Using Data Type Based Security Alert Dialogs to Raise Online Security Awareness

Max-Emanuel Maurer, Alexander De Luca, Sylvia Kempe
University of Munich
Media Informatics Group
Amalienstr. 17
80333 München
{max.maurer, alexander.de.luca}@ifi.lmu.de, kempe@cip.ifi.lmu.de

ABSTRACT

When browsing the Internet, users are likely to be exposed to security and privacy threats – like fraudulent websites. Automatic browser mechanisms can protect them only to some extent. In other situations it is still important to raise the users' security awareness at the right moment. Passive indicators are mostly overlooked and blocking warnings are quickly dismissed by habituated users. In this work, we present a new concept of warnings that appear in-context, right next to data the user has just entered. Those dialogs are displayed whenever critical data types – e.g. credit card data – are entered by the users into online forms. Since they do not immediately interrupt the users' interaction but appear right in the users' focus, it is possible to place important security information in a way that it can be easily seen.

We implemented the concept as a Firefox plugin and evaluated it in a row of studies including two lab studies, one focus group and one real world study. Results show that the concept is very well accepted by the users and that with the plugin, especially non-expert participants were more likely to identify fraudulent (or phishing) websites than using the standard browser warnings. Besides this, we were able to gather interesting findings on warning usage.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation (e.g. HCI)**]: User Interfaces - Input devices and strategies, evaluation.; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Security, Experimentation, Human Factors

Keywords

Security awareness, web browsing, data type based, in-context

Symposium on Usable Privacy and Security (SOUPS) 2011, July 20–22, 2011, Pittsburgh, PA USA

1. INTRODUCTION

Protecting users from online security threats is a non-trivial task. Automatic protection using blacklists is able to protect the user up to a certain degree but fails to capture attacking websites in real-time, leaving the first visitors to such a site vulnerable [20].

Standard browser security indicators like the padlock icon, the https indicator, the URL and the content of a website are not enough for people to detect fraudulent websites. Dhamija et al. [8] conducted a study in which participants made mistakes judging different websites 40% of the time. Even though they know that they had to watch out for fraudulent websites, 90% of the participants fell for the best phishing attack.

For malicious websites that cannot automatically be ruled out, it is important to have the user decide on what to do next. And as already stated by Whitten and Tygar [16], security is never the users' primary goal. This is why it is important to raise the users security awareness in a balanced way: At the right moment, when an undesired event might happen. Currently, this is usually done using either non-blocking status indicators to alert the user or bringing up blocking dialog windows, that have to be confirmed by the user first before being able to continue. Both approaches have their shortcomings: Most non-blocking indicators are constantly overlooked by the user who is busy with completing the primary task [19], whilst blocking indicators are quickly dismissed by the user who gets habituated to them [1]. Besides this, there is a third option: teaching the users when and how to watch out for which kind of attacks.

In this work, we describe the iterative design process of a new kind of alert dialog taking the current state of research into account. An image of the first prototype can be found in Figure 1. The dialog does not appear when the browser is started or when a website is loaded. Instead, it is triggered by different types of data entered by the user. This reduces the number of appearing dialogs to situations that incorporate critical data and thus are already somewhat critical to the user. In contrast to most of todays approaches, we do not use technical information for triggering our warning. However, since this is important information for the user, it is displayed in the body of the message.

Whenever a dialog opens, it is shown right next to the form field where the user is currently entering critical data. This ensures that the dialog is in the current field of view. The dialog is not 100% blocking – we call this semi-blocking – as the user is able to continue typing. The user then needs

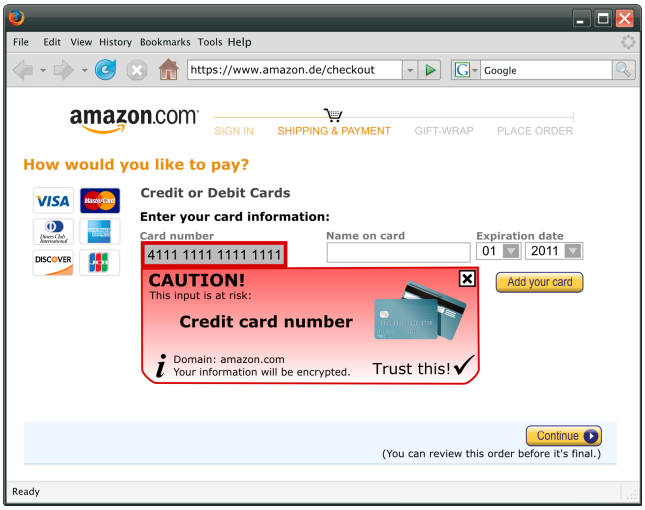

Figure 1: A screenshot of the first prototype used on an encrypted website. It informs the user that important information (credit card number) will be transmitted over a secure channel [12]. [staged for printing purposes]

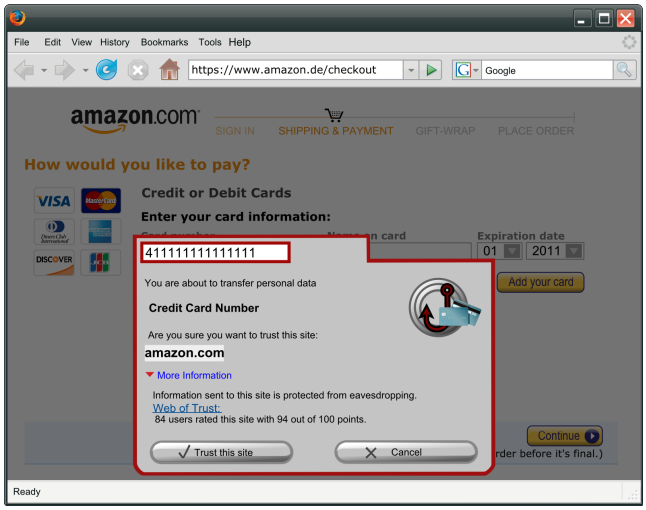

Figure 2: A screenshot of the final version of the prototype with the "Additional Information" box already unfolded. [staged for printing purposes]

to decide whether to trust the website – by adding it to a personal whitelist – or to stop entering data on that website. The dialog can be dragged away to uncover information that is potentially necessary for the user to make a decision.

We tested and refined the user interface and the concept itself over several studies (the first study has been presented as a Work-In-Progres at CHI 2011 [12]), including two lab studies, one focus group and a real world study. The lab studies showed that the concept enabled users to detect fraudulent websites much better than in a standard browser environment. Worries of some of the lab study participants that the dialog shows up too often could be diminished by a field study that proved that the concept of a dynamic whitelist quickly reduces the number of appearing dialogs.

2. RELATED WORK

Research on online security protection and warnings covers a wide range of areas that have all influenced this work in some manner.

2.1 Automated Protection

The Google Safe Browsing environment [6] is used to protect users from fraudulent websites. It is mainly used by Mozilla Firefox and Google Chrome. This blacklist-based approach can be used with a remote blacklist queried for every URL visited or by using a periodically updated local blacklist, for privacy reasons. Whenever a blacklisted page is loaded, it is completely disabled and a warning message is displayed in the browser instead. Just like all blacklist-based approaches, it requires some time to detect and enlist a fraudulent site after it shows up on the Internet. The timespan in between might already be enough for most sites to steal data from a significant amount of users. Zhang et al. [20] tested this approach using two other publicly available sources of phishing websites. Since these sources also had to get hold of the phishing URL first, it is surprising that after 24 hours only 84% for the first source and 73% of the phishing pages for the second source had been detected by Google Safe Browsing. Ludl et al. [11] did a similar study resulting in a hit rate of only 65%.

With "SpoofGuard", Chou et al. [4] created a plugin that uses different heuristics to detect malicious websites by creating a "total spoof score". The plugin has a stateless component that does not need any prior user input for its automatic detection. It uses a URL check, an image check, a link check and a password check in stateless mode to determine the spoof score. The plugin has been evaluated together with other approaches in [20] and was able to discover over 90% of the tested malicious websites. However also 42% of legitimate sites were detected as phishing attacks and for another 50% the plugin was undecided about the website.

2.2 Raising Security Awareness

As explained above, supporting the users in making their own mature security decisions remains important. To help with this, related work often relies on a non-blocking or blocking warning principle. The lock icon indicating an SSL secured connection is such a non-blocking indicator. Blocking warnings are for instance used whenever a user tries to visit a site with a self-issued certificate. The user cannot access the site until dismissing the warning. Both approaches have their advantages and drawbacks.

2.2.1 Blocking and Non-Blocking approaches

Wu et al. [19] compared different passive – or non-blocking – indicators by using different security toolbars. They simulated different test toolbars on top of potential phishing websites and measured how they would influence the participants' behavior. As a personal assistant of "John Smith" 30 subjects had to process 20 emails, five of which were phishing e-mails using standard attacks (e.g. IP-address attack). Depending on the toolbar used, 33% to 45% of the participants were successfully spoofed. Users did not obey the toolbar indicators or explained them away.

Egelman et al. [9] compared different active and passive phishing warnings in Internet Explorer and Firefox. During

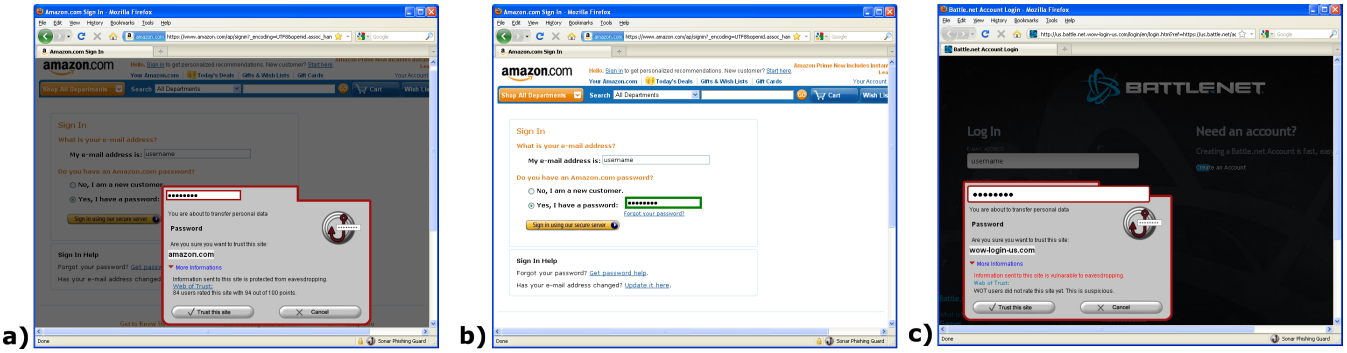

Figure 3: Screenshot examples on real web pages. a) Password Warning (expanded); b) Whitelist Match; c) Password Warning on Phishing Website

the study, participants visited two phishing websites in one of four different browser conditions (actively and passively warning browsers with a control condition not displaying a warning). Participants had to buy items at eBay and Amazon. After their purchase, they received an additional phishing email. Nearly all of the participants in each group followed the phishing link. In the control group and the passive warning condition, 90% of the participants fell for the phishing website. In the two active conditions, only 45% (Internet Explorer) and 0% (Firefox) dismissed the blocking warning and entered their information. Egelman et al. provide five recommendations for phishing indicators. The need to: interrupt the primary task; provide clear choices; fail safely; prevent habituation and alter the phishing website.

2.2.2 *Look and Feel of Security Dialogs*

When presenting a blocking warning to the user, the look and feel of the dialog plays an important role. Amer et al. [1] measured the warning dialog habituation in a browser. They had 88 participants fill out a "sales data entry form" multiple times which displayed a long warning text after each entry. The reading time of the warning message quickly declined from 15 seconds average to a two second average in about three iterations. After appearing eight times in a row, the warning was replaced by a similar one containing different text that required a different answer. Depending on the similarity of the second message up to 89% of the participants did not answer correctly.

Brustoloni and Villamarin-Salomn [3] tried to improve security decisions by creating "polymorphic and audited dialogs". Those dialogs rearranged the possible options to avoid habituation. They combined those approach with context-sensitive guidance that asks the user about the current situation and uses those answers to come to a secure decision. In a user study using e-mail attachments, significantly less unjustified risks were accepted.

To enable the user to quickly understand the contents of security dialogs, Biddle et al. [2] laid out some ground rules concerning the content. When creating those dialogs, unfamiliar terms, lengthy messages and a misleading or confusing wording should be avoided. In their work, they created a new dialog to replace the site information dialog for SSL sites used by various browsers today.

2.2.3 *Teaching*

Since automatic protection is not possible in many cases, it should be also a goal to enhance peoples ability to detect phishing websites by themselves using given indicators. Even if a browser provides additional security indicators, basic user knowledge of how URLs work would already help to detect a lot of phishing sites [14].

Sheng et al. [14] created a game called the "Anti-Phishing Phil" to teach users about which kind of domain names are usually spoofed and which ones are not. During the game, people have to avoid to eat spoofed URLs because otherwise they are "phished". This game-based approach to phishing URL education was then compared to existing training material, and a tutorial condition based on their own material used inside the game. After a teaching phase, the game condition showed the greatest improvements in user correctness when looking for phishing URLs.

A big problem with teaching material on security is the "unmotivated user property" of security decisions [16]. People do not want to have their main task being caring for security. Reading teaching material or playing a game on that subject therefore is not what most of them would voluntarily do in their spare time.

In our work, we tried to create a new kind of warning dialog incorporating the expertise generated so far in those different areas of security research. Since attacking websites are only online for a short duration [7], our approach tries to protect users of fraudulent or untrustworthy websites that do not appear on any blacklist yet and that would not have been detected automatically. However, including automatic detection algorithms can reduce the number of unnecessary warning dialogs to a large extent. To overcome the weaknesses of non-blocking and blocking warnings we try to use a semi-blocking warning that does not allow to submit form data without being noticed but still does allow a minimum of interaction with the current website. The look and feel of our dialog tries to incorporate current state-of-the art research for user interface design and to facilitate teaching information to the users where possible. We also try to detect critical data types to suppress immediate form submission. A detailed description of the concept is explained in Section 4.

3. THREAT MODEL

We assume an attacker that wants to get access to sensi-

tive and private data of a user. The attacker tries to gain access to this data through online websites. A standard example for such attacks are different kinds of phishing. Though phishing might be the most prominent threat, we explicitly include other kinds of "dangerous" websites in the threat model. This includes websites that do not have the goal of stealing personal information but that are simply not trustworthy, for instance, due to technical reasons like missing encryption of sensitive data sent to the server. Looking at the Internet from a privacy perspective, the input of crucial data, the way it is transmitted and who will be the receiver storing it, are important issues the user should reflect on.

4. CONCEPT

Whenever users enter critical data into a web from, the system notifies them of that fact using a specially designed warning dialog appearing right next to that input field (see Figure 1 or 2). This dialog does only appear if the website is not trusted (whitelisted) by the user so far. It blocks the user from further interaction with the website – e.g. submitting the form – but it remains possible to continue typing the current input. With this extra amount of awareness provided by the dialog at that very moment, the user should be made aware of the browser indicators that can be used to detect fraudulent websites and different kinds of attacks. The dialog itself also contains different indicators that help the user to make an informed decision. The user can now dismiss the dialog in case of mistrust or add the website to a list of trusted websites stored in the browser. Whenever a critical data type is detected on a trusted website, the input field is highlighted with a green border to indicate that this website has been trusted before by the user (see Figure 3).

So far, the prototype is capable of detecting three different types of critical data (see Figure 3): credit card numbers, passwords and bank transactions numbers – so called "TANs" (used in Europe to authenticate bank transactions). The prototype is designed so that new types can easily be added. The data type detection algorithms are explained further down. To avoid habituation effects, the different dialogs are clearly distinguishable depending on which data type caused the dialog to appear. This works by displaying the detected type as bold text and as an additional unique icon.

Whenever a website is added to the whitelist, the data type for which it was added is taken into account. In case a website is added for a password warning, only future password inputs on this site will be trusted. If the user inputs a credit-card number on the same website the dialog will reappear. This indicates that credit card information was not used on this website so far. We incorporated such a fine level of granularity because users potentially would not want to add credit-card information though they think it is reasonable to provide a password.

When creating the concept for the warnings, different aspects were considered. After developing and evaluating a first design, the concept was refined and tested again. Details on these changes are described further down. The next sections describe the most important properties of the concept.

4.1 Critical Data Only

An important factor when using warning messages is to avoid habituation – through repeated exposure to the same warning – where possible [18]. This means to reduce the number of unnecessarily appearing alert messages to a minimum. We classify a security dialog as unnecessary if it does not protect the user from an immediate fraud or is unlikely to be understood by the user.

Figure 4: A warning dialog appearing once a secure website is visited that fetches content from and insecure location.

One example for such a warning is the "mixed-content alert" message (see Figure 4). It is displayed for secure websites that fetch insecure content (like ads) from another server.

We therefore propose a concept which does not display warnings unless critical data is involved. This does not mean getting rid of all other browser alerts. In case the browser can guarantee the intended website is an attack (e.g. by using the blacklist approach) the user should definitely be alerted before the website is even displayed.

Displaying such alert dialogs for critical data only has not only the potential to reduce unnecessary warnings, it also makes the user understand a dangerous situation by displaying the data type involved.

4.2 Semi-Blocking Dialog

Both concepts, blocking warnings and non-blocking indicators, have their advantages and drawbacks. Although recent research showed that non-blocking indicators are rarely successful because they are simply overlooked [8], blocking ones are often dismissed without even noticing the contents [1].

Since our warnings appear during interaction with an online form, interrupting the users while typing and blocking the access to the browser would be problematic. Our first tests showed that users usually type the complete input, sometimes without even looking at the browser or the screen. A blocking warning dialog would prohibit further input and therefore much effort would be lost.

Because of this, we introduce the new concept of "semi-blocking" dialogs. Those dialogs appear during user interaction and block some part of the interaction with the website (e.g. submitting a form, or switching to another input field). But rather than being completely blocked, the user can continue typing the current input. This makes it possible to continue the current subtask before handling the warning dialog. However, submitting the form remains blocked as long as the dialog is open.

4.3 In-Context Dialogs

The warning messages in our approach do not appear as a new component of the browser centered on the screen. Instead, they are part of the current website. This has two major advantages, position and timing of the warning, which makes it highly likely that the warning will be noticed and heeded.

- **Position:** Displaying warnings depending on critical

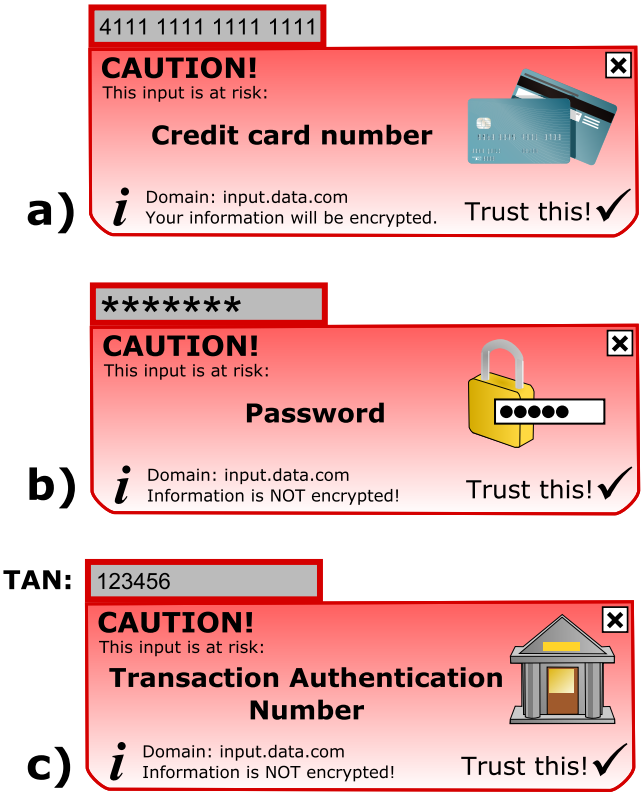

Figure 5: The three different types of warnings for the first prototype – a) Credit card; b) Password; c) Transaction Authentication Number (label is part of detection process). [staged for printing]

data types makes it possible to estimate the user's focus on the screen. The position of the current input field can be computed and the warning message can be placed right next to the field. Although some people will eventually focus on the keyboard while typing, they will look back to the position of the input field to double-check the input.

- **Timing:** The detection of critical data types is performed as the user types. This allows for displaying the message right at the moment the user's input is identified as being critical. In contrast to checking form inputs at the end of a complete interaction with the website before submitting the contents, this allows for the user to immediately reflect upon the current input.

In the current implementation, the warning dialog is injected in the code of the website. Attackers knowing that the plugin is installed would be able to use their own code to remove appearing dialogs automatically or place their own fake dialogs on top of them. Therefore, the warning messages should be rendered as part of the browser's main interface to make interference from the HTML website impossible. Another option is to block any JavaScript execution for the website as long as the dialog is open. Although the concept has been realized as a a plugin for the Firefox browser for evaluation purposes we suggest the concept should be embedded natively in all major browsers.

As mentioned in the related work section, teaching is one of the approaches of handling online security threats. Knowledgeable users will be able to make more mature security decisions but teaching them about security costs time that they are most likely not willing to spend. With the in-context dialogs, we provide minor teaching aspects whilst displaying an ordinary warning message. The message itself informs the user about when and where to take care of critical data and the additional information provided within the message can be used to learn more about the threats.

Another important factor of our approach is the dynamic creation of a personal whitelist for the user. After a warning message appeared and the website has been approved by the user – by confirming the dialog – the website/data combination is whitelisted and the respective combination will not trigger new warnings. This has two major advantages. Whenever a website/data type combination that should be whitelisted triggers a warning, this indicates that the website is most likely a fake. Additionally, it minimizes habituation effects, since over time the appearance of warning dialogs significantly goes down.

5. FIRST PROTOTYPE

To evaluate the concept, we built a first prototype as a Mozilla Firefox plugin[1].

5.1 Design

The main concern of the first prototype was to test the concept with an early version of the plugin. A screenshot of the first interface can be found in Figure 1. The graphical design and the contents of the dialog box were created during initial brainstorming sessions. Important factors of the first design were to create an eye catching alert dialog that would be suitable to be put underneath any input field on an arbitrary website and would still stand out. The finished design was not evaluated before the initial user study. Even though this study showed that there was room to improve the design, the results were very promising.

5.1.1 Data Type Indicator

With critical data types being the trigger for the warning, the type of data needs to be one of the most prominent elements of the dialog. Therefore, the type was displayed in a large font and additionally enhanced by an icon indicating the data type visually. Figure 5 shows those different dialogs for all three data types.

5.1.2 Additional Information Section

To provide some teaching effect and additionally help the user to decide how to handle a website, the interface provides a summary of important information. Namely, this is the URL of the current website and the encryption status. We added a separate section in the lower part of the dialog for this information. Providing additional information for the user is always hard as it needs to be short and yet meaningful [17].

5.2 Plugin Development

To detect data entry, the plugin monitors keyboard events throughout the whole website. Whenever a key is pressed, the values of the form elements are analyzed to find critical data types as described below. When a data type is

[1]http://www.mozilla.com

detected, the respective dialog window with the additional information is created and displayed as a new layer on top of the current website. We implemented dragging to enable moving the dialog to uncover parts of the website that might be necessary to make an informed decision.

5.2.1 Detecting Data Types

The following approaches were used to identify the data:

- **Passwords:** Every input in input fields using the HTML type "password" triggered the password warning.
- **Credit Card Numbers:** To detect credit card number input, we used the LUHN-Algorithm [15] that builds a checksum for identification numbers for most credit cards. The algorithm simply checks whether the sum of all numbers ends with a zero or is divisible by 10. Hence the algorithm is often referred to as "mod10". Since the algorithm would also identify numbers such as "8754" as a credit card number, we used a length check as well.

  This approach would also work for credit card information that is distributed over several input fields by simpling concatenating the input. This prevents an attack using intentionally split-up input fields.
- **TAN-Numbers:** TANs are usually short numbers from 4 to 6 characters length. Each number is used a single time to authenticate a single bank transaction in some European countries. Since this definition clashes with many other inputs, we added additional checks trying to find the word TAN anywhere nearby the input field on the website. Finding this label together with a potential input then triggers the warning message for a TAN.

  An attacker could fool this algorithm by placing an image in front of the input field, or by placing the text by other means in front of the input field. To counteract this, the algorithm has to be refined.

In general, we did not attempt to come up with a perfect set of algorithms and data types here. The basic detection algorithms used for the prototype had the goal to demonstrate the feasibility of the concept.

6. LAB STUDY 1

The first study was conducted with the first version of the prototype. We wanted to test whether the plugin enabled users to make informed security decision by measuring how often they fell for phishing attacks. During the study, the participants were asked to perform several tasks for their grandmother. This scenario, called "Grandma Smith", is partially based on an approach presented by Wu et al. [19]. The purpose of the study was not revealed until the debriefing and instead presented as being about "Internet behavior".

6.1 Study Design

The study was carried out using a mixed-model design, dividing subjects into two groups. The experimental group had to visit several websites using a browser modified with the plugin, whilst the control group used the same browser without modification. This represented the between-subject variable *plugin* (yes or no).

Type	URL	Phishing-Attack
CC	www.bol.de	Cousin Domain
CC	www.amazon.de	Cousin Domain
PW	www.web.de	IP-Address Attack
PW	lokalisten.de	IP-Address Attack
TAN	www.bankingportal.sparkasse-emh.de	Cousin Domain + Content
TAN	www.meine-deutsche-bank.de	Cousin Domain + Content

Table 1: List of URLs and attack types used in the first lab study. CC=credit card; PW=password; TAN=bank transaction number

Within the groups, a repeated measures design was used with the independent variable *data type* (credit card, password and TAN). As dependent variables we measured *achieved security*, the number of correctly identified phishing websites and the number of false positives – genuine sites accidentally nominated as a phishing website.

The participants were instructed to do several tasks for their grandma who was in hospital and needed help to complete some important online tasks she could not do there. Overall, each participant saw six websites, three of them being phishing websites that were created using common phishing attacks – e.g. cousin domain or IP address attack [10]. For each data type, we selected two websites that were well known to the participants and made up several small tasks that required the participants to enter data on the respective website for their grandmother. The two tasks for each data type were similar but slightly adapted with respect to the scenario.

When testing security in lab-studies, it is usually not possible to have the users use their real data for the experiment. Doing this, anxious people eventually drop out of the study, loosing feedback of this important user group [9]. When doing role-playing instead, another problem comes up. People are not willing to protect the critical data since it is only of a fictional character [13]. This is why we chose the indirect method of a fictional character the user has to do some work for, similar to the design of Wu et al. [19]. This allowed the participant to still apply their own set of ethical values whilst not using their own personal data.

The credit card number alert dialogs were displayed on two shopping websites. For TAN, we picked two well-known banking websites. Since password websites are widely deployed, we used a webmail and a community website. The different websites can be found in Table 1, which also displays the method used to create the phishing counterpart.

To minimize learning effects, a new bookmark set was used for each user. For each data type, one of the two websites was randomly assigned to be the phishing site. Twelve study settings were derived using a 6x6 Latin square two times inverting the phishing sites in the second set. In total, 24 participants in two groups with twelve different sets each containing six websites were required.

For our experiment we stated one main hypothesis:

- **H1** Participants of the plugin group will recognize more fraudulent websites than users in the control group.

6.2 Technical Setup

Participants did the tasks on a laptop computer running Firefox 3.6. During the study, all network traffic was di-

verted to a local web server running on the laptop hosting copies of the six original sites and the six modified phishing websites. We used the Microsoft Windows "hosts" file to achieve this traffic diversion. It was impossible for participants to notice the fact that the network traffic was diverted e.g. URLs still were the same. Every website seemed to come from its original location.

6.3 Ethical Considerations

The study was conducted in Germany and had thus not to pass an IRB review. Nevertheless, studies have to comply with strong German privacy regulations that were obeyed during the study. We did not use the participants' real data and collected the data anonymously. After the study, the participants were debriefed and all their questions regarding the security related issues of the study were answered.

6.4 Procedure

At the beginning of the study, participants were informed about their role within the "Grandma Smith" scenario. They were told to be working at their grandmother's laptop who had already created bookmarks for the tasks they should perform. These bookmarks did not provide any clues which URL would be opened when clicked.

Participants then received six hand-written tasks one by one and were told to process them using the grandmother's credentials written down in her secret book. The book was an actual hardcover notebook which contained the passwords and credit card information. The setup for the study was similar to the setup used in the second lab study (see Figure 9).

Participants were told to "think aloud" during the study. When participants had concerns about entering information on one website they were allowed to skip the task if they feared bad consequences for their grandma. Only fully aborting the task was counted as detecting a phishing website. After the six tasks, the participants were debriefed and had to fill out a short post-study questionnaire.

For our experiment we stated one main hypothesis:

- **H1** Participants of the plugin group will recognize more fraudulent websites than users in the control group.

6.5 Participants

24 participants were recruited for the study, most of them being students. They were randomly assigned to one of the two groups. Participants of the plugin group were in average 24 y.o., three of them being female. The control group had an average age of 23 years with four female participants.

6.6 Results

Results can be split in the main quantitative results (number of found input sites) and the findings we gathered from the questionnaires.

In total, participants of both groups were presented with 36 phishing attacks. Using the plugin, subjects refused to submit data on 20 of those 36 attacking websites (55%). In the control group, only only five phishing websites were found (13.9%). Summing up the discovered websites for each participant and comparing them using a Mann-Whitney U test is highly significant ($U = 27.5, z = -2.71, p < .01$). This confirms hypothesis H1.

Looking at the different data types that were used one by one the group difference for credit card ($U = 24.0, z = -3.24, p < .01$) and password websites ($U = 30.0, z = -3.08$, $p < .01$) was also highly significant. In contrast comparing the found websites in the TAN condition was not significant ($U = 72.0, z = 0, p = 1$). This is based on the fact that for transaction numbers an equal number of four phishing websites were found for both groups. This could have been due to the content of the TAN websites (see discussion section 6.7).

As we intended to increase the users security awareness using our plugin, it is highly important to look at how many false positives were produced. The participants of the plugin group accidentally nominated two genuine websites (5.6%) as being phishing websites whilst participants of the control group always entered data on correct websites. This is already a small number of false positives and it can be further reduced by means stated in the discussion section.

Since people were presented multiple phishing websites throughout the study, it is important to analyze how the identified phishing websites are distributed over the participants. Four participants in the control group found at least one website of the five websites identified in total by this group. In contrast, ten of the twelve participants of the plugin group found at least one phishing site. Looking only at the users who actually found phishing websites, plugin group participants found 2.0 websites (SD 0.82) and control-group participants found 1.25 websites (SD 0.5) in average.

After having debriefed the participants, we asked them several question about how they felt about the concept of this new kind of warnings. Rating the helpfulness of the concept on a Likert scale from 1-'not helpful at all' to 5-'very helpful' the participants that had used the plugin rated with a median of 4.

Asking people for advantages and drawbacks of the concept, some of them stated the fact that one has to rethink about the situation before submitting data. Another point mentioned was that habituated data entry will decrease with such a mechanism. Asking for drawbacks, many people stated that the plugin was annoying them. This was most certainly due to it showing up on every site during the study. A second point was that the look and feel of the warning felt not right for some of the users. Both problems are discussed in the next section.

6.7 Discussion

During our study, we wanted to test the impact of our plugin on the users' browsing behavior and had thus only tasks that brought up the plugin with every single website the users visited. This made some of the users experience the concept as somehow annoying. In practical use, the number of those appearing messages would be greatly reduced. First of all, limiting the plugin appearance to critical data types only already reduces the number of warnings. Since the plugin uses a whitelist approach, the number of appearances should also be reduced over time as the whitelist is populated by the user. This effect was tested in a field study using our second prototype (see section 8.6.1). To reduce the number of warnings on generally trustful websites, a public whitelist can be used. This could be queried online or downloaded with the browser for privacy reasons.

During the user study, we compared our concept to existing browser mechanisms. We did this for two reasons. Firstly, as far as we know, there is no plugin or software in related work that would make a direct comparison possible.

Secondly, we wanted to have a control group as common as possible to have a comparison to a real world setting.

As stated in the results section, participants of the control group found the same number of TAN phishing websites as the plugin group (both four out of twelve). We assume that this is due to the fact that we modified the content of the phishing banking websites to closely mimic a usual TAN attack. This modified content may have given a stronger hint to both groups than the other phishing sites did.

Although both groups had the same number of phishing websites there is a possibility that people were more alerted after finding the first phishing website. Looking at the results again showed that in most critical cases phishing websites were missed after another phishing website was detected.

6.7.1 Design Flaws

The initial design of our prototype was criticized by some of the participants during the user study. With respect to those critics and current related work, these flaws were fixed second prototype. The most important issues here was the explanatory text in the warning that relied too much on technical terms – like "encryption". As stated in [2], technical wording should be avoided whenever possible.

A second problem of the first design was the very prominent "Trust this!"-button that had no equivalent counterpart for dismissing the dialog. The little "X" in the upper right corner did non properly suggest that closing of the dialog without accepting the information would also be an option.

Those two main issue led to the decision of conducting a focus group to develop an optimized user interface.

7. SECOND PROTOTYPE

Before testing the prototype in an actual real world study, we wanted to make sure that all problems that we identified during the lab study would have been ruled out. To optimize the user interface, we conducted a focus group to create a new user interface bearing in mind the flaws of the first prototype. After that, we incorporated logging functionality into the plugin to distribute it for an actual field study. The results of this study are found at the end of this section.

7.1 Focus Group

To improve the user interface, a focus group with five participants was conducted. Before conducting the focus group we created a couple of new designs of the interface. Those were then evaluated as a part of the focus group. The new designs can be found in the left column of Figure 6. The participants were mostly students at the end of their bachelor studies which had knowledge in user interface design and HCI. The focus group took 52 minutes and was entirely recorded on video for later analysis. Additionally, one focus group leader and one person to transcribe the important statements were present.

7.1.1 Goals

The focus group should provide general insights on what the participants thought of how the design of the warnings should look like. Additionally, we wanted the participants to evaluate the old design that had been used so far and create new design ideas according to some basic limitations we had set.

Figure 6: The new design drafts created prior to the focus group (left column; a through d) and the drafts created by the participants of the focus group (right column; 1 to 5).

7.1.2 Procedure

The focus group consisted of three phases:

- **Phase 1:** At the beginning of the focus group, the participants shortly introduced themselves. Afterwards, the basic concept of warnings in context depending on certain data types was explained to them. No details on any design considerations that had been made so far were explained. Topics that were examined more closely were the colors, graphics, headlines. Other topics that were discussed in that first part included habituation effects and technical terms.
- **Phase 2:** In this second phase, participants were provided with some material to craft their own dialog. They received background plates in various colors, pencils, diverse graphics and buttons to put on their own creation.
- **Phase 3:** Participants were shown the newly created version of the dialog and should discuss about those drafts. Finally, they were asked to rate the different drafts and nominate the one that they liked best.

7.1.3 Results

The participants discussed intensely about the color of such a dialog and finally agreed on not having a bold red colored background. A reason for this was that due to the

fact that the dialog appeared inside the content of a website it needs to look more like an operating system element. The headline, if any, should not be static with the same word reappearing over and over. Talking about habituation, the participants argued that especially novice users would read the text of such a warning message.

When designing their own dialog, the participants all started to use a bold red background and then switched one after another to to less colorful backgrounds like white or gray. The final designs created during the focus group can be found in the right column of Figure 6.

When being presented the enhanced versions of the dialog, the participants proposed several changes. They did not like the close button that was still in the design although it had the same functionality as the second button. As a second issue they wanted the URL of the website to be much more prominent because this being one of the most important aspects of the dialog. They also argued that an area with more details on the security of the page would be helpful. Those could be used whenever needed.

Voting for the best dialogs, the participants selected drafts a and b of the previously enhanced versions and number two of their own artworks of being the best (see Figure 6).

7.2 Usability and Security Enhancements

To enhance the security of the overall concept, the following changes were made.

1. One big problem of phishing websites is that they do sometimes send out form values in the background prior to the final form submission. To avoid this, a new input field is created with the warning. Therefore, the original field does not receive any input until the dialog is confirmed. When the user trusts the website and the warning is dismissed, the contents are copied back to the old input field. A problem here is that some data types can only be detected after they are fully entered (e.g. credit card numbers). This could be solved by enhancing the data type recognition algorithm used to incorporate labels for example and by blocking the transmission of data recognized as critical types throughout the whole site as long as it has not been whitelisted. A similar concept is used by the "Zapper" in [5].

2. The first prototype was not compatible to auto complete mechanism filling forms. The plugin waited for key events to look for a critical data type. This was fixed in the second version and the complete form is now validated whenever different events happen.

3. The third concept enhancement came from the focus group. To make it clearer to the users that they can only interact with the current input field whenever a dialog is displayed, we darkened the rest of the website with a light gray color. This is shown in Figure 2.

8. FIELD STUDY

With the enhanced plugin, we conducted a field study with 14 volunteers who installed the plugin at their personal computers and used it for seven days. None of them had participated in the first lab study. Doing this, we wanted to test how users judge the plugin after real use. Additionally, we made the plugin send certain pieces of information to our servers that enabled us to retrace how people used the plugin and how often warnings appeared.

8.1 Procedure

We set up a website that explained what we wanted to test and that offered our plugin to be downloaded. After the users installed the plugin and restarted their browsers, they were prompted to input their email address. We needed this address to send them a post study questionnaire link after the seven days of usage. However, it was not used for the evaluation.

Participants began to use the plugin within 48 hours. For our evaluation, we used only the next seven days of log data received from the point of registration onwards. After every user had used the plugin for at least seven days, we send them a link to an online questionnaire and information on how to uninstall the plugin.

8.2 Recorded Data

The plugin collected two types of data. General data on web usage (cached and transmitted later) and detailed data on plugin appearance (transferred immediately). The general web usage data was required to be able to get data on the number of appearing dialogs in proportion to the number of visited websites. To differentiate how many different websites were visited, it was therefore not enough to only collect timestamps for visits but also to identify the website visited somehow without affecting the participants privacy. The plugin therefore reduced the URL to the top-level-domain and the server name (e.g. ebay.com) and then calculated an MD5 hash of it. This hash value was saved together with the timestamp and transmitted to the server.

For the dialog itself, we transmitted five different event types: Whenever a dialog was appearing, when the user clicked "More Information", when the dialog was dismissed either positively or negatively and whenever a website on the whitelist was found. Transmission included the type of data, why the dialog was opened, and the MD5 hash of the visited website.

8.3 Hypotheses

In the study, two main hypotheses were tested:

- **H1** The percentage of new websites will decrease over time.
- **H2** The number of warnings that will popup in a 24 hour period will decrease over time.

For our concept to work it was especially important that the second hypothesis would hold but we also wanted to make sure in how far this effect is linked to the number of new websites.

8.4 Participants

14 people volunteered for participation in the field study, ranging from 22 to 68 years (avg. 40 years). We asked the participants to rate their own level of Internet experience from 1-'not at all experienced' to 5-'very experienced'. The average experience level was 4.2 (SD 0.97). In total, we had five female participants..

8.5 Ethical Considerations

As in the first study no IRB review was needed but we took our best care to respect the participants privacy. The

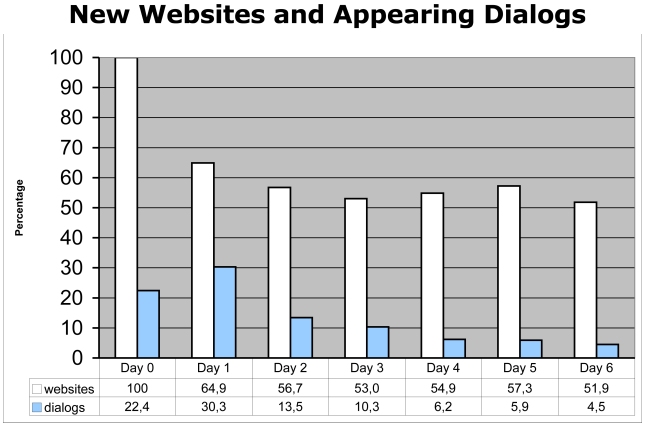

Figure 7: The percentage of new websites visited each day compared to the percentage of websites that triggered an alert dialog. Over one week, the number of warnings decreased drastically.

email addresses of the participants were deleted before analyzing the data. Data analysis was just performed based on the anonymous participation ID.

8.6 Results

In this section, we report on the results of the field study.

8.6.1 *Quantitative Data*

After every user had used the plugin for at least seven days, we stopped gathering data and prepared the evaluation. For each data entry, we calculated a timestamp relative to the moment of registration. This timestamp was then used to truncate all events to exact seven days of data. Since whitelist events appeared whenever a website contained a login form they were triggered rather often. Reasons are for instance that many websites do have their login box on all pages. We chose to allow at maximum one of those entries per website per user per hour and deleted all other events as duplicates. After this, 26,547 unique events remained. Most of those events were website visits (25,641). Those events also indicated how much the users used the Internet per day.

Figure 7 shows the percentage of new websites and the percentage of warning dialogs that were shown on distinct websites for each day. For the percentage of new websites we calculated the sum of distinct websites for each user on a day. After that, we calculated the sum of distinct websites that had not been visited on the days before. The sum of shown dialogs for each user was then calculated and divided by the sum of distinct websites visited which resulted in the percentage of dialogs.

The number of new webpages drops quickly at the beginning but stays at a certain level afterwards. All distinct websites visited in the first 24 hours are new. This does not confirm hypothesis H1. In contrast, the number of shown warnings drops quickly from 22% to 4.5%. This confirms hypothesis H2.

Another important finding is on the number of times the "More Information" panel was opened. Warnings were displayed 229 times in total but the "More Information" panel was only opened 11 times – by 4 users. This shows that important information like the encryption status should not be placed inside a hidden area that has to be uncovered by a user action.

Type	URL	Phishing-URL
CC1	www.neckermann-reisen.de	www.nerckermann-reisen.de
CC2	www.wwf.de	www.wwff.de
PW1	www.ebay.de	www.ebuy.de
PW2	www.paypal.com	www.paypal.webupdate.com

Table 2: List of domain names used for the second lab study. CC=credit card; PW=password

From the 229 dialogs that showed up, 112 were dismissed by adding them to the whitelist and 32 times the dialog was canceled. The 85 missing events are most likely caused by users that navigated away from the website without closing or acknowledging the dialog.

Looking at the number of whitelist events, a total number of 522 events remained in the database after the cleanup process. Over time, the number of those events did not noticeably rise. This may be due to the fact that people revisited most of the acknowledged sites on the first day.

8.6.2 *Questionnaire Results*

Interestingly, none of the participants reported to have ever fallen victim to a phishing attack. They did not even remembered to have ever received an email with phishing content. Asking them about their phishing knowledge from 1-'I don't know anything about it' to 5-'I know it very well', they answered with an average of 3.1. The five subjects that rated their knowledge 4 or 5 where able to give a correct answer on what phishing meant. We also asked them how much they cared for their security online (1-'I don't care at all' - 5-'I care very much') and they answered with 4.4.

Usage of the plugin was rated with 3.9 in average and asking for the concept of the plugin in general it was rated with 4.3. As additional information, two users requested an "degree of danger"-indicator but many were already happy with the information offered so far.

Asking them about what they especially liked about the plugin, five users (35%) mentioned the coloring of the input fields. Some of them emphasized the green border, which gave them a feeling of being secure. Since users in our lab study did not experience that situation, it is an important finding of the field study. As disadvantages, some users mentioned that wrong data types were detected at their input.

The users correctly experienced the number of dialogs becoming less frequent. We asked them on a Likert scale from 1-'The warnings did not get less' to 5-'The warnings got less fastly' the Median was 4.5. In the end, we asked the users whether they could imagine to use the plugin in the future. Three users agreed (4) and seven strongly agreed (5).

8.7 Discussion

Even though the number of participants was limited and the time frame we used was only one week, the field study provided us with valuable data. Within this week, users visited a certain amount of new websites each day. Still, the number of websites that required critical input decreased quickly. Installing a plugin that builds its own whitelist therefore seems feasible and should quickly stop bothering people with other decisions. With a pre-populated whitelist this number can be decreased further.

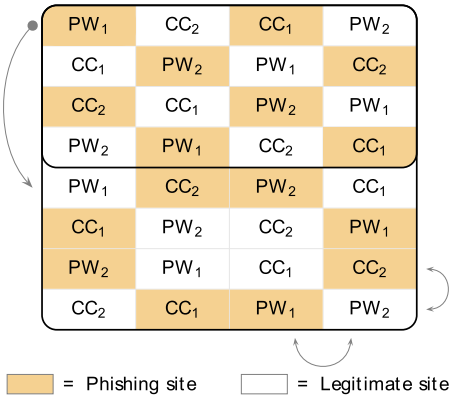

Figure 8: Order of tasks and phishing websites presented to the participants. A 4x4 latin square was used twice but flipped.

Figure 9: The setup of the second lab study.

A very important finding was that when designing an information dialog that needs decisions making, additional information should be displayed immediately. Ten of the participants did not open the "More Information" dialog at all.

9. LAB STUDY 2

In addition to the field study, we performed an additional lab study using the enhanced prototype.

9.1 Design and Procedure

We used the "Grandma Smith" scenario a second time but applied some changes to the study design. Due to the different kinds of websites needed to test a TAN phishing site, we chose to omit this data type this time. Resulting from this fact, we had only four different websites that we tested throughout the study. Compared to the first study, we also chose to use only one type of URL spoofing, namely the cousin domain attack [10].

For the credit card, we used a well known German travel operator website and the donation website of the German WWF. For the password data type we used PayPal and eBay. For those sites we used standard SSL instead. The websites and the corresponding spoofed URLs can be found in Table 2.

Again, all websites were hosted on the local machine used for the study and all traffic was diverted to a local web server. To make the genuine websites look properly encrypted, we created own SSL certificates and our own certificate authority that was added to the local browser. However, it was not possible to create fake extended validation certificates that are used by PayPal and eBay. The phishing counterparts of our websites were not delivered in an encrypted way.

The hypothesis for this experiment remained the same as it was in the first lab study:

- **H1** Participants of the plugin group will recognize more fraudulent websites than users in the control group.

Since we thought that the level of user expertise may have great impact on the outcome of such a study, we chose to screen our users before starting the first task and to balance them by security knowledge. The participant had to fill out an opening questionnaire. Since the study was disguised as being about browsing behavior only, we did not ask any security related questions. Instead we asked people to rate their "Understanding of Internet technologies" on a Likert scale from 1-'No knowledge' to 5-'Very good knowledge'. People answering 4 or 5 were considered as being experts and were evenly distributed over both groups.

Each participant used all four websites, one website of a data type always being a phishing website. We used a 4x4 latin square two times, flipping it in the second attempt to get a good distribution of our tasks. This scheme can be found in Figure 8 and was used for the plugin and the control group resulting in 16 participants.

An image of the study setup can be found in Figure 9. We used a MacBook Pro running Windows XP in a virtual machine. The MacBook was connected to an external screen at a resolution of 1024x768 pixels. An external mouse and keyboard with German layout was provided.

9.2 Participants

The youngest of the 16 participants was 19 years old the oldest 51 (avg. 28). Four of the participants were female. None of them had been taking part in any of the studies conducted so far. Using the expert classification, only three of them did not state to be an expert – two of them in the plugin group and one non-expert in the control group. In average, the expertise was rated with 4.1. The participants were all frequent Internet users. Asking them on a Likert scale from 1-'very seldom' to 5-'very often' they answered with 4.8 in average.

9.3 Results

Participants of both groups discovered more phishing websites than in the first study. The plugin group discovered twelve out of the 16 (75%) phishing attacks, the control group seven out of the 16 (44%) phishing websites. This rather large looking difference was not statistically significant. Applying the Mann-Whitney U test to the sum of found websites per user results in no significant result ($U = 19.0, z = -1.49, p = 0.14$). Looking at both data types separately neither credit card ($U = 16.0, z = -1.94, p = .053$) nor password ($U = 28.0, z = -.52, p < .60$) were significant.

Possible explanations for this are: Most importantly we had a very high number of experts in this study. This ex-

plains why the number of identified phishing websites in the control group was that high. Additionally, there was one participant in the control group who refused input on the two phishing websites not because he had detected the phishing sites but for other reasons. Considering the small sample size, one participant can already have a big influence on the result.

As in the real world study, nearly no one opened the "More Information"-field of the alert dialog. As this dialog contained the information whether the traffic was encrypted, most participants did not see this information.

In the post study questionnaires, we asked the participants for their phishing knowledge. On a Likert scale from 1-'never heard of it' to 5-'I know it very well' the knowledge was rated with 3.6 on average. Although we had evenly distributed the Internet experts the median of phishing knowledge was not the same. The plugin group had only a median of 3 whilst the control group had a median of 4. This could be another indicator for the rather high score of the control group. Seven of the participants remembered that they had at least received a phishing link once. The plugin concept was again rated with a median of 4.

We presented the concept to the control group afterwards and asked all of them for advantages and disadvantages. Most participants explicitly stated that it would be good for inexperienced users and the fact of being reminded again of the usage of a critical data type. As disadvantages, the additional click on non-fraudulent websites was mentioned. We also asked people what kinds of additional information they were missing. Some people suggested additional certificate details. Others wanted to get more information about the plugin itself. One person even stated that the encryption status was one of the most important points and should immediately be visible.

9.4 Reflecting Study 1 and 2

Comparing this study with the first study, we can identify benefits and problems of the enhanced prototype. The new design with the two selection options and the big URL in the center of the dialog seems to have helped more users to detect fraudulent URLs.

Hiding important information from being visible at first glance seems to be a mistake. As many people requested further Information, we suggest that a "More Information" box could still be used but should only contain information that is not critical for the security decision.

Many experts explicitly stated that they think the concept seems good for non-technical savvy users. When conducting a future user study to test such an approach, the participants should be selected out of such a target group.

10. CONCLUSIONS

With the large number of studies, we are able to draw different conclusions in separate areas. All-in-all, the concept of having in-context data types seems very promising as it enables users to detect more fraudulent websites. Using the field study, we were able to rule out problems with appearing dialogs.

The possible loss of critical data like credit card information seems to be a more convincing argument for an average user than a technical security warning. Throughout all questionnaires, people stated that they thought such a warning makes sense. This is also a first step to reduce the appearance of warnings and minimize habituation effects.

Semi-blocking warning dialogs appear in-context at the location of the users' focus. Even though many people look on their keyboard while typing, they notice the warning when looking back to the screen. This method enables to interrupt the users current task in a softer way. The major problem of habituation is reduced by that manner. However, the number of false alarms should still be as low as possible.

In the field study, we showed that the number of appearing warning dialogs quickly decreases over time when generating a personal whitelist for the user. As browsers today already access online blacklists to block the visits to known malicious sites, a public whitelist could be used to limit the number of appearing warnings from the beginning.

Having dialogs appear with critical data types, informing the user about important information in the browser that helps to detect malicious websites, the users are also taught about important security issues right at the moment they appear.

11. FUTURE WORK

Future work about data-type based dialogs should be done in two separate ways. Since the concept seems so promising and the field study brought up interesting facts, a long-term field study of this concept would be interesting. Such a study could determine where the concept really helped people to not fall for an attacking website or submit their data to some untrustworthy party. A big problem with this is the huge number of participants and the large amount of time that would be required to carry out the study.

The basic concept of semi-blocking dialogs could be used in other scenarios besides security. Like auto complete windows enable users in many different areas today to make input more quick, a warning message bound to a certain action the user just made, can be noticed much easier.

12. REFERENCES

[1] T. S. Amer and J. B. Maris. Signal words and signal icons in application control and information technology exception messages–hazard matching and habituation effects. *Journal of Information Systems*, 21(2):1–26, 2007.

[2] R. Biddle, P. C. van Oorschot, A. S. Patrick, J. Sobey, and T. Whalen. Browser interfaces and extended validation SSL certificates: An empirical study. In *CCSW '09: Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 19–30, Chicago, Illinois, USA, 2009. ACM.

[3] J. C. Brustoloni and R. Villamarín-Salomón. Improving security decisions with polymorphic and audited dialogs. In *Proceedings of the 3rd symposium on Usable privacy and security*, pages 76–85, Pittsburgh, Pennsylvania, 2007. ACM.

[4] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, and J. C. Mitchell. Client-side defense against web-based identity theft. In *Proc. NDSS*, 2004.

[5] S. Consolvo, J. Jung, B. Greenstein, P. Powledge, G. Maganis, and D. Avrahami. The Wi-Fi privacy ticker: improving awareness & control of personal information exposure on Wi-Fi. In *Proceedings of the 12th ACM international conference on Ubiquitous*

computing, Ubicomp '10, pages 321–330, New York, NY, USA, 2010. ACM. ACM ID: 1864398.

[6] M. Corp. Phishing protection: Design documentation. https://wiki.mozilla.org/Phishing_Protection:_Design_Documentation.

[7] A. Delmiglio. Online fraud in italy: Analysis of 5830 phishing attacks. http://www.symantec.com/connect/blogs/online-fraud-italy-analysis-5830-phishing-attacks, 2007.

[8] R. Dhamija, J. D. Tygar, and M. Hearst. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 581–590, Montréal, Québec, Canada, 2006. ACM.

[9] S. Egelman, L. F. Cranor, and J. Hong. You've been warned: an empirical study of the effectiveness of web browser phishing warnings. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1065–1074, Florence, Italy, 2008. ACM.

[10] A. Emigh. Online identity theft: Phishing technology, chokepoints and countermeasures. *Radix Labs*, 3, 2005.

[11] C. Ludl, S. McAllister, E. Kirda, and C. Kruegel. On the effectiveness of techniques to detect phishing sites. *Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 20–39, 2007.

[12] M.-E. Maurer, A. De Luca, and H. Hussmann. Data type based security alert dialogs. In *In CHI '11: Extended Abstracts on Human Factors in Computing Systems.*, New York, NY, USA, 2011. ACM.

[13] S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer. Emperor's new security indicators: An evaluation of website authentication and the effect of role playing on usability studies. *In proceedings of the 2007 IEEE symposium on security and privacy*, 2007.

[14] S. Sheng, B. Magnien, P. Kumaraguru, A. Acquisti, L. F. Cranor, J. Hong, and E. Nunge. Anti-Phishing phil: the design and evaluation of a game that teaches people not to fall for phish. In *Proceedings of the 3rd symposium on Usable privacy and security*, SOUPS '07, pages 88–99, New York, NY, USA, 2007. ACM. ACM ID: 1280692.

[15] R. Tervo. Secrets of the LUHN-10 algorithm. http://www.ee.unb.ca/tervo/ee4253/luhn.shtml, 2002.

[16] A. Whitten and J. D. Tygar. Why johnny can't encrypt: A usability evaluation of PGP 5.0. In *Proceedings of the 8th USENIX Security Symposium*, pages 169–184, 1999.

[17] M. S. Wogalter. Attention switch and maintenance. *Handbook of human factors in Web design*, pages 245–265, 2006.

[18] M. S. Wogalter and J. W. Brelsford. Incidental exposure to rotating warnings on alcoholic beverage labels. In *Human Factors and Ergonomics Society Annual Meeting Proceedings*, volume 38, pages 374–378, 1994.

[19] M. Wu, R. C. Miller, and S. L. Garfinkel. Do security toolbars actually prevent phishing attacks? In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 601–610, Montréal, Québec, Canada, 2006. ACM.

[20] Y. Zhang, S. Egelman, L. Cranor, and J. Hong. Phinding phish: Evaluating anti-phishing tools. In *Proceedings of the 14th Annual Network and Distributed System Security Symposium (NDSS 2007)*, 2007.