Service <custsup-port@yahoo.com> | Re: Office Supplies Form | UR: CTS |
| S1-M-09 | Sharon Peterson <sha-ronp1@lycos.com> | My Resume | JR |
| S1-M-10 | harris@accounting.financialservices.com | Expenses' check | UR: UAE |

**Table B.4:** Emails in set Extinction (E) used in the first scenario (S1)

| ID | Sender | Subject | Risk |
|---|---|---|---|
| S1-E-01 | Frank O'Brien <frank@acmecorp.biz> | Identity Theft Protection | JR |
| S1-E-02 | Alex Twain <alex@acmecorp.biz> | Short Leave | JR |
| S1-E-03 | Comcast Customer Service <abuse@comcast.com> | Transfer limit exceeded | UR: CTS |
| S1-E-04 | general.announcements@lists.biz | Winners of the monthly raffle | UR: UAE |
| S1-E-05 | Wayne Simmons <sylvanlearning@hotmail.com> | RE: registration | UR: UAE |
| S1-E-06 | Frank O'Brien <frank@acmecorp.biz> | About the Workflow system | JR |
| S1-E-07 | Alex Twain <alex@acmecorp.biz> | Centurion Card from American Express | JR |
| S1-E-08 | AbeBooks <news@abebooks.com> | Remember Us? Here's a Coupon to Jog Your Memory | UR: UAE |
| S1-E-09 | Pizza Hut <promo@pizzahut.com> | Enjoy even more pizza! | UR: UWA |
| S1-E-10 | Robert Gravis <robert@acmecorp.biz> | Meeting with regulators | JR |

**Table B.5:** Emails in set Learning-I (L1) used in the second scenario (S2)

| ID | Sender | Subject | Risk |
|---|---|---|---|
| S2-L1-01 | Henry Buffet <henry@securedfuture.biz> | Insurance Claims by Email | JR |
| S2-L1-02 | Theresa Goodrich <there-sa@securedfuture.biz> | Final version of your Con-tract | JR |
| S2-L1-03 | Henry Buffet <henry@securedfuture.biz> | Code of Ethics | JR |
| S2-L1-04 | Experian <reports@experian.com> | Your credit report is at-tached | UR: UAE |
| S2-L1-05 | Apple Inc <do_not_reply@apple.com> | Thanks for your iPhone purchase. Get $100 back. | UR: UAE |
| S2-L1-06 | John Carter <johnc@trainingdepartment.services.com> | Mandatory Employee training | UR: VNP |
| S2-L1-07 | Patricia Joyce <patriciar-joyce@yahoo.com> | Insurance form | JR |
| S2-L1-08 | Stephen Cobb <scobb@plibrary.org> | RE: Library access up-grade | UR: UAE |
| S2-L1-09 | Mike Smith <mike640@hotmail.com> | Claim for insurance in-voice | JR |
| S2-L1-10 | BarnesAndNoble.com <custom-ers@barnesandnoble.com> | Free Music & Videos from our new download service | UR: UWA |

**Table B.6:** Emails in set Learning-II (L2) used in the second scenario (S2)

| ID | Sender | Subject | Risk |
|---|---|---|---|
| S2-L2-01 | Jennifer Taylor <jennit@gmail.com> | Claim form | JR |
| S2-L2-02 | Henry Buffet <henry@securedfuture.biz> | Changes in insurance liability limits | JR |
| S2-L2-03 | uBid.com <info@ubid.com> | Notification profile | UR: UWA |
| S2-L2-04 | techsupport@securedfuture.biz | Upgrade of employee equipment | UR: CTS |
| S2-L2-05 | Donald Thompson <don81@aol.com> | Insurance payment claim | JR |
| S2-L2-06 | Theresa Goodrich <theresa@securedfuture.biz> | Employee satisfaction survey | JR |
| S2-L2-07 | Steve Wilkins <stevewilkins@customerservice.com> | RE: Customer info request … | UR: UAE |
| S2-L2-08 | theresa@securedfuture.biz | Give this game a try | UR: OCA |
| S2-L2-09 | Adriana Robinson <arobinson6@mail.com> | Claim for insurance payment | JR |
| S2-L2-10 | henry@securedfuture.biz | Company's Anniversary - Schedule | UR: OCM |

**Table B.7:** Emails in set Maintenance (M) used in the second scenario (S2)

| ID | Sender | Subject | Risk |
|---|---|---|---|
| S2-M-01 | Theresa Goodrich <theresa@securedfuture.biz> | Health Care enrollment form | JR |
| S2-M-02 | Henry Buffet <henry@securedfuture.biz> | Confidentiality Agreement | JR |
| S2-M-03 | The Travelocity Team <memberservices@travelocity.com> | Up to $600.00 for your next trip | UR: UAE |
| S2-M-04 | Paypal <service@paypal.com> | Status of your dispute with seller techgadgets | UR: UAE |
| S2-M-05 | Homeless Children's Fund <homelesschildrenfund@yahoo.com> | Coordinators Meeting this month | UR: UWA |
| S2-M-06 | Theresa Goodrich <theresa@securedfuture.biz> | Mandatory survey | JR |
| S2-M-07 | Raymond Howard <raymondh@live.com> | Insurance claim form | JR |
| S2-M-08 | Customer Service <customersupport@yahoo.com> | Computing Services - Accounts | UR: CTS |
| S2-M-09 | Shirley Martin <smartin@comcast.com> | Claim attached | JR |
| S2-M-10 | walker@accounting.financialservices.com | Outstanding Invoices | UR: UAE |

**Table B.8:** Emails in set Extinction (E) used in the second scenario (S2)

| ID | Sender | Subject | Risk |
|---|---|---|---|
| S2-E-01 | Theresa Goodrich <there-sa@securedfuture.biz> | 401K enrollment | JR |
| S2-E-02 | Harold Lewis <haroldlewis@aol.com> | Attaching my claim for processing | JR |
| S2-E-03 | Verizon Customer Service <abuse@verizon.net> | Transfer limit exceeded | UR: CTS |
| S2-E-04 | IRS <irs.gov@yahoo.com> | Economic stimulus Payment Notice | UR: UWA |
| S2-E-05 | Tiffany Henderson <tiffa-ny.henderson@gmail.com> | Bus and T Light Rail Passes | UR: UAE |
| S2-E-06 | Carol Griffin <cgrif-fin23@hotmail.com> | Filled out claim form | JR |
| S2-E-07 | Theresa Goodrich <there-sa@securedfuture.biz> | Click@Home pilot program | JR |
| S2-E-08 | Amazon.com <store-news@amazon.com> | Save on That Perfect Gift | UR: UAE |
| S2-E-09 | Tamara Jenkins <tjenkings@msn.com> | RE: Vacation Balance | UR: UAE |
| S2-E-10 | Henry Buffet <hen-ry@securedfuture.biz> | For preparation of your monthly report | JR |

## B.3 EMAILS USED TO EVALUATE WARNING POLYMORPHISM AND INSECURITY PUNISHMENT

We used the same emails to evaluate (i) context-sensitive guiding polymorphic dialogs (CSG-PD, chapter 4), (ii) insecurity-punishing applications (IPA, chapter 6), and (iii) vicarious insecurity-punishment (VC_IPA, chapter 7). In this section we provide details about such emails which we created for each of our two scenarios (Appendix A). All the emails described in the present section belong to sets Learning-I and Learning-II presented in the previous section, and can be recognized by their ID (see below).

### B.3.1 EMAILS IN THE FIRST SCENARIO

Table B.9 presents the email messages created for the first scenario.

### B.3.2 EMAILS IN THE SECOND SCENARIO

Table B.10 presents the email messages created for the second scenario.

**Table B.9:** Emails used in the first scenario (S1)

| ID | Sender | Subject | Risk |
|---|---|---|---|
| S1-L1-09 | Carson Wilson <cwil-son@gmail.com> | Resume for Job application | JR |
| S1-L2-20 | robert@acmecorp.biz | check this joke out | UR: OCA |
| S1-L1-10 | PNC Bank <custo-mersvc@pncbank.com> | PNC Rewards Program | UR: UWA |
| S1-L2-14 | techsupport@acmecorp.biz | Company Directory | UR: CTS |
| S1-L2-18 | alex@acmecorp.biz | Extremly iomportant | UR: OCM |
| S1-L1-07 | Maria Zimmel <ma-ria_zimmel@hotmail.com> | Response to your Job post | JR |
| S1-L2-13 | Priceline.com <bill-ing@priceline.com> | Confirmation | UR: UWA |
| S1-L2-17 | Sam Vincenzo <samvincen-zo@unlimitedbay.com> | Meeting next week | UR: UAE |
| S1-L1-06 | American Airlines <ticket-ing@aa.com> | We're sorry for the delay | UR: UAE |
| S1-L1-04 | Citizens in Action <act@citizensinaction.org> | Pennsylvania Flood Victims Relief | UR: VNP |

**Table B.10:** Emails used in the second scenario (S2)

| ID | Sender | Subject | Risk |
|---|---|---|---|
| S2-L1-07 | Patricia Joyce <patriciar-joyce@yahoo.com> | Insurance form | JR |
| S2-L1-04 | Experian <reports@experian.com> | Your credit report is attached | UR: UAE |
| S2-L1-10 | BarnesAndNoble.com <customers@barnesandnoble.com> | Free Music & Videos from our new download service | UR: UWA |
| S2-L2-17 | Steve Wilkins <stevewilkins@customerservice.com> | RE: Customer info request … | UR: UAE |
| S2-L1-06 | John Carter <johnc@trainingdepartment.services.com> | Mandatory Employee training | UR: VNP |
| S2-L1-09 | Mike Smith <mike640@hotmail.com> | Claim for insurance invoice | JR |
| S2-L2-20 | henry@securedfuture.biz | Company's Anniversary - Schedule | UR: OCM |
| S2-L2-14 | techsupport@securedfuture.biz | Upgrade of employee equipment | UR: CTS |
| S2-L2-13 | uBid.com <info@ubid.com> | Notification profile | UR: UWA |
| S2-L2-18 | theresa@securedfuture.biz | Give this game a try | UR: OCA |

# APPENDIX C

# TEMPLATES FOR AUDITORS' EMAILS

## C.1 INSECURITY-PUNISHING APPLICATION EXPERIMENT

<<recipient>>,

On <<date>> at <<time>>, you received from "<<sender>>" an email with subject "<<subject>>" and attached file(s):

<<attachment name>>

Attachments like that can contain viruses. You should open them only if you have a good work-related reason.

Your answers for opening the attachment were:

- <<list of the CSG options selected to open attachment(s)>>

We find your answers unjustified in this case. Therefore, we impose the following penalties:

- <<list of penalties imposed>>

Yours truly,

<<ORGANIZATION>>'s security auditors

## C.2 VICARIOUS INSECURITY-PUNISHMENT EXPERIMENT

Dear <<recipient>>,

We're sending you this message to emphasize [*one more time*] how important it is that you handle your email securely.

In general, you should open an email attachment only if:

- You are expecting it to complete a work related task, OR
- It is from someone you know, the message does not appear out of character for the sender, and the message body explains clearly why you need the attachment for your work.

You recently opened the following attachment, included in a message titled "<<subject>>" from “<<sender>>”:

<<attachment name>>

We consider that you should not open an attachment like this because <<specific reason>>.

Due to your insecure behavior, we impose the following penalties:

- <<list of penalties imposed>>

You cannot avoid opening all email attachments because you may need them for your job. However, you can avoid penalties by answering the email program's questions carefully.

We audit email use continuously. Penalties for unsafe use may include longer suspensions and [*higher*] fines.

Best regards,

<<ORGANIZATION>>'s security auditors

# APPENDIX D

# CLUES SHOWN AFTER SCENES OF THE VICARIOUS SECURITY-CONDITIONING INTERVENTIONS

## D.1 SCENE 2 IN VICARIOUS SECURITY-REINFORCEMENT INTERVENTION

**To avoid falling for email-borne threats, consider the following:**

- Do not open attachments in email messages that you are **not** expecting to receive in a given email account (e.g., emails related to your personal life sent to your corporate email account)
- When appropriate, ask the sender for retransmission of the attachment in a safer format (e.g., .txt) using the email program's options
- If possible, verify by other means (e.g., phone) that the sender really sent you the message
- **Open** only those attachments that are necessary to do your job, **and** that you are expecting

## D.2 SCENE 2 IN VICARIOUS INSECURITY-PUNISHMENT INTERVENTION

**To prevent falling for email-borne threats, never open attachments that:**

- are not necessary to complete your job tasks
- you are not expecting
- are of a type that may spread computer viruses

Emails with dangerous attachments may come from known and
unknown senders!!

## D.3 SCENE 3 IN BOTH VICARIOUS SECURITY-CONDITIONING INTERVENTIONS

**To avoid falling for email-borne threats, consider the following:**

- Do **<u>not</u>** open attachments in email messages that refer to events you do not remember!
- When appropriate, ask the sender for retransmission of the attachment in a safer format (e.g., .txt) using the email program's options
- If possible, verify by other means (e.g., phone) that the sender really sent you the message
- **<u>Open</u>** only those attachments that are necessary to do your job, **<u>and</u>** that you are expecting

## D.4 SCENE 4 IN BOTH VICARIOUS SECURITY-CONDITIONING INTERVENTIONS

**Only accept emails that are <u>justified</u> risks:**

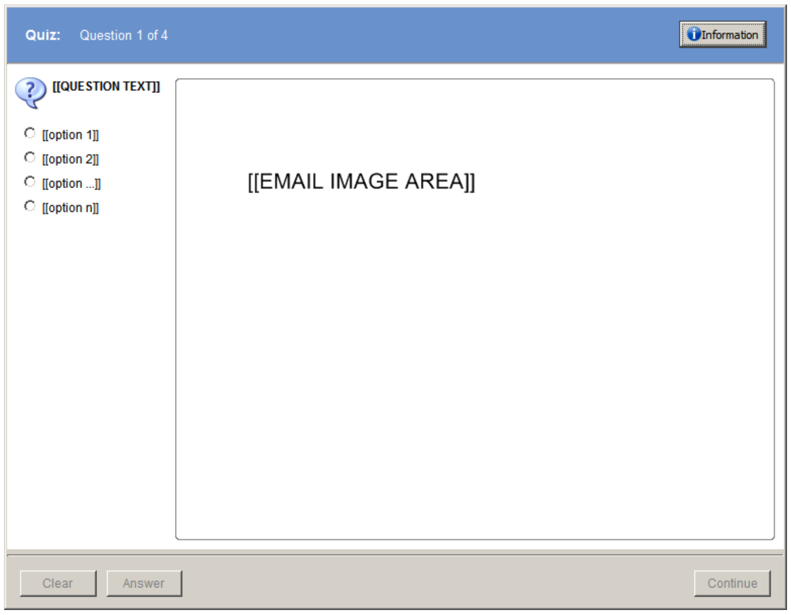
- they are necessary to do your job
- contain attachments that you are expecting, and in a file format that you are expecting
- do **<u>not</u>** seem out of character for a particular sender

Legitimate emails may come from known and unknown senders

# APPENDIX E

## QUESTIONS IN THE QUIZ OF THE VICARIOUS SECURITY-CONDITIONING EXPERIMENTS

The quiz consisted of four questions, detailed below. The text of each question was displayed in the area [[QUESTION TEXT]] of a dialog similar to the one depicted in Figure E.1. Each of the alternatives was displayed using radio buttons. In the list of alternatives below, the correct choices appear underlined and in italics. When the participant selected the right alternative, the dialog in Figure E.2 was displayed. The area labeled [[REASON]] showed the reason why the option selected was correct (the reasons are listed below). If the participant selected an incorrect alternative, the dialog of Figure E.3 was displayed, and showed the reason why it was wrong to select such option. Also, in the area labeled [[correct option]] the text of the right alternative was displayed. Finally, the image at the end of each question below was displayed in the area labeled [[EMAIL IMAGE AREA]] in the dialog depicted in Figure E.1. The button labeled “Information” in Figure E.1 displays a dialog box similar to Figure 7.5 (p. 124)


Quiz: Question 1 of 4

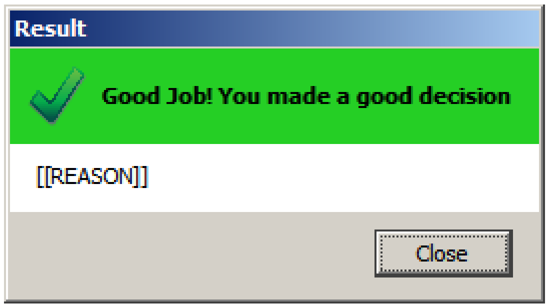
Information

[[QUESTION TEXT]]

- [[option 1]]
- [[option 2]]
- [[option ...]]
- [[option n]]

[[EMAIL IMAGE AREA]]

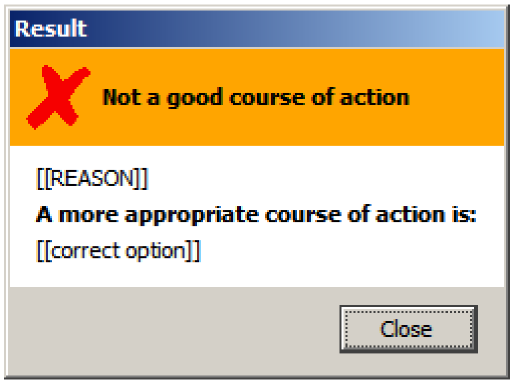
**Figure E.1: Dialog that showed the questions in the quiz**


Result

Good Job! You made a good decision

[[REASON]]

**Figure E.2: Dialog shown when participant answered a question correctly**


Result

Not a good course of action

[[REASON]]

**A more appropriate course of action is:**

[[correct option]]

**Figure E.3: Dialog shown when participant answered a question incorrectly**

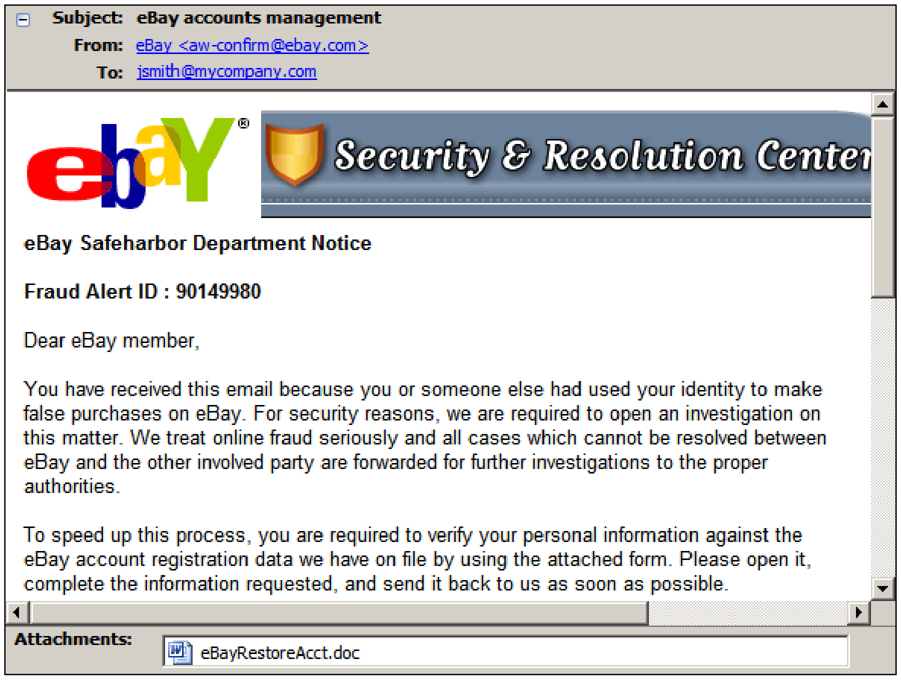
1. **You signed up with ebay.com with your personal email address. One day you receive the following email. What's wrong with it?**

   a) *<u>It was sent to my corporate email address which I didn't use to sign up with ebay.com</u>*

      **Reason**: If you use, e.g., a personal account to communicate with a site and a message supposedly from that site arrives in your work account, that message is probably a SPOOF.

   b) There is nothing wrong with it: it resembles emails sent by ebay.com and thus it must be legitimate

      **Reason**: Attackers often impersonate legitimate companies to send you potentially dangerous emails. If you use, e.g., a personal account to communicate with a site and then a message supposedly from that site arrives to your work account, that message is probably a SPOOF.



**Subject:** eBay accounts management
**From:** eBay <aw-confirm@ebay.com>
**To:** jsmith@mycompany.com

**eBay Safeharbor Department Notice**

**Fraud Alert ID : 90149980**

Dear eBay member,

You have received this email because you or someone else had used your identity to make false purchases on eBay. For security reasons, we are required to open an investigation on this matter. We treat online fraud seriously and all cases which cannot be resolved between eBay and the other involved party are forwarded for further investigations to the proper authorities.

To speed up this process, you are required to verify your personal information against the eBay account registration data we have on file by using the attached form. Please open it, complete the information requested, and send it back to us as soon as possible.

**Attachments:** eBayRestoreAcct.doc

2. **Assume you receive this email message from somebody who you don't know. Your boss told you in the morning that people will be sending you filled out forms, in Microsoft Word format, which you need to review. Such forms are necessary for applicants who want to become licensees of your employer's franchise. What should you do?**

   a) Do not open the attachment: It refers to something I do not remember

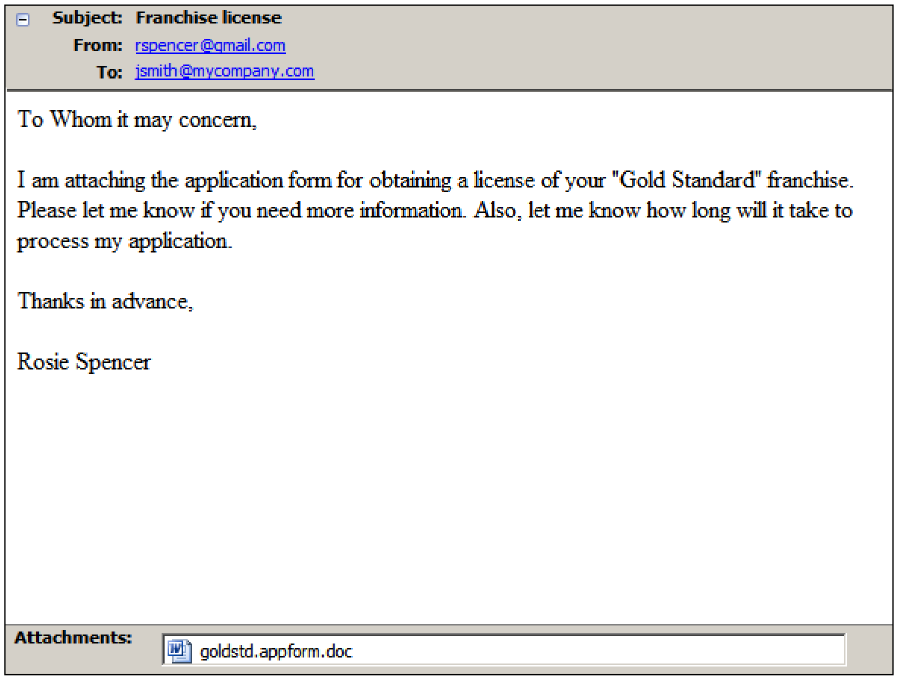
   **Reason**: This is an email about a job task you are aware of and that you are currently working on.

   b) Do not open the attachment: I do not know the sender

   **Reason**: same as a)

   c) *Open the attachment: I am expecting these attachments and are necessary to do my job*

   **Reason**: same as a)



**Subject:** **Franchise license**
**From:** rspencer@gmail.com
**To:** jsmith@mycompany.com

To Whom it may concern,

I am attaching the application form for obtaining a license of your "Gold Standard" franchise. Please let me know if you need more information. Also, let me know how long will it take to process my application.

Thanks in advance,

Rosie Spencer

**Attachments:** goldstd.appform.doc

3. **Suppose you receive this email that appears to be from another employee of the company you work for, and who you do know. It asks you to open the attached file without further explanation. You are currently not working in any project with her. What should you do?**

   a) *Do not open the attachment: It refers to an event I do not remember, or it does not convincingly explain the purpose of the attachment*

      **Reason**: Attackers often spoof legitimate email addresses, and may send infected attachments. If you are not expecting a message and attachment like this from a particular sender, it may be an attack.

   b) Open the attachment: I trust anybody working for my employer

      **Reason**: same as a)

   c) Do not open the attachment now, but will do so later

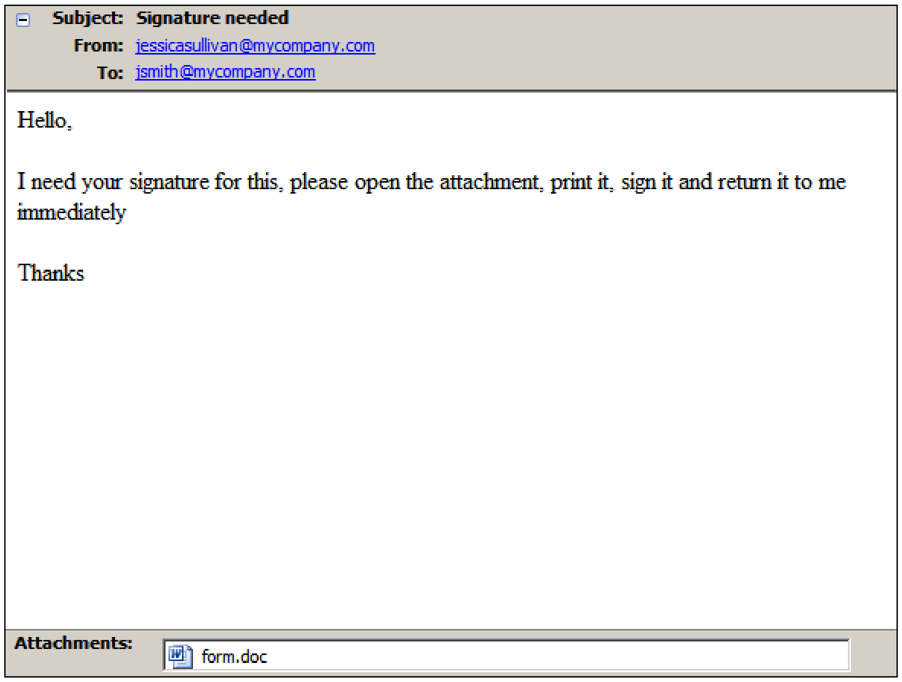
      **Reason**: same as a)

   d) Open the attachment: I know the sender

      **Reason**: same as a)

   e) Open the attachment: it's not my job to pay attention if dangerous emails arrive to my Inbox

      **Reason**: No automated security utility, e.g., antivirus software, detects all security threats, especially if they are very recent. You should always pay attention to what you receive in your Inbox. If your computer gets infected, you may not be able to complete your primary tasks on time.



**Subject: Signature needed**
**From:** jessicasullivan@mycompany.com
**To:** jsmith@mycompany.com

Hello,

I need your signature for this, please open the attachment, print it, sign it and return it to me immediately

Thanks

**Attachments:** form.doc

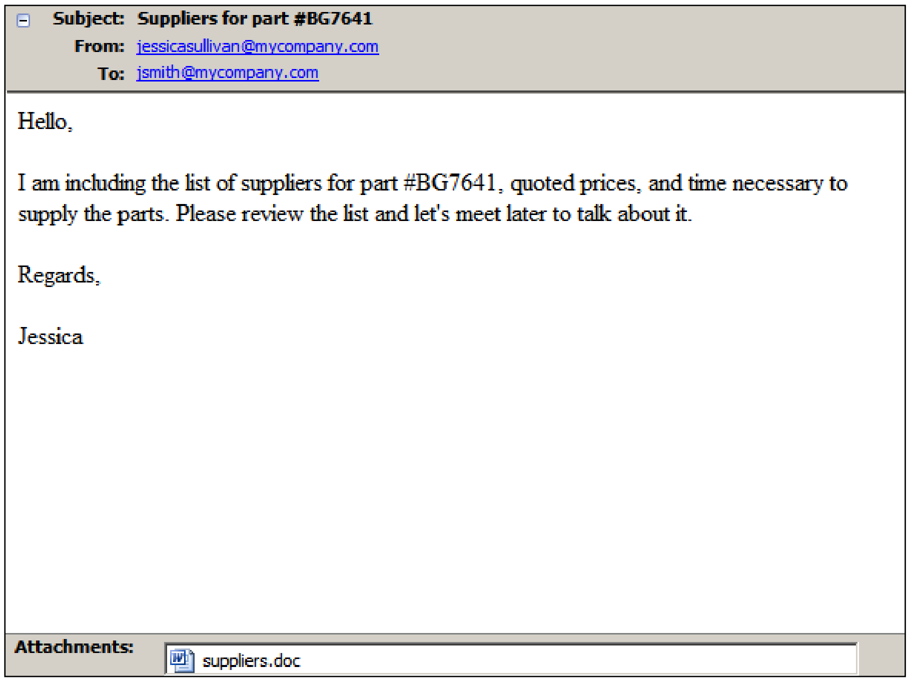
4. **Now imagine you receive this other email that appears to be from the employee of the company you work for, and who you do know and was referred to in the previous question. You both are working together to procure the parts for a new product that will be assembled and sold by your employer. What should you do?**

   a) Do not open the attachment: It refers to an event I do not remember, or it does not convincingly explain the purpose of the attachment

   **Reason**: This is an email from a known co-worker about a job task you are aware of and that you need to work on.

   b) <u>*Open the attachment: it refers to an activity of a project that I am aware of and in which I am currently working on.*</u>

   **Reason**: same as a)



**Subject: Suppliers for part #BG7641**
**From:** jessicasullivan@mycompany.com
**To:** jsmith@mycompany.com

Hello,

I am including the list of suppliers for part #BG7641, quoted prices, and time necessary to supply the parts. Please review the list and let's meet later to talk about it.

Regards,

Jessica

**Attachments:** suppliers.doc

# APPENDIX F

# FEEDBACK PROVIDED BY PARTICIPANTS ABOUT THE VICARIOUS SECURITY-CONDITIONING INTERVENTIONS

In this appendix we enumerate the comments given by participants about the vicarious security-conditioning interventions. First, we present users' remarks about the vicarious security-reinforcement video. Second, we list the users' opinions regarding the vicarious insecurity-punishment video. To preserve anonymity, the comments do not include the participants' numbers, and are listed in random order.

## F.1 VICARIOUS REINFORCEMENT INTERVENTION

Table F.11 shows the feedback given by users about the vicarious security-reinforcement video. Comments are grouped by whether they were positive, negative, or neutral.

**Table F.11:** Comments given by participants in the VC_SRA study

| *Positive* |
|---|
| The [quiz] after the video made me slow down and made me more aware of the choices I was making in choosing to open or not open attachments. |
| I thought the guidance in the sidebar was a very effective tool for making a decision when it comes to questionable emails and attachments. |
| The main [model] in the video verbalized aloud most things that I just think silently--though perhaps this was necessary for plot/exposition purposes. It was aimed at a good level of user--not a novice, but not especially savvy, either. |
| Besides the fact that the video was a bit on the cheesy side, it definitely got the point across. |
| Very interesting tactics and situations. It gave me something to think about when dealing with my email. Excellent. |
| The video was a bit cheesy, but no more so than most training videos are. The e-mail "help" was interesting and may be more helpful to people who are less suspicious than I am. |
| Amusing video. ;) Nice acting, guys. |
| *Negative* |
| The video was a bit lengthy, as a lot of ideas were repetitive. |
| The actors need to be more realistic in their depictions and a bit less exaggerated. |
| A little too much overacting |
| *Neutral* |
| Nothing |
| The video and experiment were fine |

## F.2 VICARIOUS PUNISHMENT INTERVENTION

Table F.12 shows feedback given by users about the vicarious insecurity-punishment video. For the VC_IPA study, participants were asked to state specifically what they liked about the video, and what they thought that needed improvement or did not like.

**Table F.12:** Participants' comments about what they liked or considered that needed improvement about the VC_IPA study's video

| Liked | Needs improvement |
|---|---|
| Presented very believable and common situations in the corporate workspace, as corporate mail filters aren't necessarily foolproof | If there were more examples of suspicious emails discussed, then the video may be a bit more informative for novice email users. |
| Scenarios were real-life situations. I have a habit of opening personal emails sent to my corporate account, now I will make wiser decisions. Also liked the 'rules' for opening emails with attachments | Nothing |
| How the camera changed from live person to computer screen and how it depicted real "time management" failures | Show the employee doing the non email piece work [shown in the first scene] |
| Kept my interest and while a little overdone, it was amusing. But it also clearly went over the identified security risks. Easy to understand. | Nothing, it was fine for an informational video |

| | |
|---|---|
| The [model] was pretty fun because he seemed to play up the role | A bit on the corny side, but overall corny is something that makes a video like that worth watching |
| N/A | The acting |
| It was a bit funny with the [model] talking to himself. He made stupid mistakes, but those mistakes are easily made by all of us. Good reminder to be more cautious with business AND personal accounts. | [In the video shown after the quiz], allow more time to read the choices [in the guidance] or let the narrator speak and explain the choices. |
| Humorous, but depicted information realistically | Nothing |

# APPENDIX G

# PRODUCTION OF VICARIOUS-CONDITIONING INTERVENTIONS

In this appendix we briefly explain the model recruitment process, and give details about the filming, and production stages of the vicarious interventions.

## G.1 RECRUITMENT AND SELECTION

To recruit actors, during April and May 2009 we placed flyers requesting talent for a short film in several schools of theater and performing arts in the city of Pittsburgh, such as those at the University of Pittsburgh, Carnegie-Mellon University, Duquesne University, and Point Park University. We also posted ads on several talent recruitment websites, and on craigslist.org, pittsburgh.backpage.com, and groups.google.com. Fellow PhD students, Roxana Gheorghiu and Nicholas Farnan helped with the selection of actors. From the actors interested and available, three were chosen: an actress for the female co-worker role, and two actors, one for the main model role, and the other for the boss role. The role of the male co-worker was played by Nicholas Farnan.

## G.2 FILMING, VOICE RECORDING, AND EDITING

Filming took place at a faculty office in the Computer Science department, during May 2009. Robert Hoffman from the department's tech support staff provided filming and voice recording equipment. Dr. John Ramirez, an outstanding lecturer at our department, helped with the narration of the interventions' introductory text, summaries at the end of each scene, important sections of the text in the reinforcement stimuli, and the text in the video (shown after the quiz) that introduced the guidance.

# BIBLIOGRAPHY

[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, “Influence of model's reinforcement contingencies on the acquisition of imitative responses,” *Journal of Personality and Social Psychology*, vol. 36, 1965, pp. 589-595.

[3] A. Bandura, *Principles of Behavior Modification*, Holt, Rinehart and Winston, 1969.

[4] A. Bandura, *Social learning theory*, Prentice-Hall, 1977.

[5] A. Triulzi, “Something amiss on Yahoo! Mail? No, it is a Symantec false positive!,” 2007; http://isc.sans.org/diary.html?storyid=2319.

[6] A. Whitten, and J.D. Tygar, “Safe staging for computer security,” in *Proceedings of the Workshop on Human-Computer Interaction and Security Systems*, 2003.

[7] A. Whitten, and J.D. Tygar, “Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0,” *Proceedings of the Eighth USENIX Security Symposium*, USENIX Association, pp. 169-184.

[8] A.J. DeWitt, and J. Kuljis, “Aligning usability and security: a usability study of Polaris,” *Proceedings of the second symposium on Usable privacy and security*, ACM, pp. 1-7.

[9] A.P. Goldstein, and M. Sorcher, *Changing supervisor behavior*, Pergamon Press, 1974.

[10] Agency for Workforce Innovation, “A quarter million Florida job seekers exposed,” 2008; https://www.floridajobs.org/security/security.htm.

[11] B. Klimt, and Y. Yang, “Introducing the Enron corpus,” in *First Conference on Email and Anti-Spam (CEAS)*, 2004.

[12] B. Schneier, *Beyond Fear: thinking sensibly about security in an uncertain world*, Copernicus Books, 2003.

[13] B. Schneier, *Secrets and Lies: Digital Security in a Networked World*, John Wiley and Sons, 2000.

[14] B. Tognazzini, “Design for usability,” in *Security and Usability: Designing Secure Systems That People Can Use*, L. Cranor, and S. Garfinkel eds., O'Reilly, 2005, pp. 31-46.

[15] B.F. Skinner, “Are theories of learning necessary?,” *Psychological Review*, vol. 57, no. 4, 1950, pp. 193-216.

[16] B.F. Skinner, “Operant behavior,” *American Psychologist*, vol. 18, no. 8, 1963, pp. 503-515.

[17] B.F. Skinner, *Contingencies of Reinforcement: a Theoretical Analysis*, Appleton-century-crofts, 1969.

[18] B.F. Skinner, *Science and human behavior*, Macmillan Pub Co, 1953.

[19] B.F. Skinner, *The behavior of organisms: An experimental analysis*, D. Appleton-Century Company, incorporated, 1938.

[20] C. Boulton, “Cloud Computing, Customer Wins, Microsoft Bashing Will Be Key at IBM Lotusphere,” 2009; http://www.eweek.com/c/a/Messaging-and-Collaboration/Cloud-Computing-Customer-Wins-Microsoft-Bashing-Will-Key-IBM-Lotusphere-2009/

[21] C. Jackson, D.R. Simon, D.S. Tan, and A. Barth, “An Evaluation of Extended Validation and Picture-in-Picture Phishing Attacks,” *Proceedings of Usable Security (USEC’07)*, 2007.

[22] C. Nodder, “Users and trust: A Microsoft case study,” in *Security and Usability: Designing Secure Systems That People Can Use*, L. Cranor, and S. Garfinkel eds., O'Reilly, 2005, pp. 589-606.

[23] C.B. Ferster, and B.F. Skinner, *Schedules of reinforcement*, Appleton-Century-Crofts, 1957.

[24] C.P. Pfleeger, and S.L. Pfleeger, *Security in computing*, Prentice Hall, 2006.

[25] Campaign Monitor, “Email client popularity,” http://campaignmonitor.com/stats/email-clients/

[26] Commtouch Inc., “New Trojan Variants Evade Major Anti-Virus Engines,” July 2009; http://www.commtouch.com/press-releases/new-trojan-variants-evade-major-anti-virus-engines-says-commtouch-report.

[27] Computing Technology Industry Association, “7th Trends in Information Security survey: A CompTIA Analysis of IT Security and the Workforce,” 2009; http://comptia.org/pressroom/get_pr.aspx?prid=1426.

[28] D. Balfanz, G. Durfee, D.K. Smetters, R.E. Grinter, and P.A.R. Center, "In search of usable security: Five lessons from the field," *IEEE Security & Privacy*, vol. 2, no. 5, 2004, pp. 19-24.

[29] D. Bell and L. La Padula, "Secure computer systems: Mathematical foundations," Technical Report MTR-2547, vol. I, MITRE Corporation, 1973.

[30] D. Weirich, and M.A. Sasse, "Pretty good persuasion: a first step towards effective password security in the real world," in *Proceedings of the 2001 workshop on New security paradigms*, ACM New York, NY, USA, 2001, pp. 137-143.

[31] D.K. Smetters, and R.E. Grinter, "Moving from the Design of Usable Security Technologies to the Design of Useful Secure Applications," *Proceedings of the 2002 workshop on New security paradigms*, ACM New York, NY, USA, pp. 82-89.

[32] Deloitte Touche Tohmatsu, "Protecting what matters: The 6th Annual Global Security Survey," 2009; http://deloitte.com/dtt/article/0,1002,cid%253D243032,00.html.

[33] F. Luthans, and R. Kreitner, *Organizational behavior modification and beyond: An operant and social learning approach*, Scott Foresman & Co, 1985.

[34] G. Robinson, "A statistical approach to the spam problem," Linux Journal, vol. 2003, no. 107, 2003.

[35] G. Rogers, "Microsoft says Gmail is a virus," 2006; http://blogs.zdnet.com/Google/?p=386.

[36] G.C. Walters, and J.E. Grusec, *Punishment*, WH Freeman San Francisco, CA, 1977.

[37] 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.

[38] G.S. Reynolds, *A primer of operant conditioning*, Scott, Foresman, 1975.

[39] H. Xia, and J.C. Brustoloni, "Hardening Web browsers against man-in-the-middle and eavesdropping attacks," in *Proceedings of the 14th international conference on World Wide Web*, ACM, 2005, pp. 489-498.

[40] H. Xia, J. Kanchana, and J.C. Brustoloni, "Using secure coprocessors to protect access to enterprise networks," *Lecture Notes in Computer Science*, vol. 3462, 2005, pp. 154-165.

[41] I. Flechais, J. Riegelsberger, and M.A. Sasse, "Divide and conquer: the role of trust and assurance in the design of secure socio-technical systems," *Proceedings of the 2005 workshop on New security paradigms*, ACM New York, NY, USA, pp. 33-41.

[42] I. Flechais, M.A. Sasse, and S.M.V. Hailes, "Bringing security home: a process for developing secure and usable systems," *Proceedings of the 2003 workshop on New security paradigms*, ACM New York, NY, USA, pp. 49-57.

[43] J. Cameron, and W.D. Pierce, *Rewards and intrinsic motivation: Resolving the controversy*, Bergin & Garvey, 2002.

[44] J. Cohen, *Statistical power analysis for the behavioral sciences*, Lawrence Erlbaum, 1988.

[45] J. Evers, "McAfee update exterminates Excel," 2006; http://news.cnet.com/2100-1002_3-6048709.html.

[46] J. Evers, "User education is pointless," 2006; http://news.cnet.com/2100-7350_3-6125213.html.

[47] J. Hu, C. Meinel, and M. Schmitt, "Tele-lab IT security: an architecture for interactive lessons for security education," in *Technical Symposium on Computer Science Education*, 2004.

[48] J. Nielsen, and J.N. Alertbox, "User education is not the answer to security problems," 2004; http://www.useit.com/alertbox/20041025.html.

[49] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. Cranor, "Crying Wolf: An Empirical Study of SSL Warning Effectiveness," in *The 18th USENIX Security Symposium* (to appear), 2009.

[50] J.A. Jacko, and A. Sears, *The Human-computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*, Lawrence Erlbaum Associates, 2003.

[51] J.C. Brustoloni, and R. Villamarín-Salomón, "Improving security decisions with polymorphic and audited dialogs," in *Proceedings of the 3rd symposium on Usable privacy and security*, ACM, 2007, pp. 76-85.

[52] J.H. Saltzer, and M.D. Schroeder, "The protection of information in computer systems," *Proceedings of the IEEE*, vol. 63, no. 9, 1975, pp. 1278-1308.

[53] J.J. Gonzalez, "Modeling Erosion of Security and Safety Awareness," in *Proceedings of the Twentieth International Conference of the System Dynamics Society*, 2002.

[54] J.J. Gonzalez, and A. Sawicka, "A framework for human factors in information security," in *WSEAS International Conference on Information Security*, 2002.

[55] J.J. Gonzalez, and A. Sawicka, "Modeling compliance as instrumental conditioning," in *Fifth International Conference on Cognitive Modeling*, 2003.

[56] J.J. Gonzalez, and A. Sawicka, “The role of learning and risk perception in compliance,” in *Proceedings of the 21st International Conference of the System Dynamics Society*, 2003.

[57] J.T. Reason, *Human error*, Cambridge University Press, 1990.

[58] K. Slovak, *Professional Outlook 2007 Programming*, John Wiley and Sons, 2007.

[59] K.D. Mitnick, and W.L. Simon, *The art of deception: Controlling the human element of security*, Wiley, 2002.

[60] Kumaraguru, P., Sheng, S., Acquisti, A., Cranor, L. F., and Hong, J. Teaching Johnny not to fall for phish. Tech. rep., Cranegie Mellon University, 2007. http://www.cylab.cmu.edu/files/cmucylab07003.pdf.

[61] L.F. Cranor, “A framework for reasoning about the human in the loop,” in *Proceedings of the 1st Conference on Usability, Psychology, and Security*, USENIX Association, 2008.

[62] L.F. Cranor, and S. Garfinkel, *Security and usability: Designing secure systems that people can use*, O'Reilly Media, Inc., 2005.

[63] L.F. Cranor, *Web privacy with P3P*, O'Reilly Media, 2002.

[64] L.R. Rogers, “Use Care When Reading Email with Attachments,” *news@sei*, vol. 6, no. 3, 2003.

[65] M. Bishop, “Psychological acceptability revisited,” in *Security and Usability: Designing Secure Systems That People Can Use*, L. Cranor, and S. Garfinkel eds., O'Reilly, 2005, pp. 1-12.

[66] M. Bishop, *Computer Security: Art and Science*, Addison Wesley, 2003.

[67] M. Domjan, and J.W. Grau, *The principles of learning and behavior*, Thomson/Wadsworth, 2003.

[68] M. Eysenck, *Psychology: A student’s handbook*, Psychology Press, 2000.

[69] M. McDowell, and A. Householder, “Cyber Security Tip ST04-010: Using caution with email attachments,” 2007; http://www.us-cert.gov/cas/tips/ST04-010.html.

[70] M. Oiaga, “Symantec Deems Visual Liturgy Unholy - Norton Antivirus labeled church software as spyware,” 2006; http://news.softpedia.com/news/Symantec-Deems-Visual-Liturgy-Unholy-31931.shtml.

[71] M. Wu, R.C. Miller, and G. Little, “Web wallet: preventing phishing attacks by revealing user intentions,” in *Proceedings of the second symposium on Usable privacy and security*, ACM, 2006, pp. 102-113.

[72] M. Wu, R.C. Miller, and S.L. Garfinkel, “Do security toolbars actually prevent phishing attacks?,” *Proceedings of the SIGCHI conference on Human Factors in computing systems*, ACM New York, NY, USA, pp. 601-610.

[73] M. Wu, R.C. Miller, and S.L. Garfinkel, “Do security toolbars actually prevent phishing attacks?,” *Proceedings of the SIGCHI conference on Human Factors in computing systems*, ACM New York, NY, USA, pp. 601-610.

[74] 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.

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

[76] M.E. Zurko, and R.T. Simon, “User-centered security,” *Proceedings of the 1996 workshop on New security paradigms*, ACM New York, NY, USA, pp. 27-33.

[77] 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,” *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*, pp. 371-381.

[78] M.W. Eysenck, *Psychology: an international perspective*, Psychology Press, 2004.

[79] Miami University, “Miami notifying students, alumni of privacy breach,” 2005; http://www.miami.muohio.edu/news/article/view/3435.

[80] Mozilla Corporation, “Firefox – Rediscover the Web,” http://www.mozilla.com/en-US/firefox/.

[81] Mozilla Corporation, “Thunderbird - Reclaim your inbox,” http://www.mozillamessaging.com/en-US/thunderbird/.

[82] Mozilla Developer Center, “Enhanced Extension Installation,” https://developer.mozilla.org/en/Enhanced_Extension_Installation

[83] Mozilla Developer Center, “Install Manifests,” https://developer.mozilla.org/en/install.rdf#hidden

[84] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu, “The ghost in the browser: Analysis of web-based malware,” *Proceedings of the First Workshop on Hot Topics in Understanding Botnets*, USENIX Association, p. 4.

[85] N.A. Macmillan, and C.D. Creelman, *Detection theory: A user’s guide*, Cambridge University Press, 1991.

[86] P. Dourish, and D. Redmiles, “An approach to usable security based on event monitoring and visualization,” *Proceedings of the 2002 workshop on New security paradigms*, ACM New York, NY, USA, pp. 75-81.

[87] P. Dourish, J. Delgado de la Flor, and M. Joseph, “Security as a Practical Problem: Some Preliminary Observations of Everyday Mental Models,” in *Proceedings of CHI2003 Workshop on Human-Computer Interaction and Security Systems*, 2003.

[88] 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 *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2007, pp. 905 - 914.

[89] 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 *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*, ACM New York, NY, USA, 2007, pp. 70-81.

[90] 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, pp. 323-332.

[91] P.W. Dowrick, *Practical guide to using video in the behavioral sciences*, Wiley New York, 1991.

[92] R. De Paula, X. Ding, P. Dourish, K. Nies, B. Pillet, D. Redmiles, J. Ren, J. Rode, and R. Silva Filho, “Two experiences designing for effective security,” *Proceedings of the 2005 symposium on Usable privacy and security*, ACM New York, NY, USA, pp. 25-34.

[93] R. DeCharms, *Personal causation: The internal affective determinants of behavior*, Academic Press, 1968.

[94] R. Dhamija, and A. Perrig, “Deja vu: A user study using images for authentication,” in *Proceedings of the 9th USENIX Security Symposium*, 2000, pp. 45-48.

[95] R. Dhamija, J.D. Tygar, and M. Hearst, “Why phishing works,” *Proceedings of the SIGCHI conference on Human Factors in computing systems*, ACM New York, NY, USA, pp. 581-590.

[96] R. Kanawati, and M. Riveill, “Access Control Model for Groupware Applications,” in *Proceedings of Human Computer Interaction*, University of Huddersfield, 1995, pp. 66-71.

[97] R. Villamarín-Salomón, “Videos used for vicarious security-conditioning experiments,” 2009; http://www.cs.pitt.edu/~rvillsal/vc.htm.

[98] R. Villamarín-Salomón, J. Brustoloni, M. DeSantis, and A. Brooks, "Improving User Decisions About Opening Potentially Dangerous Attachments In E-Mail Clients," in *Symposium on Usable Privacy and Security, CMU*, 2006, pp. 76 - 85.

[99] R.G. Miltenberger, *Behavior modification: Principles and procedures*, Cole Publishing Company, 1997.

[100] R.H. Hoyle, *Statistical strategies for small sample research*, Sage Publications Inc, 1999.

[101] R.K. Wong, H.L. Chau, and F.H. Lochovsky, "A Data Model and Semantics of Objects with Dynamic Roles," *Proceedings of the Thirteenth International Conference on Data Engineering*, IEEE Computer Society Washington, DC, USA, pp. 402-411.

[102] S. Brostoff and M.A. Sasse, "Safe and sound: a safety-critical approach to security," *Proceedings of the 2001 workshop on New security paradigms*, ACM, 2001, pp. 41-50.

[103] S. Egelman, L.F. Cranor, and J. Hong, "You've Been Warned: An Empirical Study of the Effectiveness of Web Browser Phishing Warnings," in *Proceedings of the Conference on Human Factors in Computing Systems*, ACM SIGCHI, 2008.

[104] S. Görling, "The myth of user education," *Proceedings of the 16th Virus Bulletin International Conference*, pp. 11–13.

[105] S. Pahnila, M. Siponen, and A. Mahmood, "Employees' Behavior towards IS Security Policy Compliance," in *Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, IEEE Computer Society, 2007, pp. 156b.

[106] 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 *Proceedings of the 3rd symposium on Usable privacy and security*, ACM New York, NY, USA, 2007, pp. 88-99.

[107] S. Sheng, B. Wardman, G. Warner, L.F. Cranor, J. Hong, and C. Zhang, "Empirical Analysis of Phishing Blacklists," in *Proceedings of The 6th Conference on Email and Anti-Spam (CEAS)*, 2009.

[108] S.R. Flora, *The power of reinforcement*, State University of New York Press, 2004.

[109] S.W. Lee, *Encyclopedia of school psychology*, Sage, 2005.

[110] SANS Institute, "SANS Top-20," http://www.sans.org/top20/#c3

[111] Trusted Computing Group, "Trusted Network Connect," https://www.trustedcomputinggroup.org/groups/network/.

[112] US-CERT, "Technical Cyber Security Alert TA06-139A -- Microsoft Word Vulnerability," 2006; http://www.us-cert.gov/cas/techalerts/TA06-139A.html.

[113] V. Anandpara, A. Dingman, M. Jakobsson, D. Liu, and H. Roinestad, "Phishing IQ Tests Measure Fear, Not Ability," *Financial Cryptography and Data Security: 11th International Conference, FC 2007, and First International Workshop on Usable Security, USEC 2007*, Springer-Verlag New York Inc, p. 362.

[114] W. Kennedy, "Blocked Attachments: The Outlook Feature You Love to Hate," 2007; http://office.microsoft.com/en-us/outlook/HA011894211033.aspx.

[115] W. McDougall, *An introduction to social psychology*, Methuen, 1908.

[116] W.K. Edwards, "Policies and roles in collaborative applications," *Computer Supported Cooperative Work: Proceedings of the 1996 ACM conference on Computer supported cooperative work*, Association for Computing Machinery, Inc, One Astor Plaza, 1515 Broadway, New York, NY, 10036-5701, USA.

[117] W.K. Edwards, E.S. Poole, and J. Stoll, "Security Automation Considered Harmful?," in *Proceedings of the IEEE New Security Paradigms Workshop (NSPW)*, 2007.

[118] Y. Blandin, and L. Proteau, "On the cognitive basis of observational learning: development of mechanisms for the detection and correction of errors," *The Quarterly journal of experimental psychology: Human experimental psychology*, vol. 53, no. 3, 2000, pp. 846-867.

[119] C. Herley, "So Long, And No Thanks for the Externalities: The Rational Rejection of Security Advice by Users," *Proceedings of the New Security Paradigms Workshop*.

[120] G.L. Blakely, E.H. Blakely, and R.H. Moorman, "The effects of training on perceptions of sexual harassment allegations," *Journal of Applied Social Psychology*, vol. 28, 1998, pp. 71-83.

[121] R. Chen, "The Old New Thing: The default answer to every dialog box is 'Cancel'," http://blogs.msdn.com/oldnewthing/archive/2003/09/01/54734.aspx.

[122] Directorate for Command, Control, Communications, and Computer Systems, *Information assurance through DEFENSE IN DEPTH*, Joint Chiefs of Staff, 2000.

[123] N.H. Azrin, and W.C. Holz, "Punishment," in *Operant behavior: Areas of research and application*, 1966, pp. 380-447.