Your Attention Please

Designing security-decision UIs to make genuine risks harder to ignore

Cristian Bravo-Lillo
cbravo@cmu.edu

Lorrie Faith Cranor
lorrie@cmu.edu

Julie Downs
downs@cmu.edu

Saranga Komanduri
sarangak@cmu.edu

Robert W. Reeder
reeder@cs.cmu.edu

Stuart Schechter
stus@microsoft.com

Manya Sleeper
msleeper@cmu.edu

ABSTRACT

We designed and tested *attractors* for computer security dialogs: user-interface modifications used to draw users' attention to the most important information for making decisions. Some of these modifications were purely visual, while others temporarily inhibited potentially-dangerous behaviors to redirect users' attention to salient information. We conducted three between-subjects experiments to test the effectiveness of the attractors.

In the first two experiments, we sent participants to perform a task on what appeared to be a third-party site that required installation of a browser plugin. We presented them with what appeared to be an installation dialog from their operating system. In both experiments, participants who saw dialogs that employed attractors were significantly less likely than those in the control group to ignore clues that installing this software might be harmful.

In the third experiment, we attempted to habituate participants to dialogs that they knew were part of the experiment. We used attractors to highlight a field that was of no value during habituation trials and contained critical information after the habituation period. Participants exposed to inhibitive attractors were two to three times more likely to make an informed decision than those in the control condition.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Evaluation/methodology, Interaction styles

Keywords

usable security, warnings, dialogs, habituation, attractors

General Terms

Human Factors; Design; Methodologies

Symposium on Usable Privacy and Security (SOUPS) 2013, July 24–26, 2013, Newcastle, UK.

1. INTRODUCTION

Like the boy who cried wolf from Aesop's Fables, today's computer systems perpetually cry for attention in the name of safety, and hundreds of cries may be heard without a real threat. *Did you want to open a file in a legacy file format? Is it OK that this certificate is out of date? Do you want to view content that was sent insecurely?* The inevitable result is that, like Aesop's villagers, users stop paying attention. When a security dialog contains information that could alert users to a real risk, they are less likely to notice it.

Reducing the onslaught of interruptive security warning dialogs might help reduce the strain on users' attention. Some warnings can be removed by re-architecting systems to reduce the potential for harm, such as by building file parsers in type-safe languages or sandboxing unsafe code. Yet inevitably, some decisions must eventually be made by the user. One type of unavoidable decision is the choice to take a risk that some users may embrace and others may reject. For example, some users may want to share their location with an application that others would not share their location with. In other cases, users have knowledge, which the system does not have, that is essential to making a correct choice. For example, the user may know that a particular wireless network is trustworthy.

Designing user interfaces to facilitate necessary security decisions is especially difficult given that the damage caused by unnecessary decisions has already been done. After years of training to ignore cries of wolf, users are unlikely to become more attentive to them overnight.

From this unfortunate starting point, we set out to test user interface elements designed for attracting users' attention to critical information in a security-decision dialog. We call these user interface elements *attractors*.

We conducted three between-subjects online experiments with Amazon Mechanical Turk workers. We designed the first two experiments to test attractors in realistic security scenarios. We asked participants to evaluate games on three third-party websites. Unbeknownst to them, we operated the third such website, which appeared to require Microsoft's Silverlight browser plugin. In some cases, the dialog contained a clue that should have made users suspicious about installing. In Experiment 1, we sometimes changed the contents of the publisher field from `Microsoft` to `MiicrOsOft`. In Experiment 2, we displayed the set of permissions required for the plugin, and some participants were shown egregious permissions being requested. In both experiments our attractors had a significant benefit over the control, reducing the proportion of participants choosing the less safe option by up to 50%.

Experiment 3 was intended to test the performance of attractors under conditions of extreme habituation. We first exposed participants to numerous repetitions of the same dialog. During this *habituation period*, the field in the dialog that attractors would direct users' attention to contained information irrelevant to the users' decision. After habituation, this field contained information essential to making a correct choice. Participants who saw our attractors were two to three times more likely to notice the updated message the first time it appeared than those in the control conditions.

2. RELATED WORK

Warnings are designed to capture attention and convey information about a hazard. As Laughery and Wogalter describe, a warning can include a variety of components that impact its effectiveness at achieving these goals, including its visual design (e.g., size, colors, graphics), use of "signal words," length, and interactivity [10].

Warning guidelines have been developed since the early 1900s, leading to standardized warning formats, including ANSI labeling standards and FDA warning regulations [10]. Laughery et al. found that warnings in a variety of categories that matched ANSI standard guidelines outperformed those that did not, across a set of effectiveness metrics [9].

When non-compliant behavior does not cause harm over time, people may develop an automated response, *habituation*, that does not take into account changes in warning context or messaging [6]. Habituation decreases warning effectiveness when people become less alert to the information presented in warnings. Kim and Wogalter found that standardization of warnings can be a factor in habituation, i.e., habituation occurred when participants were repeatedly shown the same warning design [7]. Exposure to a new design resulted in an increase in alertness; however, a return to a habituated design resulted in "recovery" of habituation.

Computer users often ignore computer security warnings. Schechter et al. asked participants to perform a banking task and provided increasingly alarming clues, such as SSL warnings, that it was unsafe to do so. Two thirds of role-playing participants and a third of participants using their own passwords ignored all the clues [14]. Sunshine et al. found that users tended to ignore SSL warnings and found evidence that habituation was often to blame [17]. Sharek et al. presented participants with real warnings and fake warnings, distinguished by visual design elements, as they performed tasks on health websites. They found that the majority of participants clicked 'OK' on the warning regardless of whether the warning was real or fake [15]. Bravo-Lillo et al. redesigned several security warning dialogs based on guidelines and mental model interviews and found that the redesigns improved user motivation and response to warnings, but did not allow users to better differentiate between high and low risk scenarios [2].

People also tend to develop "scripts" for their interactions with warnings as they become more familiar with them, which allows them to pay less attention to the warning itself [18]. Changing the appearance of warnings [3] or showing them less frequently has been shown to reduce habituation and making a warning more obvious, for example larger or brighter, can help attract attention to the warning [19]. This work evaluates warning elements for drawing user attention to critical information in computer security warnings. We focus on how well different design elements can prompt users to deviate from their "script."

One of the major challenges in studying and developing warnings is maintaining ecological validity. Study design can easily impact results. Schechter et al. showed that participants playing a role are less likely to respond to clues that indicate the presence of risk than those who believe they are actually at risk [14]. The perceived safety of a laboratory setting may cause participants to disregard the potential for harm. Sotirakopoulos et al. replicated the study by Sunshine et al. [17] and found that many participants ignored an SSL warning but attributed their behavior to the laboratory setting [16].

3. ATTRACTORS

An attractor is an interface modification designed to draw attention to a region of the screen. We investigated attractors designed to draw attention to an information field in a decision dialog that is essential to making a good decision. We call this the *salient field*. The attractors we designed are illustrated in Figure 1, in which they appear in the context of Experiment 1 (software installation with benign and suspicious publishers).

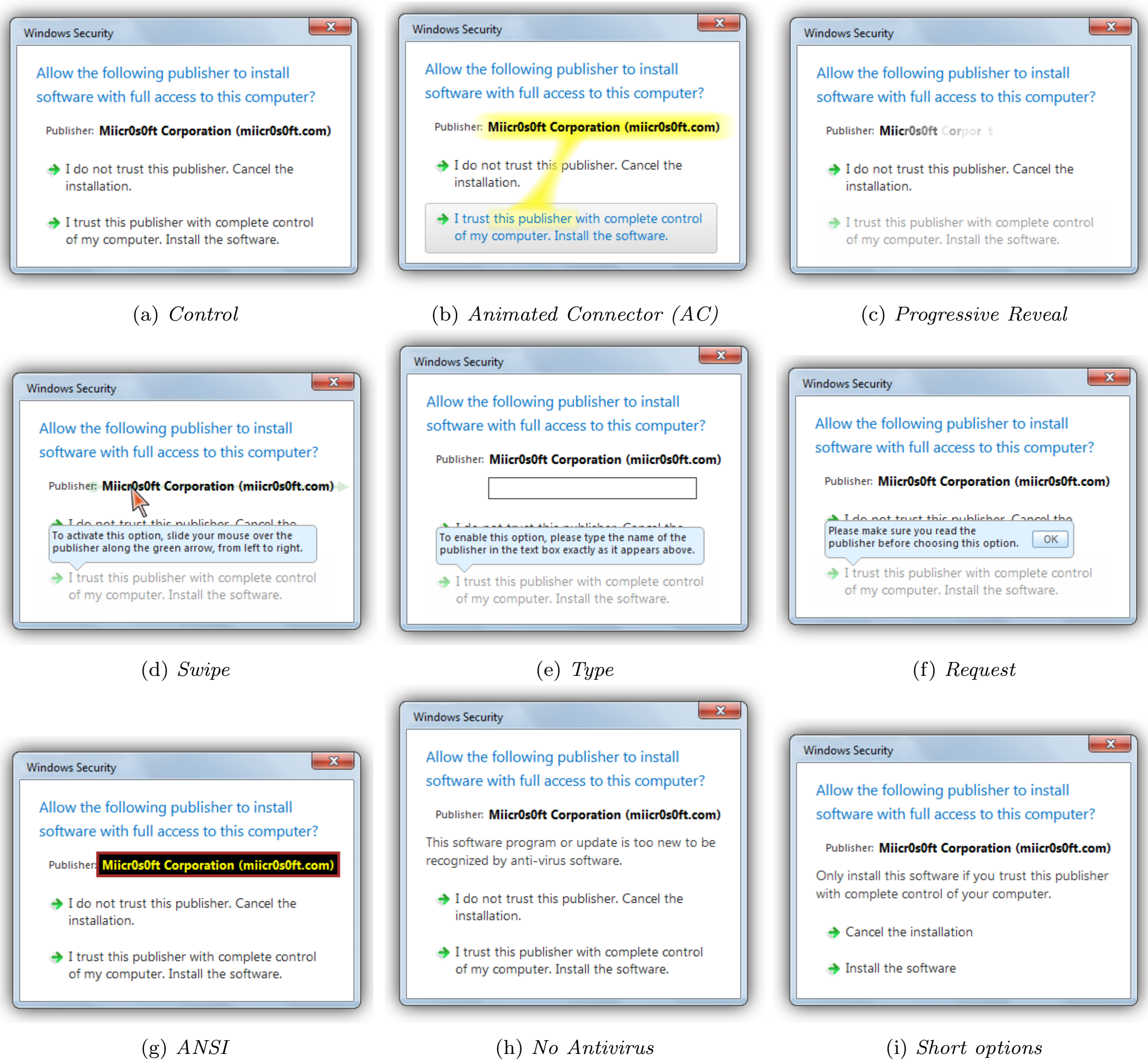
3.1 Inhibitive Attractors

We designed five *inhibitive attractors*, which prevent a user from making a potentially-hazardous choice until either some period of time has expired or the user performs a required action. These inhibitive attractors appear only when users move their mouse over the button representing the potentially-hazardous choice, which we henceforth refer to as the *triggering option*. The user is never inhibited from closing the dialog or selecting a non-triggering option.

The ***Animated Connector (AC)*** is a yellow highlight that first appears behind keywords in the triggering option that relate to the salient field. In the installation dialog of Experiment 1, the highlighted keywords in the triggering option text were "this publisher" (Figure 1a), whereas in Experiment 2 the highlighted words were "upgraded permissions" (Figure 2). Over a period of two seconds, the highlighted region progresses in the direction of the salient field, and then fills the background of the field—hopefully bringing the user's attention with it. Figure 1b shows the contents of the decision dialog in Experiment 1 after the animation completed. This attractor is inhibitive when used with a delay that prevents users from proceeding until the animation completes, or when used in combination with another inhibitive attractor.

The ***Progressive Reveal*** attractor (Figure 1c) first hides the contents of the salient field, then progressively animates it back into place over a period of four seconds. The animation is a progression in which characters are revealed mostly, but not entirely, from left to right. The motion and randomization are intended to help users notice each letter as it appears. Figure 1c shows the progressive reveal in mid-flight, and Appendix A provides details on the animation algorithm. The triggering option is disabled until the contents of the salient field have fully appeared and the animation completes.

The ***Swipe*** attractor requires users to move their mouse



Windows Security

Allow the following publisher to install software with full access to this computer?

Publisher: Miicr0s0ft Corporation (miicr0s0ft.com)

I do not trust this publisher. Cancel the installation.

I trust this publisher with complete control of my computer. Install the software.

(a) *Control*

Windows Security

Allow the following publisher to install software with full access to this computer?

Publisher: Miicr0s0ft Corporation (miicr0s0ft.com)

I do not trust this publisher. Cancel the installation.

I trust this publisher with complete control of my computer. Install the software.

(b) *Animated Connector (AC)*

Windows Security

Allow the following publisher to install software with full access to this computer?

Publisher: Miicr0s0ft Corpor t

I do not trust this publisher. Cancel the installation.

I trust this publisher with complete control of my computer. Install the software.

(c) *Progressive Reveal*

Windows Security

Allow the following publisher to install software with full access to this computer?

Publisher: Miicr0s0ft Corporation (miicr0s0ft.com)

I do not trust this publisher. Cancel the

To activate this option, slide your mouse over the publisher along the green arrow, from left to right.

I trust this publisher with complete control of my computer. Install the software.

(d) *Swipe*

Windows Security

Allow the following publisher to install software with full access to this computer?

Publisher: Miicr0s0ft Corporation (miicr0s0ft.com)

I do not trust this publisher. Cancel the

To enable this option, please type the name of the publisher in the text box exactly as it appears above.

I trust this publisher with complete control of my computer. Install the software.

(e) *Type*

Windows Security

Allow the following publisher to install software with full access to this computer?

Publisher: Miicr0s0ft Corporation (miicr0s0ft.com)

I do not trust this publisher. Cancel the

Please make sure you read the publisher before choosing this option. OK

I trust this publisher with complete control of my computer. Install the software.

(f) *Request*

Windows Security

Allow the following publisher to install software with full access to this computer?

Publisher: Miicr0s0ft Corporation (miicr0s0ft.com)

I do not trust this publisher. Cancel the installation.

I trust this publisher with complete control of my computer. Install the software.

(g) *ANSI*

Windows Security

Allow the following publisher to install software with full access to this computer?

Publisher: Miicr0s0ft Corporation (miicr0s0ft.com)

This software program or update is too new to be recognized by anti-virus software.

I do not trust this publisher. Cancel the installation.

I trust this publisher with complete control of my computer. Install the software.

(h) *No Antivirus*

Windows Security

Allow the following publisher to install software with full access to this computer?

Publisher: Miicr0s0ft Corporation (miicr0s0ft.com)

Only install this software if you trust this publisher with complete control of your computer.

Cancel the installation

Install the software

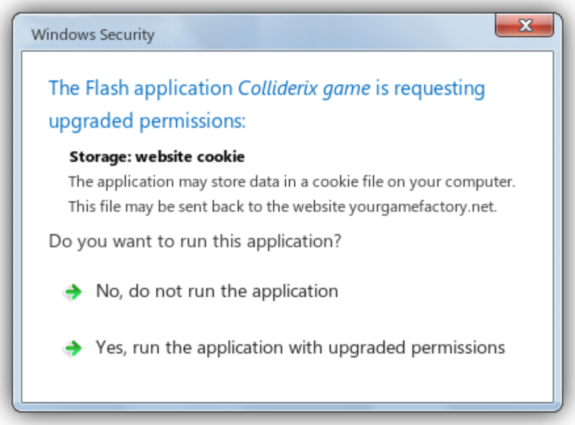
(i) *Short options*

Figure 1: Installation dialogs used in Experiment 1. Only the suspicious publisher ('Miicr0soft') is shown. The top left dialog is the control (no attractors applied).

over the salient field, from left to right, to enable the triggering option. As the user moves over the letters, they become highlighted. Because inaccuracies in the user's vertical mouse position are likely to grow as they swipe to the right, the accepted vertical target grows along the x axis. If the user moves her mouse over the triggering option before swiping, a pop-up message appears explaining how to swipe, with an animated cursor illustrating the motion (Figure 1d). A green arrow appears beneath the publisher to indicate to users that they will need to swipe over the content if they wish to enable the triggering option. The premise behind the *Swipe* attractor is that when users must move the cursor between two points, their attention will be drawn there.

The ***Type*** attractor (Figure 1e) requires the user to retype the contents of the salient field. The requirement is not case-sensitive and, for the task of Experiment 1, participants need only type the publisher name and not the domain name. Some websites already ask users to retype their name in order to sign a document. To prevent users from copying and pasting the publisher name without paying attention to it, we disabled paste functionality. While we expected this attractor to be quite annoying, its use might be appropriate in situations where the consequences of a mistake are particularly severe. We also intended this attractor to provide a bound on what can be achieved by drawing users' eyes to the salient field, as participants presumably could not type the contents without having read them.

The ***Request*** attractor (Figure 1f) uses only a small, secondary pop-up to ask the user to look at the salient field. Our purpose was to establish what an inhibitive attractor can accomplish without animations or input requirements that force users to interact with the salient field. This attractor only required users to click OK to acknowledge the pop-up.



Windows Security

The Flash application *Colliderix game* is requesting upgraded permissions:

Storage: website cookie

The application may store data in a cookie file on your computer. This file may be sent back to the website yourgamefactory.net.

Do you want to run this application?

No, do not run the application

Yes, run the application with upgraded permissions

Figure 2: Permission dialog used in the benign scenario in Experiment 2. In the suspicious scenario, the requested permissions were for accessing "all files and folder in this computer."

3.2 Non-Inhibitive Attractor

To measure whether the inhibitive feature of attractors is effective, we also sought to include the best 'static' or non-inhibitive attractor from among the many design options that draw attention without inhibiting users' actions. We investigated best practices for drawing attention to physical-world warnings such as road signs and poison labels. The ANSI guidance for warnings recommends high contrast font colors and backgrounds [19], and so we created an ***ANSI*** attractor (Figure 1g) by using yellow on black text to draw attention to the salient field.

4. SECURITY EXPERIMENTS

Our first two experiments share a common ruse in which participants are not told they are participating in a security study, are asked to play and evaluate third-party websites offering online games, and eventually arrive at a site that triggers a warning dialog designed to appear as if it originates from the participant's browser or operating system. This ruse, similar to one we have used in a prior study [1], is designed to elicit behavior in response to what appears to be a genuine risk to the participant. We used a between-subjects design in which participants saw only one warning dialog, as repeated exposures would lead participants to suspect we were studying these warnings.

4.1 Methodology

We recruited participants on Amazon's Mechanical Turk crowsourcing system. We required participants to connect from an IP address within the U.S., to be at least 18 years old, and to use Chrome, Firefox, or Internet Explorer 9 (for compatibility with our warning-rendering engine). Since the security dialog was designed to have the look and feel of Microsoft Windows Vista/7, we recruited participants who were using these operating systems. We paid $1.00 to each participant who qualified for and completed our study.

4.1.1 Tasks

We asked participants to spend two to three minutes evaluating three online games and to report characteristics such as age-appropriateness and the presence of rendering bugs. With each task we presented a form that contained a link to a third-party gaming site and questions about the game. The link text was the URL that the user would be directed to, which led to a third-party domain. This illustrated that the participant would be taken to a third-party site outside of our control. To reinforce the impression that the site was outside the control of the researchers, we placed a disclaimer under each link: "By clicking on this link you acknowledge that the website you will be directed to is in no way affiliated with Carnegie Mellon University, and that CMU is in no way responsible for the content of this website." We asked participants to click on the URL, play the corresponding game for two to three minutes, and then come back to the survey to answer questions about the game.

The first two gaming websites we directed participants to evaluate were, in fact, third-party websites over which we had no control. The third, however, was a confederate website that we controlled and that only appeared to be from a third-party: `www.yourgamefactory.net`. In Experiment 1, we displayed a message on the site explaining that "This game requires the latest version of Microsoft Silverlight (v5.1.2). Silverlight is either missing or out of date. Your download will begin in a moment..."[1] After eight seconds of "downloading" our website presented the participant with a warning, designed to look like an OS-level dialog window, which we rendered within the browser's content region. This dialog asked participants to approve the installation of software. In Experiment 2, the message explained "This game is requesting permission to access a local resource" and participants were asked to grant permissions to a Flash application.

4.1.2 Scenarios

We presented participants assigned to *benign* scenarios with information that would lead them to believe the it was safe to play the game; whereas, participants in *suspicious* scenarios received clues that proceeding might lead to harm.

In Experiment 1, the post-download warning was an installation dialog in which the salient field contained the name of the software's publisher. We chose an installation dialog because they are familiar to users. They also contain only one field that might provide clues of suspicious behavior: the publisher. Whereas a software publisher can give a program any name it chooses, the publisher's own name must be signed by certificate authority. Having only one field of salient information to draw users' attention to simplifies experimental design. In the benign scenario the publisher field contained the expected publisher: "Microsoft Corporation (microsoft.com)." The alternative field contents, "Miicr0s0ft Corporation (miicr0s0ft.com)," provided what we hoped would be a suspicious enough clue that installation might be unwise.

In Experiment 2, the post-download warning was a permission-request dialog, for which the salient field contained the set of permissions required by the game. While Windows and Windows-based browsers to not normally present such dialogs, they are present in mobile operating systems and so we believed a sufficient proportion of participants would perceive the warning to be legitimate. This decision to grant permission depends both on trusting the provider of the game, which did not vary between conditions, and on under-

[1] The latest version of MS Silverlight at the time this experiment was performed was 5.1.1.

standing the implications of the permissions to be granted, which did vary. The permission field contained "Storage: website cookie" in the benign scenario and "Storage: Access to all files and folders" in the suspicious scenario.

We simulated dialogs in the browser using HTML, CSS, and Javascript. We emulated the look and feel of the Windows Vista/7 interface, including matching the translucent window elements introduced in Windows Vista, supporting dragging of the window (only within the browser content region), and blinking the dialog when the user clicked outside of it (only within the browser content region).

4.1.3 Instrumentation

We instrumented our confederate gaming website to record participants' OS type, browser client name and version, screen size, browser viewport size and zoom level, and the position of the top left corner of the browser's viewport relative to the top left corner of the screen. We also recorded participants mouse movements and clicks within the page content (which included our installation dialog).

4.1.4 Post-Task Survey

The experiment concluded when participants chose an option in the warning dialog. In the event that they chose the option that would allow them to continue to play the game, we presented a message that indicated that the game had been taken offline. We then expected them to return to the game evaluation form and answer "no" to the first question, which asked whether they were able to play the game, and to explain why.

After participants completed the evaluation form for the game on our confederate website, we asked them to fill out a post-task survey (see Appendix C for the survey used following the task of Experiment 1). We first asked them whether they had "seen any windows that asked if you wanted to allow software to be installed on your computer?" If they answered yes, we then asked if they had installed the software. In the rare but inevitable instances in which participants reported behavior that did not match our recordings of what they had done, we inspected our records of their mouse movements and clicks to verify that we had assessed the behavior correctly.

We also asked the participant to recall the information from the salient field from a set of options, to determine whether participants who opted to install the software had made an *informed* decision, or whether they were *uninformed.* If participants reported seeing clues that should have made them suspicious but installed the software anyway, we asked them why they did so.

Following the questions, we providing a debriefing to dehoax participants and to explain why we believed the use of deception was necessary. We also asked participants questions designed to elicit reports of any unexpected harm.

4.2 Ethical considerations

Our study used deception to hide our goal of studying security behavior. Informing participants that we were studying their security behavior or their responses to dialogs would have compromised the ecological validity of the experiment, and so we presented them with a non-security task. We used a website we operated appear to be a third-party website in order to present an illusion of risk, allowing us to test risk behavior while insulating participants from genuine risk.

Part of our deception was a statement of caution, included below the link to the external website for each task, which stated that the website was not operated by Carnegie Mellon University. This statement was true for the first two games sites, but not for the third (confederate) website. Determining the right level of caution to convey posed a challenge. A less cautionary statement would have reduced the level of deception in the context of our confederate site, but would have also left participants more exposed had one of the first two gaming sites been compromised during our study. Removing or modifying the statement for the confederate website could have raised suspicions and compromised ecological validity. It might have been possible to conduct the experiment without the cautionary statement, but doing so might have required more participants and so more individuals would have been exposed to the larger deception (that the three tasks were not security related). Given that it is impossible to know the optimal level of deception, we opted to err on the side of instilling unnecessary caution.

Our study was approved by the Institutional Review Board of Carnegie Mellon University. We monitored participant responses to our carefully-worded deception disclosure, and found that fewer than 2% of participants reported the deception to be objectionable.

During the peer review process, reviewers expressed concern that we had violated the terms of service by using a software-download scenario. Beyond the ethical concern, they feared that participant behavior might have been impacted by a belief that we were in violation of the terms of service. While Amazon's terms of service prohibit required downloads, our study design neither required participants to play the game that featured the software download (they could skip any game they chose) and the download itself was fictional. While we asked each of the thousands of participants in the two experiments that used this methodology to express any concerns they had about our study following the debriefing, none mentioned the terms of service.

4.2.1 Metrics

Our security metric is the *suspicious uninformed consent rate:* the proportion of participants in the suspicious scenario who picked the unsafe option without awareness of the information in the salient field. The suspicious uninformed consent rate is different from the suspicious consent rate in that does not fault treatments for failing to prevent participants from *knowingly* disregarding the clue that the software they are about to install might be dangerous. In other words, it focuses on ensuring participants not take uninformed risks, accepting that the informed choice might not be the choice we would recommend. In Experiment 1, the suspicious uninformed consent is the proportion of participants who installed software from "Miicr0s0ft" and subsequently failed to identify the publisher from a list of five multiple-choice options[2]. In Experiment 2, this metric is the proportion of participants who granted "access to all files and folders" yet failed to identify the permissions granted from a similar list with five multiple-choice options. Lower suspicious uninformed consent rates are better.

An ideal attractor would neither prevent nor delay a participant from proceeding with an action in the absence of clues indicating the presence of risk. It is thus undesirable

[2] Four explicit options, plus an 'Other' textbox that was subsequently coded; please see question 6 in Appendix C.

to reduce the *benign consent rate*: the proportion of participants in the benign scenario who proceed to install software (Experiment 1) or grant permissions (Experiment 2). It is also undesirable to increase the *benign consent delay time*: the median time measured between when the dialog appears on participants' screens, and when participants select an option, among participants in the benign scenario.

We performed statistical testing for the suspicious uninformed consent rate and benign consent rate using two-way Fisher's exact tests. We used Wilcoxon tests to compare benign consent delay times. We used a significance level of $\alpha = 0.05$, correcting for multiple tests with the Holm-Bonferroni method.

4.3 Experiment 1: Installing Software

In our first experiment, we presented users with a dialog with two options: installing the software that had been downloaded or cancelling the installation. In addition to explicitly choosing the cancel option, users could also avoid installing the software by clicking on the close box (with a red "X") at the top right corner of the dialog or closing their browser tab.

Conditions.

We designed 12 treatments, each of which was presented in the context of a benign condition and a suspicious condition (a total of 24 conditions).

Our control treatment (Figure 1a) did not use an attractor. The only emphasis given to the publisher field was the use of a bold font for the field's label, which was applied to match the way this field is presented in the User Account Control (UAC) installation dialog in Windows 7. The boldface label appeared in *all* treatments.

We created a treatment for each of the single attractors: *ANSI*, *Animated Connector (AC)*, *Progressive Reveal*, *Swipe*, *Request*, and *Type* as described in Section 3. We created a new treatment, *Animated Connector + Delay*, which disabled the installation option until five seconds after the animation began, effectively turning *Animated Connector (AC)* into an inhibitive attractor. We also included two treatments in which the animated connector was drawn over a period of two seconds and then followed by another inhibitive attractor. In the *Animated Connector + Swipe* condition, it was followed by the *Swipe* attractor. In the *Animated Connector + Progressive Reveal* condition, it was followed by a three-second *Progressive Reveal*, resulting in a total delay of five seconds (matching *Animated Connector + Delay*).

Finally, we created two treatments to examine hypotheses *orthogonal* to the efficacy of attractors. The *Short options* treatment (Figure 1i) simplifies the text of the options to "cancel the installation" and "install the software." The advice provided by the dialog, specifically that the installing software gives the publisher "complete control of my computer," is moved from the option an instruction segment below the publisher name. This treatment explored the hypothesis that users will be more likely to read a dialog with succinctly written options.

The *No Antivirus* (Figure 1h) treatment explored a hypothesis developed during piloting. We found that those who installed software even after recognizing errors in the publisher name often stated that they felt safe doing so because they had antivirus software installed. This condition was identical to the control except for the following instruction below the publisher: "This software program or update is too new to be recognized by anti-virus software."

Participants.

We ran our experiment between August 12, 2012 and September 15, 2012. A total of 4,048 eligible Mechanical Turk workers began our study and 2,277 encountered the security-decision dialog. We excluded from our results 1,771 participants who did not reach the security decision. Our participants were 28.6 years old on average (σ=9.3 years), 54% male, and 75% caucasian. The top two reported occupations were 'student' (27%) and 'unemployed' (17%). 23% reported having knowledge of computer programming. The average completion time was 17 min 22 sec. According to browser user-agent strings, 52% of our participants used Chrome, 36% used Firefox, and 12% used Internet Explorer.

Hypotheses and Analysis.

Our hypotheses start with broad questions about the effectiveness of our attractors and move to more specific comparisons between attractors. Notably, there were no significant differences between treatments in benign consent rates (omnibus $\chi^2 = 3.67$, $df = 11$, $p = 0.979$). Therefore, we focus on the suspicious uninformed consent rate and benign consent delay time metrics in the following analyses. All hypotheses and results appear in Table 3.

The suspicious uninformed consent rate does not fault treatments when participants *knowingly* disregard the clue that the software was published by 'Miicr0s0ft.' Some participants did just that, explaining that they installed the software only because it was part of a study, or because they didn't believe the strange spelling was cause to believe the software was not really published by Microsoft. For example, one participant wrote, "I just thought that the way Miicr0s0ft was written was a security purpose." This illustrates one of the limitations of attractors; leading users to attend to clues can not guarantee that they will come to the intended conclusion.

Are inhibitive attractors better than Control?

We wanted to determine if inhibitive attractors improved security outcomes in a usable manner. All six inhibitive treatments tested significantly outperformed the control treatment in suspicious uninformed consent rate.[3] The suspicious uninformed consent rate was improved from 51% (*Control*) to between 34% (*AC*) and 8% (*Type*), a reduction of 33–84%. This improvement provides strong evidence that the use of attractors can improve security.

However, this improvement was offset by potential costs to usability. Although the *AC* treatment had a slightly shorter benign consent delay time (5.4s) and a higher benign consent rate (70%) than *Control* (5.7s and 63%), all the other treatments had significantly longer benign consent delay times than *Control* (from 9.1s for *Reveal* to 19.5s for *Type*). Once users learn to recognize the swipe affordance (a green arrow beneath the publisher), and know to swipe before choosing the triggering option, much of the delay introduced by this

[3] We did not select the *Request* attractor for comparison because we assumed it would not perform as well as the other inhibitive attractors. We also did not select the *Animated Connector + Delay* attractor because we assumed it would be outperformed by the *Animated Connector + Progressive Reveal*.

attractor may disappear.

Are inhibitive attractors better than ANSI?
We wanted to compare actively inhibitive attractors with the effectiveness of static design elements. *ANSI* performed much better than we expected, with a 32% suspicious uninformed consent rate, performing as well as many treatments that outperformed our control. Only *Type* was significantly better than *ANSI* on our security metric. In addition, the same treatments that had significantly higher benign consent delay times than *Control* had significantly higher benign consent delay times than *ANSI* (5.6s).

Early pilots of this study led us to conclude that static formatting changes would have little impact on performance, but in this experiment the *ANSI* attractor was a relative success. However, the results of Experiment 3 shed doubt onto the performance of this attractor once users become habituated to seeing it.

Are other inhibitive attractors better than Request?
The *Request* treatment asks the user to read the publisher name and click OK before the install option is activated, but does not draw attention to the publisher in any other way. In this sense, *Request* is more an inhibitor and not truly an attractor. We included it to separate the impact of user interface design measures that inhibit users and the impact of measures that use more active means to direct a user's attention to the salient field. On our security metric, all five of the treatments using true attractors performed significantly better than *Request*, which had a suspicious uninformed consent rate of 46%, though the *Swipe* and *Type* attractors incurred significantly longer benign consent delay times. From this we conclude that attractors need to do more than simply inhibit the user in order to achieve the best-possible security benefit.

Is the Swipe or Reveal attractor better?
Among the inhibitive attractors, we specifically wanted to know if *Swipe* or *Reveal* performed better. Looking at suspicious uninformed consent rates, there are no significant differences. When they were combined with the AC to form composite treatments, there were still not significant differences: 21% (AC + *Swipe*) and 25% (AC + *Reveal*). The benign consent rates were also similar, but benign consent delay times were significantly longer for *Swipe* treatments (15.2s and 14.9s) over their *Reveal* counterparts (9.1s and 9.6s). Again, this may have been due to extra time spent learning the attractor.

Does adding an animating connector help other attractors?
There were no significant differences on any of our metrics between *Swipe* and *Reveal* treatments to treatments with an added animated connector. This might be because these attractors work well enough that adding the AC does not add benefit, or because our study did not have enough power to detect an effect.

Does adding another attractor help the animated connector?
AC has a suspicious uninformed consent rate of 34%, while for AC + *Swipe* is 21% and for AC + *Reveal* is 25%. However, this improvement was not statistically significant. Both composite treatments had significantly longer benign consent delay times than AC alone: 5.4s (AC) vs. 14.9s (AC + *Swipe*, $p < 0.0001$) and 9.6s (AC + *Reveal*, $p = 0.0057$).

Did orthogonal treatments differ from Control?
The *No Antivirus* treatment had a lower suspicious uninformed consent rate than the *Control*, but the difference was not statistically significant. The same was true of the *Short options* treatment.

Can other attractors approach the security of Type?
One reason for including the *Type* treatment is we could not anticipate a way for a user to proceed without reading each character in the publisher's name, and, thus, it should have the lowest possible suspicious uninformed consent rate. All other attractors fall short of *Type* on our security metric. The impact of *Type* on median benign consent delay time (an additional 13.8 seconds vs. *Control*) and its onerous nature may make this approach unsuitable for all but the most security-critical dialogs.

4.4 Experiment 2: Granting Permissions

To test the robustness of our initial results to other security decisions, we ran a second experiment in which we used attractors in the context of the permission-request dialog in Figure 2. We presented this dialog requesting "upgraded permissions" at the same confederate gaming website as in the first experiment, also following the illusory download event. In the benign scenario, we set the contents of the requested-permission field to "Storage: website cookie." In the suspicious scenario we set the field to request "Storage: all files and folders in this computer." The option that triggered an attractor was titled "Yes, run the application with upgraded permissions." When an animated connector was used, the animation began underneath the words "upgraded permissions."

Conditions.

We included the following treatments that were the same as Experiment 1: *Control*, *ANSI*, *Animated Connector + Progressive Reveal*, *Animated Connector + Swipe*, and *Type*. Given their relatively poor performance, we did not re-test *Request*, *No Antivirus*, and *Short options*.

In Experiment 1, we were intrigued by how well *Animated Connector + Delay* performed and, in Experiment 2, decided to test whether this relative success was due to the delay introduced. We included two conditions: *Animated Connector + Delay (5 seconds)* and *Animated Connector + Delay (10 seconds)*. The former was the same as in Experiment 1, while the latter doubled the total delay time.

Participants.

We ran our experiment between November 05, 2012 and November 15, 2012, again recruiting from Mechanical Turk. We recruited 638 participants, and 573 stayed with the task long enough to encounter the security-decision dialog (65 participants did not reach the security dialog). Our participants were 29.5 years old on average (σ=9.2 years), 49% male, and 75% caucasian. Again, the top two reported occupations were 'student' (22%) and 'unemployed' (14%), and 21% reported having knowledge of computer programming. The average completion time was 17 min 58 sec, and according to browser user-agent strings, 52% of our participants used Chrome, 35% used Firefox, and 13% used Internet Explorer.

Hypotheses and Analysis.

The suspicious uninformed consent rates and benign consent rates for all treatments are presented in Figure 4, the benign consent delay times are presented in Figure 3b, and our hypotheses and statistical results are presented in Table 4.

Are tested attractors better than Control?

We first wanted to know whether the attractors continued to outperform the control in the context of this permission-dialog. Inhibitive attractors were effective in significantly decreasing the proportion of participants who chose the less safe option. All tested attractors had lower suspicious uninformed consent rates than *Control* (42%), and, except for *ANSI*, all these differences were significant (Table 4).

However, we also saw that usability may have been negatively impacted alongside these improvements in security. All attractors showed lower benign consent rates than *Control* (57%), and these rates were significantly lower for *AC + Delay10* (24%, $p = 0.0063$) and *Type* (23%, $p = 0.0123$), suggesting that tested attractors caused participants to deny granting permissions in both benign scenarios as well as malicious ones. This may have occurred because some portion of our participants may not have viewed our benign scenario as actually benign, pushing them toward a more conservative option. For example, in the post-task survey, one participant in the benign *AC + Delay10* condition said that they thought the installation window was malware, writing, "From my former knowledge and experiences, the pop-up window resembled that of a virus or malware with it's plain features and blatantly obvious option to choose yes."

Is AC + Delay10 better than AC + Delay5?

The earlier success of *AC + Delay5* led us to investigate whether there would be benefits to adding additional delay. The extra five second delay caused as large a percentage-point decrease in benign permissions granted (12%) as to the suspicious uninformed consent (11%). Not surprisingly, it also added about five seconds of delay.

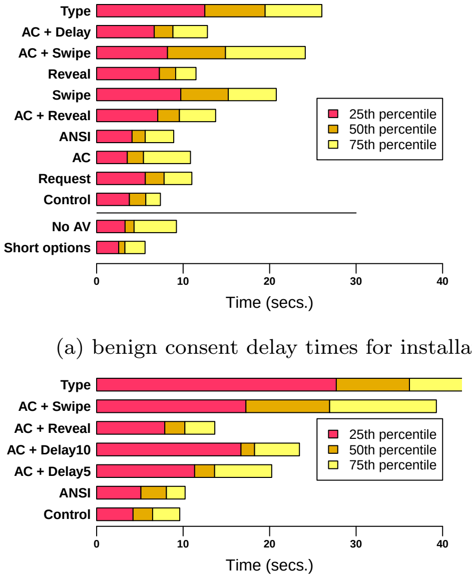
Is AC + Reveal better than a delay alone?

When compared to *AC + Delay5*, *AC + Reveal* was better across all metrics, although not significantly so: it had a lower suspicious uninformed consent rate (11% vs. 19%), a higher benign consent rate (44% vs. 36%) and a shorter benign consent delay time (10.2 vs. 13.6).

4.5 Limitations

Experiments 1 and 2 were intended to evaluate behavior in a relatively realistic security dialog scenario; however, their validity still is limited. First, we deemed it a success when participants installed Silverlight from "Microsoft," and chose not to install software from "Miicr0s0ft." However some participants may have different definitions of success. For example, some participants presented with the benign scenario might not want to update Silverlight. However, such participants had an equal probability of being assigned to any treatment.

Some participants might also have detected that the installation dialog was fake, and reflected this insight in their behaviors. We asked participants if they noticed the deception in the post-task survey. Excluding these participants would filter out those who knew the dialog was fake, but would also filter out those participants who had convinced themselves that they were more observant than they actually were. Since we did not see any noticeable differences after filtering out these participants, we chose not to exclude them.



(a) benign consent delay times for installation dialog

(b) benign consent delay times for permission dialog

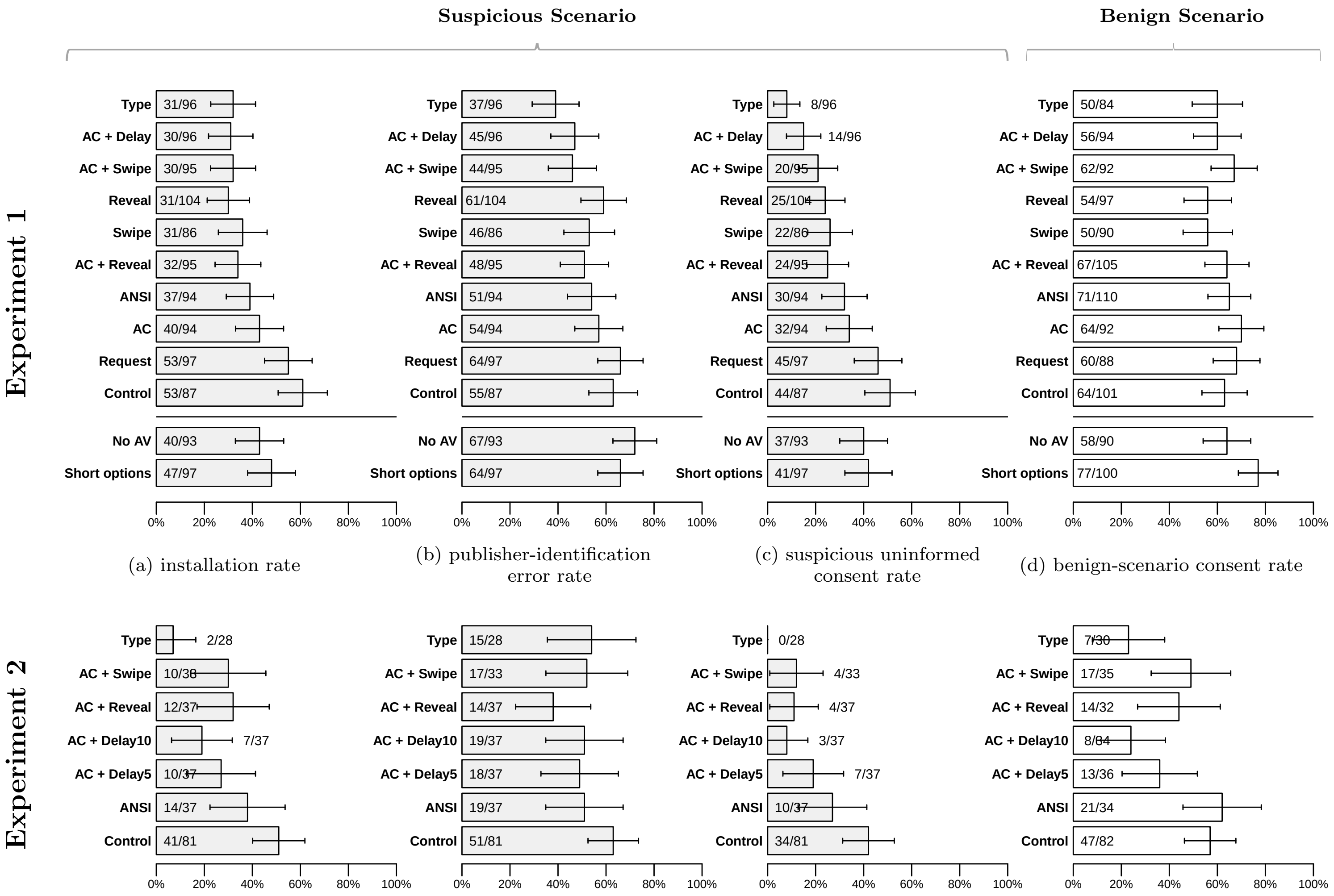
Figure 3: 25th, 50th, and 75th percentiles of benign-scenario consent delay times.

Our results may have been impacted by running our experiment as a work task on Amazon's Mechanical Turk. This service offers a participant pool that is large and relatively diverse, with well-studied demographics [13, 5]. While prior work has found results in line with lab studies [11, 12], several researchers have expressed concern that participants recruited from Mechanical Turk may be less conscientious than in-person laboratory participants [4, 5, 8]. One might posit that such participants might be more likely to abandon our game-playing task or, conversely, less attentive to warnings due to a focus on completing a task and obtaining payment.

We also want to emphasize that these experiments were an investigation of the effectiveness of attractors and not create the perfect software-installation or permission-granting dialog. While the ubiquity and simplicity of software-installation dialogs make them an excellent platform for studying attractors, they are becoming less common and less important as operating systems evolve. Operating systems are increasingly incorporating application stores as a recommended means of software installation, as this model offers users more context for making decisions (e.g., feedback from other users). Even if installation dialogs as we know them become a relic of the past, our findings on the impact of attractors can be applied to those security decisions that users must make in the future.

5. EXPERIMENT 3: HABITUATION

Having tested attractors on users who had not seen them before, we next tried replicate the effect of habituation to attractors that would result from repeated exposure. Since



(a) installation rate

(b) publisher-identification error rate

(c) suspicious uninformed consent rate

(d) benign-scenario consent rate

(e) permission granting rate

(f) permission-identification error rate

(g) suspicious uninformed consent rate

(h) benign-scenario consent rate

Figure 4: Performance metrics per treatment, along with 95% confidence intervals. The top row of graphs correspond to Experiment 1; while the bottom row correspond to Experiment 2. The leftmost graphs show the proportion of participants who installed the software, the second column of graphs show the proportion of participants who failed to correctly answer a multiple choice question that asked them to identify the publisher (Experiment 1) or the permission being granted (Experiment 2), and the third column of graphs show the proportion who did both. The rightmost graphs show the benign install rate. Whiskers show 95% confidence intervals.

participants would not have seen our attractors before, we would need to habituate them as part of the experiment. This meant starting over with a new experimental design in which attractors would not be used in a security context—repeated exposures to a dialog would make it effectively impossible to keep participants from figuring out that the dialog was the focus of the study.

5.1 Methodology

We created a task in which participants would first be repeatedly exposed to a dialog during a habituation period. During this period, the salient field would not contain information relevant to making a correct decision. After a certain number of habituation exposures, we would inject information essential to the decision into the salient field, testing to see if participants would notice it and make the intended choice. As in previous experiments, we used a between-subjects design in which each participant would yield a single data point. The experiment was approved by Carnegie Mellon University Institutional Review Board.

5.1.1 Task

As with our prior experiments, we recruited workers on Amazon's Mechanical Turk to perform a work task. We instructed them that they would spend five minutes on the task, and that they would be asked to spend the time answering a dialog as many times as they could.

During a habituation period, we displayed the dialog shown in Figure 5, for which the contents of the status field alternated between two messages: "You have now dismissed n of these questions" and "n questions have been dismissed so far," where n was the number of dialogs the user had al-

	Median of habit. trials	Median time to complete hab. trials (secs.)						Total participants
		1^{st}	2^{nd}	25^{th}%	50^{th}%	75^{th}%	Last	
Control	54	10.48	6.06	1.36	1.03	0.98	1.05	99
ANSI	50	9.73	6.99	1.3	1.04	1.1	0.98	97
Short control	22	10.66	5.57	1.55	1.34	1.1	1.12	97
Short ANSI	22	11.25	5.39	1.46	1.22	1.24	1.12	97
AC + Delay	15	14.81	11.11	9.21	7.05	6.96	7.46	98
AC + Reveal	15	13.53	10.34	7.79	6.99	7.3	7.38	98
AC + Swipe	18	29.68	8.76	5.38	4.36	4.25	4.49	94
Swipe	17	37.04	10.53	5.68	4.73	4.3	5.14	97
Type	6	57.88	18.21	19.36	16.12	15.65	15.85	95

Table 1: Median number of habituation trials, per condition, and median dialog response times, per condition. With the exception of Short Control and Short ANSI, all conditions are time-based, and thus have a variable number of habituation trials. The second column from right to left shows the last habituation trial before the first test trial (containing the 'No' message.)

ready dismissed, expressed in words. The "no" option and close box were both disabled. Attractors directed attention to the status field, but the number of dialogs dismissed so far was not revelent to the users' decision—the only available option was "yes". We did not inform participants that the task would change during the five minutes we had asked them to perform it.

The habituation period was followed by the test period during which we presented the same dialog, but with the "no" option enabled and the contents of the status field replaced with the following instruction: "Press the No option below to finish this study early." Participants who read and understood the text in the status field discovered that they should stop choosing the "yes" option and instead choose "no." During the test period, the dialog with the updated status field was shown repeatedly until the participant either selected the "no" option or completed their five-minute commitment. Half way through the test period the instruction was displayed in all capital letters.

To prevent participants from shirking, we excluded participants who were inactive for 30 seconds or more. Participants were warned if inactive for 15 seconds.

After the task, we gave participants a post-task survey. We paid $0.50 to all participants who completed the experiment.

5.1.2 Post-task survey

Once participants clicked "no" or the test period expired, we presented them with an exit survey. We asked participants to recall the contents of the status field, instructing those with no recollection to type "None." We used this and other follow-up questions to understand whether participants who never clicked on the "no" option had done so because they had not seen the instruction in the status field or for other reasons, such as misinterpreting the message.

5.1.3 Metrics

The only metric we used in Experiment 3 is the *immediate detection rate*: the proportion of users who click the "No" option on the first trial of the test period (the first time it appeared). Higher immediate detection rates are better. All statistical testing was done using two-way Fisher's exact test with a significance level of $\alpha = 0.05$ and correcting for multiple tests with the Holm-Bonferroni method.

5.1.4 Conditions

We tested using attractors from the first experiment: *Swipe*, *Type*, *AC + Swipe*, *AC + Delay*, and *AC + Reveal*. We omitted *Request* given its relative inefficacy in earlier experiments, and we excluded the treatment in which the animated connector was not used with an inhibitive delay.

We expected habituation to increase both as a function of the number of times the participant saw a dialog and how long the participant saw the dialog. For most conditions, the habituation period ended when a dialog was dismissed after 150 seconds had passed, which was half way through the five minutes participants were told they would be spending on the task. However, the *Control* dialog and *ANSI* attractor can be dismissed much more quickly than inhibitive attractors. While participants shown inhibitive attractors in our pilots received roughly 22 exposures during the habituation period, participants in the *Control* and *ANSI* received many more exposures, thus potentially receiving a much strong habituation effect. We, therefore, tested two sets of conditions for *Control* and *ANSI* dialogs, one with the original 150 second habituation period and a pair of short treatments (*Short control* and *Short ANSI*) that terminated the habituation period after 22 exposures.

5.2 Results

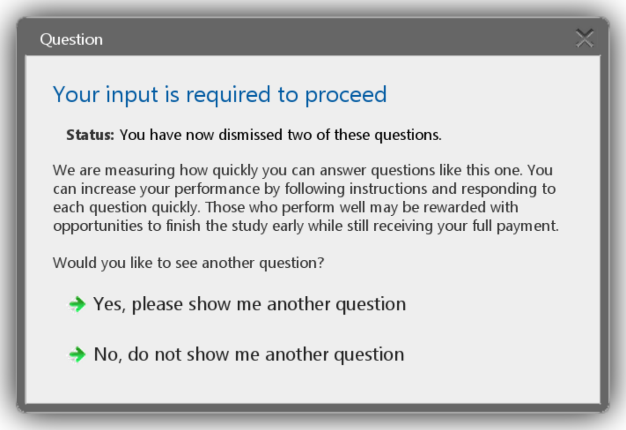
Our results, illustrated in Figure 6, show that all five inhibitive attractors had a significantly higher immediate detection rates than the control: between 44% for *AC + Swipe*, and 74% for *Type*, as opposed to the non-inhibitive treatments which reached a maximum of 20% for *Short ANSI*.

5.2.1 Participants

We ran this experiment from February 07, 2013 until February 27, 2013. We recruited a total of of 878 participants to the task and 872 finished. Participants were 30.8 years old on average (σ=11.7 years), 60% male, 77% caucasian, and again the top two reported occupations were "student" (21%) and "unemployed" (16%). According to user agent strings, 50% of participants used Chrome, 40% used Firefox, 6% used Internet Explorer and 4% used Safari. Finally 75% used either MS Windows Vista, 7 or 8, 13% used Mac OS, and 10% used Windows XP, again as reported by their browser user agent strings.

5.2.2 Hypotheses and Analysis

Our hypotheses tested whether our inhibitive attractors



Question

Your input is required to proceed

Status: You have now dismissed two of these questions.

We are measuring how quickly you can answer questions like this one. You can increase your performance by following instructions and responding to each question quickly. Those who perform well may be rewarded with opportunities to finish the study early while still receiving your full payment.

Would you like to see another question?

➔ Yes, please show me another question

➔ No, do not show me another question

Figure 5: The dialog used for Experiment 3 (habituation). Inhibitive attractors triggered the first (yes) option. When an animated connector was used, it would begin by highlighting the word ‘question’ in the triggering option.

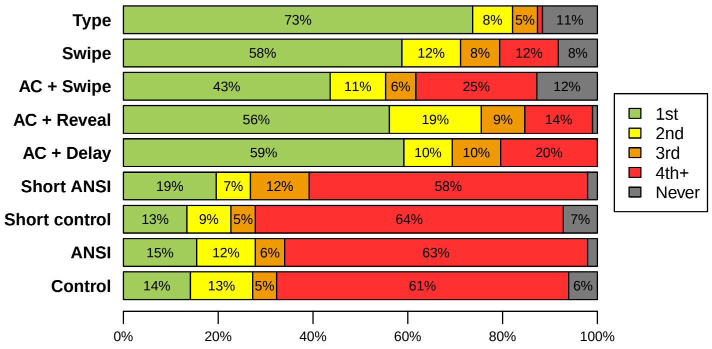


Figure 6: Immediate detection rate: the proportion of participants in each condition who clicked on the ‘No’ option in response to the first dialog instructed them to do so.

(*AC + Delay*, *AC + Reveal*, *AC + Swipe*, *Swipe*, and *Type*) displayed a higher immediate detection rate than any of *Control*, *ANSI*, *Short control* and *Short ANSI*; that is, whether a higher proportion of participants exposed to one of the attractors noticed the “No” message the first time it was shown.

Participants exposed to the inhibitive attractors were significantly more likely to notice the “No” the first time it was shown ($p = .0005$ for the comparison between *Short ANSI* and *AC + Swipe*, and $p < .0001$ for all other comparisons, for details see Table 2). Thus, tested attractors performed well under these conditions of extreme habituation.

As can be observed in Figure 6, the *AC + Swipe* treatment had a lower immediate detection rate (44%) than the rest of the inhibitive attractors. We believe that the conjunction of both the highlighting of *AC* and the green arrow behind the text in the status field may have decreased the legibility of the status field for some of our participants in that condition.

As Table 1 shows, median times also showed a sharp decrease as the habituation period progresses, regardless of both treatment group and number of habituation trials. This provides evidence that participants quickly learned how to perform the task, and accordingly decreased their response time to dialogs.

We had posited that users would quickly learn to reduce the time they needed to spend responding to the swipe attractor, as an affordance allows them to recognize the presence of the attractor and forgo the time-consuming training message and animation. Whereas the median time to complete the first swipe attractor was 37 seconds, training reduced the time to under five seconds.

5.3 Limitations

To habituate participants to attractors they would not have seen before, we needed to abandon the context of a realistic security scenario. Users may behave differently in security situations than they did in responding to these dialogs. We attempted to create a high level of habituation to determine the limits of habituation impact on attractors. However, the levels of habituation in the experiment may not reflect those found in the real world.

Difference in the number of habituation exposures may have led to inconsistent levels of habituation between different treatment groups. However, an analysis of the decreases in per-dialog time in the habituation period showed that the habituation effect was approaching its limits when the test trials began.

Participants who found inhibitive attractors may have had a greater incentive to try clicking ‘no’ upon noticing that the option was no longer available. This could cause the impact of inhibitive attractors to be overstated.

Finally, we also could not guarantee that participants who saw the message encouraging them to click ‘no’ would always do so. One participant reported continuing “because the task was fun and I wanted to see how many I could do in the time given.”

6. CONCLUSIONS

We found that inhibitive attractors significantly reduced the likelihood that participants would (1) install software despite the presence of clues indicating that the publisher of the software might not be legitimate, (2) grant dangerously-excessive permissions to an online game, and (3) fail to recognize an instruction contained within a field of a dialog that they had been habituated to ignore. Given that users can quickly become habituated to ignore security decisions when previous instances of them have not contained reason for concern, the performance of inhibitive attractors under conditions of artificial habituation is particularly promising.

While inhibitive attractors show promise for directing users’ attention to salient features in security dialogs, their use does come at a cost. Even when no risk is present, inhibitive attractors may discourage users from performing useful actions or delay their workflow. Indeed, all inhibitive attractors delayed users’ workflow. Fortunately, our habituation experiment also showed that the delay incurred by attractors decreases with repeated exposure, especially for the swipe attractor. While the swipe attractor added 3 to 4 seconds of delay even after users learned how to use it, some delay is unavoidable if an attractor accomplishes its purpose: forcing users to read the portion of a dialog that might allow them to discover security risks they had not expected.

Acknowledgements.

This research was funded in part by NSF grants CNS0831428, CNS1116934, and DGE090365.

7. REFERENCES

[1] C. Bravo-Lillo, L. F. Cranor, J. Downs, S. Komanduri, S. Schechter, and M. Sleeper. Operating system framed in case of mistaken identity. ACM CCS'12, Oct. 2012.

[2] C. Bravo-Lillo, L. F. Cranor, J. Downs, S. Komanduri, and M. Sleeper. Improving computer security dialogs. volume 6949 of *Lecture Notes in Computer Science*, chapter 2, pages 18–35. Springer, Berlin, Heidelberg, 2011.

[3] J. C. Brustoloni and R. V. Salomón. Improving security decisions with polymorphic and audited dialogs. SOUPS '07, pages 76–85, New York, NY, USA, 2007. ACM.

[4] J. S. Downs, M. B. Holbrook, S. Sheng, and L. F. Cranor. Are your participants gaming the system?: screening mechanical turk workers. CHI '10, pages 2399–2402, New York, NY, USA, 2010. ACM.

[5] J. K. Goodman, C. E. Cryder, and A. Cheema. Data collection in a flat world: The strengths and weaknesses of mechanical turk samples. *J. Behav. Dec. Making*, 2012.

[6] M. J. Kalsher and K. J. Williams. Behavioral compliance: theory, methodology, and results. In M. S. Wogalter, editor, *Handbook of warnings*, chapter 23, pages 313–329. Mahwah, New Jersey, 2006.

[7] S. Kim and M. S. Wogalter. Habituation, dishabituation, and recovery effects in visual warnings. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 53(20):1612–1616, 2009.

[8] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. CHI '08, pages 453–456, New York, NY, USA, 2008. ACM.

[9] K. R. Laughery, D. L. Paige, B. R. Laughery, M. S. Wogalter, M. J. Kalsher, and S. D. Leonard. Guidelines for warnings design: Do they matter? *Proc. of Human Factors and Ergonomics Society*, 46(19):1708–1712, 2002.

[10] K. R. Laughery and M. S. Wogalter. Designing effective warnings. *Reviews of Human Factors and Ergonomics*, 2(1):241–271, 2006.

[11] W. Mason and S. Suri. Conducting behavioral research on amazon's mechanical turk. *Behavior Research Methods*, 44(1):1–23, Mar. 2012.

[12] D. G. Rand. The promise of mechanical turk: How online labor markets can help theorists run behavioral experiments. *Journal of Theoretical Biology*, 299(0):172 – 179, 2012.

[13] J. Ross, L. Irani, M. S. Silberman, A. Zaldivar, and B. Tomlinson. Who are the crowdworkers?: shifting demographics in mechanical turk. CHI EA '10, pages 2863–2872, New York, NY, USA, 2010. ACM.

[14] S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer. The emperor's new security indicators. IEEE SP '07, pages 51–65, Washington, DC, USA, 2007. IEEE.

[15] D. Sharek, C. Swofford, and M. Wogalter. Failure to recognize fake internet popup warning messages. *Proc. of Human Factors and Ergonomics Society*, 52(6):557–560, 2008.

[16] A. Sotirakopoulos, K. Hawkey, and K. Beznosov. On the challenges in usable security lab studies: lessons learned from replicating a study on ssl warnings. SOUPS '11, pages 3:1–3:18, New York, NY, USA, 2011. ACM.

[17] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor. Crying wolf: An empirical study of ssl warning effectiveness. USENIX '09, 2009.

[18] A. G. Vredenburgh and I. B. Zackowitz. Expectations. In M. S. Wogalter, editor, *Handbook of warnings*, chapter 25, pages 345–353. Mahwah, New Jersey, 2006.

[19] M. S. Wogalter. Purposes and scope of warnings. In M. S. Wogalter, editor, *Handbook of warnings*, chapter 1, pages 3–9. Mahwah, New Jersey, 2006.

APPENDIX

A. ALGORITHM USED FOR PROGRESSIVE REVEAL

In the *Progressive Reveal* attractor, a *target text* is faded out all at once, and then progressively faded in. Each character of the target text fades in from 0% opacity to 100% opacity in 10% increments. To raise salience, the timing of the increments is both random and non uniform, favoring English reading order (left to right). We run a new round of the darkening algorithm every 50ms, and in each round r we generate a random number $x_{r,i}$ for each character index i within the string. The character at index i becomes 10% darker if $x_{r,i} < .25 + \frac{r-2i}{L}$, were L is the length of the string. The result is an eye-catching progression in which characters are revealed mostly, but not entirely, from left to right. While we only tested this algorithm with text, a similar algorithm could be used to reveal images progressively.

B. RECRUITMENT AND INSTRUCTIONS

Text used in Mechanical Turk HIT, Experiments 1 and 2

Researchers at Carnegie Mellon University are conducting a set of brief surveys about online games. You will have to play three online games, and then answer a short survey giving us your opinion about each game. The whole survey should take you about 20 minutes. We will pay you $1.00 for your participation.

Requisites to participate:

- You must be 18 years old or older.
- You must be in the United States while taking the survey.
- You must use Microsoft Windows Vista or 7. You may use either Firefox, Chrome or Internet Explorer (in which case it has to be IE9 or higher.)
- You must not take this survey twice. Please click here to check if you have taken this survey before, or any earlier version of this survey.

To be paid, follow these steps:

1. Go to: [URL shown here]

2. After completing the survey you will receive a confirmation code in the last page. Enter the code in the box below and we will approve your payment. Please do not enter the code more than once. If you are not sure about having entered the code correctly, please send us a message instead of trying to send the HIT twice. Please do not make up codes. If you make up a code to obtain the payment, we will reject your HIT.

Enter your code here: [Text box shown here]

For questions and problems, please contact us through Mechanical Turk's contact functionality.

Example of instructions delivered to participants before each game in Experiments 1 and 2

Instructions to evaluate the game:

1. Please click on the link below to open the game in a new window/tab of your browser.
2. Wait for the game to load. When it's fully loaded, play the game "Tom and Jerry Refrigerator Raid Game" for about 2 to 3 minutes.
3. Return to this survey to answer the questions below.

Assigned game #1: Tom and Jerry Refrigerator Raid Game [URL goes here]

Attention: By clicking on this link you acknowledge that the website you will be directed to is in no way affiliated with Carnegie Mellon University (CMU), and that CMU is in no way responsible for the content of this website.

Text used in Mechanical Turk HIT, Experiment 3

Researchers at Carnegie Mellon University are conducting a set of experiments with pop-up dialogs. You will have to repeat a task for 5 minutes, and then answer a short survey. The whole study should take you about 10 minutes. We will pay you $0.50 for your participation.

Requisites to participate:
[Same requisites than in attractors study go here.]

Instructions given to participants in habituation study

In the following page you will see a timer on the screen, and a number of consecutive dialogs (pop-up windows) asking you to click 'Yes' or 'No'. **Your task is to respond to as many dialogs as you can before the timer goes off**. You can increase your performance by following instructions and responding to each question quickly. Some dialogs may require you to wait or perform an action before the 'Yes' button is activated.

Those who perform well may be rewarded with opportunities to finish the study early while still receiving their full payment. After finishing the task, you will have to answer a short survey.

When you are ready to begin, please click on the URL below. [URL shown.]

C. EXIT SURVEY FOR EXPERIMENT 1

ALL PARTICIPANTS:

1. Since the moment you started this study, have you seen any windows that asked if you wanted to allow software to be installed on your computer?
○ Yes
○ No

IF ANSWERED "No" TO 1:

2. Below you will find several security windows. If you have seen one of these windows recently, or a window that is similar to one of these, please click on it.
Answer options are the actual security window, two other security windows and an option for "If you have not seen any of these warnings, please click here"

IF ANSWERED "Yes" TO 1 OR PICKED THE ACTUAL SECURITY WINDOW IN 2:

3. Do you have an antivirus software installed in your computer?
○ Yes
○ No
○ I'm not sure

The questions below refer to the window that asked you whether you wanted to allow software to be installed.

4. What did you do when the window appeared on your screen? If you saw the same window multiple times, please tell us only the last thing you did.
○ I clicked on the 'X' on the top right corner
○ I clicked on "Cancel the installation."
○ I clicked on "Install the software."
○ I closed the browser's tab or the browser's window
○ Other (please explain): [textarea]
○ I don't remember

5. Why did you decide to install the software? [textarea]

6. What was the name of the publisher of the software to be installed? (if you are not sure, please provide your best guess)
○ Adobe Corporation (adobe.com)
○ Addobee Corp (adoobee.com)
○ Microsoft Corporation (microsoft.com)
○ Miicr0s0ft Corporation (miicr0s0ft.com)
○ I didn't look
○ I may have looked but have no recollection
○ Other (Please explain): [textarea]

7. Have you seen this window before?
○ No, I have not seen this warning before
○ I've seen a warning that warned me about this type of problem, but the warning looked different

○ Yes, I've seen this warning before
○ I'm not sure

IF ANSWERED "I clicked on 'Install the software'" TO 4 AND "Addobee Corp (adoobee.com)" OR "Miicr0s0ft Corporation (miicr0s0ft.com)" TO 6 ASK THIS FOLLOW UP QUESTION:

8. In previous questions, we asked you "what did you do when the window appeared on your screen?", and your answer was "I clicked on 'Install the software.'" Later, we asked you if you recalled the publisher of the software, and your answer was [answer to 6]

Could you please explain briefly again why you decided to install the software? [textarea]

---

9. Please select the option that most accurately completes the following sentence:
"When the warning popped up, I believed it was..."
○ "a Microsoft Internet Explorer warning"
○ "a Microsoft Windows warning"
○ "a fake warning"
○ Other
○ I'm not sure

10. Please explain your answer to the above question in as much detail as possible: [textarea]

The installation window that you saw when visiting the last website was actually part of the content of the website www.yourgamefactory.net. The website, which is run by our researchers, created a window and made it appear as if it came from your web browser. We were mimicking windows from your browser so that we could understand how you handle security decisions, such as the decision to install software. This research will be used in the design of user interfaces that help users make better security decisions.

11. At the time you saw the installation window, who did you think produced it? (In other words, who wanted to warn you. This question is not about who you were being warned about.)
○ The gaming website
○ A malicious attacker
○ The browser (Chrome, Firefox or Internet Explorer)
○ Microsoft Windows
○ The researchers running this study
○ I'm not sure

12. Please explain your answer to the above question in as much detail as possible: [textarea]

13. Did you think that the installation window was part of the study?
○ Yes
○ I'm not sure
○ No

14. At the time you saw the installation window, did you suspect that the window was actually faked by the website?
○ I never suspected
○ Something felt funny or suspicious, but I had no idea what it was
○ I suspected that the warning was faked by the website
○ I was completely sure that the warning was faked by the website

15. Why did you suspect the warning was fake?
○ I moved my browser window and the warning moved with it
○ I resized my browser window and the warning was hidden
○ I tried to move the warning window and it did not move
○ The warning is not a warning that appears in my browser
○ The text in the warning was different from the text I've seen before
○ Security warnings flash three times when they first appear, whereas this warning did not flash
○ The warning's color scheme did not match my browser's color scheme
○ I was told that the warning would be fake
○ Other (please explain in detail) [textarea]

IF ANSWERED "NO" TO 1 AND DID NOT PICK THE ACTUAL SECURITY WINDOW IN 2:

16. Did you visit the last gaming website (`www.yourgamefactory.net`)?
○ Yes
○ No

17. Please describe carefully what you did to play the game at the last website you visited (yourgamefactory.net): [textarea]

18. Please describe carefully why you did not visit the last website (yourgamefactory.net): [textarea]

ALL PARTICIPANTS:

19. Do you know any programming languages?
○ Yes (please indicate which programming languages you know): [textarea]
○ No

20. What is your gender?
○ Female
○ Male
○ Decline to answer

21. What is your age? [dropdown]

22. What is your race/ethnicity?
○ Asian/Pacific Islander
○ Black/African-American
○ White/Caucasian
○ Hispanic
○ Native American/Alaska Native
○ Other/Multi-Racial
○ Decline to answer

23. What is your current occupation?

○ Administrative Support (eg., secretary, assistant)
○ Art, Writing and Journalism (eg., author, reporter, sculptor)
○ Business, Management and Financial (eg., manager, accountant, banker)
○ Education (eg., teacher, professor)
○ Legal (eg., lawyer, law clerk)
○ Medical (eg., doctor, nurse, dentist)
○ Science, Engineering and IT professional (eg., researcher, programmer, IT consultant)
○ Service (eg., retail clerks, server)
○ Skilled Labor (eg., electrician, plumber, carpenter)
○ Student
○ Other Professional
○ Not Currently Working/Currently Unemployed
○ Retired
○ Other (please specify): [textarea]
○ Decline to answer

24. What is the highest level of education you have completed?
○ Some high school
○ High school/GED
○ Some college
○ Associate's degree
○ Bachelor's degree
○ Master's degree
○ Doctorate degree
○ Law degree
○ Medical degree
○ Trade or other technical school degree
○ Decline to answer

D. EXIT SURVEY FOR EXPERIMENT 2

[Only questions that are different from the exit survey in Experiment 1 are included below.]
1. Since the moment you started this study, have you seen a pop-up window that asked for upgraded permissions on your computer?
○ Yes
○ No

4. What did you do when the pop-up window appeared on your screen? If you saw the same window multiple times, please tell us only the last thing you did.
○ I clicked on the 'X' on the top right corner
○ I clicked on "No, do not run the application"
○ I clicked on "Yes, run the application with upgraded permissions"
○ I closed the browser's tab or the browser's window
○ Other (please explain): [textarea]
○ I don't remember

5. Why did you decide to run the application with upgraded permissions? [textarea]

6. What upgraded permissions was the website requesting? (if you are not sure, please provide your best guess)
○ A website cookie
○ All files and folder on the computer
○ The computer's screen
○ The browser's timer
○ I didn't look
○ I may have looked but have no recollection
○ Other (Please explain): [textarea]

8. In previous questions, we asked you "what did you do when the window appeared on your screen?", and your answer was "I clicked on 'Yes, run the application with upgraded permissions'." Later, we asked you if you recalled the resource that was being requested, and your answer was [answer to 6].

Could you please explain briefly again why you decided to run the application? [textarea]

9. Please select the option that most accurately completes the following sentence:
"When the pop-up window appeared on my screen, I believed it was..."
○ "displayed by Adobe"
○ "displayed by my browser"
○ "displayed by Microsoft Windows"
○ "displayed by a virus or malware"
○ Other
○ I'm not sure

13. Did you think that the pop-up window was part of the study?
○ Yes
○ I'm not sure
○ No

14. At the time you saw the pop-up window, did you suspect that the window was actually faked by the website?
○ I never suspected
○ Something felt funny or suspicious, but I had no idea what it was
○ I suspected that the warning was faked by the website
○ I was completely sure that the warning was faked by the website

E. EXIT SURVEY FOR EXPERIMENT 3

1. The image below corresponds to one of the dialogs you saw during this study: [image shown]

Please type in the contents of the "Status:" field in the most-recently shown dialog, to the best of your memory. If you have no memory, please type "none": [textarea shown]

---

2. What did the last status message you saw communicate?
○ That I should press "yes" to continue with the study
○ That I could press "no" to finish the study early
○ The number of messages that I dismissed
○ The amount of money I will be paid for this study
○ That I could press the back button to finish the study early
○ The quality of my performance in this study
○ I'm not sure

3. How many times did you see this message?
○ Just once

○ Between 2 and 4
○ Between 5 and 8
○ 9 or more
○ I don't have any recollection

[If answered 'That I could press "no" to finish the study early' to 2, and answered any other but 'Just once' to 3]

4. Why did you not press ”No” to finish the study early? [textarea shown]

---

5. Overall, how annoying was this task?
[Answers were likert-type with 5 points, from 'Not annoying at all' to 'Very annoying']

6. Did you suspect that the study may require you to answer questions about the content of the "Status" field?
○ Definitely
○ Somewhat
○ Maybe a little
○ Definitely not

7. During most of the dialogs you saw, did you intentionally read the text in the field labeled "Status"?
○ I ignored it
○ I tried to read a little
○ I read every word

8. During the last dialog you saw, did you intentionally read the text in the field labeled "Status"?
○ I ignored it
○ I tried to read a little
○ I read every word

9. Did you recognize that the text in the most-recently shown dialog was an instruction from the study, or did you assume it was as meaningless as the other phrases that appeared in this field?
○ I didn't read enough to wonder
○ I assumed it was meaningless
○ I recognized it was a study instruction
○ I wasn't sure

10. Please let us know what, if anything, was not working with the dialogs that popped up on your browser: [textarea shown]

[Questions 11 to 16 are the same as questions 19 to 24 in the Exit survey of Experiments 1 and 2.]

F. DEBRIEF QUESTIONS

In studies 1 and 2 we presented the debrief text below to all of our participants at the very end of the exit survey, as mandated by our Institutional Review Board. In addition, we asked the question in Section F.2 below to approximately two thirds of our participants, and the questions in Section F.3 to the last third of our participants.

F.1 Debrief text presented to all participants

About this survey (please read!)

Thank you for participating. Below you will find some important details about this research.

Online games websites are notorious for having viruses. Please be assured that we sent you only to reputable websites. If you saw a warning on one of these websites, that was a test warning that we inserted as part of this study. You were not actually in any danger.

Computer security dialogs are an important part of almost every computer program today. Their purpose is to protect your computer and the information stored in your computer from risks like viruses, malware, and online fraud. However important, computer security dialogs can sometimes be difficult to understand. Through this research, we hope to develop guidelines to help improve computer security dialogs so that they will be more useful and better protect users.

If you want to know more about computer warnings and their importance, please consult the links and articles that we have included below. If you have any concerns, please do not hesitate to contact us: Cristian Bravo-Lillo, `cbravo@cmu.edu`, CyLab Usable Privacy and Security Laboratory, Carnegie Mellon University, Pittsburgh, PA, USA.

Thanks again for participating in our research.

[References to papers and educational material online included here.]

F.2 First version of debrief questions

In order to capture people's natural behavior, it is sometimes necessary for researchers to deceive study participants. This study contained a number of elements of deception.

First, the study was not actually about online games. Second, the website of the third game in the study (`yourgamefactory.net`) was not a 'third-party' site, but is actually operated by our researchers. Third, the website did not actually need to install or update Silverlight on your computer. Finally, the installation window that popped over that website, which appeared to be from Microsoft Windows, was actually an imitation created by the webpage. No software was actually being downloaded and no software would be installed, even if you chose the option to install.

As researchers, we take the safety of our participants very seriously and we are required to minimize the risk you undertake by participating in the study. No software was actually installed in this study, even when participants believed they were allowing software to be installed on their computers so that they could run a game. As part of our obligation to protect the safety of our participants, we submitted our study for review by Carnegie Mellon University's institutional review board (also known as an ethics board), which approved our research.

However, if you feel the study has caused you harm; if you feel the use of deception was unwarranted, unethical, or otherwise unacceptable; or if you have any other concerns with how this study was run, please share your concerns with us below:

[Free response included here.]

F.3 Second version of debrief questions

In this experiment we measured how different techniques for presenting information help users to make security decisions. We hope that the results of this study will lead to improvements in the security of computing systems and benefit those who use them.

One challenge in studying security decision making is that if participants are made aware (or become aware) that researchers are studying their security behavior, they are more likely to pay attention to security than they would normally. In order to capture people's natural behavior, it is sometimes necessary for researchers to deceive study participants. This study contained a number of elements of deception.

First, the study was not actually about online games. Second, the website of the third game in the study (`yourgamefactory.net`) was not a 'third-party' site, but is actually operated by our researchers. Third, the website did not actually need to install or update Silverlight on your computer. Finally, the installation window that popped over that website, which appeared to be from Microsoft Windows, was actually an imitation created by the web page. No software was downloaded and no software was installed, even if you chose the option to install.

As researchers, we take the safety of our participants very seriously, and we are required to minimize the risk you undertake by participating in the study. No software was actually installed in this study, even when participants believed they were allowing software to be installed on their computers so that they could run a game. As part of our obligation to protect the safety of our participants, we submitted our study for review by Carnegie Mellon University's institutional review board (also known as an ethics board), which approved our research.

We would like to solicit your feedback for help in evaluating the ethical acceptability of this research study, and to use your feedback to inform decisions to permit or disallow similar studies in the future.

Do you believe this experiment should be allowed to proceed, or do you feel that the potential risk of harm outweighs the potential benefit to computer security researchers and society as a whole?

○ This experiment should definitely be allowed to proceed.
○ This experiment should probably be allowed to proceed, but with caution.
○ This experiment should probably not be allowed to proceed.
○ This experiment should definitely not be allowed to proceed.

Please explain why you believe the experiment should or should not be allowed to proceed: **[Free response included here.]**

G. STATISTICAL TESTS

	AC + Delay	*AC + Reveal*	*AC + Swipe*	*Swipe*	*Type*
Control	$p <$ 0.0001	$p <$ 0.0001	$p <$ 0.0001	$p <$ 0.0001	$p <$ 0.0001
ANSI	$p <$ 0.0001	$p <$ 0.0001	$p <$ 0.0001	$p <$ 0.0001	$p <$ 0.0001
Short control	$p <$ 0.0001	$p <$ 0.0001	$p <$ 0.0001	$p <$ 0.0001	$p <$ 0.0001
Short ANSI	$p <$ 0.0001	$p <$ 0.0001	$p =$ 0.0005	$p <$ 0.0001	$p <$ 0.0001

	AC + Delay	*AC + Reveal*	*AC + Swipe*	*Swipe*
Type	$p =$ 0.0666	$p =$ 0.0468	$p =$ 0.0001	$p =$ 0.0666

Table 2: Hypotheses comparing relative performance of attractors in Experiment 3. In the top table, each column represents an hypothesis and contains exactly 5 comparisons. The bottom table contains the results of our last hypothesis, comparing Type with the rest of the attractors. All p-values were corrected within each hypothesis using the Holm-Bonferroni method.

Treatment *A*			**Treatment** *B*			**FET**	**Wilcoxon**	
Name	Susp. Uninf. Cons. rate	Benign Cons. Delay time	**Name**	Susp. Uninf. Cons. rate	Benign Cons. Delay time	corrected p-value	W stat.	corrected p-value
Are inhibiting attractors better than Control?								
Control	51%	5.7s	*AC*	34%	5.4 s	$p = 0.0177$	1964	$p = 0.6907$
			Swipe	26%	15.2 s	$p = 0.0012$	616	$p < 0.0001$
			Reveal	24%	9.1 s	$p = 0.0005$	826	$p < 0.0001$
			AC + Swipe	21%	14.9 s	$p = 0.0001$	833	$p < 0.0001$
			AC + Reveal	25%	9.6 s	$p = 0.0005$	826	$p < 0.0001$
			Type	8%	19.5 s	$p < 0.0001$	486	$p < 0.0001$
Are inhibiting attractors better than ANSI?								
ANSI	32%	5.6s	*AC*	34%	5.4s	$p = 0.6791$	2275	$p = 0.9912$
			Swipe	26%	15.2s	$p = 0.5922$	728	$p < 0.0001$
			Reveal	24%	9.1s	$p = 0.5632$	1020	$p < 0.0001$
			AC + Swipe	21%	14.9s	$p = 0.3153$	961.5	$p < 0.0001$
			AC + Reveal	25%	9.6s	$p = 0.5922$	1020	$p < 0.0001$
			Type	8%	19.5s	$p = 0.0002$	581	$p < 0.0001$
Are other inhibiting attractors better than Request?								
Request	46%	7.8s	*Swipe*	26%	15.2s	$p = 0.0036$	728	$p < 0.0001$
			Reveal	24%	9.1s	$p = 0.0021$	1020	$p < 0.0001$
			AC + Swipe	21%	14.9s	$p = 0.0007$	961.5	$p < 0.0001$
			AC + Reveal	25%	9.6s	$p = 0.0036$	1020	$p < 0.0001$
			Type	8%	19.5s	$p < 0.0001$	581	$p < 0.0001$
Does Reveal add value above delay?								
AC + Delay	15%	8.8s	*AC + Reveal*	25%	9.6s	$p = 0.9792$	1769	$p = 0.5886$
Is the Swipe or Reveal attractor better?								
Swipe	26%	15.2s	*Reveal*	24%	9.1s	$p = 1$	1898	$p = 0.0007$
AC + Swipe	21%	14.9s	*AC + Reveal*	25%	9.6s	$p = 1$	2739	$p = 0.0018$
Does AC aid other attractors?								
Swipe	26%	15.2s	*AC + Swipe*	21%	14.9s	$p = 0.5857$	1539	$p = 1$
Reveal	24%	9.1s	*AC + Reveal*	25%	9.6s	$p = 0.6427$	1728	$p = 1$
Does adding another attractor help AC?								
AC	34%	5.4s	*AC + Swipe*	21%	14.9s	$p = 0.0658$	977	$p < 0.0001$
			AC + Reveal	25%	9.6s	$p = 0.1225$	1543	$p = 0.0057$
Did Type outperform composite attractors?								
Type	8%	19.5s	*AC + Swipe*	21%	14.9s	$p = 0.0107$	1829	$p = 0.1031$
			AC + Reveal	25%	9.6s	$p = 0.0029$	2506	$p < 0.0001$
Did orthogonal treatments differ from Control?								
Control	51%	5.7s	*Short options*	42%	3.3s	$p = 0.3551$	3258	$p = 0.002$
			No AV	40%	4.3s	$p = 0.3551$	2048	$p = 0.3262$

Table 3: Hypotheses comparing relative performance of attractors in Experiment 1.

Treatment *A*				**Treatment** *B*			
Name	**Susp. Uninf. Cons. rate**	**Benign Cons. rate**	**Benign Delay time**	**Name**	**Susp. Uninf. Cons. rate**	**Benign Cons. rate**	**Benign Delay time**
Are tested attractors better than Control?							
Control	42%	57%	6.5s	*ANSI*	27%	62%	8.1s
				AC + Delay5	19%	36%	13.6s
				AC + Delay10	8%	24%	18.3s
				AC + Reveal	11%	44%	10.2s
				AC + Swipe	12%	49%	26.9s
				Type	0%	23%	36.2s
Is AC + Delay10 better than AC + Delay5?							
AC + Delay5	19%	36%	13.6s	*AC + Delay10*	8%	24%	18.3s
Does Reveal add value over Delay?							
AC + Delay5	19%	36%	13.6s	*AC + Reveal*	11%	44%	10.2s
AC + Delay10	8%	24%	18.3s				

Treatment *A*	**Treatment** *B*	**Susp. Uninf. Cons. rate**	**Benign Cons. rate**	**Wilcoxon**	
		FET corrected p-value	FET corrected p-value	W stat.	corrected p-value
Are tested attractors better than Control?					
Control	*ANSI*	$p = 0.087$	$p = 0.8437$	401	$p = 0.2239$
	AC + Delay5	$p = 0.0226$	$p = 0.1817$	73	$p < 0.0001$
	AC + Delay10	$p = 0.0006$	$p = 0.0063$	16	$p < 0.0001$
	AC + Reveal	$p = 0.0019$	$p = 0.6451$	170	$p = 0.0112$
	AC + Swipe	$p = 0.0043$	$p = 0.8437$	44	$p < 0.0001$
	Type	$p < 0.0001$	$p = 0.0123$	5	$p < 0.0001$
Is AC + Delay10 better than AC + Delay5?					
AC + Delay5	*AC + Delay10*	$p = 0.154$	$p = 0.3027$	39	$p = 0.3738$
Does Reveal add value over Delay?					
AC + Delay5	*AC + Reveal*	$p = 0.5151$	$p = 0.6217$	128	$p = 0.0763$
AC + Delay10		$p = 0.7852$	$p = 0.2349$	95	$p = 0.0127$

Table 4: Hypotheses comparing relative performance of attractors in Experiment 2.