2008

A Framework for Reasoning About the Human in the Loop

Lorrie F. Cranor
Carnegie Mellon University

Published In
.

A Framework for Reasoning About the Human in the Loop

Lorrie Faith Cranor
Carnegie Mellon University
lorrie@cmu.edu

Abstract

Many secure systems rely on a "human in the loop" to perform security-critical functions. However, humans often fail in their security roles. Whenever possible, secure system designers should find ways of keeping humans out of the loop. However, there are some tasks for which feasible or cost effective alternatives to humans are not available. In these cases secure system designers should engineer their systems to support the humans in the loop and maximize their chances of performing their security-critical functions successfully. We propose a framework for reasoning about the human in the loop that provides a systematic approach to identifying potential causes for human failure. This framework can be used by system designers to identify problem areas before a system is built and proactively address deficiencies. System operators can also use this framework to analyze the root cause of security failures that have been attributed to "human error." We provide examples to illustrate the applicability of this framework to a variety of secure systems design problems, including anti-phishing warnings and password policies.

"Humans are incapable of securely storing high-quality cryptographic keys, and they have unacceptable speed and accuracy when performing cryptographic operations. (They are also large, expensive to maintain, difficult to manage, and they pollute the environment. It is astonishing that these devices continue to be manufactured and deployed. But they are sufficiently pervasive that we must design our protocols around their limitations.)"
– C. Kaufman, R. Perlman, and M. Speciner, 2002 [20]

1. Introduction

It is becoming increasingly apparent that humans are a major cause of computer security failures. In the context of secure systems, humans are often thought of as "the weakest link in the chain" [31]. In 2006 the SANS Top 20 Internet Security Vulnerabilities report began listing human vulnerabilities along with software vulnerabilities [25]. A 2006 Computing Technology Industry Association survey found that security managers attribute approximately 60 percent of security breaches to human error, up from 47% the previous year [6]. Social engineering attacks and lack of compliance with organizational security policies are increasingly cited as security concerns.

With so many security failures attributed to humans, secure systems that do not rely on a "human in the loop" to perform security-critical functions are attractive. Automated components are generally more accurate and predictable than humans, and automated components don't get tired or bored [14]. Indeed, in some areas we have seen significant progress towards secure systems that "just work" without human intervention. For example, while early anti-virus programs prompted users to make a decision about every detected virus, today many anti-virus programs automatically repair or quarantine infected files in their default mode of operation. Thus, anti-virus software no longer relies on inexperienced users to make security-critical judgments. When software is likely to be able to make a better security decision than a human, removing the human from the loop may be wise. Likewise, when a user is unlikely to have relevant insights into which configuration options to choose, well-chosen default settings may result in better security-configurations than most humans would achieve on their own.

In some cases we don't seem to be able to avoid relying on humans to perform security-critical functions. There are a number of reasons why it may not be feasible or desirable to automate these functions completely [11], [14]. Some tasks rely on human knowledge that is currently difficult for a computer to reason about or process. For example, today humans tend to be better than computers at recognizing faces in crowds or noticing other humans who are acting suspiciously [30]. When tasks rely on human knowledge about context, it may be difficult to capture the entire necessary context in a way that a computer can reason about it. For example, a human may be a better judge than a computer about whether an email attachment is suspicious in a particular context. We also rely on humans to make some security-related configuration decisions and to apply policies when it is difficult to encode all of the nuances of a policy or program a computer to handle special cases.

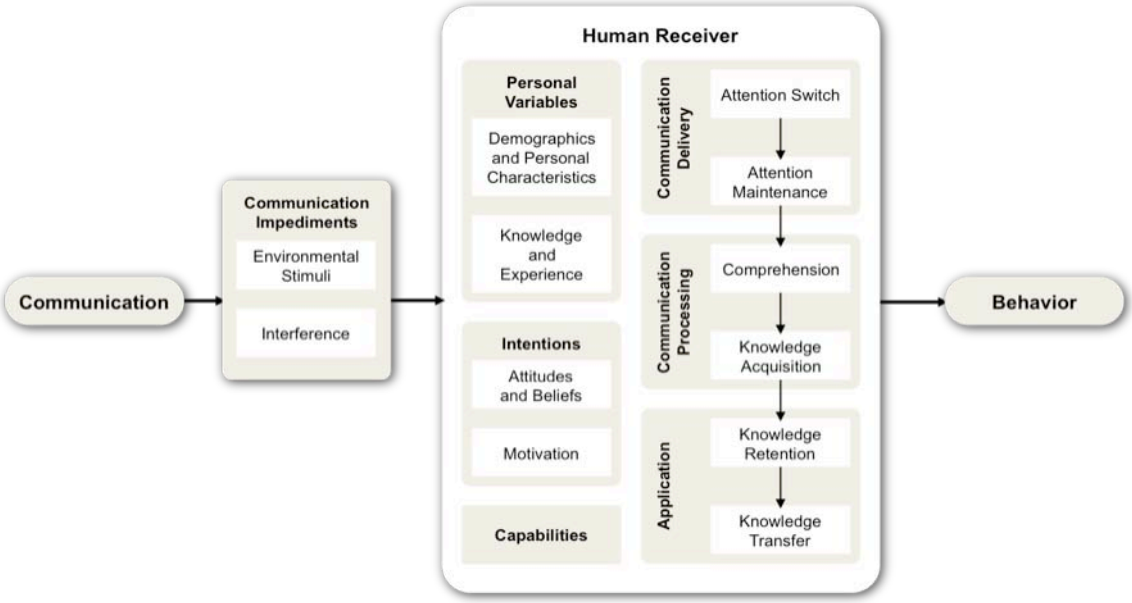

Figure 1. The human-in-the-loop security framework

In some cases a completely automated system might be too restrictive, inconvenient, expensive, or slow. Some secure systems rely on humans to manipulate or protect physical components, for example, insert a smartcard into a reader and remove it before walking away from the computer. When secure systems rely on the use of secrets, humans must typically store and protect the secrets, and produce them when needed. In addition, most systems that restrict access to only certain humans rely on those humans to participate in the authentication process.

When secure systems rely on humans in the loop to perform security-critical functions, threats to system security include not only malicious attackers, but also non-malicious humans who don't understand when or how to perform security-related tasks, humans who are unmotivated to perform security-related tasks or comply with security policies, and humans who are not capable of making sound security decisions. To protect against these threats, we must systematically analyze the human role in a secure system to identify potential failure modes and find ways to reduce the likelihood of failure. In this paper we propose a framework for reasoning about the human in the loop in a secure system. In Section 2 we provide an overview of the framework and describe its components. In Section 3 we explain how to apply the framework, and show how it can be applied to make recommendations in two different settings. We discuss related work in Section 4 and discuss our conclusions and future work in Section 5.

2. The framework

The human-in-the-loop security framework is designed to help us understand the behavior of humans whom we expect to perform security-critical functions. We are primarily interested in understanding the behavior of non-malicious humans—those who intend to maintain the security of a system as well as those who are indifferent to system security, but do not intend to attack the system. While this framework provides some insights into the behavior of attackers, traditional computer security threat modeling techniques may be better suited to identifying vulnerabilities that attackers may exploit.

The human-in-the-loop security framework is based on a simple communication-processing model in which a communication is sent to a receiver (the human user), triggering some behavior, as shown in Figure 1. What behavior occurs is dependent on the outcome of several information processing steps taken by the receiver, as well as the personal characteristics of the receiver and the influence of communication impediments. Our framework is built on the Communication-Human Information Processing (C-HIP) model from the warnings science literature [37]. More general transmission models of communication are discussed in the communication theory literature [19]. We discuss the ways our framework differs from the C-HIP model and explain the rationale behind developing a new framework in Section 4.

We based our framework on a communication processing model because security-related actions by non-

experts are generally triggered by a security-related communication—for example an on-screen alert, software manual, or security tutorial. Indeed, if a human security failure occurs and there is no associated communication that should have triggered a security-critical action, the lack of communication is likely at least partially responsible for the failure. In such situations, if it is not feasible to automate the security-critical action, a good starting point for analysis would be to consider possible communications that might be added to the system and determine whether they would have been likely to prevent the failure. Experts may initiate security-related actions on their own, based on their past training (a communication), or a self-discovered security technique. In the latter case, an individual's decision to carry out a security plan might be modeled as a communication to oneself. For example, an expert may notice that emails containing hyperlinks in them seem rather suspicious and might decide to adopt the strategy of examining these links before clicking on them. We can consider the decision to adopt this strategy as the relevant communication as we apply our framework.

The framework facilitates the analysis of a wide range of secure systems that rely on humans. For example, the warnings provided by anti-phishing toolbars or other security tools are communications that may or may not result in the user heeding the warning, and password policies are communications that may or may not result in users selecting strong passwords.

The human-in-the-loop security framework is not intended as a precise model of human information processing, but rather it is a conceptual framework that can be used much like a checklist to systematically analyze the human role in secure systems. The framework, summarized in Appendix A and described in detail below, includes factors and information processing stages that will impact security-related behaviors. However, the relationships between the various components are intentionally vague. While Figure 1 depicts a temporal flow from communication delivery, through communication processing, through application, this should not be interpreted as a strictly linear process. In practice, some of these steps may be omitted or repeated.

2.1 Communication

The first component of the human-in-the-loop framework is the communication, which if all goes well will trigger an appropriate behavior. We distinguish five types of communications that are relevant to security tasks: warnings, notices, status indicators, training, and policies.

Warnings are communications that alert users to take immediate action to avoid a hazard. For example, web browsers provide a variety of active pop-up warnings as well as passive warning indicators in the browser chrome to alert users to phishing web sites, expired SSL certificates, and other hazards. While some warnings merely alert users about the presence of a hazard, the most effective warnings generally provide clear instructions about how to avoid the hazard. Effective warnings must get the users' attention and convince them to take an action to avoid or mitigate a hazard. Warnings experts emphasize that warnings should be used only as a last resort when it is not feasible to design a system that fully protects against a hazard [36]. Thus, in cases where we can use software to accurately identify and thwart security threats without user involvement, that is generally preferable to the use of warnings.

Notices inform users about characteristics of an entity or object. For example privacy policies and SSL certificates are examples of frequently-encountered notices. Notices may be used by users to evaluate an entity and determine whether interacting with it would be hazardous or consistent with their security or privacy needs. However, generally notices are not intended to be used when an automated tool has determined that a hazard is imminent. Effective notices provide users with the information they need to make judgments.

Status indicators inform users about system status information. Generally they have a small number of possible states. Examples of status indicators include taskbar and menu bar indicators that show whether Bluetooth has been enabled or whether anti-virus software is up to date. File permissions information can also be thought of as a status indicator.

Training communications are intended to teach users about security threats and how to respond to them. They may take a variety of forms including tutorials, games, instruction manuals, web sites, emails, seminars, courses, and videos. When training is effective, users will not only learn concepts and procedures, but also remember what they learned and recognize situations where they need to apply what they have learned.

Policy communications are documents that inform users about system or organizational policies that they are expected to comply with. For example, many organizations have policies about the types of passwords people may use and what types of documents must be encrypted. Policy documents are frequently part of employee handbooks and ISP terms of service documents. Policies may also be conveyed as part of memos or training communications. If policies are to be effective, users must recognize situations where the policy is applicable, understand how to apply the policy, and

plicable, understand how to apply the policy, and have the capability and motivation to comply.

Some communications fall into more than one of the above categories. For example, notices and status indicators provide objective information for users to interpret as they wish or to consider as part of a troubleshooting process; however, some might also function as warnings when they provide information that may enable users to make risk-reducing decisions [8].

Another way to classify security communications is to put them on a scale from passive to active. The *active* communications are designed to interrupt the user's primary task and force them to pay attention, while the *passive* communications are available to the user, but easily ignored. At the most extreme active end of the spectrum, the user cannot proceed with the primary task until the user has taken an action related to the communication. For example, the Firefox anti-phishing tool prevents Firefox from loading suspected Phishing web sites unless a user clicks on a link to override the tool's recommendation. Other active indicators might play sounds or animations to get a user's attention, without blocking the primary task. Passive communications, on the other hand, may simply change the color of an icon without doing anything to attract the user's attention.

Secure systems designers should consider which type of communication will be most effective in a particular system, as well as where to place it on the active-passive spectrum. They should consider the severity of the hazards that the system is attempting to avoid, the frequency with which the hazard is encountered, and the extent to which appropriate user action is necessary to avoid the hazard. For example, frequent, active warnings about relatively low-risk hazards or hazards that ordinary users are unable to take action to avoid may lead users to start ignoring not only these warnings, but also similar warnings about more severe hazards. A more passive notice or status indicator might be a better choice than a warning in such situations, as it will provide information that may be of use to expert users without interrupting ordinary users for whom it is of minimal use. On the other hand, when hazards are severe and user action is critical, active warnings may be most appropriate, perhaps with links to relevant training.

2.2 Communication impediments

Both environmental stimuli and interference may cause a partial or full communication failure. In the most extreme cases, the receiver might not even be aware that a communication was sent.

Environmental stimuli are communications and activities that may divert the user's attention away from the security communication. These include other related and unrelated communications, ambient light and noise, and the user's primary task (which the security communication may interrupt). The more passive the communication, the more likely environmental stimuli will prevent users from noticing it. Passive indicators may also compete with each other for the user's attention. For example, a number of security-related Firefox extensions add passive indicators that clutter the browser chrome, making it difficult for users to focus on any particular one.

Interference is anything that may prevent a communication from being received as the sender intended. Interference may be caused by malicious attackers, technology failures, or environmental stimuli that obscure the communication (for example, ambient noise that prevents a user from hearing an audio alert).

Traditional secure systems analysis typically focuses on interference, examining whether a critical communication might be blocked or manipulated. In the case of security indicators, the analysis first examines whether the indicator behaves correctly when not under attack, and then whether it behaves correctly when under attack. Does the correct indicator appear at the correct time without false positives or false negatives? Is the indicator resistant to attacks designed to deceive the software into displaying an inappropriate indicator? Can the indicator be spoofed, obscured, or otherwise manipulated so that users are deceived into relying on an indicator provided by an attacker rather than one provided by their system? For example, Ye, et al demonstrated how malicious web servers could cause browsers to display SSL lock icons and certificates, even when no SSL session has been established [41].

2.3 Human receiver

The *human receiver* is the human who receives the security communication and whose actions will impact system security. The receiver brings to the situation a set of personal variables, intentions, and capabilities that impact a set of information processing steps: communication delivery, communication processing, and application. We also refer to the receiver as "the user" and "the human in the loop."

2.3.1 Communication delivery

The first information-processing step is *communication delivery*, which includes both attention switch and attention maintenance. The communication will not succeed unless users notice (*attention switch*) the commu-

nication or are made aware of rules, procedures or training messages. They must also pay attention to the communication long enough to process it (*attention maintenance*). In the case of symbolic indicators, this may mean simply focusing their attention on the indicator long enough to recognize it. For other communications this may mean spending time reading, watching, or listening to it fully. Environmental stimuli and interference, as well as characteristics of the communication (format, font size, length, delivery channel) will impact attention switch and maintenance [38]. In addition, communication delivery may also be impacted by *habituation,* the tendency for the impact of a stimulus to decrease over time as people become more accustomed to it. In practice this means that over time users may ignore security indicators that they observe frequently.

It can be easy for a system designer to overlook communication delivery failures as a potential underlying cause of human security failures, especially when the communication is properly displayed and the user simply fails to notice it. However, there is evidence that many users don't notice a variety of security indicators in software they use regularly. For example, user studies indicate that some users have never noticed the presence of the SSL lock icon in their web browser [9], [10]. A study that used an eye tracker to observe participants' behaviors when visiting SSL-enabled websites found that most users do not even attempt to look for the lock icon [35].

2.3.2 Communication processing

The next information-processing step is *communication processing*, which includes comprehension and knowledge acquisition. *Comprehension* refers to the user's ability to understand the communication. The user's familiarity with indicator symbols and their similarity to related symbols, the conceptual complexity of the communication, and the vocabulary and sentence structure will all impact comprehension. Short, jargon-free sentences, use of familiar symbols, and unambiguous statements about risk will aid comprehension [18]. *Knowledge acquisition* refers to the user's ability to not only understand a communication, but also to learn what to do in response to it. A user may comprehend a security warning and understand that they must take action to avoid a hazard, yet be unsure about what specific steps to take to avoid that hazard. Knowledge acquisition depends on the extent of training provided to the user and their involvement during the training. In many cases warnings are accompanied by little or no training in hazard avoidance, and thus unless users have received previous training they are unlikely to know what they are supposed to do when they see the warning. Thus, a good warning will include specific instructions on how to avoid the hazard.

Communicating clearly to non-experts about computer security is challenging, and communication-processing failures are common. It is difficult to write about computer security concepts without using technical jargon, and security-related concepts are difficult to represent clearly with icons. For example, most users do not understand the meaning of web browser security symbols and pop-up warnings [10], [15]. Anecdotal evidence suggests that the lock icon variations used by Firefox 2 to indicate certificate problems and the eyeball icons used by IE and Netscape Navigator to indicate blocked cookies are meaningless to the vast majority of users, including security experts.

2.3.3 Application

The final information-processing step is *application,* which consists of knowledge retention and knowledge transfer. *Knowledge retention* refers to the user's ability to remember the communication when a situation arises in which they need to apply it, and to recognize and recall the meaning of symbols or instructions. Knowledge retention is impacted by the frequency and familiarity of the communication, the user's long-term memory abilities, and the level of interactivity of training activities. *Knowledge transfer* refers to the ability of users to recognize situations where the communication is applicable and figure out how to apply it. Knowledge transfer is impacted by the level of interactivity of training activities as well as the degree of similarity between training examples and situations where knowledge should be applied. For example, users may retain knowledge from anti-phishing training, and use it to analyze email messages similar to those shown in the training materials. If they can apply this knowledge to other types of email messages, or even to suspicious messages sent through other channels then they have also transferred this knowledge. In the case of security warnings that appear automatically whenever the system detects a hazard, knowledge transfer may be unnecessary, as there is no need for the user to figure out on their own when a warning is applicable.

2.3.4 Personal variables

Personal variables include demographics and personal characteristics, as well as knowledge and experience. *Demographics and personal characteristics* that impact the receiver may include age, gender, culture, education, occupation, and disabilities. When designing a secure system that relies on humans, it is important to consider who these humans are likely to be and what their personal characteristics suggest about how they are likely to behave. It is also important to consider what relevant *knowledge and experience* these humans are likely to have. The users' education, occupation, and prior experience will impact this. Personal variables

may impact a user's ability to comprehend and apply communications, and their intention and capability to take appropriate actions. For example, expert users with computer-security-related training and experience may be more likely to understand complicated instructions than novice users. On the other hand, experts may also be more likely to second-guess security warnings and, perhaps erroneously, conclude that the situation is less risky than it actually is.

2.3.5 Intentions

Intentions includes attitudes and beliefs, as well as motivation – factors that will influence whether a user decides that a communication is worth paying attention to and acting upon. A number of theories and models of behavioral compliance are useful for understanding why users may receive and comprehend a communication, yet decide not to bother complying with it [21]. Relevant *attitudes and beliefs* include beliefs about the accuracy of the communication, whether they should pay attention to it, their ability to complete recommended actions successfully (*self-efficacy*), whether recommended actions will be effective (*response-efficacy*), how long it will take to complete recommended actions, and their general attitude towards the communication (trust, annoyance, etc.) [4]. *Motivation* addresses the incentives users have to take the appropriate action and to do it carefully or properly. Since security communications often distract users from their primary tasks, security goals may be in conflict with a user's primary goals and their attitudes and motivation may be colored by their perception of security risk and importance they place on the security communication versus the primary task. If users tend to view delays in completing the primary task as more important to avoid than security risks, then they will tend to ignore the security communication. Attitudes and beliefs might be further influenced by a user's past experience with a particular security indicator. For example, if the indicator has displayed erroneous warnings (false positives) in the past, users may be less inclined to take it seriously. Organizations might create incentives for complying (or disincentives for not complying) with security policies that serve as additional motivation.

There are a number of approaches that can be taken to motivate users to take security-related actions. Whenever possible, security tasks should be designed so that they are easy to perform and minimize disruption to a user's workflow so that users do not perceive these tasks as overly burdensome. Users should also be taught about the security risks involved so they can appreciate the consequences of security failures and that their actions might lead to security failures or hazard avoidance. Any relevant cultural norms that might serve as disincentives to good security practice should also be identified and addressed. Finally, within an organization, rewards and punishments may be useful motivational tools.

2.3.6 Capabilities

Finally, even if receivers comprehend a communication, understand how to apply it, and recognize a situation where they should apply it, a failure may still occur if they do not have the *capability* to take the appropriate actions. Depending on what these actions are, specific knowledge, or cognitive or physical skills may be necessary to complete the action. In some cases specific software or devices may also be required. For example, the ability to remember random-looking strings is a capability demanded by many password policies. Many users fail to comply with these policies because they are not capable of performing this memory task.

2.4 Behavior

The goal of a security communication is to produce the desired behavior. The receiver's information processing steps, personal variables, intention, and capability, interact with any environmental stimuli and interference to influence the human receiver's behavior. In the best case, the receiver properly understands what action needs to be taken and proceeds to take that action. However, failures may still occur at this stage when users are unable to actually complete the action successfully or determine whether they have carried out the action properly. Norman describes the gap between a person's intentions to carry out an action and the mechanisms provided by a system to facilitate that action as a *Gulf of Execution* [26]. For example, a user may be aware that anti-virus software is out of date and that they need to take action to update this software. However, they may be unable to find the menu item in their anti-virus software that facilitates this update. On the other hand, a user may complete an action, but may be unable to interpret the results of the action to determine whether it was successful. Norman refers to this as a *Gulf of Evaluation* [26]. The Gulf of Evaluation is large when it is difficult for users to determine what state a system is in. For example, Maxion and Reeder found that users have trouble determining effective file permissions in Windows XP [24]. Thus, when users change file permissions settings, it is difficult for them to determine whether they have achieved the desired outcome.

The gulf of execution and gulf of evaluation can be minimized through good design. To minimize the gulf of execution, designers should make sure security communications include clear instructions about how to execute the desired actions. They should also examine

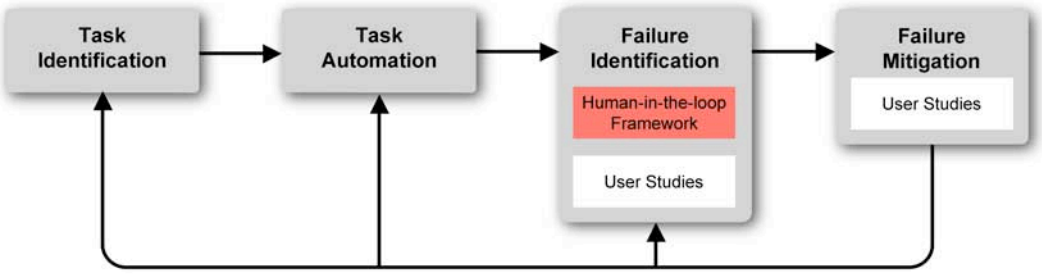

Figure 2. Human threat identification and mitigation process

the interface components or hardware that must be manipulated to make sure the proper use of these components is readily apparent. To minimize the gulf of evaluation, designers should make sure that software and devices provide relevant feedback so that users can determine whether their actions have resulted in the desired outcome. For example, after a usability study of cryptographic smart cards revealed that users have trouble figuring out how to insert these cards into card readers, Piazzalunga et al. recommended that visual cues be printed on the cards themselves (reducing the gulf of execution) and that the card readers provide feedback to indicate when a card has been properly inserted (reducing the gulf of evaluation) [27].

James Reason developed the Generic Error-Modeling System (GEMS), a theory of human error that distinguishes three types of errors: mistakes, lapses, and slips. *Mistakes* occur when people formulate action plans that will not achieve the desired goal. For example, a naïve user may decide to evaluate an email message before opening an attachment by checking to see whether the message was sent by someone the user knows. However, this plan will result in a mistake when a friend's computer is infected with a worm that propagates by email messages to everyone in her address book. *Lapses* occur when people formulate suitable action plans, but then forget to perform a planned action, for example skipping a step in a sequence of actions. *Slips* occur when people perform an action incorrectly, for example press the wrong button or select the wrong menu item [28].

Good design can reduce the chance of mistakes, lapses, and slips. Designers should develop clear communications that convey specific instructions so as to reduce the chance that users will make mistakes while completing security-critical tasks. They should minimize the number of steps necessary to complete the task and, whenever possible, provide cues to guide users through the sequence of steps and prevent lapses. To prevent slips, designers should locate the necessary controls where they are accessible and arrange and label them so that they will not be mistaken for one another.

Secure systems often rely on randomness to prevent attackers from exploiting predictable patterns to breach system security. Thus, failure can also occur at the behavior stage when users successfully perform the desired actions, but do so in a manner that follows predictable patterns that an attacker might exploit. For example, a number of studies on graphical passwords have found that users select these passwords according to predictable patterns. Davis et al. found that students using a graphical password system based on images of faces tended to select attractive faces from their own race. They demonstrated that an attacker who knows the race and gender of a user can use that knowledge to substantially reduce the number of guesses required to guess a password. Thorpe and van Oorschot found that many background images used in click-based graphical password schemes have a small number of popular "hot spots" from which users tend to select their password click points. They demonstrate techniques an attacker can use to learn what these hot spots are and substantially reduce the number of guesses required to guess a password [34]. Designers should consider whether an attacker might be able to exploit predictable user behavior, and if so, find ways to encourage less predictable behavior or prevent users from behaving in ways that fit known patterns (e.g. prohibit passwords that contain dictionary words).

3. Applying the framework

We designed the human-in-the-loop security framework to be used as part of a four-step iterative process in which human threats to system security are identified and mitigated. This process is illustrated in Figure 2. The human threat identification and mitigation process can be conducted at the design phase to proactively reduce opportunities for human security failures. It can also be conducted after a system has been implemented

and deployed to identify the cause of observed failures and find ways of mitigating them.

In the *task identification step* the secure system designer identifies all of the points where the system relies on humans to perform security-critical functions. This can be done by enumerating the points of human interaction with a secure system and considering which of these interactions are critical to security.

In the *task automation* step, the designer attempts to find ways to partially or fully automate some of the security-critical human tasks. This may or may not be possible or desirable, depending on the type of task. One way of automating tasks is to replace user decision steps with well-chosen defaults or automated decision making. Whenever software developers build systems in which end users are asked to make security configuration decisions or answer questions posed by pop-up security alerts, they should consider whether the anticipated users will have expertise or information that will allow them to make a better decision than the software developer can implement through an automated process or a default setting [29]. Edwards et al. describe the limits of security automation in detail and propose a set of guidelines for appropriate security automation [11].

In the *failure identification* step, the designer identifies potential failure modes for the remaining security-critical human tasks. The human-in-the-loop security framework offers a systematic approach to identifying these failure modes. User studies can provide empirical evidence as to which failures occur in practice and additional insights into the reasons for these failures. If empirical data is not available, the framework can suggest areas where user studies are needed.

In the *failure mitigation* step, the designer tries to find ways to prevent failures by determining how humans might be better supported in performing these tasks [17]. For example, context-sensitive help and decision-support tools might assist humans as they perform security tasks. Automated error checking tools might warn users if they appear to be making mistakes. Alerts might remind them that a task remains to be done. Visualizations might make it easier for them to spot anomalies or understand the overall system state. Training materials might help them better understand the tasks they must perform. Warnings might be better designed to communicate more effectively. In many cases it may be possible to reduce the burden on humans, even if it can't be completely eliminated. User studies can help designers evaluate the effectiveness of their failure mitigation efforts.

After completing a first pass through the four-step process, designers may revisit some or all of the steps to try to further reduce the risk of human security failures. For example, if after completing the mitigation step designers are unable to reduce human failure rates to an acceptable level, they might return to the automation step and explore whether it is feasible to develop an automated approach that would perform more reliably than humans. An automated approach known to be imperfect might have been considered and dismissed during the first pass through the process, but might be reconsidered during a second pass once it is discovered that human performance on a given task is even less reliable than the automated tool.

In Appendix B we present two case studies to illustrate how to apply the human-in-the-loop security framework in the human threat identification and mitigation process.

4. Related work

Security researchers often evaluate the effectiveness of systems as measured by their ability to resist various forms of attack. They typically envision a threat model in which a determined attacker will attempt to "fool the software" by disabling the security software, performing a malicious action in an undetectable manner, or deceiving the security software into interpreting a malicious action as an innocuous one. More recently security researchers have expanded their threat models to include semantic attacks [32], in which attackers attempt to "fool the humans" by obscuring or visually spoofing the indicators provided by their security software. However, this work tends to focus on how to prevent sophisticated semantic attacks by developing unspoofable indicators [2], [41]. But humans are often fooled by much simpler attacks and may even ignore the (usually correct) warnings provided by their security software [39]. Cranor proposed a series of questions that can be used to analyze the human factors criteria associated with security indicators and identify areas of potential failure. These questions can be mapped to models of warning processing from the warning science literature, for example the C-HIP model on which the human-in-the-loop framework is based [37].

Research on how to create the most effective warnings goes back at least 100 years—for example, research was performed at Yale University between 1904 and 1906 to determine the most effective colors for railroad signals, and the results were subsequently field tested before red, yellow, and green were adopted as standards in 1908. However, most warnings have been introduced

to address industry needs, with little or no scientific evaluation of their effectiveness [8], [13]. In the past two decades, interest in warnings research has increased, and peer-reviewed studies are now being published in this area.

Wogalter proposed the C-HIP model to identify reasons that a particular warning is ineffective [37]. Similar to the human-in-the-loop framework, the C-HIP model begins with a source delivering a warning to a receiver, who receives it along with other environmental stimuli that may distract from the message. The receiver goes through several information processing steps, which ultimately determine whether the warning results in any change in behavior. This model is directly applicable to computer security warnings. However, we have added some components to address failures that are typical in a computer security context. We have added a capabilities component because we have observed that human security failures are sometimes attributed to humans being asked to complete tasks that they are not capable of completing. We have also added an interference component because computer security communications may be impeded by an active attacker or technology failures. We have modified C-HIP to apply more generally to the five types of computer security communications outlined in Section 2.1. The knowledge acquisition, knowledge retention, and knowledge transfer steps are especially applicable to training and policy communications. Finally, we have explicitly called out two types of personal variables and restructured the human receiver representation to emphasize related concepts over temporal flow.

In his seminal book, *The Design of Everyday Things,* Don Norman proposed a seven-stage "action cycle" to describe how humans proceed from formulating a goal, through executing an action, to evaluating whether or not the action achieved the intended goal. He described how the action cycle can be used as a check-list for design so as to avoid the gulfs of execution and evaluation [26]. Norman's focus was on the design of physical objects such as radios and film projectors, but his seven stages of action are applicable to software user interfaces as well. The human-in-the-loop framework incorporates concepts from Norman's action cycle, as well as James Reasons' Generic-Error Modeling System (GEMS) [28].

Brostoff and Sasse have also applied concepts from GEMS to systematically identify and address human security failures. They focus on the distinction between the active failures described by GEMS and latent failures that predispose a system to failures. They propose a model that represents errors in five areas: decision-makers, line managers, preconditions, productive activities, and defenses [3]. Brostoff and Sasse's approach is more organization-centric, while the human-in-the-loop framework is more user-centric.

5. Conclusions and future work

There are three high-level strategies for building secure systems that humans can use. The first strategy is to find ways to get humans out of the loop and build systems that "just work" without involving humans in security-critical functions. Secure systems designers should consider what functions performed by humans might be automated, and what configuration choices might be replaced by default settings that are generally appropriate. The second strategy is to build systems that are intuitive and find ways of making them easy to use. Secure systems designers should engineer human tasks to maximize the chances that humans will perform them successfully. The third strategy is to teach humans how to perform security-critical tasks. Here we must find effective ways of teaching complicated concepts to humans who may not be all that receptive to learning them. In most cases we are unable to rely on just one of these strategies and must adopt a multi-pronged approach to secure system usability.

We have proposed a framework for reasoning about the human in the loop in secure systems that is intended to help secure systems designers and operators reduce the occurrence of human security failures. Whenever it is not possible to get humans completely out of the loop in a secure system, it is important that the human's role in performing security-critical functions be considered in any security analysis, and that potential failure modes be identified. The human-in-the loop security framework provides a systematic approach to identifying these failure modes and identifying their root causes. It can be applied as part of a human threat identification and mitigation process during system design to identify potential problem areas so that they can be addressed prior to implementation. The framework can also be used to analyze deployed systems to gain insights into why they are failing and to determine where corrective actions should be taken.

Future work is needed to validate the usefulness of the human-in-the-loop framework to security engineers, and to provide more concrete guidance on how to operationalize the human threat identification and mitigation process. So far we have focused on the identification of failure modes. Future work should develop more specific guidelines and design patterns for mitigating human threats by automating security-critical human tasks and better supporting humans as they perform these tasks.

Acknowledgements

Thanks to Serge Egelman, Jason Hong, Ponnurangam Kumaraguru, Rob Reeder and the other members of the CMU Usable Privacy and Security Laboratory for providing feedback on this framework.

References

[1] A. Adams and M. A. Sasse, 1999 Users are not the enemy: why users compromise security mechanisms and how to take remedial measures. *Communicaitons of the ACM* 42(12), 40-46.

[2] A. Adelsbach, S. Gajek, and J. Schwenk. Visual spoofing of SSL protected web sites and effective countermeasures. In *Proceedings of the 1st Information Security Practice and Experience Conference* (Singapore, 11-14 April, 2005). ISPEC2005. LNCS, Springer-Verlag, Heidelberg.

[3] S. Brostaff and M. A. Sasse. Safe and sound: a safety-critical approach to security. In *Proceedings of the 2001 Workshop on New Security Paradigms* (Cloudcroft, New Mexico, September 10 - 13, 2001). NSPW '01. ACM, New York, NY, 41-50.

[4] K. Cameron and D. DeJoy, 2006. The Persuasive Functions of Warnings: Theory and Models. In M. S. Wogalter, ed., *Handbook of Warnings.* Lawrence Erlbaum Associates, Mahwah, NJ, 301-312.

[5] R. M. Conlan and P. Tarasewich. 2006. Improving interface designs to help users choose better passwords. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems* (Montréal, Québec, Canada, April 22 - 27, 2006). CHI '06. ACM, New York, NY, 652-657.

[6] M. Crawford. Whoops, human error does it again. *CSO Online.* April 21, 2006. http://www.csoonline.com.au/index.php/id;255830211;fp;32768;fpid;20026681

[7] D. Davis, F. Monrose, and M. K. Reiter, On User Choice in Graphical Password Schemes, *Proceedings of the 13th USENIX Security Symposium* (Aug. 2004), 151-164.

[8] D. DeJoy, K. Cameron, and L. Della, 2006. Postexposure Evaluation of Warning Effectiveness: A Review of Field Studies and Population-based research. In M. S. Wogalter, ed., *Handbook of Warnings.* Lawrence Erlbaum Associates, Mahwah, NJ, 35-48.

[9] R. Dhamija, J. D. Tygar, and M. Hearst, 2006. Why phishing works. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada, April 22 - 27, 2006). R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries, and G. Olson, Eds. CHI '06. ACM Press, New York, NY, 581-590.

[10] J. S. Downs, M. B. Holbrook, and L. F. Cranor, 2006. Decision strategies and susceptibility to phishing. In *Proceedings of the Second Symposium on Usable Privacy and Security* (Pittsburgh, Pennsylvania, July 12 - 14, 2006). SOUPS '06, vol. 149. ACM Press, New York, NY, 79-90.

[11] W. K. Edwards, E. S. Poole, and J. Stoll, 2007. Security Automation Considered Harmful? *Proceedings of the IEEE New Security Paradigms Workshop (NSPW 2007).* White Mountain, New Hampshire. September 18-21, 2007.

[12] S. Egelman, L. Cranor, and J. Hong. You've Been Warned: An Empirical Study of the Effectiveness of Web Browser Phishing Warnings. ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '08). 2008.

[13] D. Egilman and S. Bohme, 2006. A Brief History of Warnings. In Wogalter, M., ed., *Handbook of Warnings.* Lawrence Erlbaum Associates, Mahwah, NJ,35-48.

[14] I. Flechais, J. Riegelsberger, and M. A. Sasse, 2005. Divide and conquer: the role of trust and assurance in the design of secure socio-technical systems. In *Proceedings of the 2005 Workshop on New Security Paradigms* (Lake Arrowhead, California, September 20 - 23, 2005). NSPW '05. ACM, New York, NY, 33-41.

[15] B. Friedman, P. Lin, and J. K. Miller, 2005. Informed consent by design. In L. Cranor and S. Garfinkel, eds. *Security and Usability.* O'Reilly, 477-504.

[16] S. Gaw and E. W. Felten, 2006. Password management strategies for online accounts. In *Proceedings of the Second Symposium on Usable Privacy and Security* (Pittsburgh, Pennsylvania, July 12 - 14, 2006). SOUPS '06, vol. 149. ACM, New York, NY, 44-55.

[17] J. B. Gross and M. B. Rosson, 2007. Looking for trouble: understanding end-user security management. In *Proceedings of the 2007 Symposium on Computer Human interaction For the Management of information Technology* (Cambridge, Massachusetts, March 30 - 31, 2007). CHIMIT '07. ACM, New York, NY, 10.

[18] H. E. Hancock, C. T. Bowles, W. A. Rogers, and A. D. Fisk. 2006. Comprehension and Retention of Warning Information. In M. S. Wogalter, ed., *Handbook of Warnings.* Lawrence Erlbaum Associates, Mahwah, NJ, 267-277.

[19] R. L. Heath and J. Bryant. *Human Communication Theory and Research: Concepts, Contexts, and Challenges.* 2nd edition. Lawrence Erlbaum, 2000.

[20] C. Kaufman, R. Perlman, and M. Speciner. *Network Security: PRIVATE Communication in a PUBLIC World.* 2nd edition. Prentice Hall, page 237, 2002.

[21] M. J. Kalsher and K. J. Williams. Behavioral Compliance: Theory, Methodology, and Results. 2006. In M. S. Wogalter, ed., *Handbook of Warnings*. Lawrence Erlbaum Associates, Mahwah, NJ, 313-331.

[22] P. Kumaraguru, Y. Rhee, S. Sheng, S. Hasan, A. Acquisti, L. Cranor and J. Hong. Getting Users to Pay Attention to Anti-Phishing Education: Evaluation of Retention and Transfer. *Proceedings of the 2nd Annual eCrime Researchers Summit,* October 4-5, 2007, Pittsburgh, PA, p. 70-81.

[23] C. Kuo, S. Romanosky, and L. F. Cranor, 2006. Human selection of mnemonic phrase-based passwords. In *Proceedings of the Second Symposium on Usable Privacy and Security* (Pittsburgh, Pennsylvania, July 12 - 14, 2006). SOUPS '06, vol. 149. ACM, New York, NY, 67-78.

[24] R. A. Maxion and R. W. Reeder. Improving User-Interface Dependability through Mitigation of Human Error. *International Journal of Human-Computer Studies*. 63 (1-2):25-50. 2005.

[25] R. McMillan. Security group ranks human error as top security worry. *Network World.* November 15, 2007. http://www.networkworld.com/news/2006/111506-security-group-ranks-human-error.html

[26] D. A. Norman. *The Design of Everyday Things.* Basic Books, 1988.

[27] U. Piazzalunga, P. Salvaneschi, and P. Coffetti. The Usability of Security Devices. In *Security and Usability: Designing Secure Systems that People Can use,* L. Cranor and S. Garfinkel, eds., O'Reilly, pages 221-242, 2005.

[28] J. Reason. *Human Error.* Cambridge University Press, 1990.

[29] B. Ross. Firefox and the Worry-Free Web. In *Security and Usability: Designing Secure Systems that People Can use,* L. Cranor and S. Garfinkel, eds., O'Reilly, pages 577-587, 2005.

[30] B. Schneier. *Beyond Fear: Thinking Sensibly About Security in an Uncertain World.* Springer-Verlag, 2003.

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

[32] B. Schneier, 2000. Semantic Attacks: The Third Wave of Network Attacks. *Cryptogram Newsletter* (October 15). http://www.schneier.com/crypto-gram-0010.html#1.

[33] S. Sheng, B. Magnien, P. Kumaraguru, A. Acquisti, L. 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 2007 Symposium On Usable Privacy and Security,* Pittsburgh, PA, July 18-20, 2007.

[34] J. Thorpe and P.C. van Oorschot. Human-Seeded Attacks and Exploiting Hot-Spots in Graphical Passwords. In *Proceedings of the 16th USENIX Security Symposium*, August 6-10, 2007, Boston, MA, USA.

[35] T. Whalen and K. M. Inkpen, 2005. Gathering evidence: use of visual security cues in web browsers. In *Proceedings of the 2005 Conference on Graphics interface* (Victoria, British Columbia, May 09 - 11, 2005). ACM International Conference Proceeding Series, vol. 112. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, 137-144.

[36] M. S. Wogalter, 2006. Purposes and Scope of Warnings. In M. S. Wogalter, ed., *Handbook of Warnings*. Lawrence Erlbaum Associates, Mahwah, NJ, 3-9.

[37] M. S. Wogalter, 2006. Communication-Human Information Processing (C-HIP) Model. In M. S.Wogalter, ed., *Handbook of Warnings*. Lawrence Erlbaum Associates, Mahwah, NJ, 51-61.

[38] M. S. Wogalter and W. J. Vigilate, Jr., 2006. Attention Switch and Maintenance. In M. S.Wogalter, ed., *Handbook of Warnings*. Lawrence Erlbaum Associates, Mahwah, NJ, 245-265.

[39] M. Wu, R. C. Miller, and S. L. Garfinkel, 2006. Do security toolbars actually prevent phishing attacks?. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada, April 22 - 27, 2006). R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries, and G. Olson, Eds. CHI '06. ACM Press, New York, NY, 601-610.

[40] M. Wu, R. C. Miller, and G. Little, 2006. Web wallet: preventing phishing attacks by revealing user intentions. In *Proceedings of the Second Symposium on Usable Privacy and Security* (Pittsburgh, Pennsylvania, July 12 - 14, 2006). SOUPS '06, vol. 149. ACM, New York, NY, 102-113.

[41] Z. Ye, S. Smith, and D. Anthony, 2005. Trusted paths for browsers. *ACM Trans. Inf. Syst. Secur.* 8, 2 (May 2005), 153-186.

Appendix A: Components of the human-in-the-loop security framework

Component		Questions to ask	Factors to consider
Communication		What type of communication is it (warning, notice, status indicator, policy, training)? Is communication active or passive? Is this the best type of communication for this situation?	Severity of hazard, frequency with which hazard is encountered, extent to which appropriate user action is necessary to avoid hazard
Communication impediments	Environmental Stimuli	What other environmental stimuli are likely to be present?	Other related and unrelated communications, user's primary task, ambient light, noise
	Interference	Will anything interfere with the communication being delivered as intended?	Malicious attackers, technology failures, environmental stimuli that obscure the communication
Personal Variables	Demographics and personal characteristics	Who are the users? What do their personal characteristics suggest about how they are likely to behave?	Age, gender, culture, education, occupation, disabilities
	Knowledge and experience	What relevant knowledge or experience do the users or recipients have?	Education, occupation, prior experience
Intentions	Attitudes and beliefs	Do users believe the communication is accurate? Do they believe they should pay attention to it? Do they have a positive attitude about it?	Reliability, conflicting goals, distraction from primary task, risk perception, self-efficacy, response efficacy
	Motivation	Are users motivated to take the appropriate action? Are thy motivated to do it carefully or properly?	Conflicting goals, distraction from primary task, convenience, risk perception, consequences, incentives/disincentives
Capabilities		Are users capable of taking the appropriate action?	Knowledge, cognitive or physical skills, memorability, required software or devices
Communication delivery	Attention switch	Do users notice the communication? Are they aware of rules, procedures, or training messages?	Environmental stimuli, interference, format, font size, length, delivery channel, habituation
	Attention maintenance	Do users pay attention to the communication long enough to process it? Do they read, watch, or listen to it fully?	Environmental stimuli, format, font size, length, delivery channel, habituation
Communication processing	Comprehension	Do users understand what the communication means?	Symbols, vocabulary and sentence structure, conceptual complexity, personal variables
	Knowledge acquisition	Have users learned how to apply it in practice? Do they know what they are supposed to do?	Exposure or training time, involvement during training, personal characteristics
Application	Knowledge retention	Do users remember the communication when a situation arises in which they need to apply it? Do they recognize and recall the meaning of symbols or instructions?	Frequency, familiarity, long term memory, involvement during training, personal characteristics
	Knowledge transfer	Can users recognize situations where the communication is applicable and figure out how to apply it?	Involvement during training, similarity of training, personal characteristics
Behavior		Does behavior result in successful completion of desired action?	*See Norman's Stages of Action, GEMS*
		Does behavior follow predictable patterns that an attacker might exploit?	Type of behavior, ability of people to act randomly in this context, usefulness of prediction to attacker

Appendix B: Case Studies

B.1 Anti-phishing tools

Anti-phishing tools that provide passive warning indicators in web browser toolbars have been found to be largely ineffective at preventing users from accessing phishing web sites [39]. Recent versions of Internet Explorer (IE) and Firefox browsers include more active anti-phishing warnings that block access to suspicious web sites unless a user explicitly chooses to override the warning. When Firefox suspects a site is fraudulent it greys out the page and displays a pop-up warning that does not look similar to other browser warnings. When IE suspects a site is fraudulent it can either load the page and display a passive warning that will be dismissed if the user types anything into the browser, or it can display an active warning instead of loading the page. Users can choose to ignore any of these warnings and proceed to the suspicious site.

We will apply the human threat identification and mitigation process and the human-in-the-loop security framework to help us determine whether the new anti-phishing warnings effectively protect users from phishing attacks and how they can be improved. Empirical evidence from user studies provides insights into the failure modes.

Task identification. The IE and Firefox anti-phishing tools rely on human users to decide whether to heed the warning and leave a suspicious site or ignore the warning and proceed to visit that site.

Task automation. This human decision could be eliminated by blocking access to suspicious sites without offering an override option. Arguably, if the false positive rate associated with the automated phishing detection tool is sufficiently low and the risk associated with visiting a phishing site is significant, it would be better to completely block users from visiting suspicious sites rather than presenting them with warnings and letting them decide what to do. However, for the time being browser vendors believe they must offer users the override option.

Failure identification. We apply the human-in-the-loop security framework as follows:

- *Communication:* Warning
- *Communication impediments:* Environmental stimuli can include other browser warnings, the user's email client and/or other applications related to the user's primary task, and other ambient conditions that may distract the user. We are not aware of attacks that interfere with the display of IE and Firefox anti-phishing warnings; however, the IE passive warning usually loads a few seconds after the page loads, and if users start filling out a web form they can inadvertently dismiss the passive warning before they notice it is there [12].
- *Personal variables:* Web browser anti-phishing tools are used by people with a wide range of knowledge, abilities, and other personal characteristics, many of whom have little or no knowledge about phishing.
- *Intentions:* A user study found that most users who read the warnings believed they should heed them and were motivated to heed them. However, a few users did not believe the warnings were important, generally because they confused them with other warnings, they did not trust them, or they did not believe the risk was severe. One user commented "since it gave me the option of still proceeding to the website, I figured it couldn't be that bad" [12].
- *Capabilities:* A user study found that users are capable of taking the appropriate action, closing the browser window or navigating to a different web site. However, those with inaccurate mental models may be incapable of making an informed decision about whether to take this action [12].
- *Communication delivery:* A user study found that users noticed the Firefox warning and active IE warning. Many users did not notice the passive IE warning. Some users in the study did not fully read the warnings. Some users erroneously believed that the IE warning was one of the frequently-encountered browser warnings such as a 404 page not found warning [12]. In another user study of three simulated passive anti-phishing warnings that displayed symbols in a browser toolbar, 25% of participants claimed they had not noticed the warnings, even after being explicitly instructed to look for them [39].
- *Communication processing:* Most of the users in the Egelman et al study understood the warnings and knew what they were supposed to do. Firefox users were more likely to correctly understand the warnings than IE users. Some users who were not previously aware of phishing attacks appeared to have erroneous mental models of the situation and thus misinterpreted the warnings. These users assumed that the email containing the link to the suspicious web site was legitimate and that the warning indicated that there was a transient problem with the web site. They clicked on the link in the email repeatedly in an attempt to reach the web site without triggering the warning [12].
- *Application:* Not applicable.
- *Behavior:* All users in the study who understood the warnings and decided to heed them were able

to do so successfully. Although users who clicked on the link in the email repeatedly were actually making a mistake, it ultimately resulted in the users not accessing the suspicious web sites, which was the desired outcome. Thus it appears that the Firefox and active IE warnings tend to fail safely [12]. Predictability of behavior is not relevant to this situation.

Failure mitigation. This analysis suggests that the Firefox and active IE warning are much more effective than the passive IE warning, and that the passive IE warning should be replaced with an active warning. The active IE warning could be made more effective by making it look less similar to non-critical warnings. A number of studies suggest that many users have formed inaccurate mental models related to phishing, and that these faulty models lead to mistakes [9], [10], [12], [39]. The warnings include links to educational materials about phishing, but we don't have empirical evidence about how frequently users consult this material and whether they understand it (the human-in-the-loop framework could be applied to analyze the effectiveness of these educational materials). Additional anti-phishing training delivered through other channels might be helpful for educating users with inaccurate mental models [22], [33]. It might also be useful to provide users with information they can use to decide whether to heed or ignore the warning, as the IE and Firefox warnings did not explain to users why they were being presented with this choice. The warning might provide additional explanations about why the site is suspicious and offer users the option of visiting the real site that the suspicious site appears to be spoofing [40].

The human threat identification and mitigation process suggested a possible approach to getting the human out of the loop altogether in this case, and a number of ways anti-phishing warnings could be improved. The failure identification step highlighted the need to find ways to correct users' inaccurate mental models about phishing and suggested that it would be useful to study the use and effectiveness of links to educational materials in anti-phishing warnings.

B.2 Password policies

Many organizations have policies that specify how users should select their passwords. Typical password policies specify minimum password lengths, mandate a combination of different types of characters, and require that users select different passwords for each system they use and remember these passwords without writing them down. Some policies also require users to change their password frequently. Password policies generally prohibit users from sharing their passwords with other people. In practice, people tend not to comply fully with password policies [1]. For example, Gaw and Felten found evidence of widespread password reuse [16].

We will apply the human threat identification and mitigation process and the human-in-the-loop security framework to help us determine how an organization might improve compliance with their password policy.

Task identification. Users must select passwords that comply with the policy, remember them without writing them down, recall them when needed, and refrain from sharing them.

Task automation. To simplify the password creation task, the system might generate random passwords and assign them to users. However, these passwords are likely to be too difficult for users to remember. Alternatively, tools might provide feedback to users on the quality of their passwords and suggest improvements [5]. Single sign-on systems or secure password vaults might be deployed to reduce the number of passwords that must be remembered. Alternative authentication mechanisms might also be considered [16].

Failure identification. We apply the human-in-the-loop security framework as follows:

- *Communication:* Policy; sometimes accompanied by training, warnings, or notices at password creation time or as periodic reminders to comply with the policy
- *Communication impediments:* Environmental stimuli combined with the desire of users to create their password quickly so they can access a system may interfere with policy communication.
- *Personal variables:* Most organizations have users with a wide range of demographics, knowledge, and experience (complete novice through security expert); however, this will depend on the organization.
- *Intentions:* Users tend to find password policies inconvenient and may not appreciate the importance of protecting their password, especially if they do not believe it protects a resource that an attacker would value. Thus, they may have little motivation to comply with a password policy. The policy may also conflict with users' goals when users view password sharing as necessary to facilitate tasks that require multiple people to edit a document.
- *Capabilities:* User studies suggest that people are capable of following typical password security guidance to create compliant passwords [Kuo et

al]. However, people are not good at remembering random-looking passwords. Most people are incapable of memorizing the numerous passwords they must typically use [16]. This problem is exacerbated when people are required to change their passwords frequently [1].

- *Communication delivery:* Most computer users appear to be aware of the typical password security guidance, indicating they have read it at least once, and likely multiple times.
- *Communication processing:* Recent user studies suggest that most people now understand typical password security guidance and know what they are supposed to do to apply it [23].
- *Application:* Users appear to be generally familiar with password security policies and know how to apply them. It is likely that they are fully aware of when they are violating these policies, although this should be verified with a user study.
- *Behavior:* Users who intend to follow policies on password creation are generally able to do so without error. However, the passwords users create may still be somewhat predictable, despite adherence to policy. For example, Kuo et al found that users advised to select passwords based on mnemonic phrases often selected well-known phrases on which to base their passwords [23].

Failure mitigation. The most critical failure appears to be a capabilities failure: people are not capable of remembering large numbers of policy-compliant passwords. This can be addressed by reducing the number of passwords people must remember through single sign-on and alternative authentication mechanisms. Organizations should also consider whether the security benefits associated with frequent, mandatory password changes make up for the tendency of users to violate other parts of the password policy because they cannot remember frequently-changed passwords. Motivation failures may become less of an issue if the capability failure can be addressed in a way that is consistent with employee workflows. Training communications that explain the rationale behind password policies may also mitigate motivation failures. Finally, the problem of users creating passwords that are not random enough can be addressed through better guidance at password creation time and software tools that support users in the password creation process.